Universitat de Girona

INTROSPECTION ON CONTROL-GROUNDED CAPABILITIES. AN AGENT-INSPIRED APPROACH FOR CONTROL

Christian Giovanny QUINTERO MONROY

ISBN: 978-84-691-0036-3
Dipòsit legal: GI-1528-2007

University of Girona
Department of Electrical Engineering, Electronics and Automation

Introspection on Control-grounded Capabilities.

An Agent-inspired Approach for Control.

By

Christian Giovanny Quintero Monroy

DOCTORAL THESIS

Advisors:
Dr. Josep Ll. de la Rosa
Dr. Josep Vehí

Girona, Catalonia, Spain
September 2007

University of Girona
Department of Electrical Engineering, Electronics and Automation

Introspection on Control-grounded Capabilities.

An Agent-inspired Approach for Control.

A dissertation presented to the University of Girona in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY

By

Christian Giovanny Quintero Monroy

Advisors:

Dr. Josep Ll. de la Rosa

Dr. Josep Vehí

Girona, Catalonia, Spain
September 2007

University of Girona
Department of Electrical Engineering, Electronics and Automation

ABSTRACT

Introspection on Control-grounded Capabilities.

An Agent-inspired Approach for Control.

By Christian Giovanny Quintero Monroy

Advisors: Dr. Josep Ll. de la Rosa
Dr. Josep Vehí

Introspective reasoning on physical agents' dynamics will have an important impact on both individual and cooperative decisions in multi-agent environments. Introspection, a self-reflection process that plays a central role in human reasoning and currently a type of cognitive ability coming from the agent metaphor, allows agents to be aware of their capabilities to perform correctly the tasks proposed by other agent-based entities or humans. Agents can then make better decisions. Introspection, mainly on physical constraints or capabilities related to dynamics, provides agents a reliable reasoning for achieving sure and trustworthy commitments in cooperative systems by means of more intelligent self-control. To that end, control-grounded capabilities, inspired by the agent metaphor, are used in this approach. Such control-grounded capabilities guarantee an appropriate and explicit agent-oriented representation of the dynamics, specifications and other relevant details encapsulated in every automatic controller of a controlled system. Currently, the conventional control techniques tend to either ignore or do implicit and naïve suppositions on the dynamics of the controllers. In this sense, it is looked for an integration vision of the agent with the environment because the agent’s physical body, its intelligence, as well as the environment itself, are continuously interacting.

This new approach is a challenge, as it changes and improves the way how agents can coordinate with each other to perform the proposed tasks and how they manage their interactions and commitments in real cooperative environments. The approach is tested on several scenarios where coordination is relevant, beneficial and necessary. Experimental results and conclusions emphasizing the advantages and importance of introspection in the improvement of multi-agent performance in coordinated tasks and task allocation problems are presented.

Acknowledgments

I would like to express my gratitude to my advisors: Josep Lluís de la Rosa and Josep Vehí for giving me the opportunity of starting my Ph.D. studies at the University of Girona, and for showing interest in my work with many valuable comments and enthusiastic support to improve it. They have invested considerable time and effort in guiding me in the right direction.

I am very grateful to Rodolfo Villamizar, Gustavo González and Luis Mujica who, in the first steps of my Ph.D. adventure, gave me a lot of useful and helpful suggestions to increase my experience as a researcher.

Likewise, I would like to thank to all members of the ARLab and MICELab, research groups of the University of Girona, for their friendship, support and encouragement during this time.

Special thanks to Salvador Ibarra, Silvia Suárez, Sonia Delfín, Jorge Flórez, Gabriel Lopardo and Martha García who, have made more comfortable my life in Girona as well as many people who directly or indirectly have collaborated with me with their brilliant ideas. I apologize for not listing everyone here.

Finally, I would like to express my gratitude and endless love to my family, who has always trusted me and has encouraged me to finish this challenge in my life.

Dedicated to my lovely family….

General Contents

Detailed Contents

PART I: INTRODUCTION AND RELATED WORK

PART II: PROPOSED APPROACH

PART III: RESULTS AND CONCLUSIONS

List of Figures

List of Tables

PART I
INTRODUCTION
AND
RELATED WORK

Chapter 1

Introduction

This chapter provides an introduction to the work presented in this thesis. Specifically, the motivation in the research area, the pursued aims and the main contributions are briefly described. Finally, the chapter concludes with an overview of the structure and contents of the thesis.

1.1 Motivation

Several recent efforts in automatic control are related to building computer-controlled systems able to solve some well-known control challenges [Halang et al., 05] [Murray et al., 03]. High levels of control, coordination and autonomy are looked for in distributed, asynchronous and networked environments. However, the recent approaches have a great deal of complexity that makes them less applicable to real-life problems. Several aspects induce the above complexity: real-time requirements, embedded and limited resources, fault-tolerant behaviors, distributed and heterogeneous components, artificial intelligence tools and large-scale structures [Sanz et al., 03]. Complex control systems are therefore, in most cases, intensive software applications and highly sophisticated control algorithms that use advanced design technologies. Moreover, these systems have generally requirements that go beyond single disciplines (*from control engineering to computer sciences*), further increasing their complexity [Sanz et al., 03]. Unfortunately, more complexity does not necessarily mean better performance, and a reorientation effort by the control research community in this respect seems necessary.

In the last years, there has been some work toward combining *A*rtificial *I*ntelligence *(AI)* approaches with traditional control theory to obtain intelligent systems. In this direction, the advances of the *AI* community in planning, adaptation, learning, logic-based theories and knowledge representation together with the techniques in the control community for modeling, analysis and design of control systems, have presented a fresh path for further progress [Murray et al., 03]. In particular, some research trends have led to managing complex control systems using agents. Agents are defined as computer systems capable of flexible and autonomous actions in dynamic, unpredictable and typically cooperative environments [Luck et al., 05]. Nevertheless, agents are also generally referred as a design metaphor. Nowadays, complex control systems must be then considered as a multi-agent system that requires coordination and cooperation to achieve global goals [Jennings and Bussmann, 03] [Stone and Veloso, 00].

Several results have been obtained for control systems designed using agent technology [Jennings and Bussmann, 03]. Agent technology helps to solve complex problems in real control scenarios by means of its cooperative problem-solving paradigm. However, **these agents lack an appropriate reasoning on their knowledge about the physical features of the controlled system**. In this thesis, such physical knowledge is directly related to the inertial dynamics and the specifications, structure and other relevant details encapsulated in the automatic controllers of the controlled system. Moreover, **such relevant knowledge is not appropriately reflected and communicated by the agents. These lacks do not allow agents to make feasible joint decisions when these are requested**. Explicitly, **lack of appropriate reasoning on physical knowledge results in a lower cooperative performance between agents, especially in coordinated tasks and task allocation problems where a proper**

communication of such information is quite relevant to achieve sure and trustworthy commitments. In fact, these lacks are currently significant impediments to reducing complexity and achieving appropriate levels of control, coordination and autonomy in control systems [Murray et al., 03].

Here, the controlled system's dynamics are mainly related to two sources: the inertial dynamics of the system components and the dynamics of the task execution resulting from the actions of automatic controllers.

The thesis argues that in the near future, any autonomous system (e.g., cars, aircrafts, mobile robots, house artifacts) controlled by agents will only complete its tasks correctly and make proper decisions, if it is able to reflect, consider and communicate its knowledge on its physical capabilities taking into account its dynamics.

For instance, Fig. 1.1 shows two agent-controlled robots trying to pass a ball between them in robot soccer. The robots have an obstacle-free movement trajectory and have a set of controllers to move in the environment. The passer must strike the ball towards the interception point in a suitable way. The shooter must intercept and shoot the ball with the intention of scoring in the opposite goal. Thus, the passer and the shooter must coordinate to perform the task successfully.

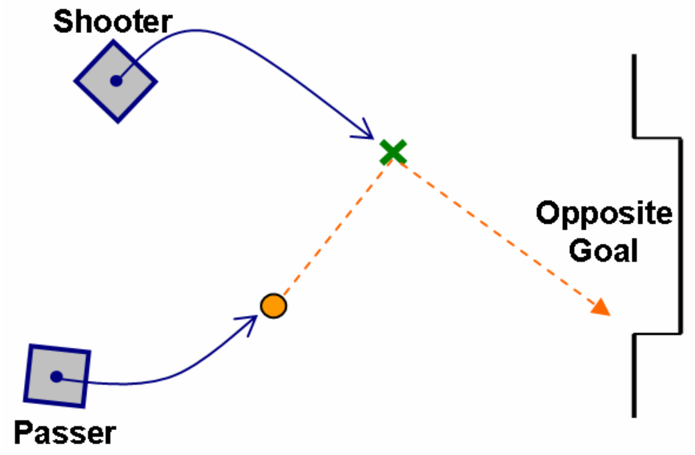


Fig. 1.1. General scheme of passing a ball in robot soccer.

In light of this, the passer can propose a pass to the shooter with specific spatial and temporal requirements (See Fig. 1.2a). The shooter must then look for and evaluate its capabilities to perform such task with its available controllers according to the established conditions (See Fig. 1.2b). The shooter tells later the passer that it can or cannot perform the proposed task (See Fig. 1.2c). Depending on the shooter reflection on its knowledge related to the dynamics resulting from the actions of its controllers, the robots can undertake or not the execution of the task (See Fig. 1.2d).

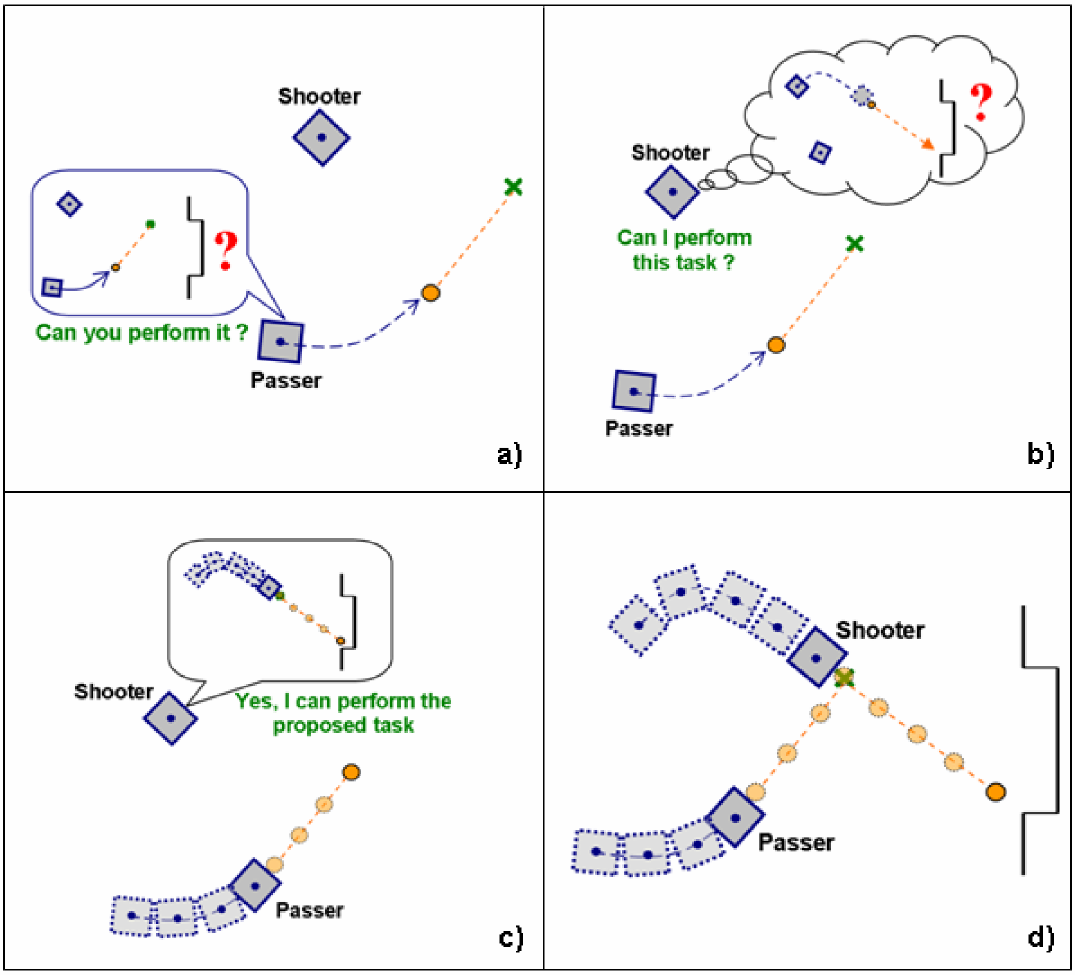


Fig. 1.2. The task of passing a ball in robot soccer: a). An agent (passer) proposes passing the ball to other agent (shooter); b). Shooter evaluates its capabilities to perform the proposed task; c). Shooter tells that it can perform the task d). Agents commit and they can perform the task successfully.

However, several passes are not physically feasible due to robots' physical limitations. In particular, sometimes there are not controllers to execute a pass or the robots' dynamics do not allow it. In this sense, robots must agree the type of control to apply for the pass and the moment to execute it, based on the knowledge they have of their dynamics. The proposal is to ensure passing between robots by physically achievable decisions. For some undesirable situations, the control system must therefore be redesigned to satisfy the dynamics of the robots. Such redesign induces generally more complexity than does not necessarily impact in a better performance.

This thesis argues that the above alternative is not a good solution. Therefore, **a solution beyond good control and perfect controllers must be implemented. In this sense, an alternative is that agents reason on their knowledge related to the robots' dynamics and consider this knowledge in their decision-making**. The agents can then negotiate, make coordinated decisions and modify their actions according to the information about these dynamics. Thus, agents' decisions will be inhibited whenever the robots' dynamics do not allow the execution of the proposed actions and will be renegotiated until the agents agree.

Therefore, explicit reasoning on robots' dynamics in the agents' decision-making will prevent, most of the time, undesirable situations. As it has been mentioned before, the dynamics of the robots' physical bodies can be modified by their automatic controllers. Here, agents are then proposed to be aware of the set of controllers of their physical bodies. In this sense, control engineers need practical tools for developing this new type of agents and their controllers, taking into account their dynamics.

Similarly, cooperative robots and humans working jointly in search and rescue operations [Murphy, 04] [Davids, 02] (see Fig. 1.3a) could optimize their multi-agent team work coordination if the robots know and they are able to reflect and communicate their knowledge on their physical limitations or capabilities.

For instance, consider a search team of three agent-controlled mobile robots trying to sweep a disaster zone within a fixed deadline. These robots have three different movement controllers, and their control laws can generate different dynamics. Therefore, the same search operation can be performed in different ways by the robots within the required time (see Fig. 1.3b). However, if the temporal constraints change, some robots cannot perform correctly the proposed tasks (see Fig. 1.3c). Thus, an alternative is that agents reflect on their knowledge related to the robots' dynamics and consider this knowledge in their decision-making to find a suitable task allocation in search and rescue operations (see Fig. 1.3d).

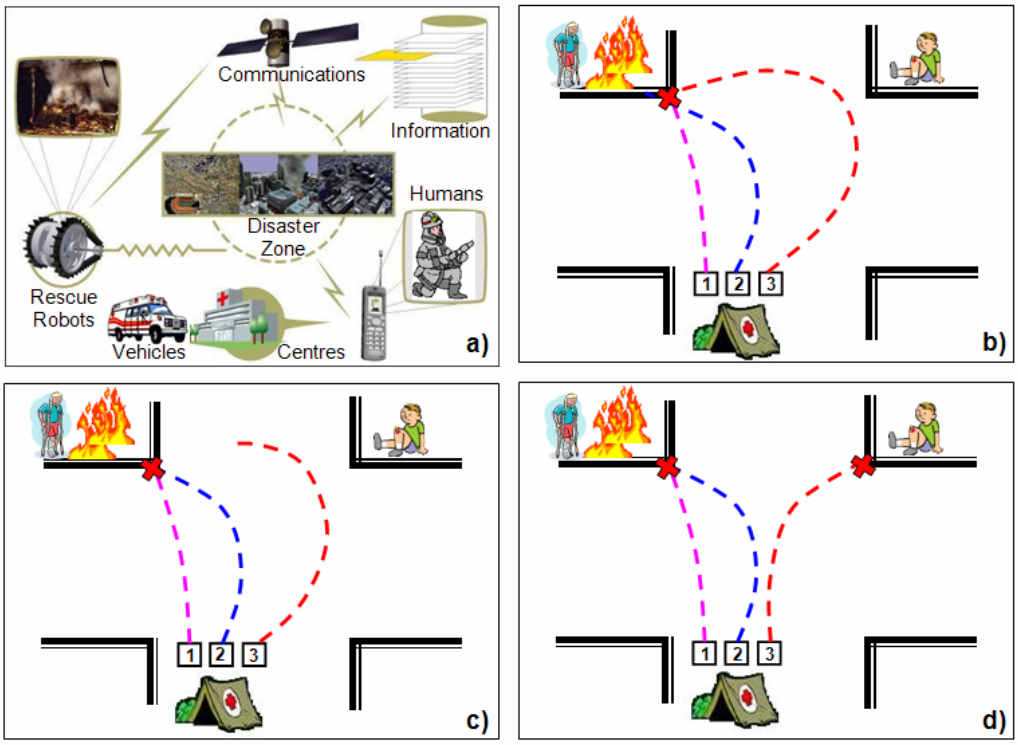


Fig. 1.3. a). General scheme of a rescue system; b). Robots trying to move towards the disaster zone; c). Only some robots can reach the disaster zone when the temporal constraints change; d). A new task allocation based on the robots' dynamics.

In summary, **agents do not reflect on their knowledge related to the controlled systems' dynamics and this knowledge is not currently properly taken into account in the agents' decision-making**. The thesis then states that reflection on dynamics is an interesting agent-oriented perspective implemented in automatic control scenarios.

In particular, the above control-oriented knowledge is directly related to the automatic controllers specifications established for a controlled system by the control engineer's criteria. To incorporate appropriately all this embedded information (*mainly* *about dynamics*) in rational decision-making, it must be first developed a suitable representation for dynamics in the agents' knowledge base which is general, accessible, understandable, comparable and computationally tractable for these agents. This agent-oriented representation makes it easier for agents to manage and communicate the controlled system's dynamics aiming at making physically feasible decisions. Such decisions improve the multi-agent performance in cooperative scenarios.

Physical agents are particular examples of controlled systems [De la Rosa et al., 07]. Here, physical agents are understood as physical and encapsulated entities with control architectures that satisfy the agent design metaphor. In recent years, mobile robots one typical representation of physical agents, have become progressively more autonomous and cooperative. So, mobile robots are used in this approach without loss of general applicability. Such autonomous mobile cooperating robots must then have reliable self-knowledge if they are to improve their performance when executing coordinated tasks. This self-knowledge must be based on an appropriate agent-oriented representation of the physical agents' dynamics in the knowledge bases. With this representation, any physical agent could reflect and consider appropriately its physical body whenever it is committed to carry out a task or assume specific behaviors in a multi-agent scenario. Thus, a physical agent is an intelligent entity, and its actions and cooperation with other agents or humans, to achieve the desired goals in a real environment, are limited and conditioned by the dynamic behavior of its physical body. Intelligence is here understood as the appropriate exploitation of knowledge about dynamics to perform better [Sanz et al., 01] and achieve enhanced levels of performance and autonomy [Sanz et al., 00].

In this sense, the thesis proposes then an introspection approach to provide agents a cognitive ability for reasoning on their dynamics, aiming at making physically feasible decisions and getting reachable and physically grounded commitments to improve the cooperative multi-agent performance. To that end, control-grounded capabilities, inspired by the agent metaphor, are used in this approach. Such control-grounded capabilities guarantee an appropriate and explicit agent-oriented representation of the dynamics, specifications and other relevant details encapsulated in every automatic controller. As will be shown, the research on introspection on control-grounded capabilities proves the impact of this agency property, and its effectiveness in cooperative intelligent agents.

