UNIVERSITÀ DI PISA
FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI SISTEMI ELETTRICI E AUTOMAZIONE
CENTRO INTERDIPARTIMENTALE DI RICERCA "E. PIAGGIO"

DOTTORATO DI RICERCA IN AUTOMATICA, ROBOTICA E BIOINGEGNERIA

-*XVIII* CICLO-

TESI DI DOTTORATO

DECENTRALIZED TRAFFIC MANAGEMENT OF MULTI-AGENT SYSTEMS

Candidato
Ing. Vincenzo Scordio

Tutore
Prof. Ing. Antonio Bicchi

April 2006

Abstract

Autonomous agents and multi-agent systems (MASs) represent one of the most exciting and challenging areas of robotics research during the last two decades. In recent years, they have been proposed for several applications, such as telecommunications, air traffic management, planetary exploration, surveillance etc.. MASs offer many potential advantages with respect to single-agent systems such as speedup in task execution, robustness with respect to failure of one or more agents, scalability and modularity. On the other hand, MASs introduce challenging issues such as the handling of distributed information data, the coordination among agents, the choice of the control framework and of communication protocols.

This thesis investigates some problems that arise in the management of MASs. More specifically it investigates problems of designing decentralized control schemes to manage collections of vehicles cooperating to reach common goals, while simultaneously avoiding collisions.

An existing decentralized policy for collisions avoidance, already proved safe for a system with three agents, has been extended up to five agents.

A new decentralized policy, the *Generalized Roundabout Policy*, has been designed and its properties analyzed. Specifically *safety* and *liveness* properties have been studied. The first one has been proved formally, while the second has been addressed by means of probabilistic approaches.

Moreover, it is addressed the problem of optimization of autonomous robotic

exploration. The problem is clearly of great relevance to many tasks, such as e.g. surveillance or exploration. However, it is in general a difficult problem, as several quantities have to be traded off, such as the expected gain in map information, the time and energy it takes to gain this information, the possible loss of pose information along the way, and so on.

Finally, software and hardware simulation tools have been developed for the analysis and the verification of the decentralized control policies. Such instruments are particularly useful for the verification of multi-agent systems which could be overwhelmingly complex to be addressed purely by a theoretical approach.

Acknowledgements

To my parents

I want to thank Vera whose love and patience helped me to go on in all circumstances. Grazie amore.

Thanks also to all my family: my brothers Mike and Andrea, my father Emanuele and my mother Maria for always supporting me during all these years.

I deeply wish to thank Prof. Antonio Bicchi for all the things he has taught to me, for all the opportunities he has given to me to increase my knowledge during these three years of Ph.D. studies.

Many thanks to all my collegues of the Centro "E. Piaggio" who became close friends: Antonio Danesi, Daniele Fontanelli, Nicola Sgambelluri, Giovanni Tonietti, Pasquale Scilingo, Antonino Previti, Pierpaolo Murrieri, Adriano Fagiolini, Riccardo Schiavi, Luca Greco and Giordano Greco.

Aknowledgment also to Walter Lucetti, Lorenzo Decaria and Alessandro Convalle for their contribution to this work during their Master thesis.

I want to thank Prof. Emilio Frazzoli for the important contribution in some part of this work.

In the end, I want to thank deeply Lucia Pallottino for all the fruitful work we have done in the last two years and for her generous friendship.

Contents

List of Figures

PREFACE

Autonomous agents are automated entities, such as computer programs or robots, with the capacity of interacting with their environment. A multi-agent system (MAS) is made up of a number of autonomous agents that share common resources and cooperate to address common goals. In recent years, MASs have attracted increasing attention and have been proposed for several applications, such as telecommunications, internet, air traffic management, planetary exploration, surveillance etc.. MASs offer many potential advantages with respect to single-agent systems such as speedup in task execution, robustness with respect to failure of one or more agents, scalability and modularity. On the other hand, MASs introduce challenging issues such as the handling of distributed information data, the coordination among agents, the choice of the control framework and of communication protocols. In past years centralized algorithms for the management of a multi-agent system have been successfully adopted. However they typically require a large amount of computational resources, and, as the complexity of systems grows, it becomes more difficult to coordinate the actions of the large number of agents. Furthermore, centralized approaches typically are very prone to faults of the decision maker. Recently, decentralized control frameworks appeared, where the approach is to divide the coordination problem into smaller and simpler subproblems. Different levels of decentralization, admitting intermediate level of coordination units can be adopted. The most challenging scenario is that where each agent executes its tasks autonomously relying only on information gathered from the environment

and exchanged with its "neighboring" agents.

This thesis deals with the problem of designing decentralized algorithms for the coordination of groups of vehicles, in order to avoid conflicts, ensuring that each of them can perform its task. Tasks can be assigned to agents depending on the specific application. We consider the case where agents have to move from an initial to a final goal configuration. The final goals can be assigned to agents by an external unit as a datum of the problem, as it happens in air traffic control, or can be generated by agents themselves accordingly to a specific task, as, e.g., in the problem of navigation in unknown environments. In this case targets are determined in order to optimize somehow the information gathered by a vehicle during the exploration of totally or partially unknown environments. This optimization problem has great relevance to many applications, such as e.g. surveillance or exploration. However, it is in general a difficult problem, as several quantities have to be traded off, such as the expected gain in map information, the time and energy it takes to gain this information, the possible loss of pose information along the way, and so on. We will introduce an algorithm for optimizing such information.

As it will be largely discussed in this thesis, decentralized policies are typically capable of managing a large scale systems, are faster to react to unexpected situations, and released from possible malfunctioning of a central authority. Nevertheless they may present some counterparts such as the degradation of performances, e.g. the loss of optimality and the complexity of verification of important properties. In the case of collision avoidance problem, important properties to be verified are *safety* and *liveness*, which are important issues of this thesis. The first should prove that no conflicts will occur between agents while executing a given policy. The second should guarantee that all agents will reach their final destination in finite time. In other words safety and liveness properties intuitively state that something bad will not happen and that something good will eventually happen. As mentioned, some properties of MASs may turn out to be overwhelmingly complex to verify the-

oretically. In this thesis we adopt a probabilistic approach based on a large number of simulation to verify properties in probability. To this purpose software and hardware test-bed platforms have been designed for the verification of the decentralized policies.

Literature Background. In recent years, one of the most addressed issues concerning multi-agent systems, both in robotics and in other application domains, has been the problem of safely coordinating the motion of several robots sharing the same environment. An important contribution to the development of multi-agent management policies have been provided by flocking literature ([1, 2, 3]), which is concerned with the problem of maintaining a formation of moving agents using only simple rules [4] and local information, without any global decision maker. Flocking is based on the observation of biological system examples, such as flock of birds, swarms of bees, schools of fishes, colonies of insects and any other flocking animal, including humans. Flocking concepts have been applied to accomplish some interesting tasks such as sensor coverage (how to dispose sensors in order to optimize some indices) , rendezvous (how to bring agents to a common meeting point), how to steer and keep agents to a formation, etc.. The main issue addressed in flocking is the stability of a formation. In [5], authors showed that a group of autonomous mobile agents, in which each agent is steered by a decentralized control law based on state information from its local flock mates, is stable in the sense that all the agents headings converge a common orientation and the agents stay close to each other while avoiding collisions. Stability is guaranteed provided that the graph representing the communication between neighbouring agents remains connected, even though its topology is time variant. Nevertheless, in flocking, agents do not address individual objectives, and they are not guaranteed to reach a pre-assigned individual destination.

Another important trend of research concerning MAS is the application of a team of robots for the exploration of unknown environments. The use of multiple robots

can bring several advantages over single robot systems [6, 7] in terms of quickness in accomplishing a task, tolerance to possible failures and capability of merging heterogeneous information, thus compensating for sensor uncertainty. For example, multiple robots have been shown to localize themselves more efficiently, especially when they have different sensor capabilities. However, when robots operate in teams there is the risk of possible interferences between them. Hence a coordination is necessary to avoid collisions among members of the team. One of the most challenging topics in SLAM is the optimization of autonomous robotic exploration. Indeed, it is often the case that robots have degrees of freedom in the choice of the path to follow, which should be used to maximize the information that the system can gather on the environment. The problem is clearly of great relevance to many tasks, such as e.g. surveillance or exploration. However, it is in general a difficult problem, as several quantities have to be traded off, such as the expected gain in map information, the time and energy it takes to gain this information, the possible loss of pose information along the way, and so on. This problem is considered in detail in this thesis.

As it can be argued, a common issue of all the applications involving multiple robots is the coordination of agents, and the collision avoidance problem is one of the basic coordination tasks. This problem concerns how to provide policies for guaranteeing that vehicles do not collide with each other while accomplishing their tasks. Traditionally it has been addressed by using a centralized approach where trajectories for all agents are computed by a unique decision maker (see e.g. [8, 9, 10, 11, 12, 13] for air traffic conflict management).

More recently decentralized approaches appeared. In [14] a hybrid control architecture is proposed with parallel problem solving which guarantees collision avoidance. In [15] the problem of path planning is divided into global and local path planning, and AI techniques are used in combination with real-time techniques. In [16] and [17], authors consider formations of robots, where a motion plan for the

overall formation is used to control a single "lead" robot while the "followers" are governed by local control laws, sensing their positions relative to neighboring robots. In [**?**] a framework exploiting the advantages of centralized and decentralized planning for multiple mobile robots with limited ranges of sensing and communication maneuvering in dynamic environments, is presented. Other decentralized conflict resolution schemes, which are often referred to as "free-flight" strategies, are reported in [18, 19, 8, 20], and [21].

Thesis organization. Following this introduction, chapter 1 focuses the problem of multi-agents systems. It analyzes differences between centralized and decentralized approach in managing such systems. It outlines important advantages provided by a decentralized approach despite of its intrinsic difficulties in verifying some properties of the system. Furthermore, it shows how a decentralized control scheme of a multi-agent system is well suited to a hybrid-system modelling, where the agent's dynamics represents the continuous part, while the time-varying topologies of agents in the environment may determine discontinuities in controls. Chapter 2 introduces a decentralized collision avoidance policy for multi-agent traffic management. Conditions ensuring the existence of maneuvers avoiding conflicts are provided, along with maneuvers that each agent can make. Proofs that the introduced policy works up to five agents are provided.

A new cooperative decentralized algorithm, called "Generalized Roundabout policy" is presented in chapter 3. Such policy allows a collection of agents moving on a planar environment, at constant speed and with limited curvature radius, to reach their assigned targets, starting form initial configurations and avoiding conflicts. Some properties of the "GRP" such as high scalability, safety in terms of collisions avoidance, absence of deadlocks and livelocks are analyzed. Furthermore, this chapter analyzes the liveness property of the "GRP", i.e. the capability of negotiating a solution in a finite time. Specifically, conditions on the final vehicles configuration are provided and conjectured to be sufficient for guaranteeing

liveness. The proof of the sufficiency is investigated in probability, because of the overwhelmingly complexity of an analytic approach. To this purpose a large number of simulation results have been examined.

Chapter 4 concerns the problem of the navigation of an autonomous vehicle in a partially or totally unknown environment. It proposes a strategy for exploration that results optimal in terms of information gathered. A closed-form solution is provided for such an optimal problem. Moreover it is provided a numerical implementation of the optimal strategy, that is necessary, despite of the loss of optimality, when the information to be gathered grows with the complexity of the environment.

Software and hardware tools which have been developed for the simulation of the developed policies are presented in chapter 5. Furthermore in this chapter some considerations about some important features of a simulation tool for multi-agent hybrid systems are reported.

Finally, chapter 6 is dedicated to conclusions.

Chapter 1

Collision avoidance of multi-agent systems

The problem of collision-free motion planning for a number of mobile agents evolving on the plane has received a great deal of attention in several areas of application such as manufacturing plants, automated factories, air traffic control, and intelligent transportation system applications. The problem has been approached through centralized methods in which safe trajectories for all agents are computed by a unique decision maker (see e.g. [8, 9, 10, 11, 13] for air traffic conflict management).

Although correct and complete algorithms for the centralized traffic management problem exist, they typically require a large amount of computational resources and, as the complexity of traffic control grows it becomes more difficult to coordinate actions of a large number of agents. Furthermore, centralized approaches are typically very prone to faults of the decision maker. Alternatively, in a decentralized control framework, the approach is to divide the coordination problem into smaller subproblems that can be solved with a minimum of coordination. This technique can lead to different degrees of decentralization, up to the challenging scenario where

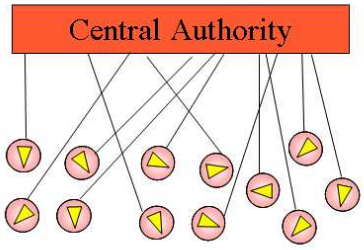


Figure 1.1: A centralized control scheme. There is no communication between agents, which interact with a central authority.

each agent plans its own trajectory based only on information limited to neighboring agents. A decentralized approach is typically capable of managing a large scale systems, is faster to react to unexpected situations, released from possible malfunctioning of a central authority. Nevertheless it may present some counterparts such as degradation of performances, i.e. loss of optimality, and complexity of safety verification, since domino effects of possible conflicts may prevent convergence to solutions in some conditions. Sometimes the performance degradation of a completely decentralized solution is unacceptable and a completely centralized solution is prohibitively complex or expensive. In such cases a compromise is adopted where each agent tries to optimize its own performance index coordinating with *neighboring* agents, but a central authority may eventually intervene to solve conflicts of objectives. It should be noticed that partially and completely decentralized control schemes are naturally suited for hybrid designs. At the continuous level, each agent pursues its own task, while discrete coordination between agents and eventually a central authority is used to resolve conflicts.

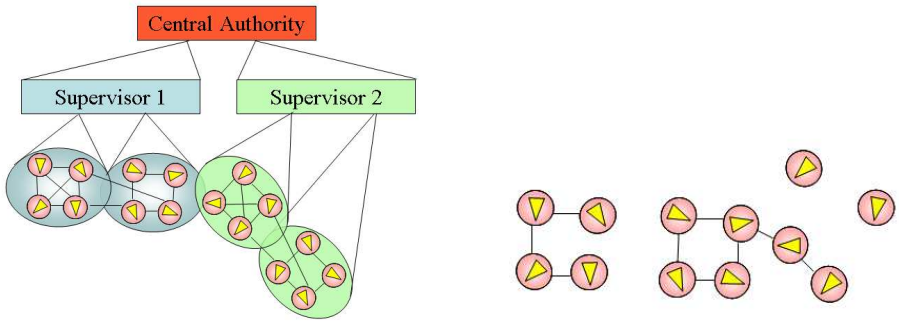


Figure 1.2: Left: A partially decentralized control scheme. Agents, clustered in subgroups communicate each others and are also coordinated by an external supervisor. Subgroubs can also communicate each others and are coordinated by an higher level central authority. Right: A completely decentralized scheme. There is no external Decision Maker. Each agent makes decisions according to information exchanged with its "neighbours" only.

1.1 Centralized approaches

Let us consider the problem of safely coordinating a number of agents with the aim of minimizing a weighted sum of costs. Consider first a centralized control scheme, whereby configurations of all agents are known by a single *Decision Maker* (DM). All possible conflicts can be solved by the DM by finding admissible controls for each of the N agents in the controlled workspace so as to minimize a given cost function. A cooperative centralized cost function is usually written as a weighted sum of individual costs,

$$J = \sum_{1}^{N} L_i,$$

where L_i may represent e.g. the path length for agent i [8], or the maximum deviation from its nominal direction [11].

1.2 Decentralized approaches

In a decentralized cooperative control scenario, each agent must make decisions autonomously, based on the information that is made available in real time through a communication network. The network does not have a fixed topology, but its connectivity is established depending on the current configuration of the overall system (*ad hoc* networking). Connections can be established and lost for different reasons, which can be modeled by e.g. *visibility graphs* for line-of-sight communications or *distance graphs* for radio-like communications.

1.2.1 Informative Structures

Let us consider n agents moving on a euclidean workspace. Suppose that each agent has access to data concerning agents within its *alert zone*, defined as sphere of radius R_a centered in the agent's position. Let $S_i(t)$ be the set of agents, referred to as "neighbours", which at time t are within the alert zone of agent i,

$$S_i(t) = \{j : j \neq i \wedge d_{i,j} < R_{a,i}\}\,. \tag{1.1}$$

where $d_{i,j}$ is the euclidean distance between agents i and j expressed by

$$d_{i,j} = \|(x_i, y_i, z_i) - (x_j, y_j, z_j)\|_2. \tag{1.2}$$

The set of data which each agent exchanges with others is referred to as "Informative Structure" (*IS*), and represents a key characteristic of decentralized control schemes for *MASs*. An informative structure is *symmetric* if $j \in S_i(t) \Rightarrow i \in S_j(t)$; it is *transitive* if $i \in S_j(t) \wedge j \in S_k(t) \Rightarrow i \in S_k(t)$. In other words symmetry implies that agents have the same alert radius while transitivity implies that two agents may share data though they are not within the alert distance, by means of a third agent which plays the role of bridge (see figure 1.3). Let $I_j(t)$ denote the information concerning agent j available to agent i whenever $j \in S_i(t)$. Let us define $IS_i(t)$ the set of information available to agent i, relative to all agents within

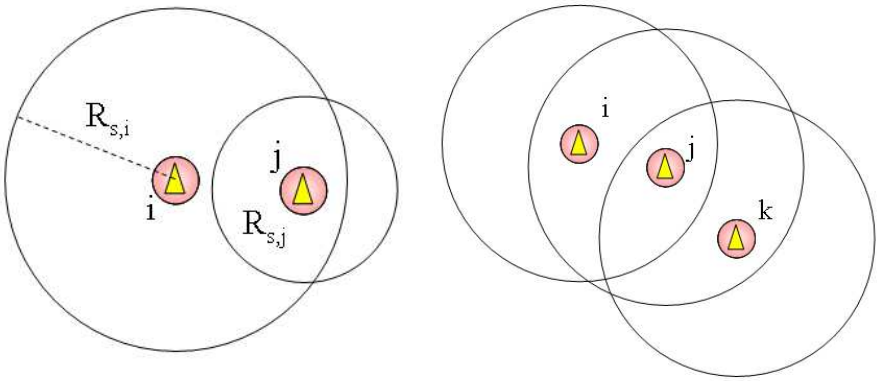


Figure 1.3: Informative structures. Left: asymmetric case. $S_i = \{i, j\}$, $S_j = \{j\}$. Right: symmetric case. If transitivity is admitted $S_i = S_j = S_k = \{1, 2, 3\}$; otherwise $S_i = \{i, j\}$, $S_j = \{i, j, k\}$ and $S_k = \{j, k\}$.

its alert distance,

$$IS_i(t) = \left\{ \bigcup I_j(t) : j \in S_i(t) \right\}. \tag{1.3}$$

A decentralized control policy for agent i consists in a feedback law of the type $u_i = u_i(IS_i)$. In a cooperative decentralized optimization approach, agent i establishes its feedback law based on the minimization of a cost function of information associated to its *neighbours*.

$$J_i = \sum_{j \in S_i} L_j, \tag{1.4}$$

1.2.2 Hybrid Control Problem

The cost (1.4), along with the agent's dynamics, input and state constraints, define an optimal control problem, which, if well-posed, determines univocally a control policy for agent i. If no optimization is considered, and only convergence towards the goal is required, then we simply set $J_i = 1$ for all agents. If multiple optimal solutions are possible, a suitable system of rules can be enforced to the purpose of

univocity. As a consequence, to each different information structure there corresponds a working mode for the system, i.e. dynamics driven by controls optimizing J_{i,S_i} subject to the non-conflict constraints for all pairs (i, j) with $j \in S_i$. However, when during execution of maneuvers that were planned based on a certain information structure $I = (S_1, \ldots, S_n)$, an agent j with $j \notin S_i$ becomes connected with agent i, or one in S_i loses connectivity, the information structure is updated, and optimal paths are replanned according to the new cost function and constraints for agent i. The resulting system is therefore hybrid, as it is comprised of a finite–state machine and of associated continuous-variable dynamic systems, transitions among states being triggered by conditions on the continuous variables.