1.2 Objectives

The research addressed in this dissertation is focused on including knowledge on physical agents' dynamics in their decision-making. Such challenge has been worked from a control-oriented viewpoint.

1.2.1 Thesis Question

The principal question addressed in this dissertation is:

Can physical agents make physically feasible joint decisions to obtain sure and trustworthy commitments and improve the multi-agent performance in coordinated control environments when they include physical knowledge, mainly related to their dynamics, in their decision-making?

More specifically, the thesis presents an appropriate alternative to include control-oriented knowledge in the physical agents' decision-making and represent explicitly such knowledge in a set of control-grounded capabilities. The thesis looks for then to bridge the gap between the high abstraction level of agents and the low abstraction level of the automatic control architectures.

It was necessary to fulfill the following goals to achieve the aim of the thesis:

- To look for a way of taking advantage of control-oriented information related to the physical agents’ dynamics and outline the way of including such knowledge in the agents’ decision-making.

- To determine relevant control-oriented knowledge related to the physical agents’ bodies (mainly about their automatic controllers) to obtain reliable low level information to use in the high level decision-making.

- To establish the requirements that the above control-oriented knowledge must achieve to be a reliable agent-oriented representation and a useful decision tool.

- To demonstrate the utility and feasibility of the overall proposed approach on several examples of coordination in physical multi-agent environments.

1.2.2 Approach

The primary interests in this research to answer the thesis question include the following topics:

- Introspective Reasoning on Physical Agents' Dynamics

- Control-grounded Capabilities inspired by the Agent Metaphor

Introspection, a self-reflection process that plays a central role in human reasoning and currently a type of cognitive ability coming from the agent metaphor, allows agents to be aware of their capabilities to perform correctly the tasks proposed by other agent-based entities or humans. Agents can then make better decisions. Introspection, mainly on physical agents’ dynamics, provides agents with a reliable reasoning for achieving sure and trustworthy commitments in cooperative systems by means of more intelligent self-control.

Control-grounded capabilities constitute the proposal aimed at closing the gap between agents and the low abstraction level of automatic control architectures. These capabilities, inspired by the agent metaphor, aim at guaranteeing an appropriate and explicit agent-oriented representation regarding the dynamics, specifications, structure and other relevant details encapsulated in every controller. Otherwise, this specific embedded knowledge about every controller is not taken into account and never reused to achieve better agent cooperation. This cooperation is measured in terms of physically grounded and reliable commitments which results in a better performance of any group of cooperating agents.

1.3 Contributions

This thesis makes the following contributions:

- A formal design methodology based on introspective reasoning to use control-oriented knowledge in an agent-oriented manner.

- A formulation based on control-grounded capabilities to represent explicitly control-oriented information of agent-controlled systems.

- A decision-making tool based on introspection on control-grounded capabilities as a bridge to the gap between the high abstraction level of agents and the low abstraction level of the automatic control architectures.

1.4 Reader's Guide to the Thesis

Following is a general description of the contents of this dissertation. This doctoral thesis is organized in three main parts constituted by several chapters.

Part I: Introduction and Related Work

Chapter 1 presented a motivational introduction on the main topics, objectives and contributions regarding this dissertation.

Chapter 2 gives a general overview of background information regarding artificial intelligence, agent technology and robotics which is required to develop the agent-inspired approach described in chapters 4 and 5.

Chapter 3 provides a general survey of the most relevant work related to the research addressed in this thesis.

Part II: Proposed Approach

Chapter 4 describes the formal aspects of the novel introspection approach presented in this thesis.

Chapter 5 presents the implementation on several test beds of the approach proposed in chapter 4. The chapter also contributes to complete the description of such proposal.

Part III: Results and Conclusion

Chapter 6 provides experimental results of the implemented approach. Empirical evaluations that evidence the utility, feasibility and reliability of the overall approach are provided in this chapter.

Chapter 7 discusses and analyses the results, summarizes the conclusions and contributions of this thesis and outlines the most promising directions for future work.

Chapter 2

Background Information

This chapter introduces and reviews general concepts of agents, multi-agent systems and robotics, such that an agent-inspired approach for automatic control scenarios is proposed and discussed later.

2.1 Agent Technology

In recent years, agent technology is one of the most relevant and useful contribution in the *I*nformation *T*echnology *(IT)* world. Agent-based systems emerge as an appropriate alternative to improve the traditional computing and the current algorithms and software applications especially in dynamic and open environments, where heterogeneous systems must interact effectively to achieve specific goals. In this sense, agent-oriented developments are seen as fundamental to enable systems to respond in a suitable, effective and reliable way to changing conditions while trying to achieve the design objectives.

The agent paradigm has found currency in several sub-disciplines of information technology, including computer networks, software engineering, **artificial intelligence**, human-computer interaction, distributed and concurrent systems, mobile systems, telematics, computer-supported cooperative work, **control systems**, decision support, information retrieval and management, and electronic commerce [Luck et al., 05]. In particular, agent technology offers fundamentally **new ways of design**, standardization and support for *IT* applications through distinct and independent software components interacting to provide better performance and valuable functionality. In such context, agent technology constitutes a proper way to conceptualise and implement the present and future computer systems.

2.2 Agent Concept

There are several agent definitions in the current literature and introducing the concept of agent in a precise and technical manner is difficult. The agent concept is a general abstraction appropriated to a large range of applications. However, several criteria allow distinguishing between what is an agent and what is not at an engineering level. Such criteria are based on a reasonable model of the agents' features and behaviors. In this sense, some of the most cited definitions are highlighted.

Agents can be defined as computer systems capable of flexible and autonomous actions in dynamic, unpredictable and typically multi-agent domains [Luck et al., 05].

More specifically, ***agents can be defined as autonomous and problem-solving computational entities capable of effective operation and flexible autonomous actions in dynamic, unpredictable and open environments***. Agents are often deployed in environments in which they interact, and maybe cooperate, with other agents that have possibly conflicting aims. Such environments are known as multi-agent systems [Luck et al., 03].

In addition, ***an agent denotes a software-based computer system that has several properties as autonomy, introspection, social ability, reactivity, pro-activeness, mobility, rationality, etc., which is capable of independent action to achieve some goals or desires*** [Wooldridge, 02].

In summary, agents are [Jennings and Bussmann, 03]:

1. Clearly identifiable problem-solving entities with well-defined boundaries and interfaces.

2. Situated (embedded) in a particular environment over which they have partial control and observability.

3. Designed to fulfill a specific role, they have particular objectives to achieve.

4. Autonomous, they have control over both their internal state and their own behavior.

5. Capable of exhibiting flexible problem-solving behavior in pursuit of their design objectives, being both reactive (able to respond in a timely fashion to changes that occur in their environment) and proactive (able to opportunistically adopt goals and take the initiative).

To avoid confusions with other agent meanings and contexts, the above agents are also commonly known as *software agents*.

2.3 Agent Metaphor

Agent technology finds a stronger applicability when is used as a design metaphor of well-structured approaches for solving real-life *IT* challenges. Currently, ***agents provide software designers and developers an appropriate way of structuring software tools and applications around autonomous, communicative, situated and problem-solving entities to achieve the required design goals*** [Jennings, 01]. In this sense, the agent metaphor offers a promising route to the development of computational systems, especially in open and dynamic environments of several real-world domains [Luck et al., 05]. In addition, ***the agent concept provides elegant tools/methods for abstraction and encapsulation.***

2.4 Agent Architectures

The internal structure of an agent is determined by its control architecture. The architecture determines the mechanisms used by an agent to interact under external and internal conditions given some specifications of its desired behavior.

There are several control architectures that allow describing the internal structure of an agent. However, four main perspectives can be mentioned: the deliberative (*think hard, then act*), the reactive (*don't think, react*), hybrids of the above two (*think and act independently, in parallel*) and a behavior-based strategy (*think the way you act*). Deliberative and reactive architectures embrace two basic ideas related to the agent concept respectively: the need of deliberation for long term reasoning based on a symbolic knowledge representation, and quick answers for suitable agent's behaviors according to the current situation.

A relevant deliberative architecture is the *BDI* (*Belief-Desired-Intention*) architecture [Rao and Georgeff, 95]. The *BDI* model has been developed to provide solutions in uncertain and dynamic environments where agents have a partial knowledge of the problem and usually manage limited resources. Beliefs, desires and intentions constitute then important parts of the agents' state in these systems under the above conditions. The beliefs represent the domain knowledge embedded in the agents. The desires represent the objectives or the expected final state. Additionally, it is necessary to define a planning mechanism that allows identifying the agents' intentions to reach the pursued objectives taking into account the current beliefs. In this sense, the plans related to the attainment of objectives constitute the intentions. The type of modelling used by a deliberative agent is usually very elaborate.

However, the associated problem to a symbolic representation has led to the study of more effective models for knowledge representation. In this sense, reactive architectures are an alternative. Subsumption [Brooks, 91] is a relevant reactive architecture. Such architecture is based on the hypothesis that "intelligence" is an emergent property of some complex systems and it allows generating suitable behaviors without symbolic models or any internal representation of the environment. Agents react to the current sensory information in a "stimulus-response" manner. This allows agents to respond very quickly to changing and unstructured environments. The Subsumption architecture manages a hierarchy of tasks for defining the agent's behaviors and they are usually organized in layers from a low to a high abstraction level. A great amount of applications of this type of architecture is found in the development of controllers in robotics. In this sense, Subsumption architecture for mobile robots is based on a given priority to different controllers under different circumstances. Here, robots can be considered as real or physical agents that act in a real and changing environment. The need of interacting in an unpredictable environment favours the adoption of reactive architectures. Limitations to this approach are that such robots, because they only look up actions for any sensory input, do not usually keep much information around, have no memory, no internal representation around them, and no ability to learn over time.

In addition, there have been some proposed hybrid architectures [Ferguson, 92] [Müller, 97] [Low et al., 02] aimed at combining aspects related to deliberative and reactive architectures and to overcome their limitations. Such architectures adopt a layered organization generally distributed in three abstraction levels [Mas et al., 05]: Reactive (low level) is related to decision-making based on real time environment conditions. Knowledge (intermediate level) is related to the domain knowledge based on a symbolic representation of the environment. Social (high level) is related to social aspects in the environment, exchange information between agents, etc. The agent's global behavior is then defined by the interaction between all the above levels. However, such interaction could be different for different hybrid architectures.

On the other hand, behavior-based approaches [Arkin, 98] [Matarić, 99] are an extension of reactive systems that fall between the purely reactive and the planner-based extremes. The behavior-based approach is a methodology for designing autonomous agents and robots. The behavior-based methodology imposes a general, biologically inspired, bottom-up philosophy, allowing for a certain freedom of interpretation. Its goal is to develop methods for controlling artificial systems (usually physical robots, but also simulated robots and other autonomous software agents) and to use robotics to model and better understand biological systems. In behavior-based approaches, the decomposition of the control system is performed in a task-oriented manner. Unlike reactive systems, behavior-based systems are not limited in their expressive and learning capabilities: behaviors themselves can have a state (internal and particular view of the world), and can form representations when networked together.

2.5 Multi-agent Systems

Several approaches, where a number of entities work together to cooperatively solve problems, fall into the area of distributed systems. The combination of distributed systems and artificial intelligence is collectively known as *D*istributed *A*rtificial *I*ntelligence *(DAI)*. Traditionally, *DAI* is divided into two areas [Stone and Veloso, 00]. The first area, distributed problem solving, is usually concerned with the decomposition and distribution of a problem-solving process among multiple slave components, and the collective construction of a solution to the problem. The second area, *M*ulti-*A*gent *S*ystems *(MAS)*, emphasizes the joint behaviors of agents with some degree of autonomy and the complexities arising from their interactions [Panait and Luke, 05].

In recent years, multi-agent systems have been studied by several research groups. There are also several multi-agent systems definitions. The most widely accepted definitions are here summarized.

Multi-agent systems are systems with a varying number of interacting, autonomous agents that communicate with each other using flexible and complex protocols, in order to achieve particular goals or perform some set of tasks. In multi-agent systems

"the intelligence" arises from the aggregation of simple competitions as well as the task assigned to every individual is as important as the collective task [Weiss, 99].

According to the distributed artificial intelligence, ***a multi-agent system is a network of entities able to solve problems, working jointly to find answer to problems that are beyond the capacity and the individual knowledge of each entity***. Thus, in multi-agent environments, agents must generally coordinate their actions and they must communicate the proper knowledge and information. In addition, there are constraints in a multi-agent environment such that agents may not at any given time know everything about the world that other agents know [Panait and Luke, 05].

In summary, the multi-agent system term is used to define all types of systems with multiple autonomous components that have the following elements and features [Jennings et al., 98]:

- A common environment.
- Agents.
- Interaction among agents.
- Interactions among agents and dynamic environment.
- Each agent has the capacity to solve the problem partially.
- There is no a global control system.
- The data are not centralized.
- The computation is asynchronous.

Three common types of interactions are described:

- Cooperation: working together towards a common goal.
- Coordination: organising problem solving activities so that harmful interactions are avoided and beneficial interactions are exploited.
- Negotiation: coming to an agreement which is acceptable to all the parties involved.

Agents interact to share information and achieve the proposed tasks and objectives in cooperative environments. In this sense, the interaction is understood as a mechanism to articulate the cooperation, coordination and negotiation between agents.

There are three key elements to obtain the multi-agent interaction:

- A common language and communication protocol.

- A common communication format.

- A shared ontology.

2.5.1 Coordination

Coordination refers to ensuring that the actions of independent agents in an environment are coherent in some way [Luck et al., 05]. The most widely accepted definition of coordination has its origins in the organization theory. In this sense, ***coordination is the management of dependences between organizational activities*** [Malone and Crowston, 94]. Taxonomy of such dependences and a set of coordination actions assigned for each dependence must be established according to the multi-agent system's features. [D'Inverno and Luck, 04] presents a formalization of possible different relations between agents in multi-agent environments. Thus, the coordination process is related to the attainment of two main tasks: To establish the dependences and to make decision on which coordination action must be performed. A coordination mechanism determines the way of how one or several agents perform the above tasks [Ossowski, 99].

From a practical perspective, ***it is possible to understand the coordination as an effort to manage the interactions between agents*** [Busi et al., 01] [Wegner, 97]. From a design perspective, the challenge is how agents can interact in an appropriate way to solve the dependences and make the related decisions. There are several approaches in the literature on the matter [Scerri et al., 04]. Multi-agent scheduling, negotiation, organizational structures, norms, trust, etc., are some of them. The aim of the above approaches is to determine the interaction space. The applications of these mechanisms depend on the characteristics of the coordination problem.

2.5.2 Cooperation and Collaboration

Cooperation refers to coordination with a common goal in mind [Luck et al., 05]. Cooperation between agents has been widely studied in the distributed artificial intelligence field. There are several works related to cooperation [Mayoh, 02] [Watson et al., 02] [Jennings, 00]. These works address the problem from a deliberative architectures viewpoint, though the cooperation has been also studied in reactive agents [Molina et al., 04]. However, there is not a global vision about cooperation and all the current contributions are related to the cooperation advantages from a perspective aimed at answering of how cooperation can be performed, or how agents must interact to cooperate. ***Cooperation embraces the allocation and coordination of tasks***. They are key factors in order that the cooperation arises. In this sense, there are studies focused on methods to allocate tasks between agents in a set of synchronized actions in time and resources. In addition, ***collaboration refers to a suitable allocation of information, tasks and resources between agents in multi-agent systems*** [Ferber, 99] [Lesser, 99]. Such allocation must take into account the agents' capabilities, the tasks' nature and the social structure of the system.

2.5.3 Negotiation

Agents in a multi-agent environment typically have conflicting goals and not all agents may satisfy their respective goals simultaneously. In this sense, agents will need to negotiate with each other to resolve conflicts [Luck et al., 05] [Beer et al., 99]. Recently, several efforts have been devoted to negotiation protocols, resource-allocation methods, and optimal division procedures based on ideas from computer science, artificial intelligence and socio-economic sciences.

Negotiation is a key coordination mechanism for interaction that allows to a group of agents to reach an agreement according to their beliefs, goals or plans.

The negotiation process can be performed of different ways as auctions, contract net, etc. The negotiation consists therefore in reaching an agreement between agents that benefits them when each one has its own interest.

2.5.4 Commitments

A commitment refers to an acquired obligation when an agent interacts with others [Mallya et al., 03]. A need of finding suitable ways to fulfil such commitment then arises. Therefore, agents will base their actions on their capabilities, the capabilities of others and the developed work framework. There are coordination mechanisms that allow an organized way to perform actions in group. Thus, an agent decides to commit to others when it is able to fulfil the proposed tasks, to interact with other agents and to communicate with its action partners.

2.6 Agents and Robots

Robotics is a research field where the agent concept can be directly applied. There is a direct equivalence between robots and agents in a rigorous sense. A robot is a real or physical agent situated in a real environment unlike an agent who just is a software entity. ***Physical agents are then understood as physical and encapsulated entities with control architectures that satisfy the agent design metaphor***.

An agent's architecture in robotics is equivalent to a robot's control architecture [Matellán and Borrajo, 01]. It is necessary to identify a set of actions (agent's capabilities) that allows robot to interact within the environment in all control architectures. The set of capabilities needs different hierarchic levels (grouping of capabilities to achieve a goal) in the control structure [Oller, 02]. Such control levels depend on the features of the tasks to perform and the available resources. There are then mainly two control levels following the above considerations. The high level performs long term reasoning and task planning while the low level performs the easiest tasks, solving the more immediate problems that not need planning.

In summary, the fact that a robot is autonomous and physically independent has driven to the utilization of the agent technology as something slightly natural.

2.6.1 Mobile Robotics and Multi-robot Systems

In particular, ***mobile robotics refers to the application field of robotics where the essential feature of robots is the ability of autonomous motion*** [Oller, 02]. The motion allows the robot the accomplishment of movements in more or less structured environments and forces it to be equipped with specific sensors to know the environment's state.

On the other hand, the study of multiple-robot systems naturally extends research on single-robot systems [Parker, 00] [Cao et al., 97]. ***Multiple-robot systems can accomplish tasks that no single robot can accomplish*** [Arai et al., 02]. Multiple-robot systems are also different from other distributed systems because of their implicit "real world" environment, which is presumably more difficult to model and reason.

There are three general problems to study in the mobile robotics that are relevant in this thesis [Parker, 00]:

- The movement control of the mobile robot like an individual entity.
- The control of a system composed by diverse robots: the cooperation.
- The planning of the actions to perform, depending on the temporal and spatial restrictions.

A more deep and extensive analysis of related works on multi-robot systems and mobile robotics, focused specifically on the research topics addressed in this thesis, is presented in Sections 3.3, 3.4, 3.5 and 3.6.

Chapter 3

Related Work

This chapter presents an overview of the main works focused on the topics addressed in this dissertation.