To illustrate application of a cooperative decentralized policy on a non-transitive, reflexive information structure, consider a $N = 3$ scenario and its associated graph reported in figure 1.4. Each node in the graph corresponds to a different status of the communication network. Switching between nodes are triggered when an agent enter or exits the alert disc of another. There are eight possible states (modes of operation), corresponding to different information structures (see figure 1.4). At each state transition, each agent evaluates in real–time the optimal control (heading angle change), from current information structure, for itself as well as for all other agent within its alert radius. Only the control policy evaluated by an agent for itself is then executed, as the one calculated for others may ignore part of the information available to them due to the non-transitivity of the information structure. By reflexivity and transitivity of the information structure, whenever agent i appears in S_j , then j also appears in S_i and $S_i = S_j$. Hence, all agents whose indexes are in the same set S_i effectively share the same information and hence execute the same policy. We will therefore refer to $S_i = S_j$ as a "team" in this case.

We describe now the structure of a general N-agents decentralized transitive scheme. Recall that transitions among different operating modes are triggered by zero-crossing conditions for variables of the type $d_{ij} - R_{a,i}$. We assume that a

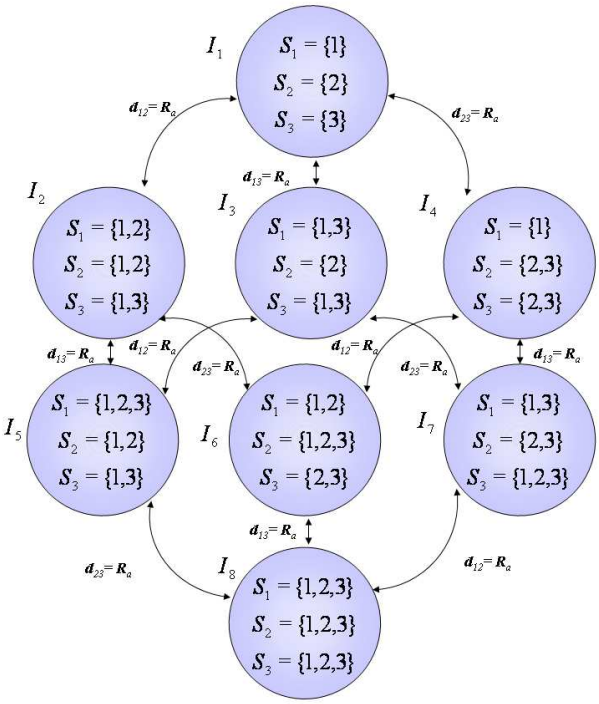


Figure 1.4: Transitions of Informative Stuctures for a system with three agents having equal alert radius. Each node in the graph corresponds to a different informative configuration. Switching between nodes are triggered when an agent enter or exits the alert disc of another.

minimum dwell time is enforced in each mode, and that no simultaneous transitions are allowed. This assumption implies for instance that, in figure 1.4 no direct arc exists between state I_1 and state I_5.

To the purposes of safety analysis, a further reduction of the cardinality of modes is instrumental. All nodes in an information graph such as that in figure 1.4, which share the same number of teams and the same number of elements per team, can be identified in a single node as represented in figure 1.5. A new graph is thus generated, named relaxed graph, in which a node is characterized by a list $([n_1], ..., [n_m], z)$, where m is the number of non–trivial teams, $n_i > 1$ is the number of elements in the i–th team, and z is the number of trivial (singleton) teams for which $S_j = j$. For example, nodes I_{21}, I_{22}, and I_{23} in figure 1.5 are identified in the relaxed graph with a $([2], 1)$. Occasionally, nodes of the relaxed graph will be labeled by I_{jk}, where the first index represents the depth of the node in the hybrid system with respect to transitions, while the second index is needed to distinguish nodes of same depth. The first node I_{11} of the hybrid system, for the N agents case, is the one represented by N. This node is thus characterized by $S_i = i$ for $i = 1, \ldots, N$, i.e. all agents are at relative distance larger than the alert distance. From node I_{11} transitions can occur only to node I_{21} that represents a team of two agents and $N - 2$ teams of single agents: $([2], \{N - 2\})$. To the purposes of safety analysis, only transitions from node I_{ik} to I_{jl} with $i \leq j$ are considered. Indeed, inverse transition corresponds to a configuration in which an agent moves at distance larger than the alert distance from each member of the team. In this case, transitions in the relaxed graph involves and modifies only two teams of the starting node. In particular, after a transition two teams are merged in the same team. In the following, we refer to transition form state I_{jk} to state $I_{(j+1)m}$ as j-th level transition. Notice that in a N agents scenario there are $N - 1$ levels of transitions in the relaxed graph. In general, at $j - th$ transition level with $j \leq N$, the nodes are characterized by the following teams and elements: $[a_i]$ for $i = 1, \ldots, k + 1$, and

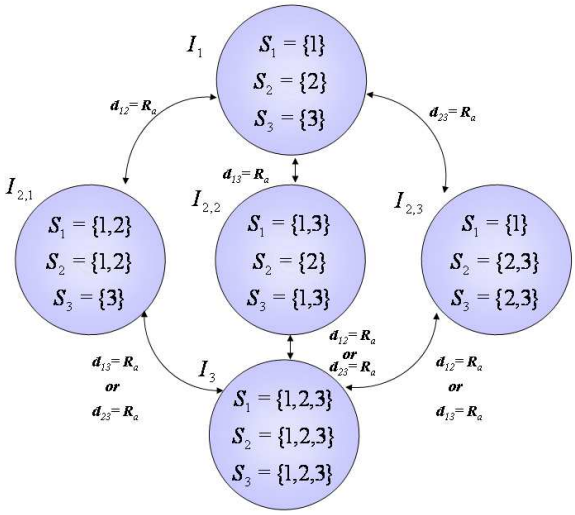


Figure 1.5: Decentralized transitive scheme with three agents. Notice that nodes I_5, I_6, I_7, and I_8 of the non-transitive scheme in figure 1.4 coincide here in a single node I_3.

$N-(j+k)$ where $a_i \neq 1$, $k = 0, \ldots, min\{j-2, N-j\}$ and $\sum_{i=1}^{k+1} a_i = j+k$. In figure 1.6, hybrid systems for $N = 4$ and $N = 5$ are exploited.

Clearly, the decrease in computational complexity of decentralized problems (when the connectivity is limited) can be such as to allow real-time implementation by simple networks and embedded controllers, and introduce a large degree of redundancy which can greatly reduce malfunctioning risks. However, a big issue with decentralized schemes is that switching among different modes can lead to situations where no feasible solution exists. Hence, even when an optimal solution to all decentralized problems is separately available, proving that executions of the corresponding hybrid system are *well posed*, *safe* and *convergent* (i.e., that dead-locks and livelocks are avoided) is quite a challenging problem.

Non transitive schemes tend to amplify both advantages and disadvantages of

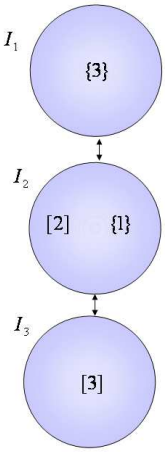


Figure 1.6: The associated relaxed graph to the decentralized transitive scheme reported in figure1.5. Notice that only the number of teams and singelton are considered.

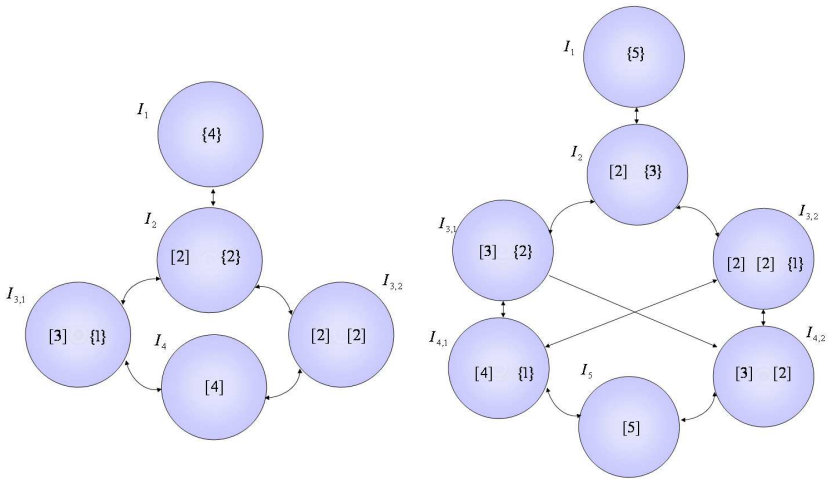


Figure 1.7: The decentralized transitive scheme for $N = 4$ and $N = 5$ agents.

decentralization. A simulative study reported in [8] has shown the increased robustness of decentralization with respect to failures in the decision making processes. An analytic study of safety of the equivalent hybrid system (for the linear model of (3)) has been presented in [21].Generalizations to more agents appear to be either overconservative, or complex.

Chapter 2

A decentralized policy for collision avoidance of MAS

In this chapter the problem of collision avoidance of a multi-agent system is studied. A definition of collision between two agents based on euclidean distance is first provided. Then an analysis on the choice of dynamic models adopted for agents is conducted. Finally a decentralized policy for conflict avoidance is reported along with its main properties.

2.1 Formulation of the problem

Consider n autonomous mobile agents, moving on a plane. Let the configuration of the i-th agent be described by a triple $g_i = (x_i, y_i, \theta_i) \in \mathbb{R} \times \mathbb{R} \times S^1$ where x_i, y_i are the coordinates of the center and θ_i is the heading angle. The agent is surrounded by two virtual discs, referred to as *safety* and *alert* discs, respectively of radius $R_{s,i}$ and $R_{a,i}$ (see figure 2.1). The *safety* disc represents the area where an agent claims its ownership. A *collision* between two agents i and j is said to occur if for some

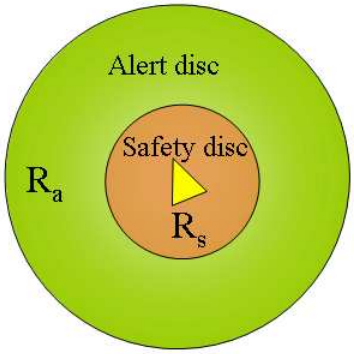


Figure 2.1: Safety and Alert discs.

value of time t, their *safety* discs overlap, or equivalently if their distance becomes less than the sum of their safety radii,

$$d_{i,j} = \|(x_i, y_i) - (x_j, y_j)\|_2 < R_{s,i} + R_{s,j}. \tag{2.1}$$

The *alert* disc represents the area within an agent can sense the presence tof other agents, referred to as *neighbours*, and exchange information with them in order to predict and resolve possible conflicts. We admit that agent i is aware of information concerning agent j if

$$d_{i,j} = \|(x_i, y_i) - (x_j, y_j)\|_2 < R_{a,i}. \tag{2.2}$$

In the following, for sake of simplicity we suppose that all agents have the same safety radius ($R_{s,i} = R_s \ \forall i \in \{1 \dots n\}$) and the same alert radius ($R_{a,i} = R_a \ \forall i \in \{1 \dots n\}$).

2.1.1 Dynamic model

Traffic coordination has been often attacked in the hypothesis that agents have rather simple dynamics, allowing them to stop rapidly to clear possible impeding

conflicts, and change direction of motion instantly, e.g., [22, 23]. This assumption is however inapplicable (or imposes conservative limits on velocities) for most practical vehicles, which have complex dynamics preventing immediate stops (as e.g. with cars or marine vessels) or not allowing stops at all (as with aircrafts).

We focus our attention on vehicles which do not have the ability to stop, backup or turn immediately. Hence we consider two simplified but realistic kinematic models for agents. The first model assumes that agents move with constant speed subject to curvature bounds. The kinematic model of the i-th agent is given by,

$$\begin{cases} x_i(t) = v_i cos(\theta_i(t)) \\ y_i(t) = v_i sin(\theta_i(t)) \\ \theta_i(t) = \omega_i(t) \end{cases} \tag{2.3}$$

where v_i and ω_i are the linear and angular velocities respectively. The maximum angular velocity is given by $\|\omega_{i,max}\| = v_i \backslash R_{c,i}^{min}$ where $R_{c,i}^{min}$ represents the minimum curvature radius (see figure 2.2). Such a model for the agent dynamics is very similar to the well-known model for car-like vehicles due to Dubins [1], with the only difference being that in our case the agents cannot vary their speed, and are therefore unable to stop (i.e, aircrafts aircrafts cruising on a planar airspace).

The second model relies on the assumption that the workspace dimensions are wide with respect to the agents linear velocity. Under this hypothesis the model (2.3) can be well approximated by one where agents are able to change instantaneously their heading of a bounded quantity. A discrete-time approximation of this model can be introduced as

$$\begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix}^+ = \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} + \begin{pmatrix} \delta_i \cos(\theta_i + p_i) \\ \delta_i \sin(\theta_i + p_i) \\ p_i \end{pmatrix}, \tag{2.4}$$

where δ_i represents the length of a forward step and p_i the heading angle change taken in a unit sampling time. Velocity and curvature constraints are implemented here by imposing bounds on $|\delta_i| \leq \hat{\delta}_b$ and $|p_i| \leq p_b$. It is important to notice

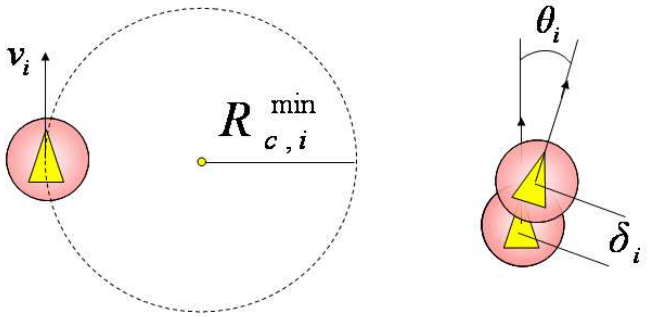


Figure 2.2: Left: agent moving at linear speed v_i with bounded curvature radius $R_{c,i}^{min}$. Right: Discrete model of an agent moving with step size δ_i and capable of streering instantaneously of a bounded quantity.

how the two models are complementary: the former (2.3) represents a smaller scale scenario with few agents, moving by closely knitted trajectories, while the latter (2.4) addresses large scale problems involving tens of agents moving in relatively large space.

In the following of this chapter we report a decentralized control policy for collision avoidance of a multi-agent system. Agents are assumed to move according to (2.4).

2.2 A decentralized policy for the conflict avoidance problem

Conflict resolution maneuvers using a simplified model allowing for bounded instantaneous changes of heading angle and velocity, have been considered in [11] and [20]. In those works authors deal with the problem of providing conditions such that, given a number of agents moving according to dynamics (2.4) there exist ma-

neuvers which ensure *safety*, i.e. avoid collisions between agents. Maneuvers have been obtained as solutions to a mixed-integer linear optimization problem (MILP), by using a centralized control scheme. In [21] a decentralized implementation of this policy has been proposed and sufficient conditions under which a 3-agents MILP-based scheme guarantees safety. This result has been extended in this thesis up to five agents. Inviting the reader to refer to [21] for details, in the following we report only main concepts.

Consider a system of n agents with dynamics 2.3. No-conflict constraints are given by non linear inequalities such as 2.1. For the purpose of safety, the problem is to find an admissible value of the steering angle p_i for agent i such that all conflicts are avoided with new heading angles $\theta_i + p_i$, for each member of the team. The construction of no-conflict constraints can be done considering agent pairwise and then combining all such conditions for all pairs of agents in the same team.

Given a pair of agents i and j, we consider the configuration of agent j with respect to agent i. Let us define the following quantities:

$$\omega_{ij} = arctan(y_j - y_i, x_j - x_i),$$

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

and

$$\alpha_{ij} = arcsin(\frac{R_s}{\rho_{ij}}),$$

With a geometrical construction and by the introduction of some boolean variables ([21]), such nonlinear constraints can be written as linear constraints in the control variables. Moreover, fixed d_{ij}, such constraints can be represented as in figure 2.4.

Let $q_{ij} = (w_{ij}, \theta_i, \theta_j, \alpha_{ij})$ and p_i, p_j the control variables. The safe set of a system of two agents in configuration (x_i, y_i, θ_i) and (x_j, y_j, θ_j) is the set of values of p_i and p_j such that $|p_i| \leq p_b$, $|p_j| \leq p_b$ and such that from configurations

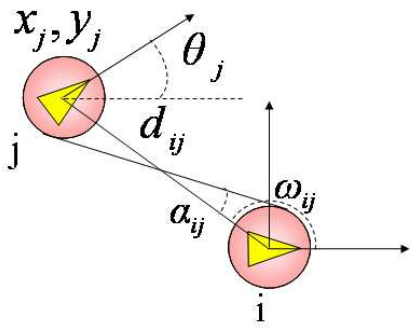


Figure 2.3: Geometrical construction of the two intersecting lines tangent to the safety discs of radius R_s for agents at distance d_{ij}

.

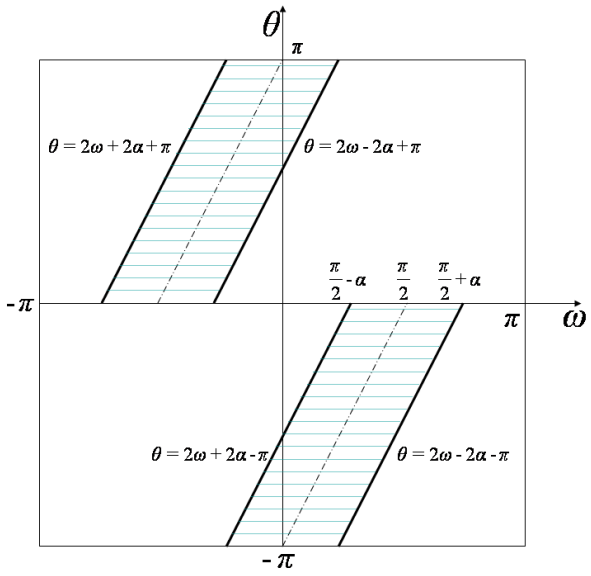


Figure 2.4: Unsafe zones: sector of the (ω, θ) for which a conflict is detected.

$(x_i, y_i, \theta_i + p_i)$, $(x_j, y_j, \theta_j + p_j)$ no conflict occurs. Referring to [21] for more details, the safe set for agents i and j can be described by a logical statement $C(q_{ij}, p_i, p_j)$ that is a set of "and" and "or" inequalities, function of q_{ij} , and linear in p_i and p_j . Choosing a linear cost function such as the 1-norm or the ∞-norm of control variables, a Mixed Integer Linear Programming problem must be solved to obtain optimal controls p_i that solve all possible conflicts ([20]). The safe set, for the pair (i, j), is thus described by $c_{ij} = q_{ij} | \exists\, p_i, p_j \in [-p_b, p_b]\,, C(q_{ij}, p_i, p_j)$. Consider a reference system with origin in the position of agent i and direction of x-axis that coincides with the direction of motion θ_i. By studying the equivalent set $\{q_{ij} \,|\, \exists\ p_{ij} \in [-2p_b, 2p_b]\,, C(q_{ij}, 0, p_{ij})\}$, where $p_{ij} = p_i - p_j$, $\omega_{ij} = \omega_{ij} - \theta_j$ and $\theta_{ij} = \theta_i - \theta_j$, the unsafe set represented in the plane $(w = w_{ij}, \theta = \theta_{ij})$ is reported hased in figure 2.4.