3.1 Challenges in Control Systems

In recent years, control is an increasingly essential element for managing systems with enormous amounts of data to process and communicate, providing high performance, high confidence, and reconfigurable operation in the presence of uncertainties [Murray et al., 03]. Control systems are then constituted as a heterogeneous collection of physical and information systems. Such systems must have intricate interconnections and interactions with higher levels of decision making especially in dynamic and uncertain environments. In this sense, among the challenges currently facing the field are [Murray et al., 03]:

1. ***New formalisms for ensuring stability, performance, and robustness in the control of distributed, asynchronous and networked environments.***

2. ***Advances toward high-levels of decision-making, coordination and autonomy in control systems if they are to perform reliably in realistic settings.***

3. ***Next-generation systems must combine logical operations (such as symbolic reasoning and decision making) with continuous quantities, i.e., control of systems with both symbolic and continuous dynamics.***

4. ***Researchers need to develop much more powerful design tools that automate the entire control design process. Namely, future engineering systems will require the ability to rapidly design, redesign, and implement control software.***

5. ***Researchers need to build very reliable systems from unreliable parts. Increasingly, this requires designs that allow the systems to automatically reconfigure themselves so that their performance degrades gradually rather than abruptly.***

In particular, advances in robotics are needed in many fields to improve the robots ability to locomote, interpret complex sensory inputs, perform higher level reasoning, and cooperate together in teams. Here, the opportunity to combine the advances of the *AI* community in planning, adaptation, and learning with the techniques in the control community for modeling, analysis, and design of control systems presents a renewed path for progress. Therefore, it is possible to think in the "*agent technology*" as an appropriate alternative to manage complex control system and solve some challenges presented in the modern control systems to obtain better results and performance [Halang et al., 05].

3.2 Agent-based Control Systems

Current control systems are highly complex. They generally have a large number of interacting parts. Designing and implementing such complex control systems as a collection of interacting, autonomous and flexible components (*as agents*) affords software and control engineers several significant advantages over contemporary methods [Jennings and Bussmann, 03].

The role of any new software engineering paradigm is to provide structures and techniques that make the complexity easier to handle. Software and control engineers have several fundamental tools as decomposition, abstraction and organization to help manage this complexity.

When adopting an agent-oriented view, it soon becomes apparent that most problems require or involve multiple agents (*multi-agent systems*). From a control perspective, this view of software systems has several similarities to work on heterarchical system in distributed control. The work on heterarchical control tends to concentrate on the distributed systems nature and the autonomy of the individual components of these control systems. ***The agent metaphor applied in complex control system contributes to satisfy then the above first challenge in automatic control.***

In general, complex systems consist of a number of subsystems organized in a hierarchical structure. Such subsystems work together to achieve the functionality of the whole system. For the autonomous components to fulfill both their individual and collective objectives, they need to interact. However, the system's inherent complexity means it is impossible to a priori know about all potential interactions. For that reason, it is necessary to endow the components with the ability to make decisions about the nature and scope of their interactions in a flexible manner. It is apparent that the natural way to modularize a complex system is in terms of multiple autonomous components that act and interact in flexible ways to achieve their objectives. In particular, given this, ***agent-inspired approaches are simply the best fit to satisfy the second challenge in automatic control.***

In the case of complex systems, subsystems naturally correspond to agent organizations. Agents systems are invariably described in terms of "cooperating to achieve common objectives", "coordinating their actions", or, "negotiating to resolve conflicts". Complex systems involve changing relationships among their various components according to their role in the existing subsystems in the organization.

Agent-oriented approaches emerge then as a suitable alternative to manage control systems according to the third challenge in automatic control.

Additionally, agent-oriented systems are evolutionary and incremental, as legacy software can be incorporated in a relatively straightforward manner. The technique used is to place wrapping software around the legacy code to serve as an agent interface to the other software components. Thus, from the outside, the wrapper looks like any other agent; on the inside, it performs a two-way translation function: taking external requests from other agents and mapping them into calls in the legacy code and taking the legacy code's external requests and mapping them into the appropriate set of agent communication commands. This ability to wrap legacy systems means ***agents may initially be used as an integration technology according to the specifications in automatic control commented in the fourth challenge.***

As new requirements are placed on the system, however, bespoke agents may be developed and added. ***This feature enables a complex system to grow in an evolutionary fashion while continually maintaining a working version of the system and allow systems continue to operate even when individual components fail so that its performance degrades gradually rather than abruptly as is mentioned in the fifth challenge of automatic control.***

In summary, this section has sought to justify precisely why agent-oriented approaches are well suited to developing complex software systems in general and control systems in particular. Agent technology provides a way to conceptualise these systems as comprising interacting autonomous entities, each acting, learning or evolving separately in response to interactions in their local environments [Luck et al., 05] [Jennings and Bussmann, 03].

[Jennings and Bussmann, 03] believes that agent-based systems provide several advantages to the next generation of control systems. They provide a decentralized solution based on local decision making that gives the system a high degree of flexibility, autonomy and robustness.

According to the above advantages of an agent-inspired design for control systems, this thesis presents particularly agent-inspired techniques focused on the control system of the physical agents within cooperative multi-agent environments. Such physical agents are represented in the implementation as cooperative mobile robots. Thus, some relevant related works on this subject are presented next.

3.3 Origins of the Control-grounded Capabilities

The design of a control system for an autonomous physical agent has an important impact on the agent's overall functionality. The control architecture constraints the way a physical agent senses, reasons and acts, thus affecting its task performance. However, an agent-based framework provides a well-structured way for a better understanding of relevant aspects related to the arisen complex control problems. In this sense, Dynamical Physical Agents Architecture *DPA*^{2} [Oller et al., 99] [Oller, 02] is a layered architecture aimed at combining the requirements of control systems architectures with the requirements of multi-agent systems architectures. Currently, physical agents have to fulfil real time and real world requirements when performing tasks in a multi-agent environment. Situated behaviours, goal-oriented behaviours, efficiency and coordination are among them. Such requirements are closely related to the control architectures and the multi-agent design. The *DPA*^{2} uses three main modules (control, supervisor and agent) for integrating the requirements. Fig. 3.1 shows the different layers and the different abstraction levels of this architecture.

According to *DPA*^{2}, the agents must check some external and internal parameters to decide their behaviours after other agents' requests. The external parameters can be obtained by information exchange with other agents. The internal parameters must describe the different states of the physical agents' body, at both low and high abstraction levels.

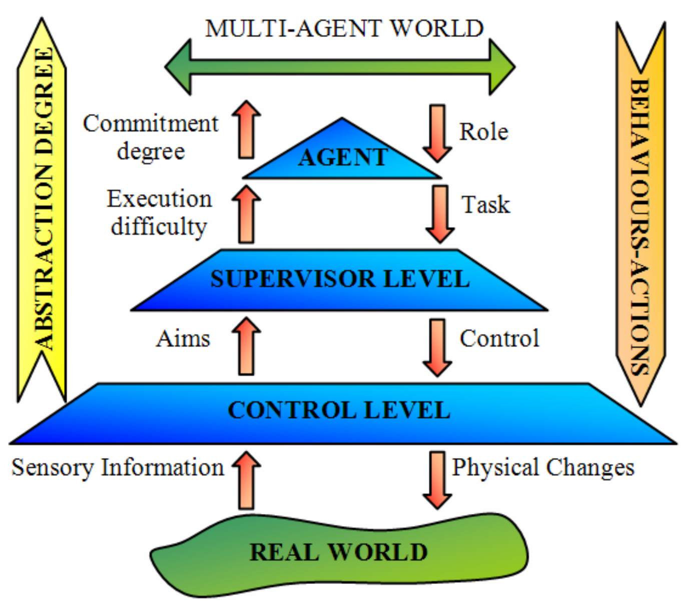


Fig. 3.1. DPA^2 architecture.

The following set of capabilities is proposed in *DPA^2* to represent the internal parameters depending on the information abstraction level:

Atomic Capabilities: These represent control-oriented knowledge that describes the physical agents' controllers. This knowledge helps the physical agents to increase the awareness about their bodies and the perception of the environment through these bodies from a control-oriented viewpoint. Such self-knowledge enhances the adaptation, performance and learning skills of the physical agents in a real environment.

Basic Capabilities: These represent task-oriented knowledge that emerges from different sets of atomic capabilities. This knowledge helps the physical agents to select the most suitable resources to perform the proposed tasks according to the requirements of the tasks.

Symbolic Capabilities: These represent role-oriented knowledge that emerges from different sets of basic capabilities. This knowledge helps the physical agents to perform collective behaviours. Such behaviours take into account the certainty related to the execution of the assigned roles in the acquisition of commitments.

In summary, knowledge based on capabilities provides physical agents with reliable information about their physical features. Thus, physical agents are able to decide with a high certainty level if their physical bodies allow them to perform the requested tasks. In this sense, the relevance of the atomic capabilities like key support of the *DPA*^{2} architecture is evident. However, it is therefore necessary to obtain a more adequate, accurate, reliable and general definition that gathers control-oriented knowledge in an agent-oriented manner. This definition is summarized in the thesis in the control-grounded capabilities, a specialized set of atomic capabilities focused on the physical agents' dynamics.

3.4 Approaches for Coordinated Tasks

Several authors have studied the problems related to the control, coordination and cooperation between physical agents when executing coordinated tasks. These approaches take into account the physical features of the physical agents' bodies from a control-oriented viewpoint. However, a general formalization based on control-grounded capabilities has not been completely carried out. For instance, [Oller et al., 99] introduces dynamic aspects into the design of physical agents. The approach is introduced into this concept that takes dynamics into account to evaluate the difficulty of agent actions.

Reference [De la Rosa et al., 01] shows an approach applied to a ball passing experiment between two robots. The purpose of the example is to show the usefulness of inter-agent negotiation with explicit representations of dynamics in the decision-making structure. This approach also shows the improvement in the decision of *when* and *how* to carry out the passing with respect to static knowledge.

References [Oller et al., 99] and [De la Rosa et al., 00] consider a convoy of two autonomous mobile robots controlled by agents. The rear agent has the responsibility of avoiding collisions, but both are responsible for the reliability of sure decisions based on dynamics. The cooperative decisions based on dynamics provide the controllers with safer set points and a better coordinated control.

The aim of [Innocenti et al., 01] is to find some attributes to describe the dynamics of the physical agent's body. Such attributes are used in a decision algorithm to let the agent know about its physical limitations for deciding feasible actions.

Reference [Quintero et al., 04] shows an example of a set of capabilities and how it is a proper option to represent knowledge related to the physical agent's body. Thus, [Quintero et al., 04] focuses on introspective reasoning on these capabilities to show how the performance of the multi-agent system is improved. In this approach, the physical agents can manage their bodies by taking into account the capabilities associated with their automatic controllers.

An example of a set of capabilities is used in [Zubelzu et al., 04] to represent the dynamics of the physical agents as well as to generate and obtain diversity in dynamics.

The mentioned works present suitable approaches to represent the knowledge related to the physical features of the agent-controlled systems. However, it is still difficult to choose necessary and enough information to include in the agents' decision-making. In spite of this, it is possible to assume that such knowledge must be directly related to the automatic controllers of the physical agents. Thus, reliable information must be extracted from the controllers to obtain an appropriate control-oriented knowledge of the physical agent's body. In this sense, such knowledge can be represented by means of specific attributes (capabilities) focused mainly on control-oriented features as it will be shown in the next chapters.

3.5 Task Allocation Approaches

Several authors have studied the problems related to task allocation, especially in multi-robot environments, based on utility/cost functions. These approaches mainly take into account domain knowledge in the agents' decision-making. However, an approach based on control-oriented features has not been completely carried out.

For instance, [Goldberg and Matarić, 00] presents a behaviour-based controller for a multi-robot collection task that is easily modifiable to obtain new controllers. However, it does not perform any controller/agent selection.

Reference [Dias and Stentz, 00] introduces a free market architecture for distributed control of multi-robot systems solving decomposable tasks. The free market approach defines revenue and cost functions across the possible plans for executing a specified subtask. The robots negotiate amongst themselves to execute the tasks while trying to minimize their costs and maximize their profits.

Reference [Gerkey and Matarić, 02] presents the MURDOCH system for the allocation of tasks using auctions. Regarding *metrics*, it only states that "it should represent the robot's fitness for a task" and that "it could perform any arbitrary computation". It gives as examples of metrics: (a) Cartesian distances from the robot's position to the goal position, and (b) the offset of an object in the robot's camera image.

An effective approach to action selection is presented in [Scheutz, 02]. It does not refer to utility/costs at all.

The aim of [Balakirsky and Lacaze, 02] is to describe a graph search technique to select appropriate behaviours for single and multiple robots. Regarding costs, it computes the cost of each different option in order to select the best one. This cost is computed as the weighted sum of several features (such as road conditions, risk and path length).

Reference [Goldberg et al., 03] shows a market-based architecture for multi-robot coordination. The robots compute their cost according to the *distance* to travel and the *opportunity cost* (that is, how much each time step is valued).

A short review of different task allocation methods, analyzing their efficiency (solution quality versus computation and communication costs) is provided in [Gerkey and Matarić, 04]. It defines utility as the difference between the *quality* of task execution and the *cost* of executing the task. However, it does not indicate how this quality and cost should be computed.

The approach presented in [Dahl et al., 04] uses Q-learning to establish task utilities (which task is the most profitable among a set of possible tasks) in a multi-robot transportation scenario. This utility is computed as the reward (fixed) obtained by executing the task (weighted according to the execution time), and is used to decide which task to engage in next.

Reference [Lagoudakis et al., 05] establishes several bidding rules for auction based multi-robot coordination. The different rules use *path costs* (minimum, maximum or average) to compute the bid for a given target.

Continuing the work in [Lagoudakis et al., 05], [Koenig et al., 06] presents another auction based method to coordinate robots, and the bids are also based on *shortest paths* to target locations.

An auction method for multi-robot exploration tasks is explained in [Sariel and Balch, 06]. The different heuristics used to compute cost are based on *distances* between robots and goal locations.

Reference [Ramos et al., 06] presents a fuzzy approach to action (behaviour) selection. Each behavior computes its cost using a set of fuzzy rules and activation levels, and the one with the lowest cost is the one executed. These rules use several features, such as who has the ball, where a given player is, etc.

The above-mentioned works present suitable approaches to task/action selection where the criteria to bid in such multi-agent task allocation are usually classified by: cost (spatial/temporal) and embodiment (physical features, actuator and preceptor capabilities, etc). In this sense, such basic utility/cost functions are only related to the physical components of the physical agents where dynamics are not taken into account. However, the next chapters show how introspection on control-grounded capabilities contributes to a more suitable task allocation by considering the physical agents' bodies in a better and more reliable way. Such consideration is directly related to the automatic controllers of the physical agents. Thus, appropriate control-oriented knowledge must be extracted from the controllers of the physical agent's body. In this sense, such knowledge is represented by means of specific control-grounded capabilities as it will be also experimentally shown later where physical agents' dynamics are included in task allocation problems.

3.6 Dynamics in Agents' Decisions

The performance of a robot is related to the dynamics of the interactions among its control system, its physical body and its environment. The conventional control techniques for robots tend to either ignore or do implicit and naïve suppositions on the dynamics of their controllers. In this sense, it is looked for an integration vision of the robot with the environment because the robot's physical body, its intelligence, as well as the environment itself, are continuously interacting.

In particular, the effects of the restrictions of the real world as imprecision or uncertainty are very important in mobile robotics due to that they influence the execution quality of the movements/actions. Therefore, it is necessary to take into account all these effects in the decision stage. Thus, software agents must incorporate the problems that characterize the physical systems to analyze such effects [Asada et al., 97].

The above cognitive integration forces to design and implement robotics systems with methodologies of artificial intelligence and control theory. Preliminary interfaces among the high level employed at the *AI* (based generally on the calculation off-line) and the low level of the control theory (based generally on the calculation on-line) have been obtained [Oller, 02]. The coordination among these two levels is not completely carried out and it provokes difficulties to analyze the behavior of this kind of system [Beer, 00].

For some years, several authors as [Müller, 97] [Asada et al., 97] [Beer, 00] and [Oller, 02] have proposed to join the numerical knowledge common of the dynamics of the systems with the symbolic knowledge usual of the dialectical reasoning, the automatons, and the structures of arguments and planning of the agents. In this direction, this thesis presents an approach focused on introspection on control-grounded capabilities that look for then to bridge the gap between the high abstraction level of agents and the low abstraction level of the automatic control architectures.

3.7 Final Remarks

Agent must know the implications of the commitments with other agent-based entities or humans and they must know if they can carry out them. To that end, it is necessary to have some physical knowledge of the system to know what it is physically possible to perform and what it is not possible. In this sense, physical inputs and outputs towards and from the environment must be integrated to the agent's knowledge base. This is due to the fact that the agent is contained in a physical body (*embodied and situated*) which it must control and move by means of decisions.

The current works with physical agents do not appropriately evaluate the associated problems with the agent's dynamics and they only interpret the arisen problems from the viewpoint of supervision or damages detection [Oller, 02]. Therefore, if it is taken into account that the movements of a robot can be commonly described with differential equations, the automatic control theory is then necessary for the analysis of the dynamics of such physical agent, and its effects in the decision-making that generally is implemented by using artificial intelligence techniques.

In summary, the physical knowledge is obtained from the dynamics of the physical body. This dynamics is represented of a declarative way through ***control-grounded capabilities*** extracted by means of ***introspection***, which are the two principal research topics in this doctoral dissertation.

PART II
PROPOSED APPROACH

Chapter 4

Introspection Approach

This chapter presents the introspection approach proposed in this dissertation to provide agents a cognitive ability for reasoning on their dynamics, aiming at making physically feasible decisions and getting reachable and physically grounded commitments that improve the cooperative multi-agent performance. The main definitions, formalization aspects and the algorithms for control and decision used in this work are introduced in this chapter. As will be shown in next chapters, the research on introspection proves the impact of this agency property, and its effectiveness in cooperative intelligent agents.

4.1 Problem Statement

The main problem addressed in this dissertation is concerned to the agents' lack of appropriate reasoning on *physical knowledge mainly related to their dynamics*. Such relevant knowledge is not properly taken into account in the current agents' decision-making and it is not appropriately reflected and communicated by the agents. Agents cannot then make feasible joint decisions when these are requested. Explicitly, lack of appropriate reasoning on physical knowledge results in a lower cooperative performance between agents, especially in coordinated tasks and task allocation problems where a proper communication of such information is quite relevant to achieve sure and trustworthy commitments. The above lacks are currently significant impediments to reducing complexity and achieving appropriate levels of control, coordination and autonomy in control systems.

The cause of the mentioned lack of reasoning is the absence of an appropriate representation related to the physical features of the physical agents, namely the absence of an explicit agent-oriented representation about the agents' inertial dynamics and the specifications, structure and other relevant details encapsulated in their automatic controllers. Otherwise, this specific embedded knowledge about every controller is not taken into account and never reused to achieve better agent cooperation. This cooperation is measured in terms of physically grounded and reliable commitments which results in a better performance of any group of cooperating agents.

Decisions made by agents concerning their actions depend on the information they receive. Therefore, receiving and sending the right information related to physical knowledge is essential for a proper performance and a coherent behaviour of the overall multi-agent system.

To incorporate appropriately physical knowledge in rational decision-making, a suitable representation for dynamics must be first developed in the agents' knowledge base which is general, accessible, understandable, comparable and computationally tractable for these agents. To that end, control-grounded capabilities constitute an alternative. These capabilities aim at guaranteeing an appropriate and explicit agent-oriented representation of the physical agents' dynamics. The thesis claims that the introspection approach on such control-grounded capabilities makes then easier for agents to reflect and communicate the above knowledge, aiming at making physically feasible and safer decisions, getting secure, reachable and physically grounded commitments, preventing from undesirable situations, and achieving a better coordinated control. Such advantages result in the improvement of the multi-agent performance in coordinated control scenarios.

4.2 Introspection on Control-grounded Capabilities in Agents' Decisions

Several researches in artificial intelligence try to build computer-controlled systems that imitate conscious-level reasoning and problem solving of humans [Bolander, 03]. Humans use sentences to express the things they know and sequences of sentences to express reasoning. In this sense, artificial intelligent systems (*agents*) aim at simulating human conscious-level reasoning and problem solving by representing facts internally as sentences and using formal derivations from these sentences as the reasoning mechanism of the system [Bolander, 03].

The set of facts represented as sentences internally in an agent is usually known as its knowledge base. The knowledge base of the agent is a model of the agent's environment, since the objects in the knowledge base represent (or model) properties of the objects in this environment. For that, the knowledge base contains information about the world that the agent takes to be true, facts known or propositions believed by the agent.

Before an agent starts a task, it should make a plan for how to reach a given goal. This planning requires the agent to have knowledge about the environment, knowledge that can be represented in the agent's knowledge base. It is the agent's ability to model its own environment that makes it able to reason about this environment, to plan its actions and to predict the consequences of performing these actions. However, ***much intelligent behavior seems to involve an ability to model not only the agent's external environment but also itself and the agent's own reasoning. Such ability is called introspection*** [Bolander, 03].

4.2.1 Agent Introspection

Introspection is a self-reflection process of human reasoning by which people come to be attentively conscious of mental states they are currently in [Wilson and Keil, 01].

We know that self-reflection plays a central role in human cognition - it is one of the primary abilities setting us apart from animals - and we would therefore expect this ability to play an equally important role in artificial intelligence. We use introspection whenever we reason about the way we carry out certain tasks, and whenever we reason about how to improve our routines for carrying out these tasks. Thus, introspection is fundamental for our ability to consciously improve ourselves. Specifically, to have introspection in an artificial intelligence system means that the system is able to reflect on its own knowledge (or ignorance), its own reasoning, actions, tasks and planning [Bolander, 03].

For instance, before an agent commits in the execution of a task, the agent should register the fact of knowing if it can or cannot perform the task, this needs introspection, due to the agent has to look introspectively into its own knowledge base and from it to arrive at a suitable decision. In addition, in order to decide how well the agent is doing or will do the proposed task, an agent will also need this self-examination capability (*introspection*) [McCarthy, 99].

To express introspective reasoning, the agent should refer to its own knowledge as objects in its world. In this sense, the ***agent is non-introspective when no information in the knowledge base expresses facts concerning the agent itself. Any non-introspective agent only models its external environment.*** This mean that there is a complete separation between the model (the knowledge base) and the reality being modelled (the external environment).

Otherwise, ***introspective agents differ from non-introspective ones by modelling not only their external environment but also themselves. It is by also have models of themselves they are given the ability to introspect*** [Bolander, 03].

Humans also have models of themselves, since we generally believe that we can *predict* our own reactions to most situations that can occur to us. We rely heavily on this ability when we plan our actions. So it turns out then desirably to have agents with this cognitive skill.

In particular, introspection on physical agents' dynamics is a previously unexplored research area. So this thesis focused the work just on this topic for examining its impact in the performance of cooperative multi-agent decisions.

4.2.2 Introspection on Physical Agents' Dynamics

Physical agents require a sense of themselves as distinct and autonomous individuals able to interact with others in cooperative environments, i.e., they require an identity [Duffy, 04]. A complete concept of identity therefore constitutes the set of internal and external attributes associated with any given physical agent based on introspection of its physical and "mental" states and capabilities. In this work, the notion of internal and external relates to the attributes of a single embodied physical agent analogous to Shoham's notion of capabilities in multi-agent systems [Shoham, 93]. It follows that in order to address the issue of embodiment; there are two distinct attributes that are local and particular to each physical agent within a cooperative system:

•*Internal Attributes*: beliefs, desires, intentions, the physical agent's knowledge of self, experiences, a priori and learned knowledge.

•*External Attributes*: the physical presence of the agent in an environment; its actuator and preceptor capabilities (e.g., automatic controllers), the physical features (e.g., physical dimensions).

In this sense, an agent's knowledge of its attributes (models of themselves) therefore allows a sufficient degree of introspection to facilitate and maintain the development of cooperative work and social relationships between groups of agent entities [Duffy, 04]. When an agent is "aware" of itself, it can explicitly communicate knowledge of self to others in a social and cooperative environment to reach a goal. This makes introspection particularly important in connection with multi-agent systems.

In particular, a subset of internal attributes (*control-grounded capabilities*) is used to describe the physical agents' dynamics. Thus,

Definition 1: *Introspection on physical agents' dynamics refers to the self-examination by a physical agent of a subset of internal attributes (control-grounded capabilities) to perform tasks. This self-examination mainly considers the agent body's dynamics. Introspection on control-grounded capabilities is a self-reflection process that refers then to a self-examination that allows agents to be aware of what they are able to do.*

In this context, physical agents must reach an agreement in cooperative groups to obtain sure and trustworthy commitments in the execution of coordinated tasks. Thus,

Definition 2: *Sure and trustworthy commitments refer to commitments accepted by the agents only when they have a high certainty about correctly performing the related task. Such commitments are directly related to a better system response to undesired events and better coordinated control in cooperative decisions.*

To achieve sure and trustworthy commitments, each physical agent must be aware of its ability to perform the requested tasks before committing to them. Therefore, if an agent proposes a coordinated task to another agent, both must introspect, consider and communicate their capabilities before performing the task. Thus, agents would have a high certainty about the correct performance of the task when they acquire commitments.

4.2.3 Formalization Aspects

Let us suppose that a physical agent A_α is a part of a cooperative group *G*. A cooperative group must generally involve more than one physical agent for the execution of a task (see Fig. 4.1). That is,

$$\exists A_i, A_j \in G_{task_k} \mid A_i \neq A_j \quad \text{and} \quad G_{task_k} \subseteq AA$$

Where *AA* is the set of all possible physical agents in the environment.

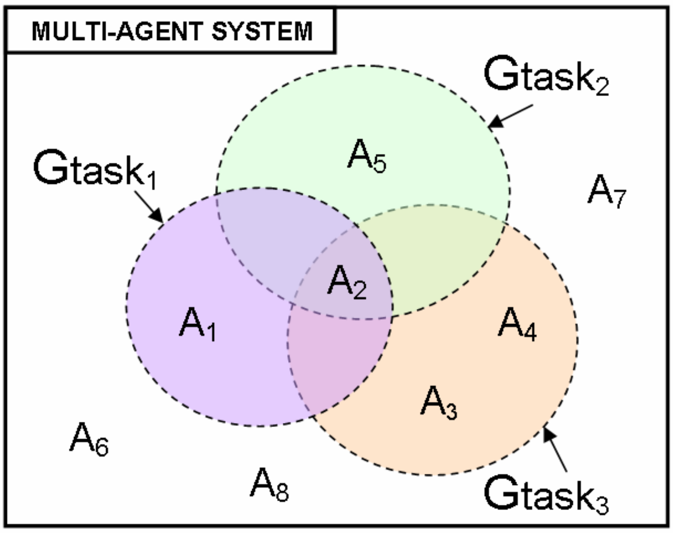


Fig. 4.1. General scheme of the physical agents within the environment.

Let us define the set of automatic controllers *C*, with different control algorithms and different control laws, as a subset of the external attributes *EA* of a physical agent A_α, each one designed to control the plant under particular operating conditions [Breemen, 01], and whose actions provoke the physical agent's dynamics, such that:

$$C(A_\alpha) \subseteq EA(A_\alpha)$$

$$\text{Where } \exists c_i, c_j \in C(A_\alpha) \mid c_i \neq c_j \text{, thus } C(A_\alpha) = \{c_1, c_2, c_3, \ldots c_m\}$$

The controllers allow and limit the tasks' executions. So they are key at the moment physical agents introspect on their capabilities to perform tasks.

Let us consider the domain knowledge *DK* for a physical agent A_α made up of a set of environmental conditions *EC* (e.g., agents' locations, targets' locations), a set of available tasks to perform *T* (e.g., foraging, object transportation, exploration, flocking) and a set of tasks requirements *TR* (e.g., achieve the target, avoid obstacles, time constraints, spatial constraints, energy costs) as is described by (4.1).

$$\mathrm{DK}(A_\alpha) = \mathrm{EC}(A_\alpha) \cup \mathrm{T}(A_\alpha) \cup \mathrm{TR}(A_\alpha) \quad (4.1)$$

$$\text{Where } \exists \mathrm{ec}_i, \mathrm{ec}_j \in \mathrm{EC}(A_\alpha) \mid \mathrm{ec}_i \neq \mathrm{ec}_j,$$

$$\exists \mathrm{task}_i, \mathrm{task}_j \in \mathrm{T}(A_\alpha) \mid \mathrm{task}_i \neq \mathrm{task}_j \text{ and } \exists \mathrm{tr}_i, \mathrm{tr}_j \in \mathrm{TR}(A_\alpha) \mid \mathrm{tr}_i \neq \mathrm{tr}_j$$

$$\text{Thus } \mathrm{EC} = \{\mathrm{ec}_1, \mathrm{ec}_2, \mathrm{ec}_3, \ldots \mathrm{ec}_o\}, \ \mathrm{T} = \{\mathrm{task}_1, \mathrm{task}_2, \mathrm{task}_3, \ldots \mathrm{task}_p\} \text{ and}$$

$$\mathrm{TR} = \{\mathrm{tr}_1, \mathrm{tr}_2, \mathrm{tr}_3, \ldots \mathrm{tr}_q\}$$

$$\text{This gives: } \mathrm{DK}(A_\alpha) = \{\mathrm{ec}_1, \ldots, \mathrm{ec}_o, \mathrm{task}_1, \ldots \mathrm{task}_p, \mathrm{tr}_1, \ldots, \mathrm{tr}_q\}$$

Here, domain knowledge refers to specific knowledge to model the environment in which the physical agents operate and interact. In this context, environmental conditions refer to the needed information to describe the state of the environment. The available tasks refer to well-defined set of real-time actions that usually need coordination to achieve the goals of the multi-agent system. The task requirements refer to the set of conditions over the proposed task that must be accomplished under the current environmental conditions.

The information acquired by the agents' sensors usually constitutes the environmental conditions, while the tasks and their requirements refer generally to the information communicated by other agents in the multi-agent system.

Here, each task has associated a subset of possible controllers for its execution (see Fig. 4.2) such that:

$$\forall \mathrm{task}_k \in \mathrm{T}(A_\alpha), \ \exists C_{\mathrm{task}_k}(A_\alpha) \subseteq C(A_\alpha)$$

$$\text{Where } \exists c_i, c_j \in C_{\mathrm{task}_k}(A_\alpha) \mid c_i \neq c_j, \text{ thus } C_{\mathrm{task}_k}(A_\alpha) = \{c_1, c_2, c_3, \ldots c_{m(\mathrm{task}_k)}\}$$

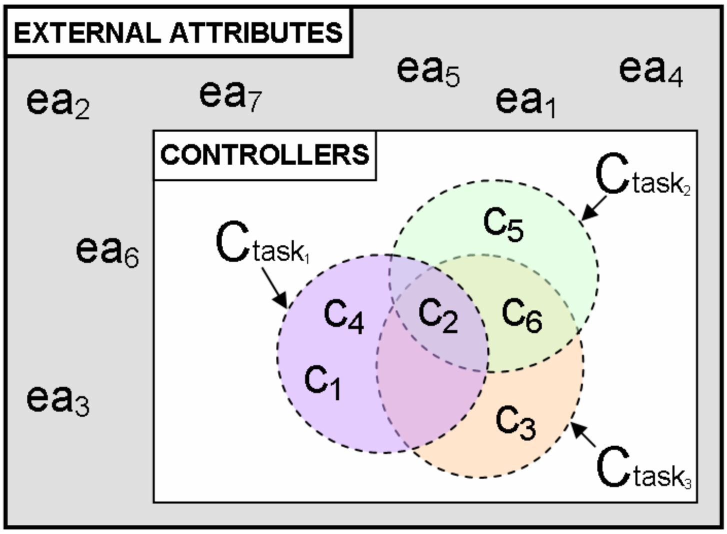


Fig. 4.2. General scheme of the physical agent's external attributes and controllers.

Let us define the set of control-grounded capabilities *CC* to represent the physical agent's dynamics as a subset of the internal attributes *IA* of a physical agent A_α such that:

$$\mathrm{CC}(\mathrm{A}_\alpha) \subseteq \mathrm{IA}(\mathrm{A}_\alpha)$$

$$\text{Where } \exists \mathrm{cc}_i, \mathrm{cc}_j \in \mathrm{CC}(\mathrm{A}_\alpha) \mid \mathrm{cc}_i \neq \mathrm{cc}_j \text{, thus } \mathrm{CC}(\mathrm{A}_\alpha) = \{\mathrm{cc}_1, \mathrm{cc}_2, \mathrm{cc}_3, \ldots \mathrm{cc}_n\}$$

Definition 3: *Control-grounded capabilities constitute a set of internal attributes that form part of the internal state of a physical agent. Such capabilities represent (or model) the physical agent's dynamics in its knowledge base, allowing computational treatment to be accessible and understandable by the agents in the system, i.e., an explicit agent-oriented representation of the physical agents' dynamics. Such representation allows comparing the same kind of capabilities and combining them with other different capabilities to be exploited as a decision tool by the cooperative group where the agents are involved.*

The control-grounded capabilities representation provides then the possibility of representing different automatic controllers with different control algorithms and different control laws. Such effective representation enables the physical agents to use the capabilities as a decision tool and to manage their bodies in a more reliable way.

All controllers involve in the same task has associated the same kind on control-grounded capabilities (see Fig. 4.3) such that:

$$\forall c_i \in C_{task_k}(A_\alpha),\ \exists CC_{c_i,task_k}(A_\alpha) \subseteq CC(A_\alpha)$$

$$\text{Where } \exists cc_i, cc_j \in CC_{c_i,task_k}(A_\alpha) \mid cc_i \neq cc_j \text{, thus}$$

$$CC_{c_i,task_k}(A_\alpha) = \{cc_1, cc_2, cc_3, \ldots cc_{r(task_k)}\}$$

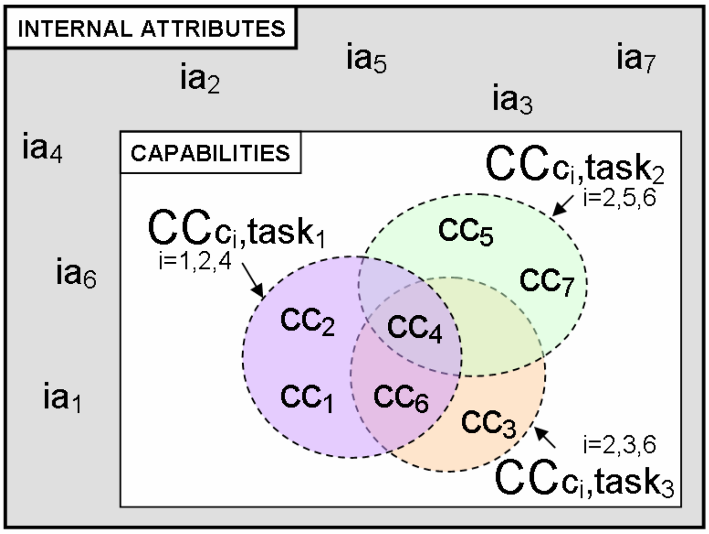


Fig. 4.3. General scheme of the physical agent's internal attributes and control-grounded capabilities.

The control-grounded capabilities $CC_{c_i,task_k}$ for a controller *i* in the execution of a particular task *k* are obtained, as in (4.2), taking into account the agent's domain knowledge DK_{task_k} (see Fig. 4.4) related to the proposed task such that:

$$CC_{c_i,task_k}(A_\alpha) \subseteq CC(A_\alpha) \subseteq IA(A_\alpha) \text{ and } DK_{task_k}(A_\alpha) \subseteq DK(A_\alpha) \mid$$

$$CC_{c_i,task_k}(A_\alpha) = \Psi_{c_i,task_k}(DK_{task_k}(A_\alpha)) \quad (4.2)$$

$\Psi_{c_i,task_k}$ is a self-inspection function that allows physical agents *introspect* on their capabilities using the controller *i* for the task *k*.

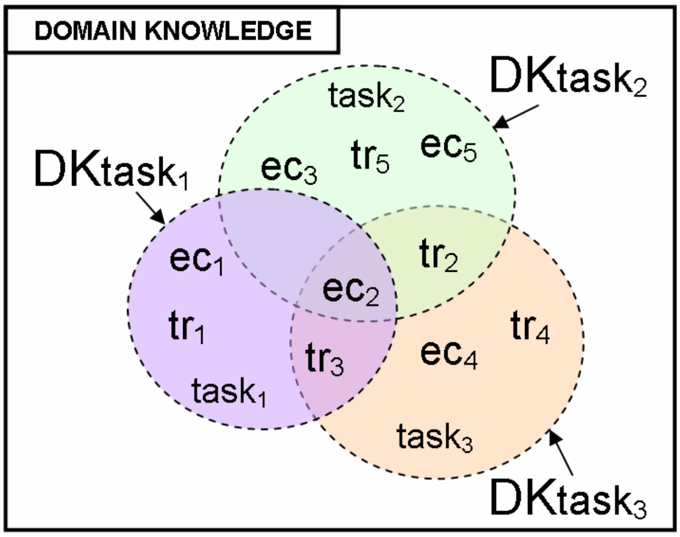


Fig. 4.4. General scheme of the physical agent's domain knowledge.

The capabilities of all controllers within the same task can be grouped as a subset such that:

$$CC_{c_i,\text{task}_k}(A_\alpha) \subseteq CC_{\text{task}\,k}(A_\alpha)$$

$$\text{Where } \exists cc_i, cc_j \in CC_{\text{task}_k}(A_\alpha) \mid cc_i \neq cc_j, \text{ thus}$$

$$CC_{\text{task}_k}(A_\alpha) = \{cc_1, cc_2, cc_3, \ldots cc_{m(\text{task}_k)*r(\text{task}_k)}\}$$

In addition, each controller has then associated a subset of capabilities (see Fig. 4.5) such that:

$$\forall c_i \in C(A_\alpha),\ \exists CC_{c_i}(A_\alpha) \subseteq CC(A_\alpha)$$

$$\text{Where } \exists cc_i, cc_j \in CC_{c_i}(A_\alpha) \mid cc_i \neq cc_j, \text{ thus } CC_{c_i}(A_\alpha) = \{cc_1, cc_2, cc_3, \ldots cc_{r(c_i)}\}$$

Likewise, each controller has associated a subset of tasks such that:

$$\forall c_i \in C(A_\alpha),\ \exists T_{c_i}(A_\alpha) \subseteq T(A_\alpha)$$

$$\text{Where } \exists \text{task}_i, \text{task}_j \in T_{c_i}(A_\alpha) \mid \text{task}_i \neq \text{task}_j, \text{ thus}$$

$$T_{c_i}(A_\alpha) = \{\text{task}_1, \text{task}_2, \text{task}_3, \ldots \text{task}_{p(c_i)}\}$$

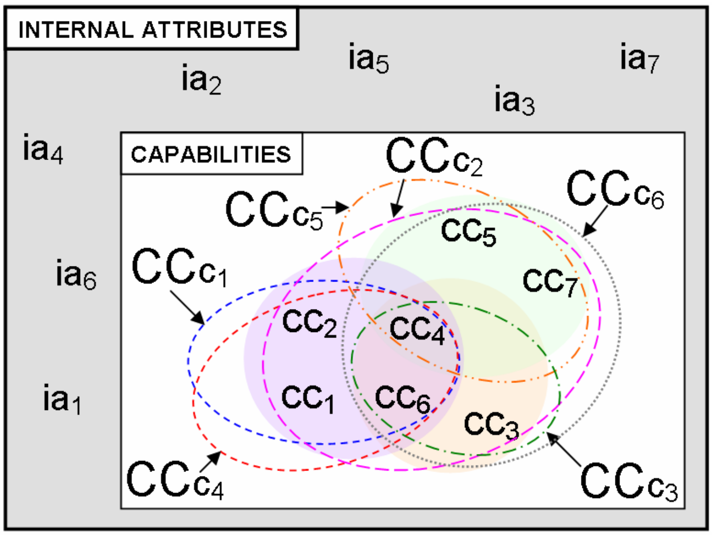


Fig. 4.5. General scheme of the physical agent's internal attributes and capabilities for a specific controller.