Consider now the width Δ_{ij} of the unsafe set band, we have that $\Delta_{ij} = 4\alpha_{ij}$ and decreases with α_{ij} . As a consequence, it decreases as the distance d_{ij} between i and j increases. The value of the bandwidth will be used in the theorem proof.

2.3 Safety of a decentralized 5-agents system

Consider configurations for which a solution of the relative MILP problem exists within each team of agents, we will refer to those as safe configurations. In other words, safe configurations are such that no conflict is detected or if a conflict is detected it is solvable with maneuvers of amplitude bounded by p_b. A transition from a state of the hybrid system to another state is a safe transition if it starts in a safe configuration and it ends in safe configurations of the new state of the system. Our aim is to compute minimum values of the alert distance to ensure safety for all possible transitions in the hybrid system.

Remark 1. Assume that a minimum alert distance has been computed for the case $N = k$, so that all transition of the associated hybrid system are safe. Consider

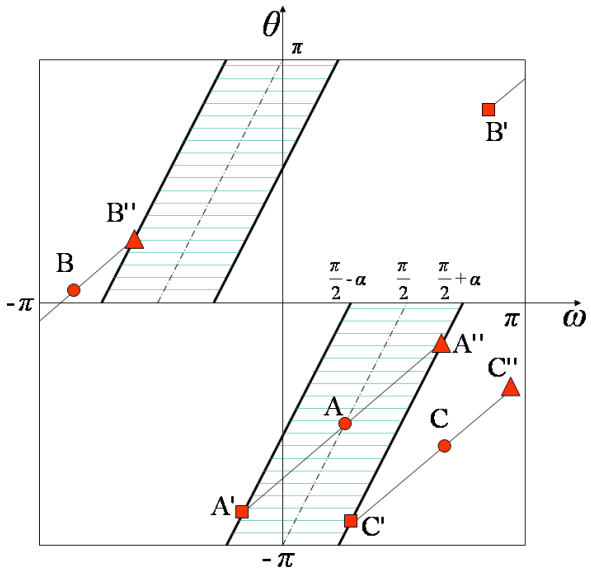


Figure 2.5: Case $N = 4$, transition from $([3], 1)$ to $[4]$.

the case $N = k + 1$, all i-th level transitions with $i < k$ are safe (safety conditions on the alert distance have already been obtained in the case $N = k$). For example, based on results obtained for $N = 3$, for case $N = 4$ only transition of type $T_{31} : ([3], 1) \rightarrow ([2], [2])$ and $T_{22} : ([2], [2]) \rightarrow ([4])$ must be exploited (see figure 1.7, left). In the following, we propose conflict resolution maneuvers in the worst cases of all transitions in the hybrid systems. Our purpose is to provide an admissible maneuver for the worst-case transitions of the hybrid system, thus proving its safety. Optimal maneuvers (with respect to the relative cost function) are computed by agents in the same team by solving a MILP problem.

Theorem. Consider N agents with $N \leq 5$ with safety distance d in a common workspace such that initial relative distances are larger than the alert radius R_a or such that they are in a safe configuration of a node of the relaxed graph. Consider

the upper bound on possible instantaneous heading angle changes as p_b. If $R_a \geq d/sin(p_b/10)$ then each transition that can occur in the hybrid system is safe (i.e. for each transition there exist admissible maneuvers solving conflicts).

Proof: Based on remark 1, we first give conditions on safety for 3 and 4 agents and finally for 5 agents taking into account only 2nd 3rd and 4th level transitions respectively.

Case $N = 3$: in [21] safety has been demonstrated for the decentralized cooperative non-transitive scheme in the $N = 3$ case. The obtained alert distance that ensure safety transitions is $R_a = d/sin(p_b/4)$ and depends on the safety distance and the bound of admissible controls. A similar demonstration can be applied to the transitive scheme obtaining, for the $N = 3$ case, the same value of the alert distance. Demonstration is omitted for space limitations.

Case $N = 4$: consider the worst case for the first transition, all three agents of [3] (named agents A, B, C) are at the minimum distance R_a with respect to 1 (named agent 1), in this case we have $\alpha_{1A} = \alpha_{1B} = \alpha_{1C} = \alpha$. This is a worst case since we have supposed that only two agents (e.g. agent 1 and A) can be at distance R_a at each time t, hence immediately after the transition we have $A_{1B} > R_a$ and $A_{1C} > R_a$. Therefore, $\alpha_{1A} = \alpha > \alpha_{1B}$ and $\alpha > \alpha_{1C}$. Hence, the unsafe sets of pairs $(1, B)$ and $(1, C)$ would be smaller than the unsafe set of $(1, A)$. Since transition T_31 starts from a safe configuration, conflicts within team [3] have already been solved. In order to do not generates other conflicts, we do not want agents of team [3] to maneuver. We will assume that the maneuver will be done by agent 1. Let then consider the non collision constraints in coordinates relative to agent 1, (see figure 2.4). Assume that, after the transition, agent 1 detect a conflict with agent A, the worst case (reported in figure 2.5) is when the minimum maneuver for 1 to avoid the conflict with A generates a conflict with B or C. If a positive deviation is done by agent 1 of amplitude 2α then agents A, B, C will move in configuration A', B', C', while the conflict with A is solved there is a new conflict with C. Otherwise, if a

right deviation is done by 1 then agents A, B, C will move in configuration A'', B'', C'', while the conflict with A is solved a conflict with B is detected. For example, let agent 1 to maneuver with $p_1 = 2\alpha$, in order to solve the generated conflict with C another maneuver of amplitude 4α would be needed for agent 1. Hence, in the worst case a singular maneuver of amplitude 6α solves all conflicts. The chosen maneuver is admissible if $6\alpha \leq p_b$, and the transition is safe if $R_a \geq sin(p_b/6)$. Regarding transition T_{22} : $[2], [2] \rightarrow [4]$, agents of the two teams are named 1, 2 and A, B respectively. The worst case occurs when agent 1 is at distance R_a from A and a conflict occurs, between 1 and A, such that a maneuver of amplitude $\pm 2\alpha_{1A}$ is needed by agent 1 to solve the conflict with A (or by agent A to solve the conflict with 1). In worst case both maneuvers generate conflicts between agents 1 and 2. This happens when agent 2 has $w = \pi$ and $\theta \leq 2\alpha_{1A}$, both w and θ in coordinates relative to agent 1. If this is the case, we let maneuver agent A, instead of agent 1 of amplitude $2\alpha_{1A}$ or $-2\alpha_{1A}$. In worst case also this two maneuvers are such that a conflict between A and B is generated. This worst case configuration is reported in figure 2.5 in the coordinates relative to agent 1. Larger unsafe sets are for agent 2 ($\alpha_{12} = \pi/2$) while smaller ones are for A and B agents. Referring again to figure 2.6, a conflict avoidance maneuver will consist in let both 1 and 2 maneuver with amplitude at worst $+6\alpha_{1A}$. With respect to agent 1 this maneuver produce a diagonal displacement for agent A and B and an horizontal displacement of 6α in the (w, θ) plane without generating other conflicts, see figure 2.7.

Concluding, in worst case the transition T_{22} is safe if $R_a \geq d/sin(p_b/10)$. Once transitions of $N = 4$ case are safe, regarding $N = 5$ case, transition that need an exploitation are $T_{41} : I_{41} \rightarrow I_{51}$ and $T_{32} : I_{42} \rightarrow I_{51}$, (see figure 1.7).

Case $N = 5$: consider the worst case for transition T_{41}, from $([4], \{1\})$ to $[5]$. This is similar to transition T_{31} of the $N = 4$ case, with respect to the $N = 4$ case, in addition there is another agent D at distance $A_{1D} = R_a$. As we have shown previously, with a maneuver of amplitude 4α (for agent 1) conflicts between agent

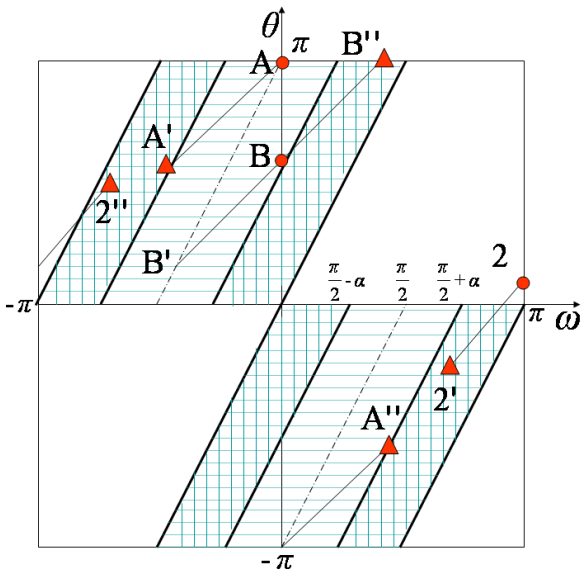


Figure 2.6: Case $N = 4$, transition from $([2], [2])$ to $[4]$, in coordinates relative to agent 1.

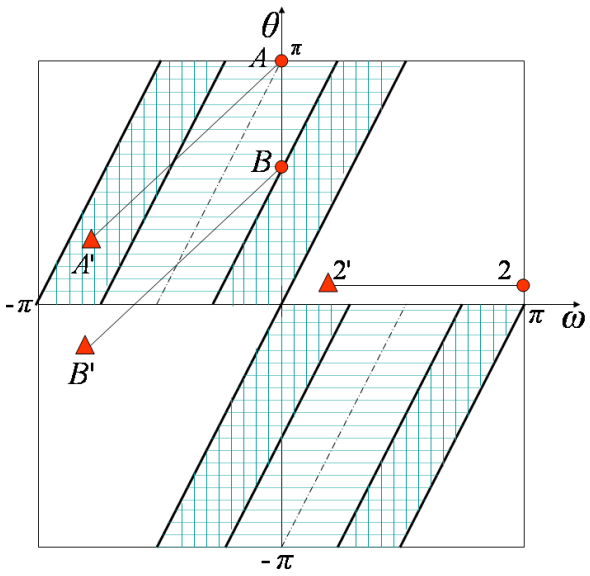


Figure 2.7: Case $N = 4$, transition from $([2], [2])$ to $[4]$, in coordinates relative to agent A.

1 and agents A, B and C are solved. The worst case is when with such maneuver a new conflict between 1 and D is detected. Hence, a total maneuver of amplitude 8α solves all conflict of 1 with A, B, C and D. Concluding, transition is safe if $R_a < p_b/8$ or equivalently if $R_a \geq d/sin(p_b/8)$. Transition T_{32} from $([3],[2])$ to $[5]$ is similar to transition T_{22} of the $N = 4$ case. The worst case is the same we reported for T_{22} (see figure 2.6), in addition there is another agent C such that once conflicts between 1 and A and B are solved, a new conflict between 1 and C is detected. In order to solve also this conflict a total maneuver of amplitude 10α is needed by both 1 and 2. Concluding, transition is safe if $\alpha < p_b/10$ or equivalently if $R_a \geq d/sin(p_b/10)$. Concluding, the most restricting condition on the alert distance obtained in the proof is $R_a \geq d/sin(p_b/10)$ that proves the theorem. To give an idea on the lower bound obtained to the alert distance, if admissible maneuvers are of amplitude smaller than $p_b = 0.35rad$, for $N \leq 5$ agents it is sufficient to impose an initial relative distance larger than the alert distance $R_a \geq 28.6d$ to ensure safety for every transition that can occur. For example if $d = 10$cm it is sufficient to impose an initial relative distance larger than 3m.

Chapter 3

The Generalized Roundabout Policy

The policy introduced in the previous chapter has some drawbacks. The assumption made on the dynamics of agents might not be acceptable when the agents' speed is high with respect to the size of the environment. Furthermore the policy is hardly scalable to a large number of agents, and the effort to be done in order to extend the results to few more units is remarkable. Here it is introduced a new spatially decentralized policy which aims at avoiding conflicts between a large number of agents and guaranteeing, under certain assumptions, that each agent may reach its final goal starting from an initial configuration. Agents are assumed to move according to (2.3), modeling a scenario where aircrafts moving at cruising speed, but is also suited for some earthly applications with unicycle-like vehicles.

3.1 Problem Formulation

Let the configuration of agent i be defined by coordinates $g_i = (x_i, y_i, \theta_i)$. Each agent enters the environment at the initial configuration $g_i(0) = g_{0,i}$ and is assigned a final goal $g_i(f) = g_{f,i}$. Agents move along a continuous path $g_i : \mathbb{R} \to \mathbb{R} \times \mathbb{R} \times S^1$ according to equations

$$\begin{cases} x_i(t) = v_i cos(\theta_i(t)) \\ y_i(t) = v_i sin(\theta_i(t)) \\ \theta_i(t) = \omega_i(t) \end{cases}, \tag{3.1}$$

where v_i is the linear speed, assumed constant, unitary and the same for all the agents ($v_i = 1, \forall i = \{1 \dots n\}$), and $\omega_i \in [-1, 1]$ is the angular velocity, that represents the control bounded input. This choices implies that the minimum steering radius for each agent results

$$R_{c,i}^{min} = R_c = 1 \forall i \in \{1 \dots n\}$$

We assume that agents have the same safety radius,

$$R_{s,i} = R_s, \ \forall i \in \{1 \dots n\};$$

hence according to (2.1) a collision between agents i and j occurs if

$$d_{i,j} < 2\, R_s.$$

We want to define a spatially decentralized control policy such that each agent i may reach a final configuration avoiding possible collision, relying only on its current and final configuration and on its informative structure $S_i(t)$. Informative structure is supposed to be symmetric and non-transitive and to contain only the current configuration of agents within the alert radius $R_{a,i}$ (see Cap. 2). In other words, agent i makes decisions based on the position and orientation of its neighbours, neglecting other informations such as their target configurations. We suppose that agents have the same alert radius

$$R_{a,i} = R_a \forall i \in \{1 \dots n\},$$

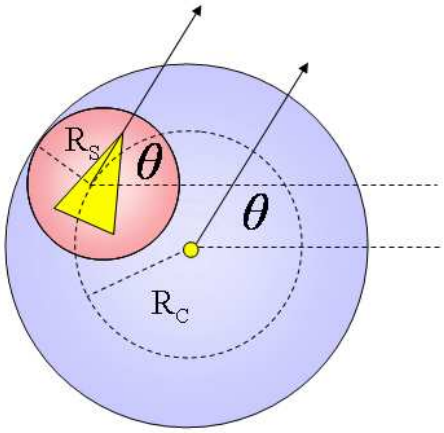


Figure 3.1: The reserved disc.

whose size of R_a, which, as previously mentioned, deeply influences the level of decentralization, will been chosen according to some issues of the policy, as it will be reported below along with some preliminary definitions.

The Reserved Disc. The hinge of the Generalized Roundabout policy is the concept of *reserved* disc ([24]), which is a circle of radius $(R_s + R_c)$ over which an agent claims its exclusive ownership. Given the configuration g_i of agent i, the configuration of its associated *reserved* disc is $g_i^c(x_i^c, y_i^c, \theta_i^c)$, where $(x_i^c, y_i^c) = (x_i + sin\theta_i,\ y_i - cos\theta_i)$ is the center of the disc, and $\theta_c^i = \theta^i$.
The dynamics of the reserved disc is given by:

$$\begin{cases} \dot{x}_i^c(t) = (1 + \omega_i(t))cos(\theta_i(t)) \\ \dot{y}_i^c(t) = (1 + \omega_i(t))sin(\theta_i(t)) \\ \dot{\theta}_i^c(t) = \omega_i(t). \end{cases} \tag{3.2}$$

According to (3.2), the input $\omega_i = -1$ determines an immediate stop of the reserved disc. Moreover it can be moved in any direction, provided one waits long enough

for the heading θ_i to reach the appropriate value. As a consequence, for example, the reserved disc can be assimilated to a holonomic vehicle than can follow any continuous path, unlike model (2.3).

Admissible Cone. A sufficient condition to ensure *safety* is that the interiors of reserved regions are disjoint at all times. In fact, if such a condition is met, also the safety discs of agents do not overlap. Let J_i denote the set of indices of agents whose reserved discs are in contact with the reserved disc of agent i. Each contact determines the following constraint to the motion of agent i

$$\dot{x}_i^c\,(x_i^c - x_j^c) + \dot{y}_i^c\,(y_i^c - y_j^c) \geq 0 \quad \forall j \in J_i. \tag{3.3}$$

In other words the velocity of the i-th reserved region is constrained to remain in the convex cone, referred to as *admissible* cone and denoted by Θ_i, determined by the intersection of a number of closed half-planes. Let us define $\max(\Theta_i)$ and $\min(\Theta_i)$ as the boundaries of Θ_i, respectively in the positive and negative direction with respect to the bisectrix of Θ_i (see fig. 3.2). Finally, define $\Theta_i^- = \Theta_i \setminus \min(\Theta_i)$.

In order to guarantee that the *reserved* discs of any two agents do not overlap, it must be provided that each agent has to be aware of the configuration of all surrounding agents within $R_a \geq 4\,(R_c + 2\,R_s)$ (fig. 3.3, left). The choice of $R_a = (4\,R_c + 2\,R_s)$ bounds the amount of information needed to each agent, independently from the total number of agents in the system: in fact, the maximum number of agents whose reserved region is in contact with the reserved region of the computing agents is six (fig. 3.3, right). This provides high scalability to the GRP policy.

It is to be noticed that the minimum value for safety radius is theoric and relies on the assumptions that communications between agents are performed without delay and that the control can be computed instantaneously by each agent. In practical applications, alert radius must be taken greater than that value in order to cope with communication delays and finite time computations.

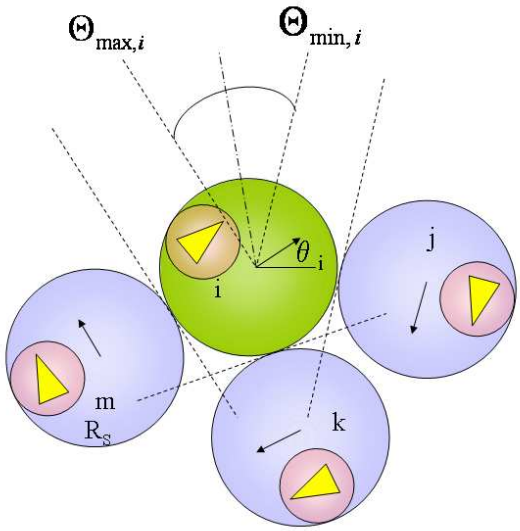


Figure 3.2: Admissible Cone.

Straight. When $\omega = 0$ agent moves `straight` along with its reserved disc. In such case we say the agent is in the `straight` mode.

Holding. As previously said, setting $\omega = -1$ causes an immediate stop of the reserved disc. When $\omega = -1$ agent is said to be in the *`hold`* mode.