A self-evaluation function $\Phi_{c_i,task_k}$ uses the control-grounded capabilities in an appropriate way to allow physical agents *know* a certainty index $ci_{c_i,task_k}$ related to the correct execution of the proposed task *k* using the controller *i* as is described in (4.3).

$$ci_{c_i,task_k}(A_\alpha) = \Phi_{c_i,task_k}(CC_{c_i,task_k}(A_\alpha)) \quad (4.3)$$

The set of all certainty indexes for a specific task *k* is constituted by all $ci_{c_i,task_k}$ of the possible controllers in this task (see Fig. 4.6) such that:

$$\forall c_i \in C_{task_k}(A_\alpha),\ \exists ci_{c_i,task_k}(A_\alpha) \subseteq CI_{task_k}(A_\alpha) \mid \ CI_{task_k}(A_\alpha) \subseteq CI(A_\alpha)$$

$$\text{Where } \exists ci_i, ci_j \in CI_{task_k}(A_\alpha) \mid ci_i \neq ci_j,\ \text{thus}\ \ CI_{task_k}(A_\alpha) = \{ci_1, ci_2, ci_3, \ldots ci_{m(task_k)}\}$$

CI constitutes the set of all certainty indexes related to the available tasks *T* for the agent A_α. Thus,

Definition 4: *A certainty index provides physical agent a measure of conviction concerning its knowledge and physical actual ability to perform a particular task.*

Each controller has also associated a set of certainty indexes (see Fig. 4.6) such that:

$$\forall c_i \in C(A_\alpha),\ \exists ci_{c_i}(A_\alpha) \subseteq CI_{c_i}(A_\alpha) \mid \ CI_{c_i}(A_\alpha) \subseteq CI(A_\alpha)$$

$$\text{Where } \exists ci_i, ci_j \in CI_{c_i}(A_\alpha) \mid ci_i \neq ci_j, \text{ thus } CI_{c_i}(A_\alpha) = \{ci_1, ci_2, ci_3, \ldots ci_{s(c_i)}\}$$

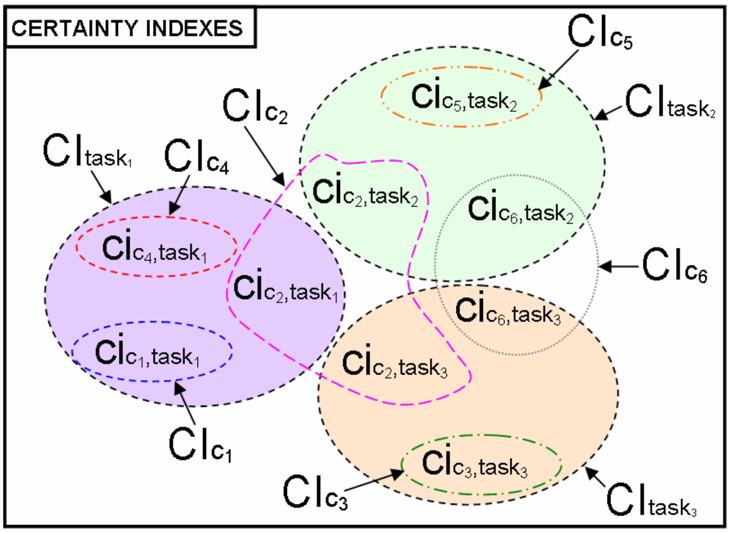


Fig. 4.6. General scheme of the physical agent's certainty indexes.

In addition, a suitability function $\Theta_{c_i,task_k}$ provides physical agent an appropriate alternative to measure the suitability of each possible controller *i* in the execution of a proposed task *k* according to its capabilities. For that, the certainty index of each controller is jointly used with its respective control-grounded capabilities as is described in (4.4).

$$sr_{c_i,task_k}(A_\alpha) = \Theta_{c_i,task_k}(ci_{c_i,task_k}(A_\alpha), CC_{c_i,task_k}(A_\alpha)) \quad (4.4)$$

The set of all suitability rates for a specific task *k* is constituted by all $sr_{c_i,task_k}$ of the possible controllers in this task such that:

$$\forall c_i \in C_{task_k}(A_\alpha),\ \exists sr_{c_i,task_k}(A_\alpha) \subseteq SR_{task_k}(A_\alpha) \mid SR_{task_k}(A_\alpha) \subseteq SR(A_\alpha)$$

$$\text{Where } \exists sr_i, sr_j \in SR_{task_k}(A_\alpha) \mid sr_i \neq sr_j, \text{ thus } SR_{task_k}(A_\alpha) = \{sr_1, sr_2, sr_3, \ldots sr_{m(task_k)}\}$$

Thus, each controller has then associated a set of suitability rates such that:

$$\forall c_i \in C(A_\alpha),\ \exists sr_{c_i}(A_\alpha) \subseteq SR_{c_i}(A_\alpha) \mid SR_{c_i}(A_\alpha) \subseteq SR(A_\alpha)$$

$$\text{Where } \exists sr_i, sr_j \in SR_{c_i}(A_\alpha) \mid sr_i \neq sr_j, \text{ thus } SR_{c_i}(A_\alpha) = \{sr_1, sr_2, sr_3, \ldots sr_{s(c_i)}\}$$

SR constitutes the set of all suitability rates related to the available tasks *T* for the agent A_α. Thus,

Definition 5: *A suitability rate provides physical agent a fitness measure of its capabilities for the execution of a specific task.*

So a comparative analysis ζ_{task_k} between all possible suitability rates of $\text{SR}_{\text{task}_k}$ of the controllers in a specific task allows physical agents to select the most suitable controller for the execution of this task. Similarly, an analysis of $\text{SR}_{\text{c}_\text{i}}$ allows physical agents to identify the task where the controller *i* has its better performance.

Let us define then the physical agent's knowledge of self related to its dynamics as in (4.5).

$$\text{SK}(\text{A}_\alpha) = \text{C}(\text{A}_\alpha) \cup \text{CC}(\text{A}_\alpha) \cup \text{CI}(\text{A}_\alpha) \cup \text{SR}(\text{A}_\alpha) \quad (4.5)$$

In particular, the physical agent's knowledge of self in a specific task *k* (see Fig. 4.7) is given by (4.6).

$$\text{SK}_{\text{task}_k}(\text{A}_\alpha) = \text{C}_{\text{task}_k}(\text{A}_\alpha) \cup \text{CC}_{\text{task}_k}(\text{A}_\alpha) \cup \text{CI}_{\text{task}_k}(\text{A}_\alpha) \cup \text{SR}_{\text{task}_k}(\text{A}_\alpha) \quad (4.6)$$

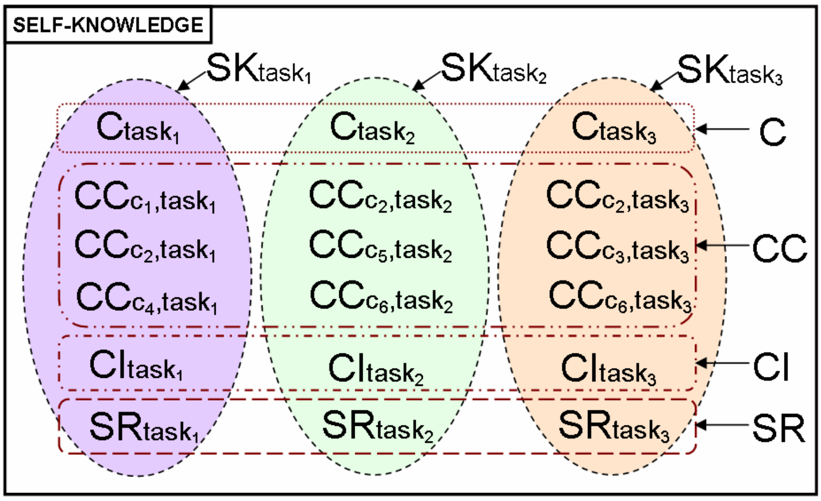

SELF-KNOWLEDGE
$\text{SK}_{\text{task}_1}$
$\text{SK}_{\text{task}_2}$
$\text{SK}_{\text{task}_3}$
C_{task_1}
C_{task_2}
C_{task_3}
C
$\text{CC}_{\text{c}_1,\text{task}_1}$
$\text{CC}_{\text{c}_2,\text{task}_1}$
$\text{CC}_{\text{c}_4,\text{task}_1}$
$\text{CC}_{\text{c}_2,\text{task}_2}$
$\text{CC}_{\text{c}_5,\text{task}_2}$
$\text{CC}_{\text{c}_6,\text{task}_2}$
$\text{CC}_{\text{c}_2,\text{task}_3}$
$\text{CC}_{\text{c}_3,\text{task}_3}$
$\text{CC}_{\text{c}_6,\text{task}_3}$
CC
$\text{CI}_{\text{task}_1}$
$\text{CI}_{\text{task}_2}$
$\text{CI}_{\text{task}_3}$
CI
$\text{SR}_{\text{task}_1}$
$\text{SR}_{\text{task}_2}$
$\text{SR}_{\text{task}_3}$
SR

Fig. 4.7. General scheme of the physical agent's knowledge of self from a task-oriented perspective.

In the same way, the physical agent's knowledge of self related to each controller *i* (see Fig. 4.8) is given by (4.7).

$$SK_{c_i}(A_\alpha) = CC_{c_i}(A_\alpha) \cup CI_{c_i}(A_\alpha) \cup SR_{c_i}(A_\alpha) \cup T_{c_i}(A_\alpha) \quad (4.7)$$

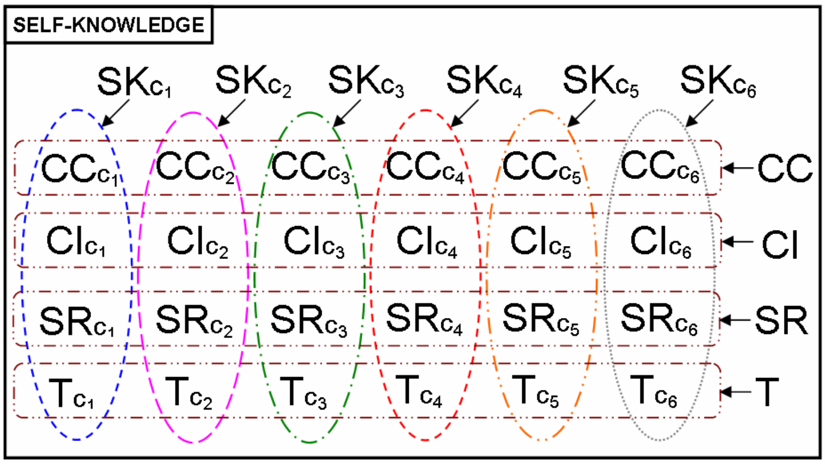


Fig. 4.8. General scheme of the physical agent's knowledge of self from a controller-oriented perspective.

This self-knowledge varies then along four main parameters: content (physical agent's dynamics), manner or mode of representation (control-grounded capabilities, certainty indexes, and suitability rates), domain of application (physical multi-agent systems) and means of acquisition (functions, algorithms and techniques of soft-computing and automatic control).

The physical agent's knowledge base *KB* (see Fig. 4.9) is therefore founded on the union of both its domain knowledge *DK* and its self-knowledge *SK* as in (4.8).

$$KB(A_\alpha) = DK(A_\alpha) \cup SK(A_\alpha) \quad (4.8)$$

In summary, the functions (Ψ, Φ, Θ) provide physical agents powerful tools, independent of particular implementation technologies, for introspection-level reasoning and suitable model of themselves. Such functions constitute a novel design methodology of physical agents establishing the degree of introspection *I* (see Fig. 4.10) used by such agents in the reasoning related to the execution of proposed tasks as is defined in (4.9).

$$I = f(\Psi, \Phi, \Theta) \quad (4.9)$$

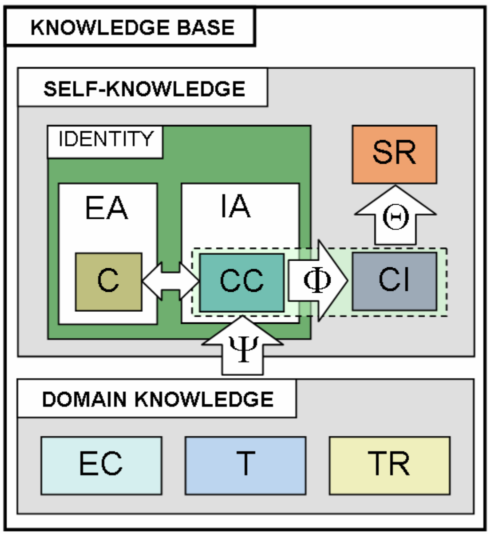


Fig. 4.9. General scheme of the physical agent's knowledge base.

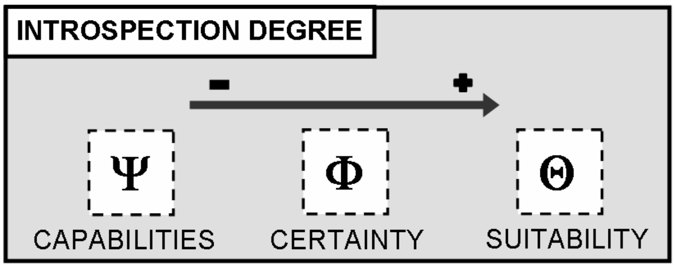


Fig. 4.10. General scheme of the physical agent's introspection degree.

The design methodology is then constituted by three main components:

1. Agent's reasoning on its own environment makes it able to know its *capabilities* to perform the available tasks in such environment $I = f(\Psi)$.

2. Agent's reasoning on its capabilities makes it able to predict the *certainty* related to its physical actual ability to perform such tasks $I = f(\Psi, \Phi)$.

3. Knowledge on the above agent's capabilities and certainties allows it reason on its *suitability* for the proper execution of the proposed tasks $I = f(\Psi, \Phi, \Theta)$.

So, the physical agent's introspection degree is going to depend on how much knowledge physical agent wants to consider or need in its decision-making. In this sense, physical agents might need to acquire more knowledge or increase their introspection degree (*increase the abstraction level*) to carry out certain tasks.

Currently, there are several alternatives to implement independently or jointly the above functions. Thus, soft-computing techniques (e.g., neural networks, case-based reasoning and fuzzy logic) and control techniques (e.g., model-predictive control, multiple model adaptive controllers and switching control systems) are commonly used.

In consequence, the proposed design methodology implicates a change in the current way of designing physical agents by means of a declarative and explicit knowledge of their automatic controllers. Such methodology is inspired by the KISS principle of engineering that states that design simplicity should be a key goal and unnecessary complexity should be avoided. In this sense, the practical work addressed in this dissertation constitutes an engineering approach easily applicable.

4.2.4 Decision Algorithm for Coordinated Tasks

One of the most important physical agent's jobs is to make decisions, that is, to commit to particular actions or tasks. To that end, the connection between logical reasoning and decision making is simple: the physical agent must conclude, based on its knowledge, that a certain action or task is best. In addition, this decision is influenced by the goal of the agent, the skills required to accomplish its tasks and the resources that are needed. So, agents make rational decisions. In this sense, with introspection, this thesis argues that agents can discriminate between the trials in which they have a chance of performing a proposed task and those in which they have no chance.

For instance, let us consider two physical agents A_α and A_β aiming at undertaking a coordinated task. It is assumed that agents use an agent communication language to communicate information and knowledge and they follow the *BDI* philosophy

[Shoham, 93]. So the decision algorithm (see Fig. 4.11) executed by the agents for the execution of a particular task is as follows:

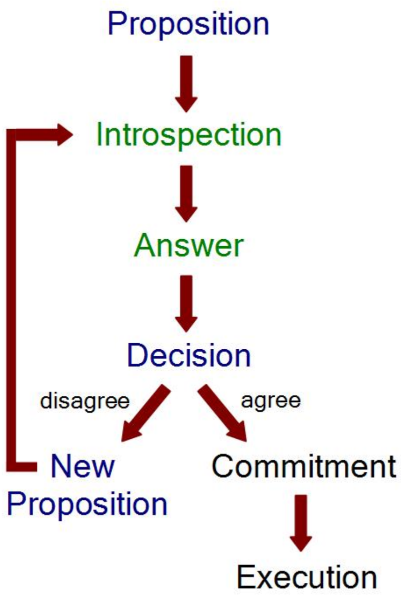


Fig. 4.11. Decision algorithm executed by the agents for the execution of a particular task.

Proposition: An agent A_α proposes to other agent A_β a coordinated task k with some requirements (e.g., time constraints).

$$\mathrm{REQUEST}\left(\mathrm{A}_\alpha, \mathrm{A}_\beta, \mathrm{t}_1, \mathrm{task}_\mathrm{k}\right)$$

Introspection: A_β looks for and evaluates its capabilities $\mathrm{CC}_{\mathrm{task}_\mathrm{k}}$ to perform the proposed task with its available resources (e.g., controllers) $\mathrm{C}_{\mathrm{task}_\mathrm{k}}$ and domain knowledge $\mathrm{DK}_{\mathrm{task}_\mathrm{k}}$.

$$\mathrm{KNOW}^{\mathrm{t}_2}_{\mathrm{A}_\beta}\left\{\mathrm{A}_\beta, \mathrm{task}_\mathrm{k}, \mathrm{SK}_{\mathrm{task}_\mathrm{k}}(\mathrm{A}_\beta)\right\} \quad \mathrm{AND} \quad \mathrm{BELIEF}^{\mathrm{t}_2}_{\mathrm{A}_\beta}\left\{\mathrm{CAN}^{\mathrm{t}_2}_{\mathrm{A}_\beta}\left[\mathrm{task}_\mathrm{k}, \zeta_{\mathrm{task}_\mathrm{k}}(\mathrm{A}_\beta)\right]\right\}$$

Answer: A_β tells A_α that it can perform the task with reliability $\zeta_{\mathrm{task}_\mathrm{k}}$.

$$\mathrm{INFORM}\left(\mathrm{A}_\beta, \mathrm{A}_\alpha, \mathrm{t}_3, \mathrm{BELIEF}^{\mathrm{t}_2}_{\mathrm{A}_\beta}\left\{\mathrm{CAN}^{\mathrm{t}_2}_{\mathrm{A}_\beta}\left[\mathrm{task}_\mathrm{k}, \zeta_{\mathrm{task}_\mathrm{k}}(\mathrm{A}_\beta)\right]\right\}\right)$$

Decision: If A_α agrees that ζ_{task_k} is large enough then the agents commit (*Commitment*) and they can perform the task successfully (*Execution*).

If A_α considers that ζ_{task_k} is too low then A_α inhibits the commitment in the execution of the task and proposes to A_β the same task with other requirements or a new task (*New Proposition*) until the agents agree.

$$\begin{gathered}\text{IF}\quad \text{BELIEF}_{A_\alpha}^{t_4}\left\{\text{ENOUGH}\left(\zeta_{\text{task}_k}(A_\beta)\right)\right\}\\ \text{THEN}\quad \text{COMMIT}\left(A_\alpha, A_\beta, t_5, \text{task}_k\right)\\ \text{ELSE}\quad \text{REQUEST}\left(A_\alpha, A_\beta, t_5, \text{task}_{k'} \vee \text{task}_l\right)\end{gathered}$$

4.2.5 Decision Algorithm for Task Allocation

Several authors have studied the problems related to task allocation, especially in multi-robot environments, based on utility/cost functions. These approaches mainly take into account domain knowledge in the agents' decision-making. However, an approach based on control-oriented knowledge has not been completely carried out. In this sense, the thesis aims at showing how introspection on control-grounded capabilities contributes to a more suitable task allocation by considering the physical agents' bodies in a better and more reliable way. With introspection, the approach looks for that agents can discriminate between the tasks that they can perform and those in which they have no chance.