Clockwise-only steering. If the reserved disc of an agent never becomes in contact with the reserved disc of other agents, it can reach its target by switching between the `hold` and `straight` states only . The proof can be given constructively. Let g_0 and g_f be initial and final configurations of the agent. Let ϕ be the angle between the line joining the center of the reserved disc of an agent with its target, and the horizontal axis. The agent stays in the `hold` mode until $\theta = \phi$; then it switches to the `straight` mode until $d(g_c, g_{c,f}) = 0$. Once the centers of current and target reserved discs coincides, the agent switches back to the `hold` mode until

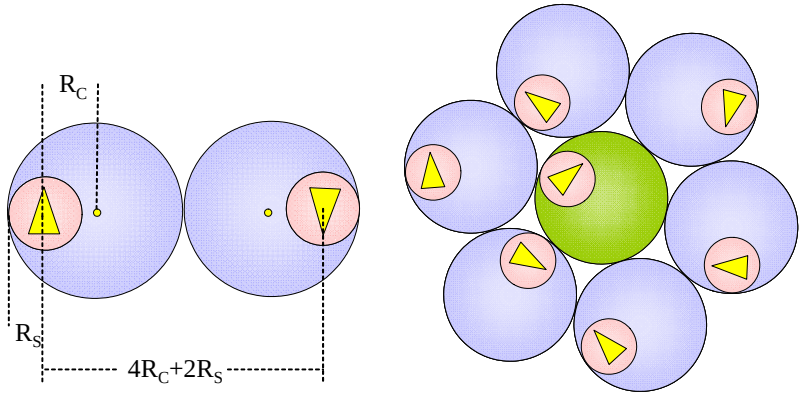


Figure 3.3: Left: worst case for the computation of the minimum alert radius. Right: the maximum number of agents in contact with the computing agent is six and does not depend form the total number of agents in the system.

its final configuration is reached (see fig. 3.4). This policy can be summarized as

$$\omega = \begin{cases} \texttt{straight} & \text{if } d(g_c, g_{c,f}) \geq 0 \text{ and } \theta = \phi, \\ \texttt{hold} & \text{otherwise.} \end{cases} \tag{3.4}$$

The policy generates Dubins-like paths, except that only clockwise steering is admitted. The choice of steering agents only in the clockwise direction brings the policy away from optimality in terms of length of the path to reach the target. Nevertheless, optimality is not an issue for our policy, which aims to be "safe" and "live" as it will be reported in the following.

Rolling on a stationary reserved disc. If the path of the reserved region to its position at the target is blocked by another reserved region, agent can choose an input such that its reserved disc may `roll` in the positive direction on the boundary of the blocking region. Since in our setup agents communicate only information on their states, not on their future intentions, care must be exercised in such a way that the interiors of reserved regions remain disjoint. Let us start by assuming that

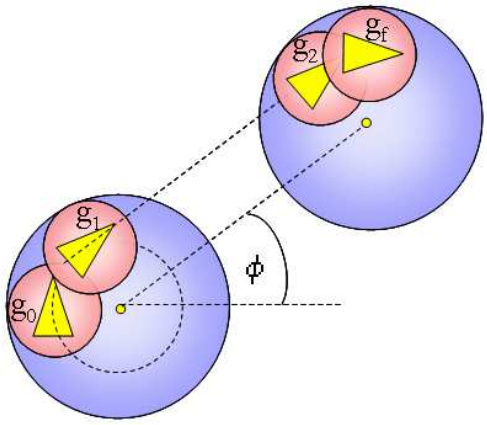


Figure 3.4: Right-only steering. Agent is initially in configuration g_0. By setting $w = -1$ it reaches configuration g_1 where $\theta = \phi$. Agent switches then in `straight` mode until $d(g_c, g_{c,f}) = 0$. At this moment agent, which is in configuration g_2 sets $\omega = -1$ thus reaching its final configuration g_f.

the reserved region of the neighboring agent remains stationary; in order to `roll` on such region, without violating safety constraints, the control input must be set to

$$\omega = \begin{cases} (1 + R_S)^{-1} & \text{if } \Theta^- \neq \emptyset \text{ and }, \theta = max(\Theta^-) \\ -1 & \text{otherwise.} \end{cases} \tag{3.5}$$

The above policy is obtained by switching between the `hold` state and a `roll` state; note that when in the `roll` state, the agent is not turning at the maximum rate.

Not stationary reserved disc. In general, the reserved region of an agent will not necessarily remain stationary while an agent is rolling on it. While it can be recognized that the interiors of the reserved regions of two or more agents executing (3.5) will always remain disjoint, it is possible that contact between two agents is lost unexpectedly (recall that the control input of other agents, their constraints, and their targets, are not available). In this case, we introduce a new state, which we call " ", in which the agent turns in the positive direction at the maximum rate, i.e., $\omega_i = +1$, unless this violates the constraints. The rationale for such a behavior is to attempt to recover contact with the former neighbor, and to exploit the maximum turn rate when possible. The `roll2` state can only be entered if the previous state was `roll`.

Generalized Roundabout policy. Accordingly to definitions given above, we can model each agent as a finite-state hybrid automa where the continuous part is represented by the four modes of operation, `straight`, `hold`, `roll` and `roll2` obtained by applying respectively inputs $w = 0$, $w = -1$, $w = (1 + R_S)^{-1}$, and $w = +1$, while the discontinuities are represented by switchings between these modes, ruled by conditions determined by the current configuration of the agent, its final destination and the configurations of agents whose reserved disc are tangent to the reserved disc of the considered agent. Reporting explicitly the GR policy and its guards, would be long. For the sake of clarity we report in fig. 3.5 a graphical

representation of the finite-state hybrid automa with its four modes of operation and its switching conditions.

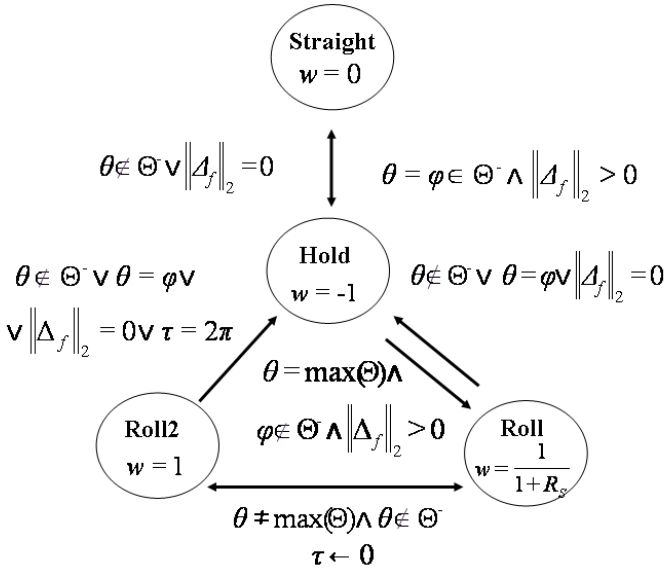


Figure 3.5: A hybrid automaton describing the Generalized Roundabout policy.

3.2 Analysis

In this section, we will analyze the properties of the closed-loop hybrid system GR defined in the previous one.

3.2.1 Well-posedness

Verifying that GR it is a well posed dynamical system, implies to guarantee that a solution exists and is unique, for all initial conditions within a given set. Indeed,

Theorem. *The hybrid system GR is well posed, for all initial conditions in which*

the interiors of reserved disks are disjoint, i.e., $d(g_{c,i}, g_{c,j}) \geq 2(1 + R_s)\ \forall\, i, j\ \in \{1, ..., n\}$.

Proof: The hybrid system represented in fig. 3.5 is globally Lipschitz in the state and in the control input; moreover, control inputs are constant within a discrete mode. The parallel composition of n copies of the continuous dynamics 3.5 is also globally Lipschitz. Hence, in order to establish well posedness of it is sufficient to show that there is no accumulation point of switching times, i.e., the number of switches in an open time interval is bounded, and the control input signal ω is piecewise continuous. First of all, note that the number of instantaneous switches is bounded by three: the specification of invariant conditions in figure 3.5 prevents infinite loops without time advancement. This can be verified by inspection of the invariants. Let t_0 denote the time at which a switch in the discrete state has occurred, we need to show that there exists a $t_1 > t_0$ such that there are no switches in the open interval (t_0, t_1). For simplicity, assume that the discrete state at time t_0 is the terminal state of the sequence of instantaneous switches occurring at t_0. In the following, we will consider the i-th agent, and compute bounds on the time separation between switches, based on the current state of all agents. We have the following cases:

Case 1: $q_i(t0) =$ `hold`. A switch can be triggered by the following:

- $d(g_{c,i}, g_{c,f}) = 0, \quad \theta_i = \theta_{f,i}$: The agent reaches its final configuration, and is removed from the system.
- $d(g_{c,i}, g_{c,f}) > 0, \quad \theta_i = \phi_i \in \Theta_i^-$: the agent transitions to the discrete state `straight`.
- $d(g_{c,i}, g_{c,f}) > 0, \quad \theta_i = max(\Theta_i)$: the agent transitions to the discrete state `roll`.

None of the three above events can occur in the time interval $(t_0, t_0 + \delta_{1,i})$, with $\delta_{1,i} = min\ \{\delta_{f,i} - \delta_i, \phi_i - \delta_i, max(\Delta_i) - \delta_i\}$; the angle differences are meant to be

counted in the direction of angular motion of the agent, modulo 2π.

Case 2: $qi(t_0) = \texttt{straight}$. A switch can be triggered by the following:

- $d(g_{c,i}, g_{c,f}) = 0$: the reserved disk has been steered to its final configuration, and the agent transitions to the discrete state `hold`.

- $\phi_i \notin \Theta_i^-$: a new constraint on the motion of the reserved disk is activated, as the consequence of a contact with another agent's reserved disk.

Neither of the two above events can occur in the time interval $(t_0, t_0 + \delta_{2,i})$, with $\delta_{2,i} = \min\{||\Delta_{f,i}||_2, min_{j \neq i} \{||c(g_i) - c(g_j)||_2 - 2(1 - R_s)\}\}$.

Case 3: $q_i(t_0) = \texttt{roll2}$. A switch can be triggered by events that have already been considered above, plus the time-out condition $\tau < 2\pi$. Hence no switches can occur in time interval $(t_0, t_0 + \delta_{3,i})$, where $\delta_{3,i} = \min\{\delta_1, \delta_2, 2\pi - \tau\}$.

Case 4: $q_i(t_0) = \texttt{roll}$. This is the only delicate case, as instantaneous transitions can be triggered by other agents' actions. Let us indicate with j the index of the agent generating the constraint corresponding to $max(\Theta_i)$. If $q_j = \texttt{hold}$, then the invariant $\theta_i = max(\Theta_i)$ is preserved as the reserved disk of the i-th agent rolls on the reserved disk of the j-th agent; switches can be triggered by events considered above. If $q_j \neq \texttt{hold}$, the reserved disks of the two agents will detach at time zero-thus triggering a transition of the discrete state of the i-th agent to `roll2`; however, since the motion of the j-th agent is constrained by agent i, in such a way that the envelope of the reserved disk of agent j forms an angle $\chi_{ij} > 0$ (since Θ_j^- has been defined as an open set), the time at which the next switch can occur in this case is no sooner than $t_0 + 2sin(\chi_{ij}/2)$. Hence, an additional switch cannot happen in the interval $(t_0, \delta_{4,i})$, with $\delta_{4,i} = min\{\delta_2, (\phi_i - \delta_i)(1 + R_s), 2sin(\chi_{ij}/2)\}$. Summarizing, for the whole system, if t_0 is a switching time for at least one of the agents, no other agents can switch within the interval $(t_0, t_0 + \delta)$, where $\delta = min\{\delta_{1,i}, \delta_{2,i}, \delta_{3,i}, \delta_{4,i}\} > 0$.

3.2.2 Safey

Theorem. *For all initial conditions for which the interiors of the agent's reserved discs are disjoint, i.e.,* $d(g_{c,i}, g_{c,j}) \geq 2(1 + R_s)\ \forall\, i, j\ \in \{1, ..., n\}$*, the GR policy is safe, that is,* $\forall\, t \geq 0,\ d(g_i(t), g_j(t)) > 2\, R_s\ \forall\, i, j\ \in\ \{1, ..., n\}\,,\ i \neq j$.

Proof: The proof of the theorem follows directly from the fact that trajectories $g_i(t)$, $i = 1, ..., n$ are continuous functions of time. Moreover, within each state the feedback control policy has been chosen so that reserved discs never overlap: a transition is always enabled to the `hold` state, which stops the reserved disk instantaneously. Since the agents are always contained within their reserved disk, at a distance R_s from its boundary, safety is ensured.

3.2.3 Liveness

Safety is not sufficient for guaranteeing that agents reach the intended destination in finite time. We want to verify if the GR policy has the livenss property, i.e, the capability of negotiating a solution in finite time. Liveness is a big issue for multi-agent systems where lots of interdependent actions must be taken simultaneously into account. Though providing a general condition which guarantees liveness for the GR policy is still an open problem, it is possible to give sufficient conditions on the target location for liveness in the simple case $n = 2$.

Theorem. Consider two vehicles such that the center of the reserved disc in final configurations are at distance larger than $4(1 + R_s)$. The GR policy allows the vehicles to reach their final destinations in finite time, from all initial conditions such that the interiors of the reserved disks are disjoint.

Proof: If the reserved disks of the two vehicles do not touch each other the two vehicles will reach their goal with the sequence of controls $\omega = -1$, $\omega = 0$, $\omega = -1$. Otherwise, when a contact between the reserved discs occurs, we have six different cases:

Case 1: $q_1 = q_2 =$ `straight`.

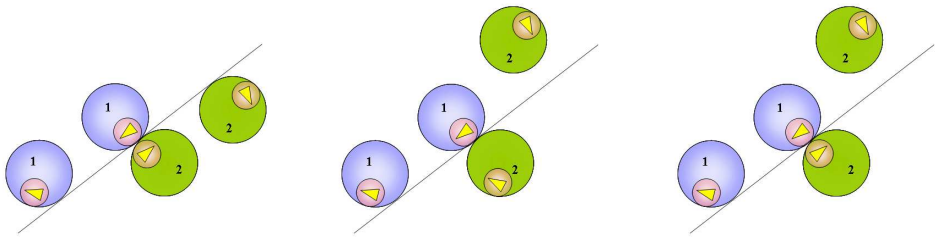


Figure 3.6: Three possible situation for two agents with reserved discs in contact, agent 1 is such that q_1 = `straight`. On the left q_2 = `straight`, in the middle q_2 = `hold`, on the right q_2 = `roll`.

Case 2: q_1 = `straight`, q_2 = `hold`.

Case 3: q_1 = `straight`, $q2$ = `roll`.

Those cases, reported in figure 3.6 are such that the contact will be immediately lost. In the first case no other contacts will be generated and the goals will be reached with a sequence of transitions `straight`, `hold` for both vehicles. In the second case the first vehicle will reach its final destination with a sequence `straight`, `hold`, while the second one will maintains control `hold` until it is no longer blocked by the first vehicle, and can move towards its goal; the reserved disks will no longer touch. In the third case, the second agent will transition to the `roll2` state as soon as contact is lost. The reserved disks will not touch again. The second agent will reach its final destination with a sequence `roll2`, `hold`, `straight`, `hold`, or `roll2`, `straight`, `hold`, depending on the initial and final configurations.

Case 4: $q_1 = q_2$ = `hold`.

Case 5: q_1 = `hold`, q_2 = `roll`.

It is sufficient to discuss the second case, since if both vehicles are in state `hold` they will reach a configuration that is equivalent to the second case unless one of them can move through its final configuration without contacts of the reserved disks. If this occurs, one of the vehicles will be in state `straight` and this is the **case 2**

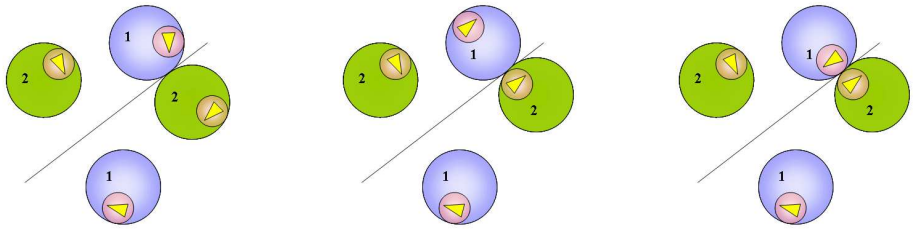


Figure 3.7: Three possible situation for two agents with reserved discs in contact, on the left and in the middle, agent 1 is such that $q_1 =$ `hold`, while agent 2 $q_2 =$ `hold` and $q_2 =$ `roll`. On the right $q_1 = q_2 =$ `roll`.

discussed above. In the second case the second vehicle will turn on the left so that the second reserved discs will slide along the first one until one of the two vehicle are able to move through the goal or they reach the configuration of **case 6** (that will be discussed below).

Case 6: $q_1 = q_2 =$ `roll`.

In this case the contact will be lost immediately, and both vehicles will switch to `roll2`; reserved disks may touch again. If a new contact occurs, the point of contact between the reserved discs has moved counterclockwise in the first vehicle's frame and clockwise on the second one. After this new contact both vehicles are in the `hold` state. If one of the vehicle can move through its final configuration by switching to the `straight` state, the configuration is equivalent to **case 2**. Otherwise this procedure is repeated. But after enough time if the distance between target configurations is larger than $4(1 + R_s)$, one of the two vehicles will be able to move through its goal since at least one of the goals is not covered by the cluster movements. In this case for one vehicle $\omega = 0$ and the configuration is equivalent to one of the previous cases. A similar proof can be provided for the three-vehicle case, but is not reported here since it does not provide any additional insight into the problem, and is quite laborious.

3.3 Probabilistic verification of liveness

In the previous section we proved for the simple case of $n = 2$ agents that the GR policy ensures liveness if the reserved discs of all agents in the initial configuration are disjoint and final configurations are at distance larger than $4(1 + R_s)$.

Unfortunately, it appears overwhelmingly complex to develop an inductive step to extend the result to a general number of agents, or provide a proof that is valid in the general case. As it resulted from considerations on the policy and from some simulation results, the liveness property of GR policy hinges upon the spatial distribution of the final configurations of agents. Intuitively, targets are required to be sparse in the workspace: if some targets are too close, the associated agents can hamper each other while attempting to reach their destinations.

Nevertheless the target configuration is not the only responsible for determining liveness. In fact, given a final configuration, a livelock may occur depending also on the set of initial configurations of agents. In fact, if we consider a scenario where agents leave the workspace, once they have reached their targets, i.e. aircrafts in the proximity of airports, it could happen that according to some initial conditions two agents may arrive to their targets in different instants. In this case they do not disturb reciprocally for reaching their goals. Otherwise, if the evolution of the whole system is such that they reach the target area simultaneously, they may generate a "livelock", i.e., a situation where they repeat indefinitely a sequence of commands, without reaching their targets. In this terms the problem is untractable, because it is quite complex to manage a-priori the evolution of the whole system if some parameters are not fixed. Before proceeding it results necessary to define properly the problem we want to solve.

Let us consider a framework in which new agents may issue a request to enter the scenario at an arbitrary time and with an arbitrary "flight plan" consisting of an initial and final configuration. In this case, it is important to have conditions to efficiently decide on the acceptability of a new request, i.e. whether the new

proposed plan is compatible with safety and liveness of the overall system. The decision whether a new flight plan is admissible may be made by a centralized decision maker, based only on information on the current and final configurations of all agents. In such case the overall control scheme is then partially decentralized since a central authority partially coordinates agents which perform real-time collision avoidance in a strictly decentralized control scheme, however.