For instance, let us consider a group of physical agents $G = \{A_1, A_2, \ldots, A_N\}$ aiming at undertaking several tasks $T = \{\text{task}_1, \text{task}_2, \ldots, \text{task}_M\}$ under the supervision of an omniscient and omnipotent centralized agent *SA* [Stone, 00]. So the decision algorithm executed by the agents for allocating the available tasks is as follows:

Proposition: A request related to all proposed tasks with some requirements (e.g., time constraints, spatial constraints) is done by the *SA* to all agents in the group *G*.

$$\text{REQUEST}\left(\text{SA},\{A_1,A_2,\ldots,A_N\},t_1,\{\text{task}_1,\text{task}_2,\ldots,\text{task}_M\}\right)$$

Introspection: Agents ($A_i\ i=1,...,N$) look for and evaluate their control-grounded capabilities $CC_{task_k}\ k=1,\ldots,M$ to perform each proposed task with their available resources (e.g., controllers) $C_{task_k}\ k=1,\ldots,M$ and domain knowledge $DK_{task_k}\ k=1,\ldots,M$.

$$\text{KNOW}_{A_i}^{t_2}\left\{A_i,\{\text{task}_1,\text{task}_2,\ldots,\text{task}_M\},\{SK_{task_1}(A_i),SK_{task_2}(A_i),\ldots,SK_{task_M}(A_i)\}\right\}$$
$$\text{AND}\quad \text{BELIEF}_{A_i}^{t_2}\left\{\text{CAN}_{A_i}^{t_2}\left[\{\text{task}_1,\text{task}_2,\ldots,\text{task}_M\},\{\zeta_{task_1}(A_i),\zeta_{task_2}(A_i),\ldots,\zeta_{task_M}(A_i)\}\right]\right\}$$

Agents sort in a decreasing order their introspected reliabilities $\{\zeta_{task_1}(A_i),\zeta_{task_2}(A_i),\ldots,\zeta_{task_M}(A_i)\}$ in the execution of the proposed tasks. In this sense, agents are able generally to perform a set of tasks $T_{OK}(A_i)$ whose reliabilities $\zeta(A_i)$ are greater than a decision threshold *Th*. The used reliabilities are directly related to the introspection degree established. It means that the reliabilities can embrace a comparative analysis of certainty indexes $CI(A_i)$ or a comparative analysis of suitability rates $SR(A_i)$.

Answer: Each agent informs to *SA* the set of tasks *TOK* that it can perform and its respective set of reliabilities $\zeta(A_i)$.

$$\text{INFORM}\left(\{A_1,A_2,\ldots,A_N\},\text{SA},t_3,\text{BELIEF}_{A_i}^{t_2}\left\{\text{CAN}_{A_i}^{t_2}\left[T_{OK}(A_i),\zeta(A_i)\right]\right\}\right)$$

Decision: If more than one agent can perform the same task then the supervisor agent *SA* allocates such task to the agent with the greater reliability (*the most suitable agent*). The selected agent commits (*Commitment*) and can then perform the task successfully (*Execution*). For the remaining tasks and remaining agents, the *SA* repeats the above process taking into account the remaining reliabilities of the agents for these tasks. The above task allocation algorithm aims at allocating the most possible amount of remaining tasks for all remaining agents. Such process depends on the number of available agents and the reliability of each agent in relation to each proposed task.

Fig 4.12 shows an example of the decision algorithm for task allocation. In this case, there are six agents $G = \{A_1, A_2, A_3, A_4, A_5, A_6\}$ and six tasks $T = \{task_1, task_2, task_3, task_4, task_5, task_6\}$ and it is established, for illustrative reasons, a decision threshold *Th* = 0.75. The example shows the set of tasks that each agent can perform with their respective reliabilities in a decreasing order. In three cycles of the algorithm, the *SA* allocates several tasks to the most suitable agent (*$task_1 \rightarrow A_1$, $task_3 \rightarrow A_5$, $task_4 \rightarrow A_6$ and $task_5 \rightarrow A_3$*). However, in this example, the agents A_2 and A_4 cannot perform any task according to their capabilities (*it means that their reliabilities are not enough to perform correctly any task*). Additionally, the *$task_6$* cannot be performed by any agent and the *$task_2$* can not finally be allocated.

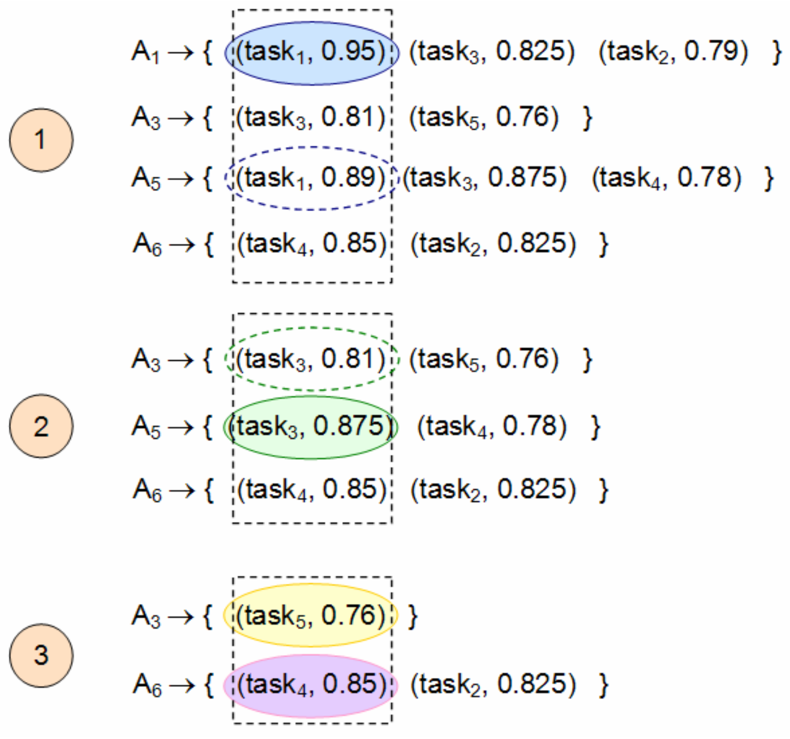

1
A₁ → { (task₁, 0.95) (task₃, 0.825) (task₂, 0.79) }
A₃ → { (task₃, 0.81) (task₅, 0.76) }
A₅ → { (task₁, 0.89) (task₃, 0.875) (task₄, 0.78) }
A₆ → { (task₄, 0.85) (task₂, 0.825) }
2
A₃ → { (task₃, 0.81) (task₅, 0.76) }
A₅ → { (task₃, 0.875) (task₄, 0.78) }
A₆ → { (task₄, 0.85) (task₂, 0.825) }
3
A₃ → { (task₅, 0.76) }
A₆ → { (task₄, 0.85) (task₂, 0.825) }

Fig. 4.12. Task allocation example.

Here, only the reliability parameter is used for allocating tasks. However, the task allocation algorithm can be implemented in the same way if several coordination parameters are jointly used with the reliability in the utility/costs functions for allocating tasks. More details about how it can be implemented are provided in Sections 5.1.4 and 5.1.6.3.

Roughly speaking, introspective skills in physical agents provide them a suitable alternative for appropriate task allocation and appropriate task execution in cooperative multi-agent environments. The big interest is then to see how this physical knowledge of dynamical capabilities affects the cooperative behavior of the whole cooperative world. To that end, the decision algorithms for coordinated tasks and task allocation, presented in Sections 4.2.4 and 4.2.5 respectively, constitute particular and illustrative alternatives. Therefore, other decision algorithms, depending on the objectives of the multi-agent system, could be perfectly designed.

4.2.6 Control-grounded Capabilities for Linear Control Systems

At the automatic control level, the control-grounded capabilities definition must take into account the design of the control system. In this sense, the study and design of a control system are generally based on the system's response. Control engineers should know the specifications that the system's response must achieve before designing the controllers for the system. However, controllers design must also involve considerations of the dynamical behaviour of the plant. In addition, the specifications are usually given in terms of the transient and the steady-state performance, and controllers are designed to meet these specifications. Such specifications describe then the response of the controlled system. They can be used to complete the control-grounded capabilities according to the control theory foundations. However, this information must be complemented in order to accomplish the requirements mentioned for the control-grounded capabilities (see Definition 3).

The following set of control-grounded capabilities has been proposed and defined according to the scope of this work, to be applied in linear control systems (Single-Input Single-Output, Multi-Input Single-Output, Single-Input Multi-Output and Multi-Input Multi-Output, i.e., SISO, MISO, SIMO and MIMO systems respectively). Such set of capabilities is adopted, expecting them to have direct relevance to represent the controllers, although this set is not complete. (Note: *m* represents the number of outputs and *n* represents the number of inputs of the controlled system in the definitions).

A. *Overshoot* (μ):

The overshoot is related to the maximum value reached by the transient response of the controlled system with reference to its steady-state value. Let $y_s(j,k)$ be the unit-step response taking into account the *k-th* input and the *j-th* output of the controlled system. Let $y_{max}(j,k)$ denote the maximum value of $y_s(j,k)$; and $y_{ss}(j,k)$, the steady-state value of $y_s(j,k)$. The overshoot is defined as $\mu(j,k) = y_{max}(j,k) - y_{ss}(j,k)$. The μ capability is often represented as a percentage of the final value of the step response. However, the desired condition is a low μ capability as shown in (4.10).

$$\mu(\%) = \frac{\sum_{j=1}^{m} b_\mu(j) * \left(\frac{\sum_{k=1}^{n} a_\mu(j,k) * \mu(j,k)(\%)}{n - \sum_{k=1}^{n}\left(1 - a_\mu(j,k)\right)} \right)}{m - \sum_{j=1}^{m}\left(1 - b_\mu(j)\right)} \tag{4.10}$$

$$\text{where} \quad \mu(j,k)(\%) = 100\% - \frac{y_{max}(j,k) - y_{ss}(j,k)}{y_{ss}(j,k)} \times 100\%$$

$a_\mu(j,k)$ represents the *k-th* weight coefficient, between [0,1], for the *k-th* overshoot $\mu(j,k)(\%)$ in the *j-th* output, and $b_\mu(j)$ represents the *j-th* weight coefficient, between [0,1], for the *j-th* overshoot.

B. *Speediness* (σ):

The speediness represents an indicator of the controlled system's speed when it reaches the set point. The speediness is defined in (4.11) as a percentage relation between the settling time of the closed-loop system $t_s(j,k)$ and the settling time of the open-loop system $t_{s\text{-}ol}(j,k)$, taking into account *k-th* input and the *j-th* output.

$$\sigma(\%) = \frac{\sum_{j=1}^{m} b_{\sigma}(j) * \left(\frac{\sum_{k=1}^{n} a_{\sigma}(j,k) * \sigma(j,k)(\%)}{n - \sum_{k=1}^{n}(1 - a_{\sigma}(j,k))} \right)}{m - \sum_{j=1}^{m}(1 - b_{\sigma}(j))} \qquad (4.11)$$

$$\text{where} \quad \sigma(j,k)(\%) = \left(1 - \frac{t_s(j,k)}{t_{s-ol}(j,k)} \right) \times 100\%$$

$a_{\sigma}(j,k)$ represents the *k-th* weight coefficient, between [0,1], for the *k-th* speediness *$\sigma(j,k)(\%)$* in the *j-th* output, and *$b_{\sigma}(j)$* represents the *j-th* weight coefficient, between [0,1], for the *j-th* speediness.

If *$\sigma(\%)$*>0 for all controllers, then the fastest controller is the one with the largest value of *$\sigma(\%)$* between them. If *$\sigma(\%)$*<0 for all controllers, the closed loop systems are slower than the open loop system, then the largest negative value of *$\sigma(\%)$* between them correspond to the fastest. Controlled systems with *$\sigma(\%)$* > 0% are always faster than those with *$\sigma(\%)$* < 0%.

C. *Persistence* (γ):

This represents the capability of the controlled system to follow the set point when there are external signals affecting the value of the system's output. The persistence is related to the capability of the controlled system to reject disturbances and maintain the output signal at a suitable value. Rejecting disturbances is sometimes a specification of the controller design. The evaluation of such rejections depends on the control engineer's criteria. However, (4.12) and (4.13) provide the persistence for the two most common disturbances, the step signal and the pulse signal in the SISO case.

For the step disturbances:

$$\gamma(\%) = \left[\left(1 - \frac{IAE}{\tau * A} \right) \times (\%\_\text{disturbance}) \right] \% \qquad (4.12)$$

IAE is the integrated absolute error $IAE = \int_{t_1}^{t_2} |e(t)| dt$, *τ* is the open-loop time constant, *A* is the amplitude of the step disturbance and *%_disturbance* is the percentage of times that this disturbance affects the system.

For the pulse disturbances:

$$\gamma(\%) = \left[\left(1 - \frac{IAE}{B}\right) \times (\%\_\text{disturbance})\right]\% \quad (4.13)$$

B is the area of the pulse signal, B = (Amplitude of the pulse)*(Duration of the pulse).

If *γ(%)* < 0% then *γ(%)* = 0% and the system does not reject then disturbances. If there is more than one kind of disturbance, then the value of *γ(%)* will be the maximum of the corresponding persistence values. Equation (4.14) shows the MIMO case.

$$\gamma(\%) = \frac{\sum_{j=1}^{m} b_\gamma(j) * \left(\frac{\sum_{k=1}^{n} a_\gamma(j,k) * \gamma(j,k)(\%)}{n - \sum_{k=1}^{n}(1 - a_\gamma(j,k))}\right)}{m - \sum_{j=1}^{m}(1 - b_\gamma(j))} \quad (4.14)$$

a_γ(j,k) represents the *k-th* weight coefficient, between [0,1], for the *k-th* persistence *γ(j,k)(%)* in the *j-th* output, and *b_γ(j)* represents the *j-th* weight coefficient, between [0,1], for the *j-th* persistence.

D. Aggressiveness (α):

This capability represents the quickness of the system to react to the set point changes. The aggressiveness is defined in (4.15) as the percentage relation between the rise time $t_r(j,k)$ and the settling time $t_s(j,k)$ for the *k-th* input and *j-th* output of the controlled system.

$$\alpha(\%) = \frac{\sum_{j=1}^{m} b_{\alpha}(j) * \left(\frac{\sum_{k=1}^{n} a_{\alpha}(j,k) * \alpha(j,k)(\%)}{n - \sum_{k=1}^{n} (1 - a_{\alpha}(j,k))} \right)}{m - \sum_{j=1}^{m} (1 - b_{\alpha}(j))} \tag{4.15}$$

$$\text{where} \quad \alpha(j,k)(\%) = 100\% - \frac{t_r(j,k)}{t_s(j,k)} \times 100\%$$

$a_\alpha(j,k)$ represents the *k-th* weight coefficient, between [0,1], for the *k-th* aggressiveness *$\alpha(j,k)(\%)$* in the *j-th* output, and *$b_\alpha(j)$* represents the *j-th* weight coefficient, between [0,1], for the *j-th* aggressiveness.

E. Precision (δ):

This represents the capability of the controlled system to follow the changes of set point. The precision is related to the error of the controlled system when it is excited by a ramp input signal *r(j,k)* with slope τ, after 2τ seconds. Let *$y_r(j,k)$* be the ramp response and let τ be the time constant of the open-loop system taking into account the *k-th* input and the *j-th* output of the system. Precision is defined in (4.16).

$$\delta(\%) = \frac{\sum_{j=1}^{m} b_{\delta}(j) * \left(\frac{\sum_{k=1}^{n} a_{\delta}(j,k) * \delta(j,k)(\%)}{n - \sum_{k=1}^{n} (1 - a_{\delta}(j,k))} \right)}{m - \sum_{j=1}^{m} (1 - b_{\delta}(j))} \tag{4.16}$$

$$\text{where} \quad \delta(j,k)(\%) = 100\% - \lim_{t \to 2\tau} \frac{r(j,k) - y_r(j,k)}{r(j,k)} \times 100\%$$

$a_\delta(j,k)$ represents the *k-th* weight coefficient, between [0,1], for the *k-th* precision *$\delta(j,k)(\%)$* in the *j-th* output, and *$b_\delta(j)$* represents the *j-th* weight coefficient, between [0,1], for the *j-th* precision.

F. *Robustness* (ρ):

This describes the capability to maintain the output of the controlled system inside an acceptable range when there are internal disturbances in the system. The phase margin and gain margin provide a measurement of the system's stability. These margins without variations in the open-loop system parameters (*MP(j,k)_{nom}* and *MG(j,k)_{nom}*) and with the maximum variations of these parameters (*MP(j,k)* and *MG(j,k)*), taking into account the *k-th* input and the *j-th* output of the system, are used in the robustness definition proposed in (4.17).

$$\rho = \frac{\sum_{j=1}^{m} b_\rho(j) * \left(\frac{\sum_{k=1}^{n} a_\rho(j,k) * \rho(j,k)}{n - \sum_{k=1}^{n} (1 - a_\rho(j,k))} \right)}{m - \sum_{j=1}^{m} (1 - b_\rho(j))}$$

$$\rho(j,k) = \frac{\frac{MF(j,k)}{MF(j,k)_{nom}} + \frac{MG(j,k)}{MG(j,k)_{nom}}}{2} \qquad (4.17)$$

$a_\rho(j,k)$ represents the *k-th* weight coefficient, between [0,1], for the *k-th* robustness *$\rho(j,k)$* in the *j-th* output, and *$b_\rho(j)$* represents the *j-th* weight coefficient, between [0,1], for the *j-th* robustness.

G. *Control Effort* (ε):

This capability describes the controlled system's effort in driving the output towards the desired value. The control effort is defined in (4.18).

$$\varepsilon = \sum_{j=1}^{m} b_\varepsilon(j) * \left(\frac{\sum_{k=1}^{cs} a_\varepsilon(k) * \varepsilon(k)}{cs - \sum_{k=1}^{cs} (1 - a_\varepsilon(k))} \right)$$

$$\text{where} \quad \varepsilon(k) = \frac{\int_{t_1}^{t_2} \left| \frac{du(k)}{dt} \right| dt}{u(k)_{max} - u(k)_{min}} \qquad (4.18)$$

u(k) is the *k-th* control signal, *u(k)*max denotes the maximum value of *u(k)*, *u(k)*min denotes the minimum value of *u(k)*, *cs* represents the number of control signals, $a_\varepsilon(k)$ represents the *k-th* weight coefficient, between [0,1], for the *k-th* control effort $\varepsilon(k)$ in *j-th* output, and $b_\varepsilon(j)$ represents the *j-th* weight coefficient, between [0,1], for the *j-th* control effort.

A comparison between the controllers with respect to the ε capability is defined in (4.19). In this sense, we have assigned the highest percentage (100%) to the controller with the lowest ε capability.

$$\varepsilon_{c_i}(\%) = \frac{\min(\varepsilon_{c_i})}{\varepsilon_{c_i}} \times 100\%, \forall c_i, \quad i = 1 \ldots m \quad (4.19)$$

H. Control Kind (κ)*:*

This capability identifies the type of controller that is being analysed. Thus, only controllers of the same kind are compared. (E.g., position controllers → κ = 1, velocity controllers → κ = 2, etc.).

In the above definitions, the weight coefficients are used to assign the relevance degree of every *k-th* and *j-th* component in the calculation of the capabilities. Therefore, such relevance degrees determine the influence of every input and every output of the system in the agents' decision-making.