The problem of certifying the admissibility of a requested plan can be dealt by decoupling the safety and liveness aspects of current and final configurations. Indeed, for a given policy π, consider the two properties:

$\mathbf{P_1}$: A configuration set $G = \{g_i,\ i = 1, \ldots, n\}$, is *unsafe* for the policy π if there exists a set of target configurations $G_f = \{g_{f,i},\ i = 1, \ldots, n\}$ such that application of π leads to a collision;

$\mathbf{P_2}$: A target configuration set $G_f = \{g_{f,i},\ i = 1, \ldots, n\}$, is *blocking* for the policy π if there exists a set of configurations $G = \{g_i,\ i = 1, \ldots, n\}$ from which the application of π leads to a dead- or live-lock.

A plan $(G(t), G_f)$ is *admissible* if it verifies the predicate $\neg\mathbf{P}_1\left(G(t)\right) \wedge \neg\mathbf{P}_2(G_f)$. A simple test to check the first property is provided by the following proposition.

Property $\mathbf{P_1}(\mathbf{G})$ is verified for the GR policy if and only if the reserved disks of at least two agents in G overlap.

Proof: The proof is a straightforward extension of the safety results provided in the previous section. The analysis of property P_2 is more complex, and hinges upon the definition of a condition concerning the separation of reserved discs associated with target configurations. Let $G_c^f = \{g_{f,i}^c,\ i = 1, ..., n\}$ denote the set of configurations of the reserved discs corresponding to G_f, and $P_f^c = \{(x_{f,i}^c, y_{f,i}^c),\ i = 1, ..., n\}$ be the set of their center coordinates.

Sparsity Condition

Consider a system of n agents whose final configurations are distributed such that for all $(x, y) \in \mathbb{R}^2$ and for $m = 2, \ldots, n$,

$$\text{card}\{(x^c_{f,i}, y^c_{f,i}) \in P^c_f : \|(x^c_{f,i}, y^c_{f,i}) - (x, y)\|_2 < \rho(m)\} < m, \tag{3.6}$$

where

$$\rho(m) = \begin{cases} 2(1 + R_S) & \text{for } m \leq 4, \\ (1 + \cot(\frac{\pi}{m}))(1 + R_S) & \text{for } m \geq 4. \end{cases} \tag{3.7}$$

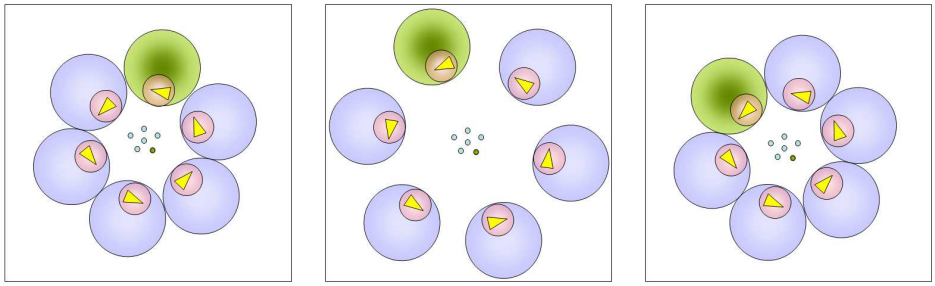
In other words, any circle of radius $\rho(m)$, with $1 < m \leq n$, can contain at most $m - 1$ reserved disk centers of targets.

In the following there are reported examples that show possible blocking situations, if final configurations do not satisy the sparsity condition (3.6).

Let $\widehat{m} \geq 2$ denote the maximum cardinality of subsets of P^c_f that violate the sparsity condition (3.6), and let $P^c_{f,\widehat{m}} \subset P^c_f$ denote one such subset. Take initial conditions for the $n - \widehat{m}$ agents corresponding to $P^c_f \setminus P^c_{f,\widehat{m}}$ to coincide with their respective targets.

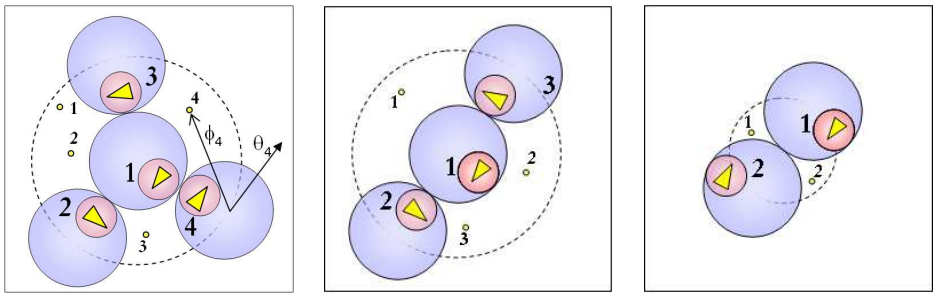
Case $\widehat{m} \geq 5$. Consider the smallest circle containing $P^c_{f,\widehat{m}}$ and the concentric circle $C_{\widehat{m}}$ of radius $\rho(\widehat{m}) - (1 + R_S)$. Take initial conditions for the $\widehat{m}$ agents such that their reserved discs are centered on $C_{\widehat{m}}$ and head in the tangent direction (see figure 3.8-a). By applying the GR policy to this configuration, the $\widehat{m}$ agents start and stay in `hold` mode until they all reach $\theta_i = \max(\Theta^-_i)$ and switch to the `roll` state. Immediately after the switch, contact between agents is lost, and all switch to `roll2` (figure 3.8-b) until contact is re-established, and all switch simultaneously back to `hold`. At this time, agents are in the initial configuration rotated by $2\pi/\widehat{m}$ (figure 3.8-c). A livelock cycle is thus obtained after $\widehat{m}$ such sequences.

Case $2 < \widehat{m} \leq 4$. The construction is analogous to the previous case, but $C_{\widehat{m}}$ has now radius $\rho(\widehat{m})$. Take initial conditions for $\widehat{m} - 1$ agents so that their reserved discs are centered on $C_{\widehat{m}}$ $2\pi/(\widehat{m} - 1)$ radians apart and head in the tangent direction (see figure 3.9-a and -b). Place the initial position of the reserved disc of the remaining



a) b) c)

Figure 3.8: Livelock-generating conditions for the GR policy with $\widehat{m} = 6$.



a) b) c)

Figure 3.9: Blocking executions of the GR policy with $\widehat{m} \leq 4$.

agent in the center of $C_{\widehat{m}}$. By applying the GR policy, this agent remains indefinitely in the `hold` state while the other $\widehat{m} - 1$ remain in the `roll` state. Indeed, while in `roll`, the admissible cone coincides with the half plane determined by the tangent to the reserved disc of the inner agent, hence $\theta_i \equiv \max\{\Theta_i^-\} \in \Theta_i^-$. Moreover, by the same reason, $\phi_i \notin \Theta_i^-$. Therefore, no guard leaving `roll` is ever active for these agents.

Case $\widehat{m} = 2$. The construction and behaviour in this case is completely analogous to the case $\widehat{m} \geq 5$ (see figure 3.9-c).

We have thus proved that sparsity of target configurations is a necessary condition to rule out the possibility of blocking executions of the GR policy. A proof of

sufficiency appears to be very complex. In the next section, we describe a method to approach the problem from a probabilistic point of view ([25]).

Probabilistic approach

Consider the following statement:

Conjecture The GR policy provides a non-blocking solution for all admissible plans (G_0, G_f).

Let the predicate $\mathbf{P}_{GR}(G_0, G_f)$ be true if the generalized roundabout policy provides a non-blocking solution for initial and final configurations G_0 and G_f, respectively.

A probabilistic verification of the conjecture can be obtained following the approach described below (for more details, see e.g. [26]).

Consider a bounded set $\mathcal{B} = \mathcal{B}_0 \times \mathcal{B}_f$ where the uncertainty $\Delta = (G_0, G_f)$ is uniformly distributed. Let $\mathcal{G} = \{(G_0, G_f) \in \mathcal{B} | \mathbf{P}_{GR}(G_0, G_f)\}$ denote the "good" set of problem data for which the predicate applies. Also, let $\mathcal{C} = \{(G_0, G_f) \in \mathcal{B} | \neg \mathbf{P}_1 (G_0) \wedge \neg \mathbf{P}_2(G_f)\}$ denote the set of admissible plans.

Using the standard induced measure on $\mathcal{B}$, the volume ratio

$$r := \frac{\mathrm{Vol}(\mathcal{G} \cap \mathcal{C})}{\mathrm{Vol}(\mathcal{C})},$$

can be regarded as a measure of the probability of correctness of the conjecture. A classical method to estimate r is the Monte Carlo approach, based on the generation of N independent identically distributed (i.i.d.) random samples within $\mathcal{C}$, which we denote by $\mathbf{\Delta}^i$, $i = 1, \ldots, N$. An estimate of r based on the empirical outcomes of the N instances of the problem is given by $\hat{r}(N) = \frac{1}{N} \sum_{i=1}^{N} I_{\mathcal{G} \cap \mathcal{C}}(\mathbf{\Delta}^i)$ where $I_{\mathcal{G} \cap \mathcal{C}}(\mathbf{\Delta}^i) = 1$ if $\mathbf{\Delta}^i \in \mathcal{G} \cap \mathcal{C}$ and 0 otherwise.

By the laws of large numbers for empirical probabilities, we can expect that $\hat{r}(N) \to r$ as $N \to \infty$. Probability inequalities for finite sample populations, such

as the classical Chernoff bound [26], provide a lower bound N such that the empirical mean $\hat{r}(N)$ differs from the true probability r less than ϵ with probability greater than $1-\delta$, i.e. $Pr\{|r-\hat{r}(N)|<\epsilon\}>1-\delta$, for $0<\epsilon,\delta<1$. The Chernoff bound is given by

$$N > \frac{1}{2\epsilon^2}\log\left(\frac{2}{\delta}\right). \tag{3.8}$$

Notice that the sample size N, given by (3.8), is independent on the size of $\mathcal{B}$ and on the distribution.

To obtain an empirical estimate of r through execution of numerical experiments in our specific problem, the predicate can be modified in the finitely computable form

$$\mathbf{P}'_{GR}(G_0, G_f) = \{J(G_0, G_f) \leq \gamma\},$$

where $J(G_0, G_f)$ denotes the time employed by the last agent to reach its goal, and γ is a threshold to be suitably fixed.

An exhaustive probabilistic verification of the conjecture for wide ranges of all the involved variables remains untractable. To provide a meaningful set of results, however, some of the experimental parameters can be fixed according to criteria indicating the complexity of problems. In other terms, for a given size of the workspace $\mathcal{B}$, the safety distance d_s and the number of agents n can be chosen so that

1. the area occupied by the agents and their reserved discs is a significant portion of the available workspace, and

2. the average worst arrival time of agents is substantially larger than the time necessary for a solution computed disregarding collision avoidance.

The second criterion provides a qualitative information on the amount of deviations from nominal paths caused by collisions, hence on the amount of conflicts occurred.

Several experiments have been conducted to assess how these two indicators vary with the parameters (see figure 3.10). With the choice $\mathcal{B} = ([0,800]\times[0,700]\times[0,2\pi))^{2n}$,

$d_s = 18$ and $n = 10$, the area occupied by agents is 7% of the workspace, and the average worst arrival time is 80% longer than the unconstrained solution time. Another set of preliminary experiments have been conducted to choose a threshold time γ which was computationally manageable, yet sufficiently long not to discard solutions. The percentage of successes of the policy as a function of the threshold γ is reported in figure 3.10. From results obtained, it appears that only minor modifications of the outcomes should be expected for threshold above $\gamma = 1600$. Finally, an estimate of the ratio r has been obtained by the probabilistic approach previously described. In order to have accuracy $\epsilon = 0.01$ with 99% confidence ($\delta = 0.01$), it was necessary by (3.8) to run 27000 experiments, with initial and final conditions uniformly distributed in the configuration space $\mathcal{C}$. Samples were generated by a rejection method applied to uniform samples generated in $\mathcal{B}$. None of these 27000 experiments failed to find a solution within time $\gamma = 4000$, hence $\hat{r}(N) = 1$. Hence, we can affirm with 99% confidence that the sparsity condition is sufficient to guarantee admissible plans for the generalized roundabout policy to within an approximation of 1% in case of $n = 10$ agents with safety disc of diameter $d_s = 18$.

Some extensions to those preliminary results are herein reported (see [27] for details). In order to have accuracy $\epsilon = 0.0075$ with 99.25% confidence ($\delta = 0.0075$), it was necessary by (3.8) to run 50000 experiments, with initial and final conditions uniformly distributed in the configuration space $\mathcal{C}$. Samples were generated by a rejection method applied to uniform samples generated in $\mathcal{B}_n = ([0, 800] \times [0, 700] \times [0, 2\pi))^n \subset \mathcal{B}$, with $n \in \{2, \ldots, 10\}$ and $R_S \in \{2, \ldots, 18\}$. None of these 50.000 experiments failed to find a solution within time $\gamma = 4000$, hence $\hat{p}_D(N) = 1$. Hence, with 99.25% of confidence we can state that the sparsity condition is sufficient to guarantee liveness of the generalized roundabout policy to within an approximation of 0.75% for systems with a number of agents varying from 2 to 10 and safety radius from 2 to 18.

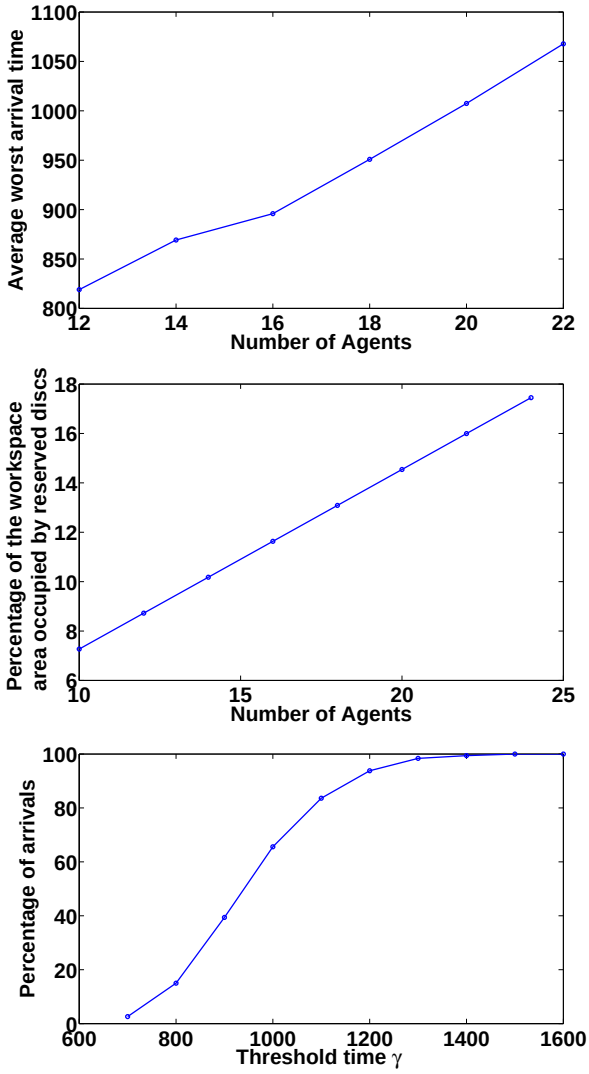


Figure 3.10: Left: Average worst arrival time (over 300 experiments) vs. safety distance, for a system of 10 agents. The average unconstrained solution time is close to 520. Center: Percentage of workspace area occupied by agents and their reserved discs for different numbers of agents. Right: Percentage of arrivals with respect to threshold time γ.

3.4 Evaluation of the Roundabout Policy

We are now interested in providing qualitative evaluations on the chosen policy regarding liveness apart from the sparsity condition. The dimension of $\mathcal{C}$ in $\mathcal{B}$ depends on the value of the number of agents n and the safety radius R_S. The figure 3.11 represents the normalized dimension of $\mathcal{C}$ in $\mathcal{B}_n$ with respect to variation of $n \in \{2, \ldots, 20\}$ and $R_S \in \{2, \ldots, 40\}$. In figure 3.12 the z-axis view is reported. Projections of the isodimensional curves on the (n, R_S) plane appear to be hyperbolas, i.e. $n\,R_S = const.$.

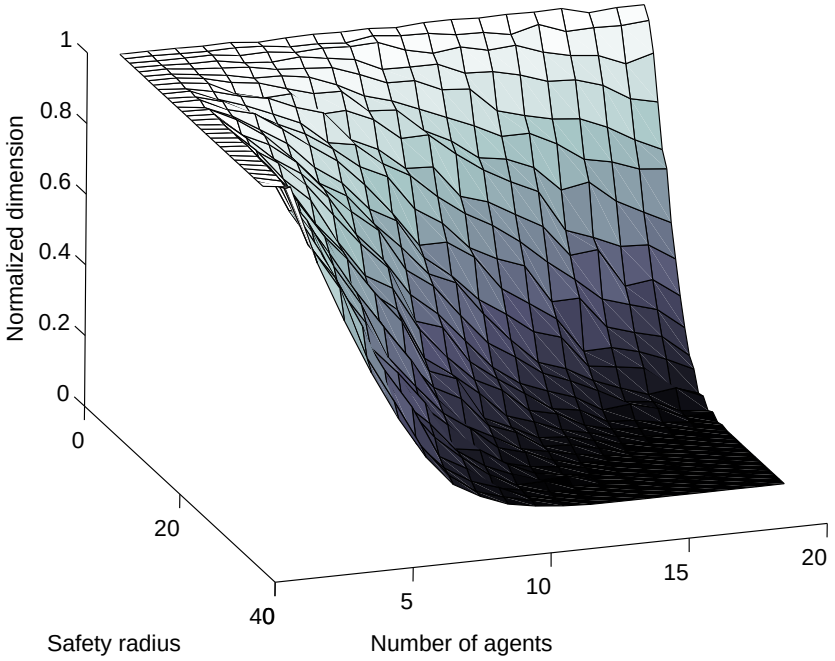


Figure 3.11: The normalized dimension of $\mathcal{C}$ in $\mathcal{B}$ with respect to variation of n and R_S.

Using values of n and R_S such that the dimension of $\mathcal{C}$ in $\mathcal{B}_n$ is larger or equal

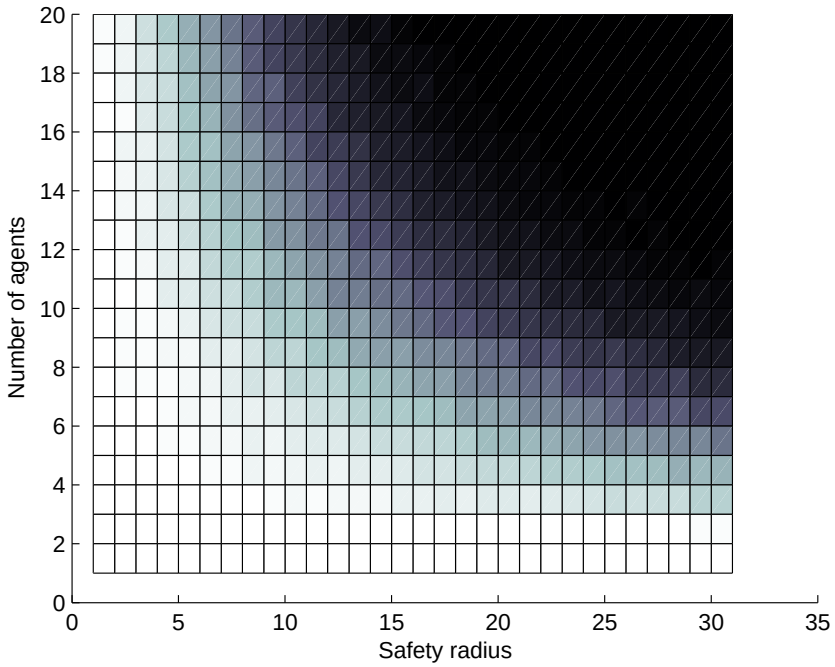


Figure 3.12: Projections of the isodimensional curves on the (n, R_S) plane appear to be hyperbolas.

to 95% we have verified, with the proposed probabilistic approach, that with 99% confidence the sparsity condition is sufficient to guarantee liveness of the generalized roundabout policy to within an approximation of 1%. For the remaining 5% of $\mathcal{B}_n \setminus \mathcal{C}$ more than 20000 simulations have been run. In the 96.433% of cases such simulations have termined with the reaching of the goal configurations. Concluding, we can affirm that more of $0.99 \;\; 0.95 + 0.96433 \;\; 0.05 = 99.8\%$ of cases all agents will eventually reach the goal configurations.

Furthermore, notice that for (n, R_S) couples for which $\mathcal{C}$ is at least 95% of $\mathcal{B}$ the total space occupied by agents is around the $4 - 5\%$ of the workspace. For example, in terms of agents occupancy this means that in a workspace of dimension

7meter $\times$ 8meter we are able to manage safely 10 agents with a safety disc diameter of 60 centimeters.

Chapter 4

Optimal navigation of an autonomous vehicle

In the previous chapters two decentralized collision avoidance policies for multi-agent systems have been considered. It was assumed that initial and final configurations may be arbitrarily assigned to agents. In the case of air traffic management they can be considered, for example, departure and arrival airports. Nevertheless, such strategies could be exploited in a different scenario where agents move in the environment for accomplishing to some tasks, even different from each other, and may use collision avoidance policies when they become closer. In such cases goals can be considered as the result of an optimizing strategy. Consider for example the case of a team of agents, exploring an unknown environment: each agent can choose individually or by using a common strategy paths that optimize the gathered information concerning the environment. In such case goals are directly computed by agents which can move, ad example according to GR policy, to reach them avoiding collision with other agents.

Exploration represents a very important trend in research related to mobile

robots. Three of the main issues in explorative applications are the localization of the vehicle with respect to the environment, the construction of a map of the environment itself, the planning and the control of the vehicle to desired postures relative to the environment. Naturally, these problems are closely interconnected. The first two issues are generally studied under a common framework known with the acronym of SLAM (Simultaneous Localization And Map building). Indeed, in the SLAM literature, vehicles are often commanded in open loop and not always a motion strategy aiming at optimizing some quantity is specified. In this chapter it is considered the problem of maximizing the gathered information during the exploration of an unknown or a partially unknown environment by an agent.

In practical applications of automated vehicle control, however, one is confronted with the problem of estimating the current position and orientation of the vehicle only through indirect, noisy measurements by available sensors. I focused on aspects concerning the existence of solutions to the SLAM problem, and to the choice of optimal exploratory paths to elicit SLAM information.

4.1 Formulation of the SLAM problem

Consider a system comprised of a unicycle like vehicle moving in an environment with the aim of localizing itself and the environment features. For simplicity, we assume that features are distinctive $2D$ points in the environment where the vehicle moves. The vehicle is endowed with sensors, such as a radial laser rangefinder or video cameras. Both the vehicle initial position and orientation, and the feature positions, are unknown or, more generally, known up to some a priori probability distribution. Among the features that the sensor head detects in the robot environment, we will distinguish between those belonging to objects with unknown positions, referred to as targets, and those belonging to objects whose absolute position is known, denoted as markers. Indeed, as it can be argued, this distinction is only useful for simplicity of description, as in general the case is that there exist

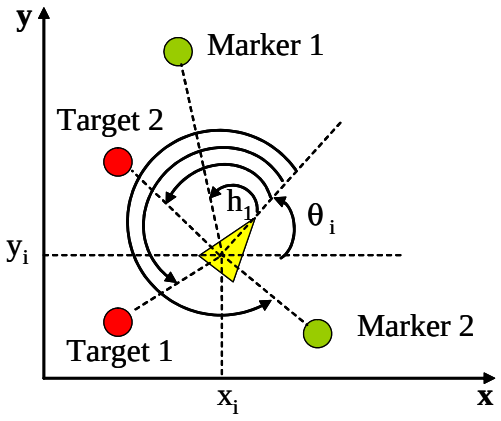


Figure 4.1: A vehicle in an unknown environment with markers and targets.

features that are more or less uncertain. The vehicle dynamics is supposed to be slow enough to be neglected (dynamics does not add much to the problem structure, while increasing formal complexity). Let the kinematics of vehicle i be written as a nonlinear system of the type $\dot{g}_i = F(g_i)u_i$, where $g_i = (x_i, y_i, \theta_i) \in \mathbb{R} \times \mathbb{R} \times S^1$ is the robot pose and $u_i \in \mathbb{R}^2$ are the input velocities. It is often the case where the system velocities are affected by disturbances (such as e.g. slippage of the wheels), and the model is accordingly modified to include process noise as $\dot{g} = F(g)(u + \mu)$. Let the i-th target absolute coordinates be denoted by $p_i \in \mathbb{R}^2$ and use $p \in \mathbb{R}^{2\,n_f}$ to denote the collection of all features. According to the sensor equipment specifics, the relative position of the vehicle and of the features form sensor readings, or observables, described by the map $h : \mathbb{R} \times \mathbb{R} \times S^1 \times \mathbb{R}^2 \to \mathbb{R}^q, (g, p) \to y = h(g, p)$, where q is the number of observables. We assume that the sensor data are affected by additive noise as $y = h(g, p) + \nu$. In system-theoretic terms, the problem can be described by referring to the input-state-output system

$$\begin{aligned} \begin{bmatrix} \dot{g} \\ \dot{p} \end{bmatrix} &= f(g,p,u,\mu) = \begin{bmatrix} F(g) \\ 0 \end{bmatrix} (u+\mu) \\ y &= h(g,p) + \nu \end{aligned} \tag{4.1}$$

In this framework (see [28] for details) localization and mapping are modeled as observability problems, dealing with the reconstruction of the present pose g and feature map p, respectively, from current and past observables, from model and input knowledge, and from statistics on process noise and measurement noise. Uncertainty can be dealt with in basically two ways, i.e. deterministically or by using probabilistic models. The first approach assumes that all uncertainty sources may generate errors that are unknown but bounded, and seeks for bounds on how these error can propagate through the reconstruction process. Naturally, the problem tends to be overly complex from the computational and memory-occupation viewpoints, hence efficient algorithms to approximate the worst-case bounds are in order. An application of this approach to robot localization is reported in [29], where an efficient, recursive algorithm to approximate the set of robot poses compatible with present and past measurements is presented. Deterministic algorithms tend to suffer from excessive conservativeness, and are typically not very suited to take into account the existence of large, sporadic errors in sensor readings (outliers), which are common in some types of sensors used in SLAMS (e.g. spurious reflections of lasers or sonars, feature mismatch, etc.). When an excess of conservatism is not justified by particularly risk-sensitive applications, it is often preferred to adopt probabilistic models of uncertainty. The basis for virtually all probabilistic methods is Bayesian theory of inference, which assumes that the statistical properties of the data space and of the model space are well defined. These are the vector spaces, of suitable dimension, where observables y and unknowns (and estimates thereof, denoted for brevity as x) take their values, and where a probability density function (p.d.f.) is defined for the variables of interest. The a priori state of information consists in a p.d.f. defined over the model space X, $f_{prior}(x)$, which models any knowledge one

may have on the system model parameters independently from the present act of measurement, due e.g. to physical insight or to independent measurements carried out previously.

In the formation of estimates, two information sources are to be considered, i.e. the forward solution of the physical model, and the act of measuring itself. The state of information on the experimental uncertainties in measurement outputs can be modeled by means of a p.d.f. $f_{exp}(y)$ over the data space Y (this should be provided by the instrument supplier), while modeling errors (due to imperfection of (4.1), or to process noise) can be represented by a conditional p.d.f. $f_{mod}(y|x)$ in the data space Y (or, more generally, by a joint p.d.f $f_{mod}(y, x)$ over $X \times Y$). Fusing the different information in an estimate of x leads to a posterior p.d.f over X, that is described by Bayes' formula

$$f_{post}(x) = f(x|y) = \alpha_b f_{prior}(x) \int_Y f_{exp}(y) f_{mod}(y|x) dy, \tag{4.2}$$

where α_b is a normalization factor such that $\int_X f_{post}(x)dx = 1$. Although the posterior p.d.f. on the model space represents the most complete description of the state of information on the quantity to be measured one may wish, a final decision on what is the "best "estimate of x needs usually be taken. Several possibilities arise in general, such as the maximum a posteriori estimate (MAP), maximum likelihood estimates (MLE, which coincides with MAP if no priors are available), the minimum variance estimate (MVE) alias minimum mean square (MMSE). While very little can be said in general about the performance of such estimators, well known particularizations apply under certain assumptions on the prior distributions. Thus, if a normal distribution (an order-2 Gaussian) can be assumed for all prior information, the MAP estimate enjoys many useful properties: first (and perhaps most importantly for the problem at hand), since the convolution in (4.2) of two Gaussian distributions is Gaussian, the modeling and experimental errors in measurements simply combine by addition of the covariance matrices of experimental and modeling errors, $C_Y = C_{exp} + C_{mod}$. Roughly speaking, errors in the model

knowledge (kinematic model of the systems and odometry errors) can be ignored, provided that experimental measurement errors in y are suitably increased. This result holds for nonlinear sensor models as well. For linearized measurement models ($y = Hx$), the a posteriori p.d.f. would also be Gaussian, the MVE and MAP estimates would coincide and evaluate to

$$\begin{aligned} \hat{x} &= C_{post}(H^T C_Y^{-1} y + C_{prior}^{-1} x_{prior}), \\ C_{post} &= \left(\mathcal{F} + C_{prior}^{-1}\right)^{-1} \end{aligned} \tag{4.3}$$

where $\mathcal{F}$, is the *Fisher information matrix* for the linear case, and is defined as

$$\mathcal{F} = H^T C_Y^{-1} H \tag{4.4}$$

As a final remark, the Gauss-Markov theorem [30] ensures that the estimate (4.3) is the best linear unbiased estimate in the minimum-variance sense even for non-Gaussian a priori distributions. This result may seem to indicate some "absolute optimality " of the least-squares estimate. However, the MVE of a non-Gaussian distribution may not be a significant estimate. This is the case for instance when a few measurements are grossly in error (*outliers*): the MVE in this case can provide meaningless results. This fact is sometimes used to point out the lack of robustness of the MVE. In the literature on mobile robot localization and mapping, methods to evaluate an estimate of the posterior p.d.f. over the space of unknown robot poses and targets have been studied extensively. While for an exhaustive review the reader is referred to [**?**], we limit ourselves to point out that methods proposed so far can be roughly classified in two main groups: batch and recursive. Batch methods attempt as accurate a solution of the posterior as possible, by taking into account that often in SLAM the posterior p.d.f. is a complex multimodal distribution. To such complexity contribute different factors, among which the nonlinearity of dynamics and measurement equations (4.1), and the fact that measurement noise in different measurements is statistically correlated, because errors in control accumulate over time, and they affect how subsequent measurements are interpreted ([**?**]).

A crucial aspect of SLAM is indeed that, when features are not distinctive, multiple correspondences are possible, a problem also known as data association. The correspondence problem, consisting in determining if sensor measurements taken at different times correspond to the same physical object in the world, is very hard to be tackled, since the number of possible hypotheses can grow exponentially over time. A family of methods recently introduced to deal with these problems, which is based on Dempster's Expectation Maximization Algorithms (EM) [24,41], represent the current state-of-art in this regard. However, since EM have to process data multiple times they are not suitable to real time implementation, as needed e.g. to interface with servoing algorithms. On the other hand, most often new updates of model estimates are needed in real-time, without referring to the whole history of sensed data. To cope with this requirement, further simplifications are usually done: for instance, assuming a Gaussian posterior distribution, the given record of data can be completely described by the mean vector and the covariance matrix. When a new datum is available, all prior information can be extracted from those statistics. A method that does not use prior information explicitly, but through its statistics only, is called recursive. The Kalman filter is one such recursive method, implementing the optimal minimum variance observer for a linear system subject to uncorrelated, zero-mean, Gaussian white noise disturbances. Unfortunately, these assumptions are unfulfilled in SLAM applications. Hence, different simplifying assumptions and approximations are employed. Filters resulting from repeated approximate linearization of (4.1) are commonly referred to Extended Kalman Filters (EKF). Although extended Kalman filters for the SLAM problem do not guarantee any optimality property, they remain the most widely used filters in SLAM. EKF maintain all information on the estimated posteriors in the vector of means and in a covariance matrix, whose update at each step is a costly operation (quadratic with the number of features). In practical implementations, a key limitation of EKF is the low number of features it can deal with. Algorithms have been recently

proposed to overcome this limitation. The Fast- Slam [31] algorithm is based on the assumption that the knowledge of the robot path renders measurements of individual markers independent, so that the problem of determining the position of K features could be decomposed into K estimation problems, one for each feature [31]. Compressed EKF (CEKF), see [32], stores and maintains all the information gathered in a local area with a cost proportional to the square of the number of landmarks in the area. This information can then be transferred to the rest of the global map with a cost that is similar to full SLAM, but in only one iteration. Sparse Extended Information Filter (SEIF), see [33], is an algorithm whose updates require constant time, independent of the number of features in the map. It exploits the particular form of the information matrix, i.e. the inverse of the covariance matrix. Since the information matrix is sparse, it possesses a large number of elements whose values, when normalized, are near zero and can be neglected in the updating process. Some algorithms, see [34, 35, 36], based on incremental update of uncertain maps, use a fuzzy logic approach to manage uncertainty on obstacle poses and successively implement obstacle avoidance strategies. An interesting possibility in SLAM is the possibility of using multiple vehicles in a cooperative way in order to perform tasks more quickly and robustly than a single vehicle can do. In [6, 7], the problem of performing concurrent mapping and localization with a team of cooperating autonomous vehicles is considered, and the advantages of such a multi-agent cooperation are illustrated. One of the most challenging topics in SLAM is the optimization of autonomous robotic exploration. Indeed, it is often the case that robots have degrees of freedom in the choice of the path to follow, which should be used to maximize the information that the system can gather on the environment. The problem is clearly of great relevance to many tasks, such as e.g. surveillance or exploration. However, it is in general a difficult problem, as several quantities have to be traded off, such as the expected gain in map information, the time and energy it takes to gain this information, the possible loss of pose information along

the way, and so on. This problem is considered in detail in the next section.

4.2 Solvability and Optimization of SLAM

As already mentioned, simultaneous localization and mapping amounts to estimating the state of system (4.1) through integration of input velocities (odometry) and knowledge of the observations y. Input velocities and observables are affected by process and measurement noise, respectively. We start by observing that system (4.1) is nonlinear in an intrinsic way, in the sense that approximating the system with a linear time-invariant model destroys the very property of observability: this entails that elementary theory and results on linear estimation do not hold in this case. The intrinsic nonlinear nature of the problem can be illustrated directly by the simple example in figure 4.1 of a planar vehicle with M markers and N targets. Outputs in this examples would be the $q = M + N$. The linear approximation of system (4.1) at any equilibrium $x = x_0$, $p = p_0$, $u = 0$, would indeed have a null dynamic matrix

$$A = \left.\frac{\partial f(\cdot)}{\partial (g,p)}\right|_{eq.} = 0 \in \mathbb{R}^{(2\,N+3)\times(2\,N+3)},$$

and

$$C = \left.\frac{\partial h(\cdot)}{\partial (g,p)}\right|_{eq.} = 0 \in \mathbb{R}^{(M+N)\times(2\,N+3)}.$$

Hence, in any nontrivial case (i.e., whenever there is at least one targets ($N \neq 0$) or there are less than three known markers ($M < 3$) the linearized system is unobservable.

On the other hand, it is intuitively clear that simple triangulation calculations using two or more measurements from different positions would allow the reconstruction of all the problem unknowns, except at most for singular configurations. Analytically, complete observability of system (4.1) can be checked, as an exercise in nonlinear system theory, by computing the dimension of $< f(\cdot) \,|\, span\,\{dh(\cdot)\} >$, the smallest codistribution that contains the output one-forms and is invariant un-

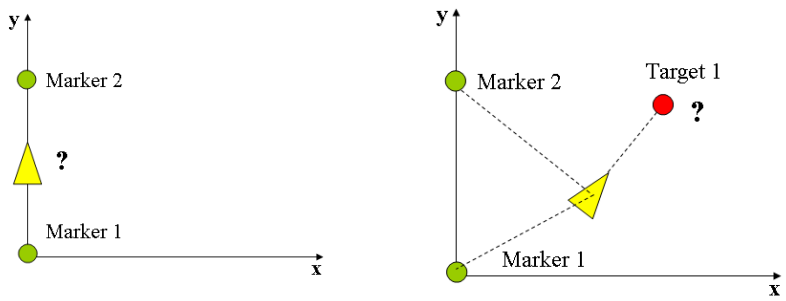


Figure 4.2: A vehicle triangulating with two markers cannot localize itself if the inputs are such that it remains aligned with the markers; it cannot localize a target if it aims at the target directly.

der the control vector fields (see [37] for details on calculations). By such nonlinear analysis, it is also possible to notice that observability can be destroyed by choosing particular input functions, the so-called *bad inputs*. A bad input for our example is the trivial input $u = 0$: the vehicle cannot localize itself nor the targets without moving. Other bad inputs are illustrated in figure 4.2 In order to drive a rover to explore its environment, it is clear that bad inputs should be avoided. Indeed, the existence of bad inputs suggests that there should also be *good*, and possibly optimal, inputs. To find such optimal exploratory strategies, however, the differential geometric analysis tools such as those introduced above are not well suited, as they only provide topological criteria for observability. What is needed instead is a metric information on the *distance* of a system from unobservability, and to how maximize it. More generally, it is to be expected that different trajectories will elicit different amounts of information: a complete SLAM system should not only provide estimates of the vehicle and feature positions, but also as precise as possible a description of the statistics of those estimates as random variables, so as to allow evaluation of confidence intervals on possible decisions. To provide a

better understanding of how two different states can be distinguished via dynamic measurements, let us consider the output $y(t) = h(g, p) = y(g_o^0, u, t)$ as a function of the initial conditions g_o and of the inputs u. Let g_0 and g_0^1 denote two different initial conditions with $\|g_0^0 - g_0^1\| < \epsilon$, and let us consider

$$y(g_0^1, u, t) - y(g_0^0, u, t) = \left.\frac{\partial y}{\partial g_0}\right|_{g_0 = g_0^0} (g_0^1 - g_0^0) + \mathcal{O}^2(\epsilon) \tag{4.5}$$

i.e. a linear measurement of the form

$$\tilde{y} + \delta_y = M(t)\tilde{g} \tag{4.6}$$

where $\tilde{g} = (g_0^1 - g_0^0)$ is unknown, $\tilde{y}$ comes from measurements, and the perturbation term δ_y accounts for measurement noise and approximation errors. Notice explicitly that the linear operator $M = \left.\frac{\partial y}{\partial g_0}\right|_{g_0 = g_0^0}$ depends in general on applied inputs, as only for very special systems (in particular, linear) superposition of effects of initial states and inputs holds. By premultiplying both sides of (4.6) by $M^T W$, with $W > 0$ a suitable positive definite matrix weighing accuracy of different sensors, and by integrating from time 0 to T , we obtain

$$\mathcal{Y} + \Delta_y = \mathcal{F}\tilde{g} \tag{4.7}$$

where $\mathcal{Y} = \int_0^T M^T(t) W \tilde{y} dt$ and $\mathcal{F} = \int_0^T M(t) W M(t) dt$ is the Fisher Information Matrix for our system. Singularity of $\mathcal{F}$ (for some input choice) clearly implies that distinct initial values of the state exist which provide exactly the same measurements over the time interval, hence is tantamount to unobservability of the system. A different argument to support the same conclusion can be derived from Kalman estimation theory. Indeed, in the linear case, for the covariance matrix P of a Kalman filter, the C.R. Rao inequalities [30] hold:

$$(\mathcal{F} + \mathcal{N}) \leq P \leq \mathcal{F}^{-1} + \mathcal{N} \tag{4.8}$$

where F is the Fisher Information Matrix (defined in (4.8) in this framework), and N, the covariance matrix of process noise, is assumed to be independent of

the trajectory. According to this, minimization of $\mathcal{F}^{-1}$ can be considered as an instrument to minimize P. This is further justified by the fact that, in the absence of process noise and of prior information, the Riccati equation solution for the filter is exactly $P(t) = \mathcal{F}^{-1}(t)$.