Hence, the set of all control-grounded capabilities $CC_{c_i}(A_\alpha)$, associated with the controller *i,* where *i = 1,…,m* for a physical agent A_α, is shown in (4.20):

$$CC_{c_i}(A_\alpha) = \{\mu, \sigma, \gamma, \alpha, \delta, \rho, \varepsilon, \kappa\}, i = 1, \ldots, m \quad (4.20)$$

However, as it was shown in Section 4.2.3, the set of control-grounded capabilities associated with a specific controller depends on if such capabilities are needed in the involved tasks for such controller. Fig 4.13 depicts an example scheme of the physical agent's capabilities for a linear control system.

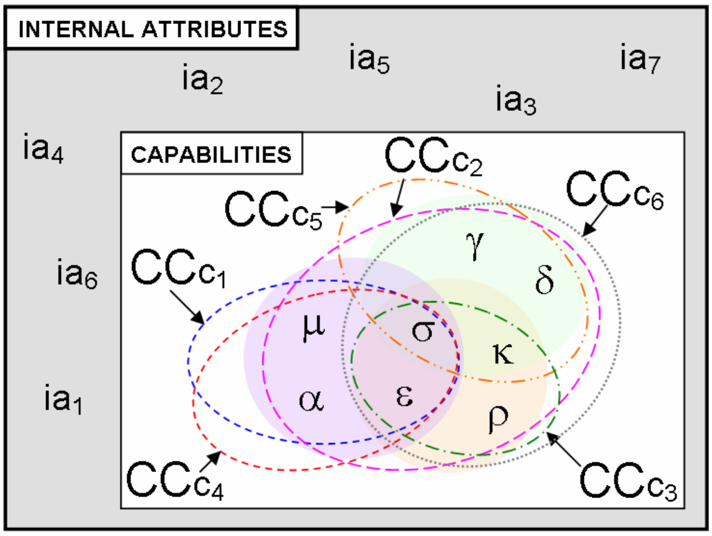


Fig. 4.13. General scheme of the physical agent's capabilities for a linear control system.

4.3 Final Remarks

This chapter argues the need for introspective skills about dynamics in physically grounded agents to improve the agents' decision-making performance in both individual and cooperative decisions. Here, introspection allows agents to achieve sure and trustworthy commitments, to prevent undesirable situations, to make safer decisions, to drive a better coordinated control and to obtain enhanced levels of performance and autonomy in any group of cooperating agents especially in coordinated tasks and task allocation problems.

This work claims the relevance of introspection for both individual and cooperative decisions about commitments between physical agents. In such decisions, the introspection allows an agent to know about its physical body's ability to perform the proposed tasks. Therefore, physical agents can behave intelligently when they negotiate commitments with other agents or humans. Here, intelligence is understood as the appropriate exploitation of knowledge about dynamics to perform better [Sanz et al., 01] and achieve enhanced levels of performance and autonomy [Sanz et al., 00]. In this sense, introspection contributes to increasing the level of intelligence in physical agents by means of a suitable self-examination capability. In particular, with introspection, physical agents will have a great deal of flexibility and self-control that will make them more intelligent. Likewise, introspection is closely related to self-awareness. Research on self-aware control systems aims at building systems that exhibit flexible, autonomous and goal-directed behaviours and provides an application domain for research and development of agent technologies [Sanz et al., 04] [Luck et al., 05]. The emergence of the self-aware control systems' behaviors is based on a deep understanding of the world and the self. Since introspection is related to self-knowing, it could help to fulfill the aims of self-aware control systems and also be a contribution to agent-based computing theory and practice.

The chapter considered a representation based on capabilities related to the agent body's dynamics. These capabilities are managed in an introspective manner when agents are required to make a decision or to commit to the fulfillment of a task. Nevertheless, it is still difficult to choose the necessary information to include in the capabilities to represent dynamics. Here, introspection on dynamics is closely related to automatic controllers of physical agents. From the controllers, suitable information is extracted to obtain reliable control-oriented knowledge of the agent body's dynamics. However, in many cases the correspondence among controllers and capabilities is not possible of establishing in an analytical way by using the traditional control theory, that is generally concerned with the analysis of the dynamical behavior of controlled systems, often in terms of differential equations [Breemen, 01]. The above can be due to the characteristics of the plant (plants of dynamics of high order, multi-variables, not linear, etc), of the controllers (non-linear controllers, fuzzy controllers, neural networks controllers, etc), or the environment (very noisy environments with high disturbances). In these cases, it will be necessary to establish the correspondence by using, as alternatives, techniques of machine learning or soft-computing to establish an appropriate relation as it is shown with some examples in the following chapters.

Chapter 5

Implementation

This chapter presents the application of the proposed approach on several coordinated control scenarios. It describes how physical agents can introspect on their control-grounded capabilities in coordinated tasks and task allocation problems of three different test beds. Specifically, robot soccer has been used as the test bed 1. The passing a ball, the offside maneuver and a team-work coordination case are presented in robot soccer. A search team of agent-controlled mobile robots in a rescue environment represents the test bed 2. Finally, convoys of agent-controlled vehicles in a traffic flow environment constitute the test bed 3. The chapter provides a general description of these three test beds while specifying their details as they are used for empirical testing.

5.1 The Test bed 1: Robot Soccer

Autonomous, cooperating mobile robots represent multi-agent systems. The robot soccer test bed is a challenge for autonomous mobile cooperating robots [Burkhard et al., 02] . This test bed is a rich domain for the study of topics related to multi-agent systems [Kim and Vadakkepat, 00]. Robot soccer has many features found in a real-world system such as complexity, dynamism, uncertainty and goal variability, together with both cooperating and competing robots [Oller et al., 97]. In addition, working with robot soccer is a great opportunity to deal with a lot of different kinds of technical subject areas. It is possible to deal with every technology, which is necessary for an autonomous system [Novak, 02]. The emergence of robot soccer competitions as RoboCup (http://www.robocup.org/) and FIRA (http://www.fira.net/) is then an interesting trend that control researchers should explore for developing new approaches. In this case, stronger links with the *AI* community should be explored, because that community is currently at the forefront of many of these applications [Murray et al., 03].

In particular, the agent paradigm is commonly proposed as a solution to controlling a robot community [De la Rosa et al., 97] [Rocher and Duhaut, 98]. The global behavior problem of a robot soccer team provides the opportunity to apply agent theory because of the distributed architecture of the mobile robots. The problem itself implies coordination, competition and cooperation by means of communication between the robot soccer players. Specifically, the multi-agent control algorithm, in such active environment, must then comprise a low level kinematics and dynamics and high level decision-making.

Robot soccer has been used as the main test bed for these reasons. Robots employed in disaster control and response operations, household activities, traffic control and industrial operations can profit from the results gained by researching and enhancing the game of these small mobile robots.

5.1.1 Robot Soccer System

Basically, robots, a vision system, a host computer and a communication system are needed for a robot soccer game (see Fig. 5.1). A vision-based soccer robot system has been used as operating method implemented as a remote-brainless soccer robot system [Kim and Vadakkepat, 00]. Each robot has its own driving mechanism, communication board and CPU board in the remote-brainless soccer robot system. The computational part controls the robot's velocity according to the command data received from a host computer. All calculations on vision data processing, decision-making, strategies, position control of robots, are done in a host computer which controls the robots via radio communication. In robot soccer different kinds of system configurations exist. The configuration used in this work is called MiroSOT (Micro Robot Soccer Tournament) [Kim and Vadakkepat, 00]. Specifically, Fig. 5.2 shows the team of real MiroSOT robots used in the experiments.

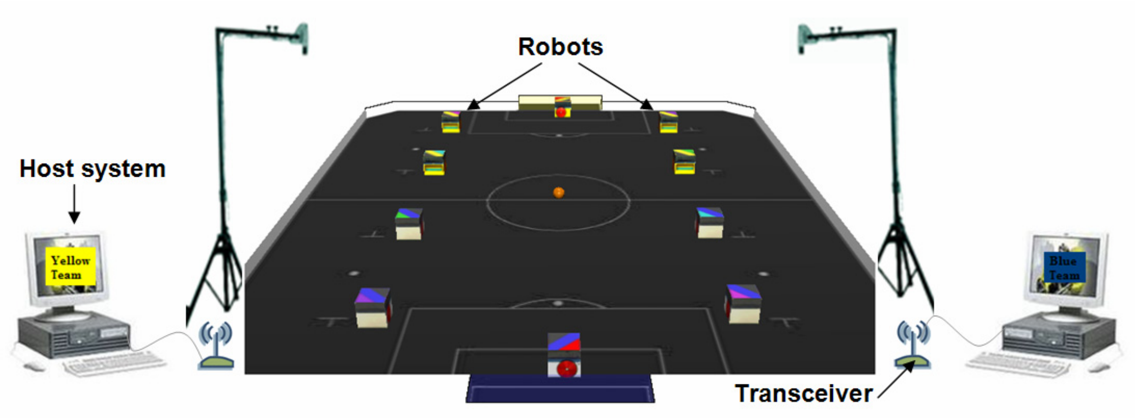


Fig. 5.1. Overall robot soccer system.

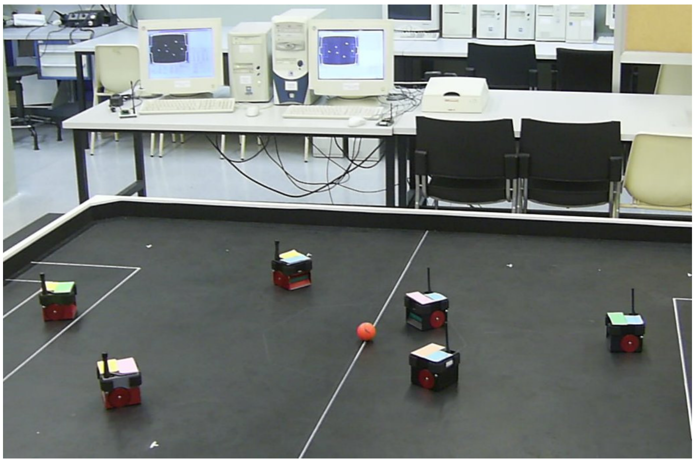


Fig. 5.2. Team of real MiroSOT robots.

5.1.2 Robot Modeling

The state *S(t)* of the MiroSOT robots can be established by any set of the following representations: (see Fig. 5.3)

$$S(t) = \{x, y, \theta, v_l, v_r\} \text{ or } S(t) = \{x, y, \theta, v, \omega\} \text{ or } S(t) = \{x, y, \theta, v_x, v_y\}$$

The relation among the different representations is established from the following kinematics relations:

$$v_l = R\omega_l; \quad v_r = R\omega_r$$

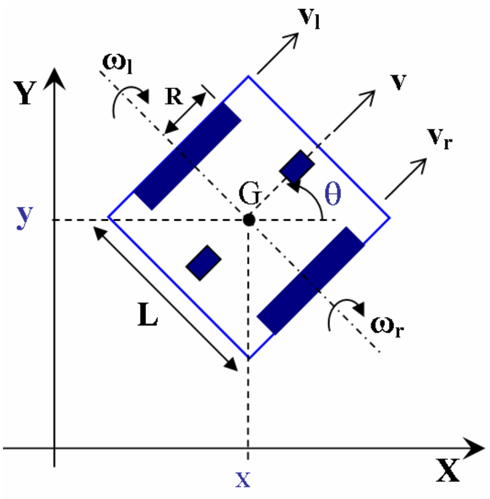


Fig. 5.3. Variables that describe the robots' state, L=7.5 cm, R=2.25 cm, G: geometric center.

Where v_l and v_r are the linear velocities of the wheels left and right respectively, ω_l and ω_r are the angular velocities of the wheels left and right respectively and *R* is the radius of the wheels. Also, it can be shown that:

$$v = \frac{v_l + v_r}{2} \quad \text{and} \quad \omega = \frac{v_l - v_r}{L}$$

Where v is the linear velocity of the mobile robot; ω is the robot's angular velocity and *L* is the distance between the wheels. The projections of the linear velocity on the *X* and *Y* axes are given by:

$$v_x = v\cos(\theta) \quad \text{and} \quad v_y = v\sin(\theta)$$

From the above relations is observed the need of controlling the linear velocities of each wheel (v_l, v_r) to be able of controlling the movement of the geometrical center of the robot (G) represented by means of the coordinates (x, y, θ).

A mobile robot is then a MIMO (Multi-Input Multi-Output) system and its control is typically too complex to be developed and operated when it must include the specifications of the system's response. These specifications must take into account the dynamical limitations and the non-holonomic features of the mobile robot and the geometric and kinematics properties of the movement path. In this sense, Equation (5.1) provides the robot model used.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{2.3}{0.2833\,s+1} & 0 \\ 0 & \dfrac{2.07}{0.0687\,s+1} \end{bmatrix} \begin{bmatrix} \dfrac{1}{2} & \dfrac{1}{2} \\ \dfrac{1}{L} & -\dfrac{1}{L} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} \qquad (5.1)$$

Other higher order transfer functions, non-linearities and other variables will be analysed in future work.

5.1.3 Selected Coordinated Tasks

Two coordinated tasks of the robot soccer test bed have been selected to evaluate the proposed approach: passing a ball (*task$_1$*) and the offside maneuver (*task$_2$*).

5.1.3.1 Passing a Ball

The passing experiment is here described as follows: Two physical agents, *passer* (*A_p*) and *shooter* (*A_s*), are involved in passing a ball. The physical agents have an obstacle-free movement trajectory and have a set of controllers to move in the environment. The passer must strike the ball towards the interception point in a suitable way. The shooter must intercept and shoot the ball with the intention of scoring in the opposite goal. Thus, the passer and the shooter must coordinate to perform the task successfully. Fig. 5.4 shows an example of this task.

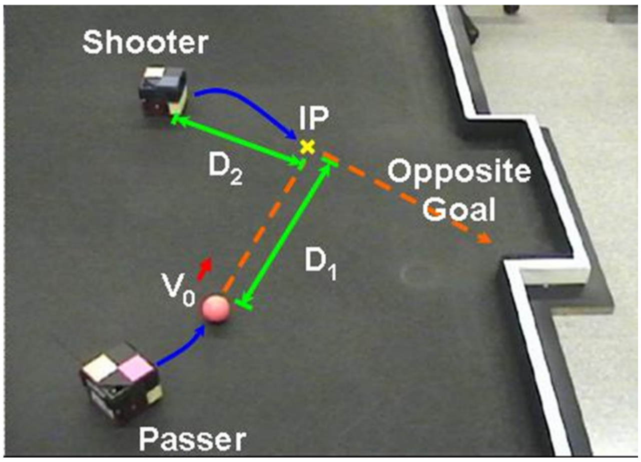


Fig. 5.4. Task1: passing a ball between physical agents.

Passing a ball is then represented as follows: the initial distance (*D_1*) between the ball and the interception point (*IP*), the initial velocity of the ball (*V_0*) and the initial distance (*D_2*) between the shooter and the *IP*. For the sake of simplicity, the passer and shooter are not moving at the beginning of the task. The *IP* is arbitrarily selected in a region near the opposite goal. The *V_0* determines the behavior of the ball and depends on the impact of the passer. Additionally, this task takes into account dynamic and non-holonomic constraints inherent to the physical agents' bodies, and time constraints of the physical agents' controllers. Time constraints are considered because the environment's dynamics impose time limitations on passing a ball.

Here, the passer's orientation when it strikes the ball is important to execute a well-coordinated task. The impact orientation determines the future path of the ball as shown in Fig. 5.5. This determination is required because the passer's orientation is influenced by the above dynamic, non-holonomic and time constraints. Therefore, the interception point is recalculated when the passer strikes the ball and it is informed to the shooter. Equation (5.2) shows this recalculation.

$$x_{IPnew} = x_B + D_1 \cos(\theta_P), \quad y_{IPnew} = y_B + D_1 \sin(\theta_P) \quad (5.2)$$

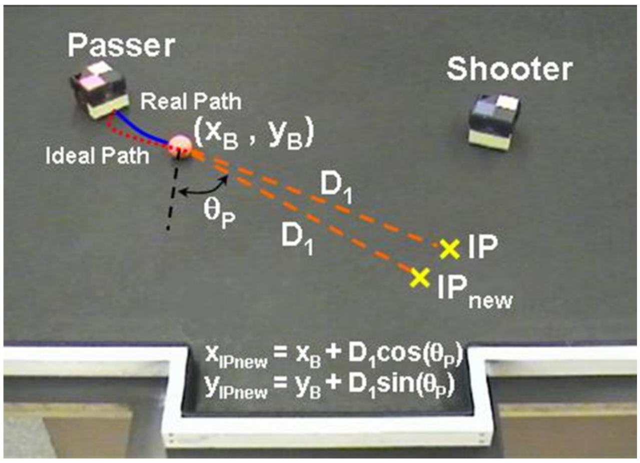


Fig. 5.5. Path of the ball when the passer strikes it.

In the same sense, the variables to represent the shooter are the following: the minimal time that needs the shooter to perform the task (*Time$_A$*), the initial distance (*D_2*) between the shooter and the *IP*, and a compound orientation of the shooter (θ_A). The compound orientation is described in (5.3). Fig. 5.6 shows the variables that describe the shooter's state.

$$\theta_A = |\alpha_1| + |\alpha_2| = |\theta_L - \theta_I| + |\theta_L - \theta_F| \quad (5.3)$$

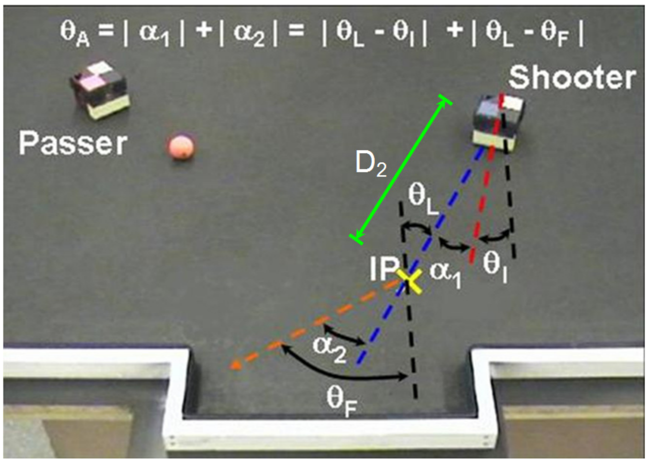

$\theta_A = |\alpha_1| + |\alpha_2| = |\theta_L - \theta_I| + |\theta_L - \theta_F|$
Shooter
Passer
D_2
θ_L
IP
α_1
θ_I
α_2
θ_F

Fig. 5.6. Variables that describe the shooter's state.

5.1.3.2 Offside Maneuver

In the offside maneuver, two physical agents, *defender1* (A_{d1}) and *defender2* (A_{d2}), are involved. Such physical agents have also a set of controllers to move in the environment. Defender1 and defender2 must coordinate between them to perform a maneuver able to avoid a successful passing a ball between two opposing physical agents. Fig. 5.7 shows an example of this task.