From the above considerations on state reconstruction and on Cramèr-Rao inequalities, it is clear that the information matrix can provide the desired notion of "distance" from unobservability, that is, a merit figure for different inputs (hence trajectories) of the exploring rover. Indeed, the smallest eigenvalue $E = \lambda_{min}(\mathcal{F}) = 1/\|\mathcal{F}^{-1}\|_2$, the determinant index $D = (n_v + d\,n_f)\sqrt{det\,\mathcal{F}}$, the trace index $T = \frac{trace(\mathcal{F})}{n_v + d\,n_f}$ and the average-variance index $A = \frac{n_v + d\,n_f}{trace(\mathcal{F}^{-1})}$ are among the most often used such criteria (known as E–, D–, T–, and A– criterion, respectively). Notice that information based criteria do not reflect any particular choice in the estimator or filter adopted in the actual localization procedure, rather it is intrinsic to the reconstructibility of the state from the given trajectory. This is a very useful property, in view of the fact that several different estimators and filters can be applied to the SLAM problem.

4.3 Closed-form Solution

The problem of choosing exploratory paths of fixed length L to maximize SLAM information can be formalized as an optimal control problem, i.e.

$$\text{maximize } J(u) = \lambda_{min}(\mathcal{F}) \tag{4.9}$$

subject to the constraints

$$\begin{aligned} &L = \textstyle\int_0^T \sqrt{\dot{x}^2 + \dot{y}^2}\, dt \\ &\dot{g} = F(g)u; \;\; g(0) = g_0 \\ &y = h(g). \end{aligned}$$

Solving this problem can be expected to be quite difficult in general. Using system. theoretic tools, an analytic solution was given in [38] for the simplified case of an omnidirectional vehicle moving in a planar environment with only two markers. Extremal paths for the functional J were shown to be contained in the pencil of curves spanned by the parameter α as

$$(cos(\alpha)\ sin(\alpha)) \left(\frac{\partial y}{\partial g_0}^T \frac{\partial y}{\partial g_0} - \frac{\partial y}{\partial g_0}^T \Big|_{g=g_0} \frac{\partial y}{\partial g_0} \Big|_{g=g_0} \right) \begin{pmatrix} cos(\alpha) \\ sin(\alpha) \end{pmatrix} \tag{4.10}$$

where the actual value of α depends on L. It can be easily seen that the obtained pencil is a set of conics (some examples of optimal exploratory paths, for different lengths, are represented in figure4.3).

4.4 Numerical Methods

Extensions of the analytic solutions to nonholonomically constrained vehicles with unknown target features are feasible (work in this direction is undergoing). However, to obtain solutions in most general cases, efficient numerical methods are in order. In a recent overview [**?**], where the importance of the SLAM optimization problem is acknowledged, currently available solutions are reported to be mostly limited to heuristic, greedy algorithms. Furthermore, most known methods often disregard the nonlinear character of the SLAM problem, which on the contrary is of large momentum, as we discussed. The main limitation of gradient-descent methods in this framework is of course the presence of local minima in the information return function: application of methods from receding horizon optimal control theory in this context can be expected to offer a substantial edge. In the follwoing, we illustrate application of such techniques to a few examples of on-line, numeric SLAM trajectory optimization. To apply numerical methods, continuous–time system equations (4.1) are first discretized, so that the information matrix is rewritten as the sum of products

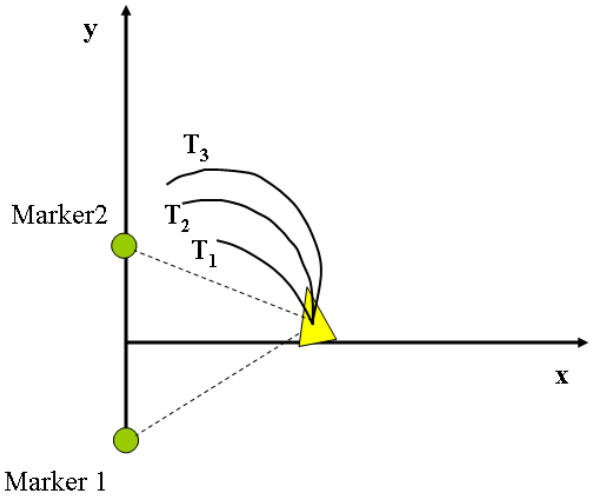


Figure 4.3: Optimal trajectories for three different path lenghts $T_1 = 1\,sec$, $T_2 = 2\,sec$ and $T_3 = 3\,sec$

.

$$\mathcal{F} = \sum_{i=0}^{k} \frac{\partial y}{\partial g_0}\Big|_{x_i}^{T} \frac{\partial y}{\partial g_0}\Big|_{x_i} \tag{4.11}$$

evaluated at each point of a candidate trajectory. Using techniques developed in [39], we furthermore introduce a quantization of the input space (i.e., the set of possible incremental moves of the vehicle), thus inducing a discretization of the configuration space. It can be shown that, for vehicles with chained-form kinematics, the reachable set is indeed a lattice in this case, which is a very convenient structure to apply numerical search methods to.

If d is the cardinality of the input set, there are d^k paths of length k stemming from a generic configuration, for which the contribution to information is given by

(4.11). An exhaustive search of the most informative path is possible for moderate values of d and k. The receding-horizon optimal control policy consists then in applying only the first control of the locally optimizing sequence, to recompute the next optimizing sequence, thus proceeding iteratively. The method can be easily used in conjunction with other techniques for e.g. obstacle avoidance. How practical the method is depend very much on the affordable horizon length for which real-time computations are feasible, hence choices concerning time and input quantization, information representation, etc., are an important area of research.

4.4.1 Simulation results: comparison between greedy and receding horizon approach.

Simulation results reported in figure 4.4 compare performance of a greedy algorithm with the receding horizon method. Walls are considered here as pure obstacles, i.e. they are detected if the vehicle "bumps" into them, while information for self localization and mapping is only extracted from measurements relative to two markers (black circles) and to four target features. Results show how the receding horizon methods collects richer information in this case.

More simulation results relative to different environments are reported in figure 4.5. While these results show how the method is quite versatile in navigating in a cluttered environment fetching for information where that is available, it is of course an open research issue to provide a provable, quantitative assessment of the advantages of this method with respect to others, and to design the numerous parameters that play an importnat role in its implementation

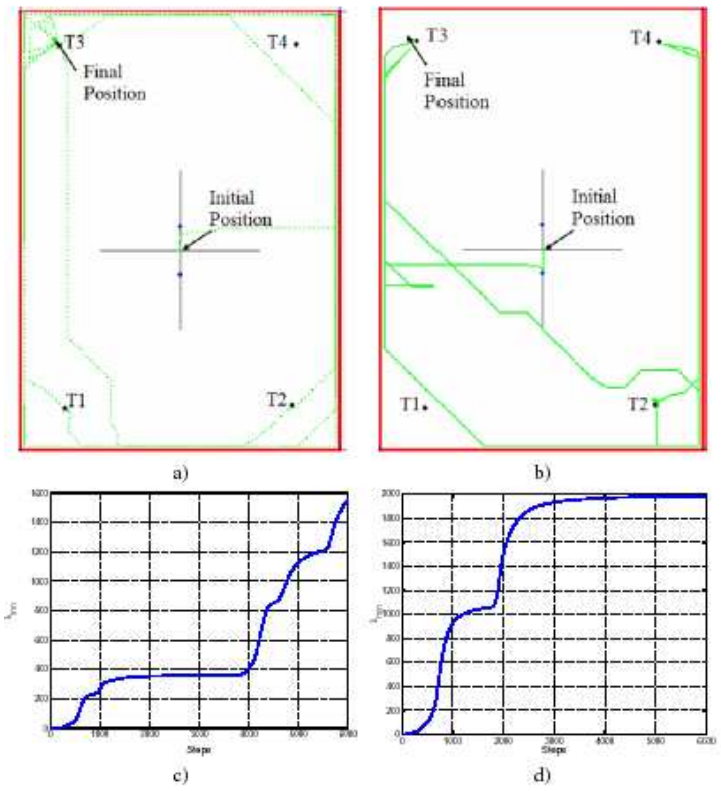


Figure 4.4: Trajectory of a vehicle during the exploration of a rectangular environment with 2 markers and 4 target features, using gradient-descent (a) and a 3-steps receding horizon (b),respectively. Time evolutions of the corresponding information function E = min(F) are reported in c) and d).

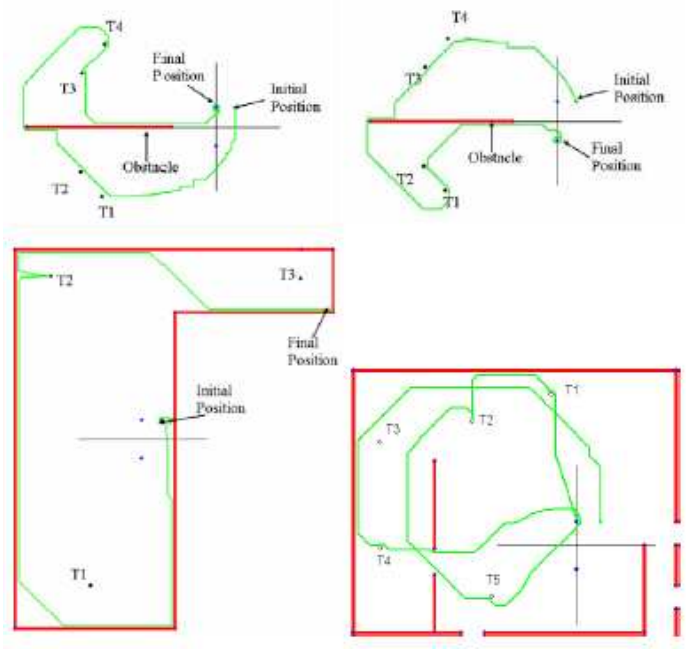


Figure 4.5: Receding-horizon optimal trajectories in different environments, whereby the task of maximizing the information return function leads the vehicle to cover target areas. Observe how slightly different initial conditions may lead to completely different exploration strategies (upper right and left), however with similar characteristics. More complex environments are also dealt with satisfactorily (bottom left and right).

Chapter 5

Platforms for the test-bed of a networked multi-agents system

Despite of their large scalability, robustness to failures and other advantages, decentralized policies present some counterparts such as the degradation of performances, i.e. the loss of optimality, and mainly complexity of verification of some important properties. In the case of collision avoidance problem, we have seen that important properties to be verified are *safety* and *liveness*. The second property has been addressed by means of a probabilistic approach thus requiring a large number of experiments. To this purpose software and hardware test-bed platforms have been designed.

5.1 Software platform

The simulation of hybrid systems has been always a problem for classical discrete time methods. The problem is intrinsic to hybrid systems whose trajectories can be viewed as a concatenation of continuous flows and discrete jumps. A multi-agent system is a collection of interacting hybrid systems, individually referred to as agents. The system studied in chapter 3 is a typical example of multi-agent hybrid system. There the dynamics of each agent is decoupled while switching conditions depend on the state of neighbouring agents.

It is well known that the switching between different modes of operation may determine disastrous results in the simulation on hybrid systems. Therefore a primary requirement for hybrid systems simulation tool is an accurate detection of *zero-crossing*, i.e. events that may determine discrete jumps. The proper way of simulating such systems is to determine the time to the closest event which may give rise to a switching condition. This time can be then used to compute the right integration step for integrating the dynamics of the system up to switching time and continuing the simulation with new differential equations and initial conditions given by the values of the state at the event. In multi-agent systems event detection is a very difficult problem, especially when the number of agents is large. In such cases a remarkable amount of possible events must be foreseen which leads to an unavoidable loss of performances. In [40] authors adopt a step selection algorithm which allows the dynamics of agents to be integrated asynchronously when the state is far form switching conditions. Conversely when the state approaches to the switching event dynamics are integrated synchronously slowing down the simulation.

In this thesis we developed a software platform which consists in a simulation tool for networked mobility systems, capable of handling continuous systems of type (2.3) comprised of tens of autonomous agents (see figure 5.1). The simulator is able to handle effectively the hybrid nature of the system, with accurate detection in

time of events such as network topology switching conditions and collisions.

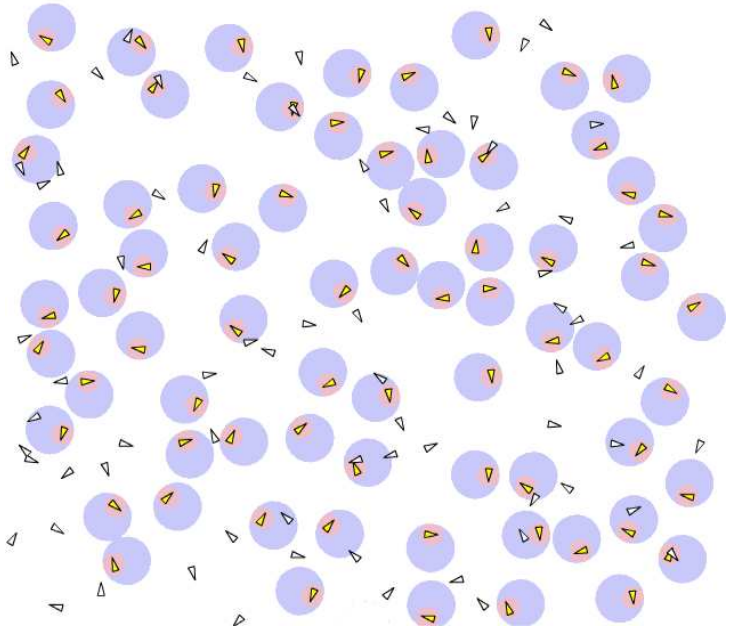


Figure 5.1: Simulation running with 70 agents.

In the following it will be introduced the software tool for the simulation of the Roundabout Policy, with deep attention to the *Event Detector* component which allowed a huge number of simulations to be performed with an high level of fidelity with respect to the theoretical mode of operation of the policy.

5.1.1 Components of the simulation tool

The simulation tool is composed essentially of five components: the Randomization module, the Decision Maker, the Event Detector, the Equation Integrator and the User Interface. The software components are reported in figure 5.2

Event Detector. This module represents along with the Decision Maker the core of the simulation tool. The most challenging issue of this module is to detect the

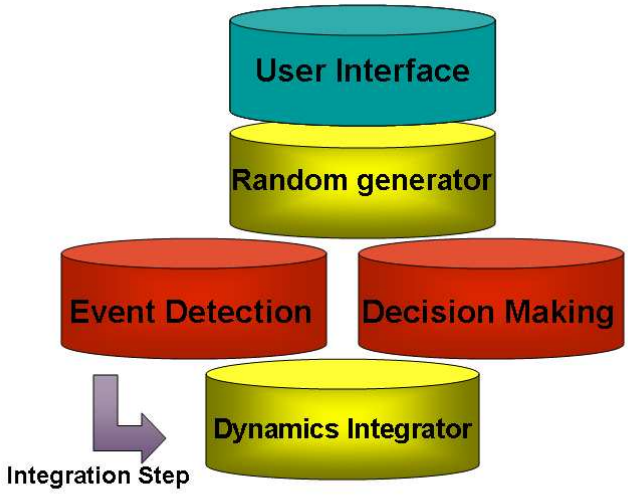


Figure 5.2: Structure of the software tool for the simulation of the GR policy. It is important to notice that the Event Detector module computes the integration step which is used to integrate the continuous agent's dynamics.

time when the first switching condition is met in all the system. This task is critical for two main reason: the correctness of the algorithm and the simulation speedness. Errors in the detection of such event may determine a malfunctioning of the system, e.g. two agents collide, or an agents does no execute the right transition. In some cases malfunctioning causes a just a divergence of the evolution of the simulated system from the theoretical one, while in others they can determine dramatic errors like the violation of the safety constraints.

This module may also determine a loss of performance in the system since it requires a large slice of time to perform the "detection task". The computational cost of the Event Detector module is $\mathcal{O}(n^2)$ where n is the number of agents. In fact,

switching conditions are generally determined by contacts between agents. Hence for each agent all possible contacts with others have to be foreseen. Certainly, some kind of optimization is possible. For example contacts between distant agents are impossible in a short period and could be neglected.

In the following we report the critical events detected by the the Event Detector, without giving implementation details, which would not add interesting issues.

- collision between two agents both in `straight` mode;
- collision between two agents respectively in `straight` and `hold` mode;
- collision between two agents respectively in `straight` and `roll` mode;
- collision between two agents respectively in `straight` and `roll2` mode;
- collision between two agents respectively in `hold` and `roll` mode;
- collision between two agents respectively in `roll` and `roll2` mode;
- transition for `hold` to `roll` mode. In this case if the transition time is not perfectly identified, the switching between the two modes can be delayed or exectuted in advance, determining the overlapping of the agents' reserved disks;
- transition for `roll` to `roll2` mode. It happens when an agent is performing `roll` on another and the contact between their reserved discs is lost.

By checking the above events, this module determines if in a time less than a specified threshold, chosen as the "default integration time", a switch may occur in the modes of operation of any agent of the system. If such condition is met, the integration step takes a new lower value corresponding to the instant when the first switch happens. More precisely the step size T passed to the Integration module is computed as follows:

$$T = min\left\{T_{default}, min\left\{T_i \,|i = 1 \ldots n, \right\}\right\}, \tag{5.1}$$

where T_i corresponds to the "switching time' for agent i and n is the number of agents in the system.

The Randomization module. The randomization component generates uniformly distributed initial an final configuration of agents, guaranteeing that initial conditions are "safe" and checking if final configuration verify the "sparsity condition" (please refer to cap 3). It is possible to configure the size of the workspace, the number of agents, the size of the "safety radius", and of the "reserved disc radius". This module can be used both to perform randomized experiments, and to generate statistical data concerning the number of successful generations, i.e. "safe" and "sparse" with respect to the total number of generated samples, by varying some parameters of the system.

Decision Maker. This module implements the GR automa, i.e. the finite state hybrid machine reported in figure 3.5 associated to each agent. Hence there are as many instances of the decision maker as the number of agents. Each instances makes decisions by following the steps:

- computation of the agent's neighbours;
- computation of the admissible cone;
- evaluation of switching guards.
- evenutally switching of the agent current mode of operation (`straight`, `hold`, `roll`, and `roll2`);

Equation Integrator. This module simulates the dynamic part of the hybrid system. Specifically by integrating equations (2.3), we obtain

$$\begin{aligned} x(\bar{t}+T) &= x_{\bar{t}} + \textstyle\int_{\bar{t}}^{\bar{t}+T} cos(\theta(\tau))\, d\tau \\ y(\bar{t}+T) &= y_{\bar{t}} + \textstyle\int_{\bar{t}}^{\bar{t}+T} sin(\theta(\tau))\, d\tau \\ \theta(\bar{t}+T) &= \theta_{\bar{t}} + \textstyle\int_{\bar{t}}^{\bar{t}+T} w(\tau)\, d\tau \ , \end{aligned} \tag{5.2}$$

where $x_{\bar{t}}, y_{\bar{t}}$, and $\theta_{\bar{t}}$ represent the agent's configuration at time $\bar{t}$, and T is the integration step computed by the Event Detector module. Considering that during the integration period T no switches between the agent's mode of operations occurrs, w is constant during the integration phase and integration (5.2) can be easily performed as it follows

$$\begin{aligned} x(\bar{t}+T) &= x_{\bar{t}} + \int_t^{t+T} cos(\theta_{\bar{t}} + \tilde{w}\,T)\,d\tau = x_{\bar{t}} + sin(\theta_{\bar{t}} + \tilde{w}\,T)\,T \\ y(t+T) &= y_{\bar{t}} + \int_t^{t+T} sin(\theta_{\bar{t}} + \tilde{w}\,T)\,d\tau = y_{\bar{t}} - cos(\theta_{\bar{t}} + \tilde{w}\,T)\,T \\ \theta(t+T) &= \theta_{\bar{t}} + \tilde{w}\,T \end{aligned} \tag{5.3}$$

where $\tilde{w}$ represents the current input value.

User Interface. This module lets the user to choose what kind of simulation to perform. For example, a single randomized simulation can be executed. It is also possible to save a set of configuration in order to make a debug of the system. To this purpose a step by step mode of operation is available, which is extremely useful to debug a simulation with a large number of agents. The user interface provides also a real-time visualization of results and also the possibility of storing results in a file for future analysis or movie generation

5.2 Hardware platforms

Two main hardware platforms have been developed in this thesis. The first one is composed by a set of small radio-controlled agents, moving on a table top with a camera supervising the scene. Agents have unicycle-like kinematics with two actuators for driving forward and steering, with no embedded computational capabilities. Decisions are made by an external computational unit and are communicated to agents through a radio transmitter connected to the I/O of the controller. All controllers are implemented as separate tasks on the central CPU, emulating decentralization by a message-passing protocol mimicking different networking protocols such

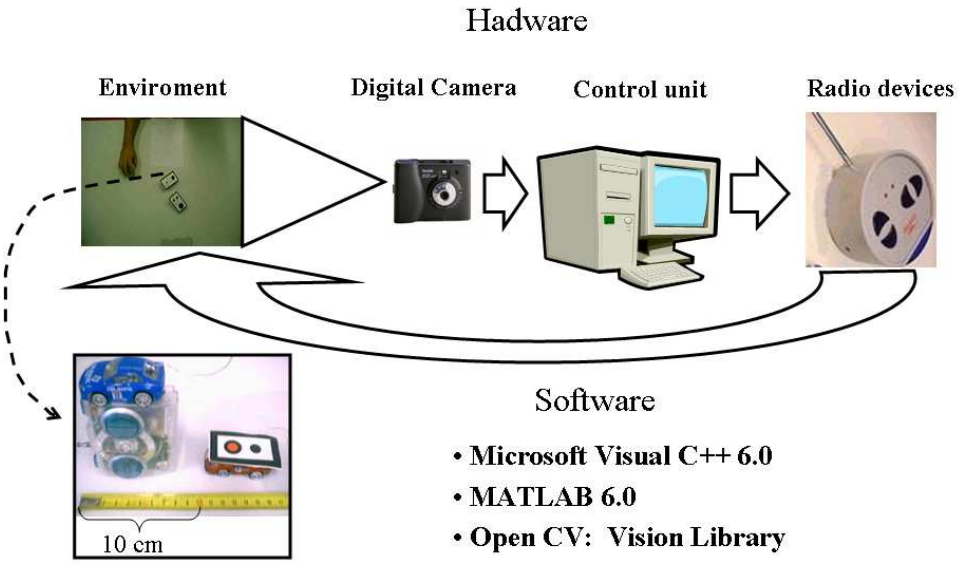


Figure 5.3: Structure of the first experimental platform.

as e.g. Bluetooth. The network architecture provides middleware services including localization and communication. The structure of the platform is reported in figure 5.3. It contains the vision module, the interface with the Radio control module, and the interface with the modules which implement the policy to be tested.

Figure 5.4: Toy commercial cars used in the first experimental setup.

5.2.1 Centralized localization system

The localization service is provided to vehicles by a vision-based module. A camera placed on the ceiling monitors the workspace, provides frames to a software component, which is able to recognize a pattern located on the top of each and to compute its position and orientation. Each frame image is thresholded to filter noise, searched for specified patterns on cars, identification of the car and localization. Usage of multiple cameras to homogenize resolution in different workspace regions is envisioned. The vision system has been implemented on Windows XP operative system. It relies on Intel OpenCV library which have been integrated inside a *MFC* project developed in the Microsoft Visual C++ 6.0 IDE. Since some control algorithms have been previously developed in Matlab, an integration between the Visual C++ and the Matlab environment has been done.

5.2.2 Radio Control System.

The radio control system comprises a hardware and a software component. The first one is made up of a radio device connected to the I/O of a Personal computer. Since each vehicles can receive 4 distinct commands (forward, backward, right, and left) it is possible to use the 8 data pin of a Parallel port to control separately two vehicles. In order to control more than two vehicles, it is possible to use some additive hardware, e.g. PCI-Parallel adapters, to extend the number of parallel ports available The second part is composed of an interface with the I/O for the communication of commands to the agents, of an interface with the localization subsystem, of a module which generates the informative structure for all agents and of a module which implements the policy to be tested. The system has is very flexible because of its multithreaded structure. In fact, it is possible tu run all modules on a single PC, or easily to distribute them on several CPUs, with few coordination efforts.

An extension of this testbed will provide agents with an higher level of autonomy.

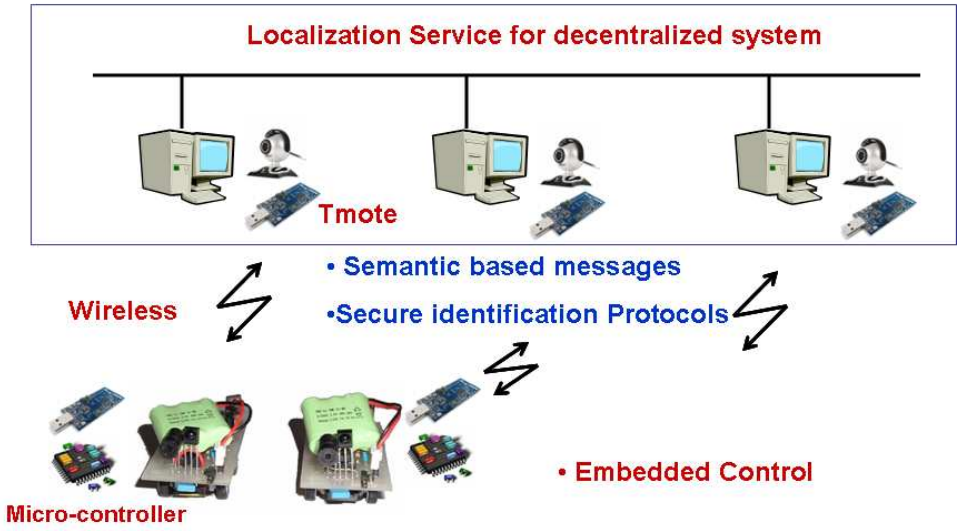


Figure 5.5: Structure of the second experimental platform.

To this purpose we aim that each agent can

- sense its immediate environment;
- communicate with other agents;
- process the information gathered;
- make a decision and consequently take an action.

In order to implement such requisites we designed small agents endowed with microcontrollers and wireless communication capability. Microcontrollers have limited computational power. Specifically, they are PSoC's processors, produced by Cypress MicroSystems. These are controllers at 24 MHz endowed with 16 kb Rom and 256 bytes Ram for code and data storage respectively. Because of the limited computational power, an architecture must be designed which is able to integrate a number of PSoCs for speed control, interface with the communication device and

for implementation of the control strategy. The integration of a number of microcontrollers requires efforts for distributing optimally the control modules between hardware computational resources. Currently an external unit which provides localization services to the vehicles is kept. Each agent is able to receive from the localization server its position and to communicate it to close agents. The communication with the localization unit and with the other agents is implemented via Tmotes Sky devices, produced by Moteiv Inc. This architecture is flexible and has been thought for the test of different decentralized policies. In fact, to change the policy it is only required to modify the module which implements it. The step towards agents, capable of localizing autonomously is a work in progress and does not require any modification to the structure of the platform. Practically, the localization which is currently a service provided by an external unit, could be substituted by a module which uses odometry to compute the position and the orientation of agents. We are studying the possibility of using the same technology of an optical mouse to exploit the localization task.

In order to extend the size of the working area and to improve the quality of the localization service a system which integrates data coming from different cameras has been used . A prototype of the autonomous vehicle which we are developing is reported in figure 5.6.

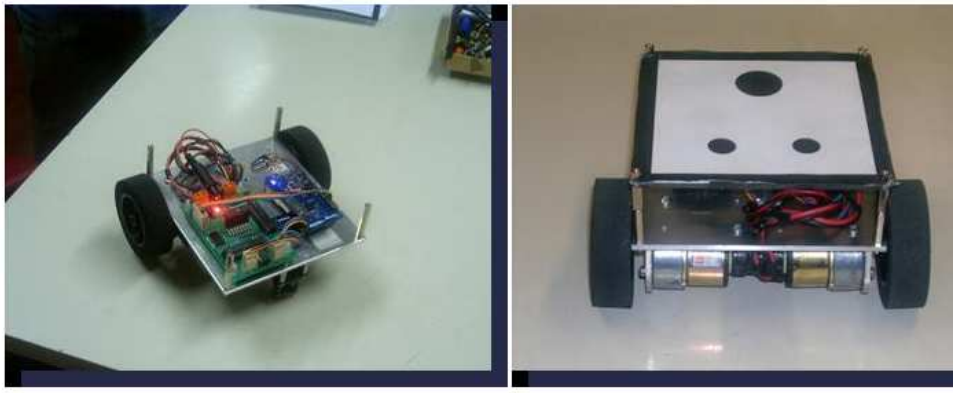


Figure 5.6: Prototype of an autonomous vehicle in the experimental testbed.

Chapter 6

Conclusions

In this thesis is concerned with the problem of designing decentralized policies for coordinating multi-agent systems. A decentralized extension of an already existing policy for collision avoidance has been provided. Conditions for the existence of collision-free maneuvers have been reported for a system up to 5 agents. A new highly scalable decentralized policy, called Generalized Roundabout policy, has been designed to manage a remarkable number of agents moving on a planar environment, at constant speed and with limited curvature radius, to reach their final configurations. Safety and liveness properties of GR policy have been studied. The former has been proved analitically, while for the latter a probabilistic approach has been addressed. To this purpose software simulation tools and test-bed platforms have been developed in order to run a large number of simulations. The problem concerning the navigation of an autonomous vehicle in a partially or totally unknown environment has been investigated with the purpose of defining a strategy for optimizing the information gathered. A closed-form solution and a numerical implementation of the optimal strategy, have been proposed.

Bibliography

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules", *IEEE Trans. Aut. Control*, Vol. 48, No. 6, June 2003, pp. 988-1001.

[2] H. Tanner, A. Jadbabaie, and G. J. Pappas. "Flocking agents with varying interconnection topology", *Submitted to Automatica*.

[3] R. Olfati-Saber. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory", *Submitted to the IEEE Trans. on Automatic Control* (Technical Report CIT-CDS 2004-005).

[4] C. Reynolds. "Flocks, Herds and Schools: A Distributed Behavioral Model". *Computer Graphics*, 21(4), pp 25–34, 1987.

[5] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. "Stable flocking of mobile agents, Part II:Dynamic topology". In *Proceedings of the IEEE Conference on Decision and Control*, 2003.

[6] J. W. Fenwick, P. M. Newman, and J.J. Leonard. "Cooperative Concurrent Mapping and Localization", *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, pp. 1802–1809, 2002.

[7] S. Thrun, W. Burgard, and D. Fox. "A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots", *Machine Learning and Autonomous Robots*, nr. 31/5, 125, 1998.

[8] A. Bicchi and L. Pallottino. "On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems". *IEEE Transactions On Intelligent Transportation Systems*, vol. 1, no.4, pp.221–231, Dec. 2000.

[9] E. Frazzoli, Z.H. Mao, J.H. Oh and E. Feron. "Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming". *AIAA, Journal of Guidance, Control and Dynamics*, 2001.

[10] P. K. Menon, G. D. Sweriduk, and B. Sridhar. "Optimal strategies for free-flight air traffic conflict resolution". *Journ. of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 202-211, MarchApril 1999.

[11] L. Pallottino, E. Feron and A. Bicchi. "Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming". *IEEE Transactions On Intelligent Transportation Systems*, vol. 3, no.1, pp.3–11, March 2002.

[12] S. M. LaValle, and S. A. Hutchinson. "Optimal motion planning for multiple robots having independent goals", *IEEE Trans. on Robotics and Automation*, Vol. 14, No. 6, 1998, pp. 912-925.

[13] C. Tomlin, G. J. Pappas, and S. Sastry. "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems". *IEEE Trans. on Aut. Contr.*, vol. 43, no. 4, pp. 509-521, April 1998.

[14] K. Azarm and G. Schmidt. "Conflict-Free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation", IEEE Int. Conf. on Robotics and Automation, pp 3526-3533, 1997.

[15] L. Chun, Z. Zheng and W. Chang. "A Decentralized Approach to the Conflict-Free Motion Planning for Multiple *Mobile Robots", IEEE Int. Conf. on Robotics and Automation*, pp 1544–1549, 1999.

[16] J.P. Desai, J. Ostrowski and V. Kumar. "Controlling Formations of Multiple Mobile Robots", *IEEE Int. Conf. on Robotics and Automation*, pp 2864–2896, 1998.

[17] H. Yamaguchi and J.W. Burdick. "Asymptotic Stabilization of Nonholonomic Mobile Robots Forming Group Formations", *IEEE Intern. Conf. on Robotics and Automation*, pp 3573–3580, 1998.

[18] RTCA Task Force 3, "Final tech. rep.: Free flight implementation Tech. Rep". *Radio Technical Commission for Aeronautics, Washington*, D.C., October 1995.

[19] Ed. Board, " Special report on free flight". *Aviation Week and Space Technology*, July 1995.

[20] L. Pallottino, E. Feron, and A. Bicchi. "Mixed Integer Programming for Aircraft Conflict Resolution". *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2001.

[21] L. Pallottino, A. Bicchi, and S. Pancanti. "Safety of a decentralized scheme for Free-Flight ATMS using Mixed Integer Linear Programming". *American Control Conference*, pages 742-747, May 2002.

[22] V. J. Lumelsky, and K. R. Harinarayan. "Decentralized motion planning for multiple mobile robots: the cocktail party model", *Autonomous Robots*, Vol. 4, No. 1, 1997, pp. 121-35.

[23] E. Klavins. "Communication Complexity of Multi-Robot Systems", *Proc. Fifth International Workshop on the Algorithmic Foundations of Robotics*, Nice, France, 2002.

[24] L. Pallottino, V. G. Scordio, E. Frazzoli, and A. Bicchi. "Decentralized Cooperative Conflict Resolution for Multiple Nonholonomic Vehicles". *In Proc. AIAA COnf. on Guidance, Navigation and Control*, 2005.

[25] L. Pallottino, V.G. Scordio, E. Frazzoli, and A. Bicchi. "Probabilistic verification of a decentralized policy for conflict resolution in multi-agent systems". *In Proc. IEEE Int. Conf. on Robotics and Automation*, 2006.

[26] R. Tempo, G. Calafiore, and F. Dabbene. "Randomized Algorithms for Analysis and Control of Uncertain Systems". *ser. Communications and Control Engineering*, Springer-Verlag, 2003.

[27] L. Pallottino, V.G. Scordio, E. Frazzoli, and A. Bicchi. "Decentralized and scalable conflict resolution strategy for multi-agents systems". *In Int. Symp. on Mathematical Theory of Networks and Systems*, 2006.

[28] A. Bicchi, F. Lorussi, P. Murrieri, and V. G. Scordio. "On The Problem of Simultaneous Localization, Map Building, and Servoing of Autonomous Vehicles". *In B. Siciliano, A. De Luca, C. Melchiorri, and G. Casalino, editors, Advances in Control of Articulated and Mobile Robots, number 10 in STAR - Springer Tracts in Advanced Robotics*, pages 223-239. Springer Verlag, 2004.

[29] A. Garulli, and A. Vicino. "Set Membership Localization of Mobile Robots via Angle Measurements", *IEEE Transaction on Robotics and automation*, Vol. 17, nr. 4, AUGUST 2001

[30] C. R. Rao. "Linear Statistical Inference and Its Applications", *Wiley*, New York, 1973.

[31] M. Montemerlo, S. Thrun, D. Koller, and Ben Wegbreit. "A Factored Solution to the Simultaneous Localization and Mapping Problem", *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, 2002.

[32] J. E. Guivant and Eduardo Mario Nebot. "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time", *Proc. IEEE*

[33] S. Thrun, D. Koller, Z. Ghahmarani, and H. Durrant-Whyte. "SLAM Updates Require Constant Time", *Tech. rep., School of Computer Science*, Carnegie Mellon University, 2002.

[34] F. Gambino, G. Oriolo, and G. Ulivi. "A comparison of three uncertainty calculus techniques for ultrasonic map building", *SPIE*, Orlando, pp. 249-260, 1996.

[35] F. Gambino, G. Ulivi, and M. Vendittelli. "The transferable belief model in ultrasonic map building" *6th IEEE Conference on Fuzzy Sys*, Barcellona, pp. 601-606, 1997.

[36] L. Jetto, S. Longhi, and D. Vitali. "Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted Kalman filter", *Control Engineering Practice*, vol. 7, pp. 763–771, 1999.

[37] A. Bicchi, D. Pratichizzo, A. Marigo, and A. Balestrino. "On the observability of mobile vehicles localization". *Proc. IEEE Mediterranean Conf. On Control And Systems*, 1998.

[38] F. Lorussi, A. Marigo, and A. Bicchi. "Optimal exploratory paths for a mobile rover", *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2078-2083, 2001.

[39] S. Pancanti, L. Leonardi, L. Pallottino, and A. Bicchi. "Optimal control of quantized input systems", *C. Tomlin and M. Greenstreet, editors, Hybrid Systems: Computation and Control*, volume LNCS 2289 of Lecture Notes in Computer Science, pages 351-363. *Springer-Verlag*, Heidelberg, Germany, 2002.

[40] J. Esposito, R.V. Kumar, and G.J. Pappas. "Multi-Agent Hybrid System Simulation". In *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001, Volume 1, pages 780–785.