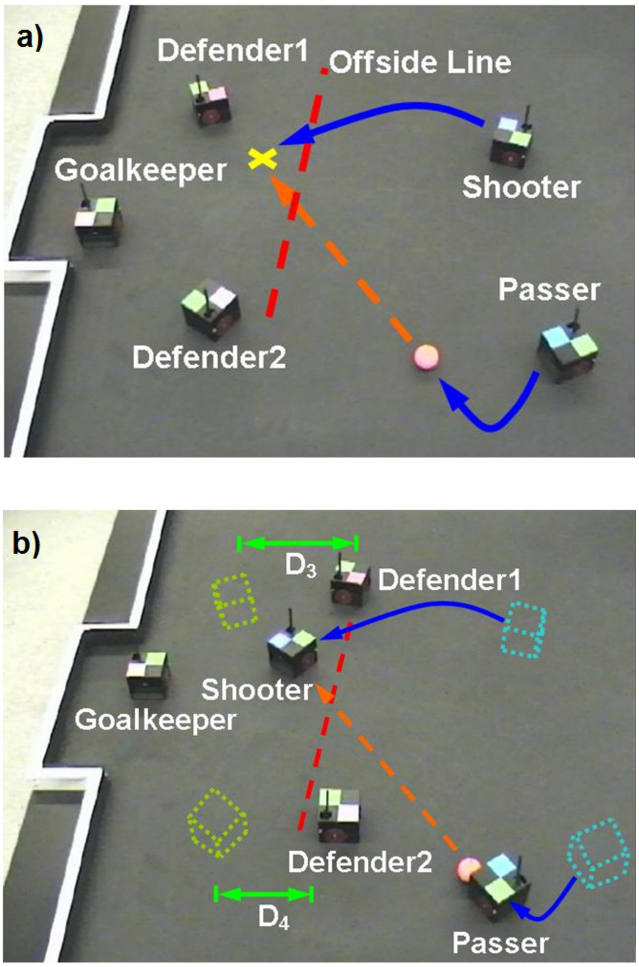


Fig. 5.7. Task$_2$: offside scheme. a). Before the play; b). After the play.

It is possible to describe the offside maneuver by means of the following variables: the time that the opposing passer strikes the ball ($Time_P$), the distances (D_3) and (D_4) between each defender and the offside line (*OL*) as well as their respective generic orientations (θ_B) and (θ_C), (similar to θ_A, see Equation 5.3, but using *OL* as reference).

Here, as a simplification, the offside line is selected by the defenders knowing the *IP* of the passed ball, and taking into account the size of their bodies, around 7.5 cm x 7.5 cm as described in (5.4).

$$x_{OL} = x_{IP} + 7.5\text{cm} \quad (5.4)$$

In summary, there are two cooperative groups: G_{task_1} *-attack-* and G_{task_2} *-defense-*, each one is involved in the execution of a coordinated task of the set of tasks *T* within the multi-agent system *AA* such that:

$$G_{task_1} = \{\text{passer}, \text{shooter}\} = \{A_p, A_s\}$$

$$G_{task_2} = \{\text{defender}_1, \text{defender}_2\} = \{A_{d1}, A_{d2}\}$$

$$T = \{\text{pas}\sin\text{g}, \text{offside}\} = \{\text{task}_1, \text{task}_2\}$$

$$G_{task_1} \subseteq AA \quad \wedge \quad G_{task_2} \subseteq AA$$

$$\text{Where } AA = \{\text{passer}, \text{shooter}, \text{defender}_1, \text{defender}_2, \text{goalkeeper}\}$$

5.1.4 Team-work Coordination Case Study

A task allocation problem in the robot soccer test bed has been also selected to evaluate the proposed approach. To that end, a heterogeneous team of physical agents has been designed such that G = {A_1, A_2, A_3, A_4, A_5}, where each agent has a set of movement controllers to interact in the environment. There are three scenes S = {*attack*, *midfield*, *defense*} in such environment as it is shown in Fig. 5.8.

Here, scenes refer to the spatial regions where agents must meet and work jointly to perform the proposed tasks. The current scene is established taking into account the current ball's location.

For the sake of simplicity, the main task to allocate in each scene is to kick the ball with the intention of scoring in the opposite goal. In this sense, the physical agents must coordinate between them by using several coordination parameters in their utility/cost functions to select the most suitable agent for the main task. The other remaining agents follow a fixed strategy according to the conditions established for the current scene. Fig. 5.9 shows an example of this task allocation case study.

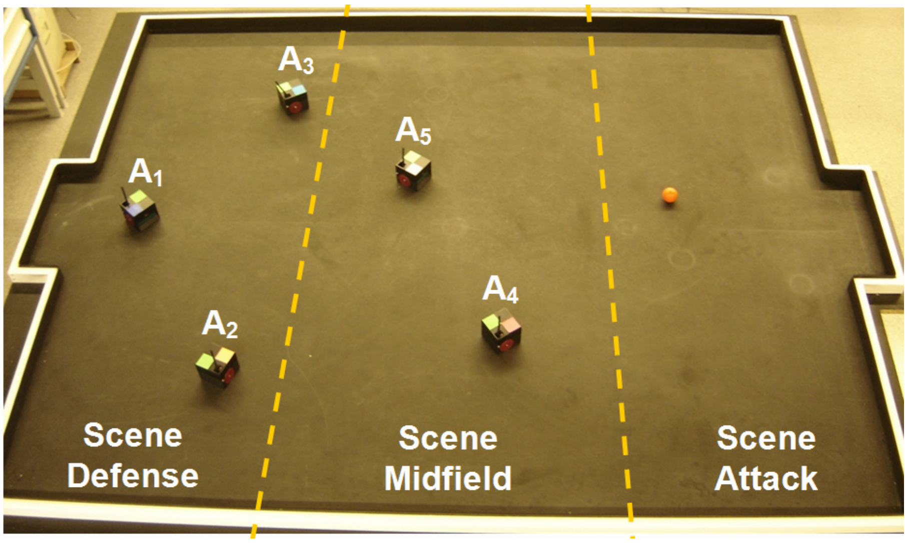


Fig. 5.8. Scenes and agents in the robot soccer environment.

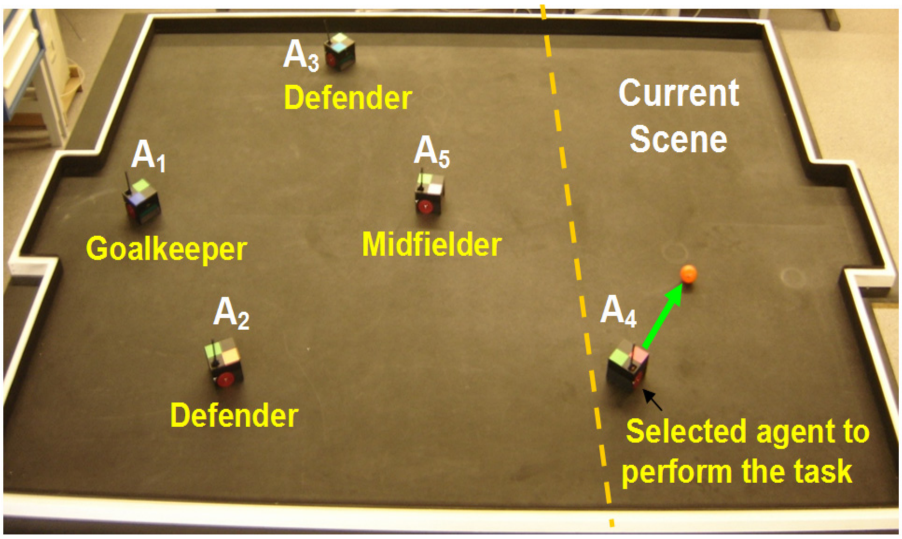


Fig. 5.9. Example of the task allocation case study in robot soccer.

5.1.5 Linear Control System Case Study

In addition, a linearized second-order model [Oller and Garcia, 02] of the mobile robot's dynamics has been also used in this approach. The movement of each robot $(x(t), y(t), \theta(t))$ is then controlled such that the robot follows the horizontal axis *x* with a constant linear velocity *v*. A control law based on the poles location method in which the values of the angular velocity *ω* are obtained in terms of the robot location $(y(t), \theta(t))$ is proposed in (5.5).

$$\omega(t) = -\frac{\alpha_1\alpha_2}{v} y(t) + (\alpha_1 + \alpha_2)\theta(t) \quad (5.5)$$

Where $\alpha_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$, $\alpha_{1,2}$ are the poles of the system, ζ is the damping factor and ω_n is the natural frequency of the characteristic equation of a second-order system. Thus, the linear controlled system using the movement variables *(y, θ)* of the robot is represented by (5.6) and (5.7).

$$Y(s) = \frac{s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2} y(0) + \frac{vs}{s^2 + 2\zeta\omega_n s + \omega_n^2}\theta(0) \quad (5.6)$$

$$\theta(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\theta(0) - \left(\frac{\omega_n^2}{v}\right)\frac{s}{s^2 + 2\zeta\omega_n s + \omega_n^2} y(0) \quad (5.7)$$

The step responses expressions of the above linearized model are shown in (5.8) and (5.9).

$$y(t) = \frac{-\omega_n e^{-\zeta\omega_n t}}{\omega_d}\left[\sin\left(\omega_d t - \tan^{-1}\frac{\omega_d}{\zeta\omega_n}\right) - 2\zeta\sin\omega_d t + \frac{1}{\omega_n}\sin\omega_d t\right]u(t) \quad (5.8)$$

$$\theta(t) = \frac{\omega_n e^{-\zeta\omega_n t}}{\omega_d}\left[\cos\left(\omega_d t + \frac{\pi}{2} - \tan^{-1}\frac{\omega_d}{\zeta\omega_n}\right) - \omega_n\sin\omega_d t\right]u(t) \quad (5.9)$$

Where $\omega_d = \omega_n\sqrt{1-\zeta^2}$, ω_d is the damping natural frequency.

In this case, the above linear model for each mobile robot has been used in the coordinated tasks described in Sections 5.1.3.1 and 5.1.3.2. Here, different situations can appear in the execution of such proposed tasks. These situations must be taken into account when the physical agents make cooperative decisions. In this domain, space limitations *SL*: reduced space for movement due to the presence of other agents, motion disturbances *MD*: collisions with other physical agents, time constraints *TC*: deadlines in the tasks due to the environment's dynamics, energy performance *EP*: different energy expenses according to the tasks, and special behaviours like aggressiveness *AB* and quickness *QB* are considered and their combinations in the coordinated tasks are examined. Every situation has an influence degree *ID* to establish the relevance of each one on the agents' decisions. The sum of all *IDs* of the examined situations is equal to 100%.

5.1.6 Implementations in Robot Soccer

In the implementations, each physical agent has a set of movement controllers to execute the proposed coordinated tasks. Three different controllers (c_1, c_2, and c_3) have been designed such that:

$$C(A_p) = C(A_s) = C(A_{d1}) = C(A_{d2}) = \{c_1, c_2, c_3\} \text{ and}$$

$$C_{task_1}(A_p) = C_{task_1}(A_s) = C_{task_2}(A_{d1}) = C_{task_2}(A_{d2}) = \{c_1, c_2, c_3\}$$

Fig. 5.10 shows how these controllers produce different dynamics in the execution of the tasks. The results of the tasks' executions will be different if the controllers have different control laws under the same environmental conditions and task requirements. Thus, it is possible to obtain a set of capabilities associated with each controller for the current environmental conditions and the proposed task requirements. These capabilities describe the dynamic features of the system during the execution of the tasks with a specific controller. The representation of dynamics based on capabilities could be by means of performance indicators according to features of the path.

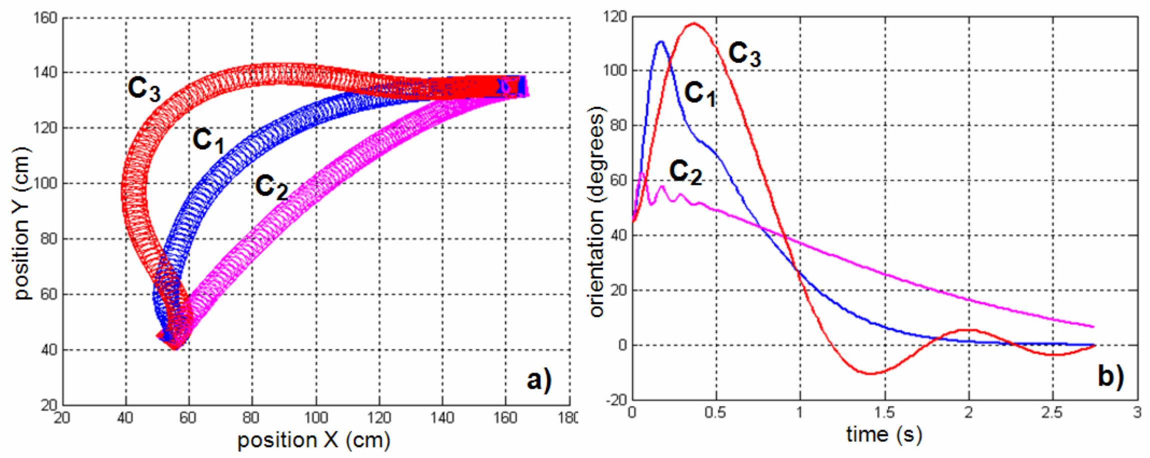


Fig. 5.10. a). Spatial evolution of the physical agent with each controller; b). Temporal evolution of the physical agent's orientation with each controller.

For instance, Fig. 5.11 shows the spatial evolution of a physical agent with each designed controller (c_1, c_2, and c_3) under some specific environmental conditions (*initial location: (150cm, 50cm, 0°), final location: (50cm, 150cm, 190°)*) and task requirements (*temporal constraint of 4 seconds*).

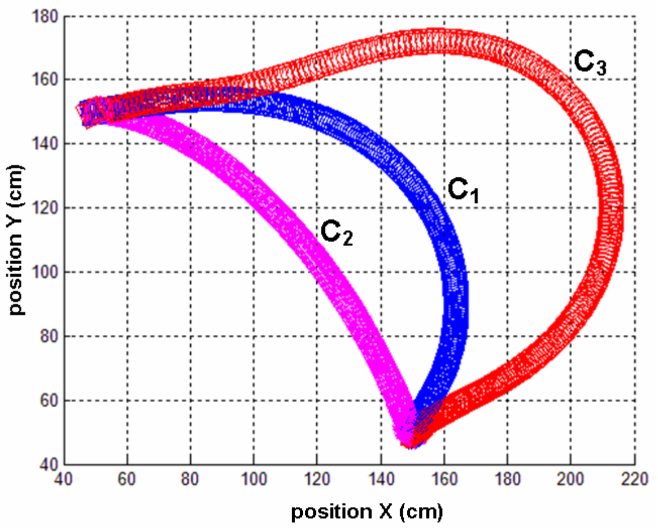


Fig. 5.11. Spatial evolution of the physical agent for a temporal constraint of 4 seconds from an initial location of (150cm, 50cm, 0°) to a final location of (50cm, 150cm, 190°).

The following set of capabilities have been then supposed for this task: the minimal time to perform the task ($Time_A$), the distance error (ΔD_A) and the orientation error ($\Delta\theta_A$) with respect to the final target, the final linear velocity (v_A), the maximal linear and angular velocities (max_v_A and max_ω_A), and the control effort (CE_A) in the execution of the task. Table 5.1 presents the set of the above capabilities associated with each controller for the conditions shown in Fig. 5.11.

Table 5.1. Set of capabilities of the physical agent for a temporal constraint of 4 seconds.

$C_{task_k}(A_\alpha)$	Time$_A$(s)	ΔD$_A$(cm)	$\Delta\theta_A$(°)	v$_A$ (cm/s)	max_v$_A$ (cm/s)	max_ω_A (rad/s)	CE$_A$
c_1	1.0272	0.6506	6.9624	18.5919	25.5216	31.3531	3.7282
c_2	0.6242	0.3705	19.0600	10.5883	18.1886	32	2.7704
c_3	1.7045	1.0735	27.9195	30.6761	48.6848	32	3.9055

However, there are many cases where is not possible to fulfil simultaneously all the required conditions. Fig 5.12 shows, for instance, how the physical agent's dynamics are different when the task requirements change (*temporal constraint of 1 seconds*) under the same environmental conditions (*initial location: (150cm, 50cm, 0 °), final location: (50cm, 150cm, 190 °*)).

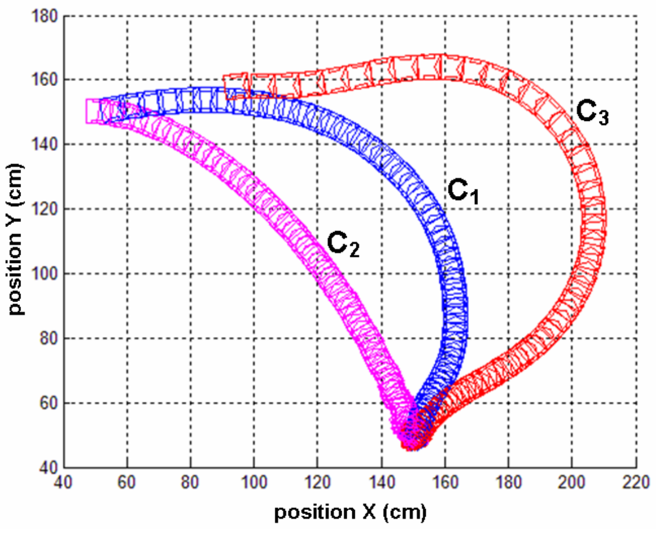
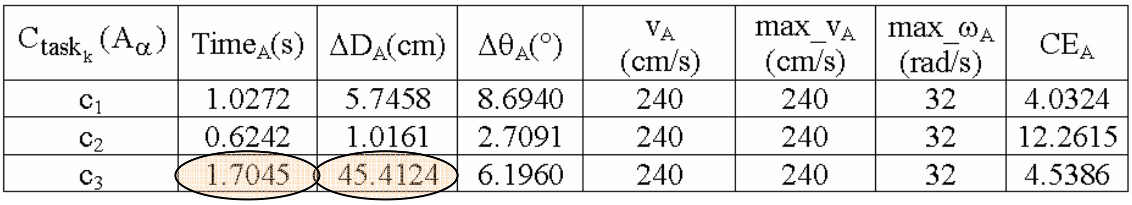


Fig. 5.12. Spatial evolution of the physical agent for a temporal constraint of 1 second from an initial location of (150cm, 50cm, 0°) to a final location of (50cm, 150cm, 190°).

Likewise, Table 5.2 shows how the controllers have different capabilities to perform the same task. In some cases, the proposed task is not correctly performed since the associated capabilities are not the most appropriate to perform it.

Table 5.2. Set of capabilities of the physical agent for a temporal constraint of 1 second.



$C_{task_k}(A_\alpha)$	$Time_A$(s)	ΔD_A(cm)	$\Delta\theta_A$(°)	v_A (cm/s)	max_v_A (cm/s)	max_ω_A (rad/s)	CE_A
c_1	1.0272	5.7458	8.6940	240	240	32	4.0324
c_2	0.6242	1.0161	2.7091	240	240	32	12.2615
c_3	1.7045	45.4124	6.1960	240	240	32	4.5386

Thus, it should be noted that the information provided by the capabilities about dynamics results quite relevant for physical agents to avoid undesirable situations and to fulfill correctly the proposed tasks. The implementation aspects and examples of the algorithms for control and decision used in this work for the passing a ball, the offside maneuver and the team-work coordination case in robot soccer are introduced next.

5.1.6.1 Implementation in the Passing a Ball

The following set of capabilities has been used for passing a ball described in Section 5.1.3.1: the minimal time that needs the shooter to perform the task (*$Time_A$*), the distance error (*ΔD_A*) and the orientation error *($\Delta\theta_A$*) with respect to the final target, and the control effort (*CE_A*) in the execution of the task such that:

$$CC_{c_i,task_1}(A_s)=\{Time_A,\Delta D_A,\Delta\theta_A,CE_A\},\quad i=1,2,3$$

The environmental conditions are related to the shooter's initial state, represented by (*D_2, θ_A*). The task requirements are related to the proposed execution time of the task (*$Time_T$*). It takes into account the ball initial state represented by (*V_0, D_1*) in the approach presented in [De la Rosa et al., 04] where a neural network calculates *$Time_T$* such that:

$$EC(A_s)=\{D_2,\theta_A\},\quad TR=\{Time_T\}\ \mid\ DK_{task_1}(A_s)=\{D_2,\theta_A,task_1,Time_T\}$$

The self-inspection functions $\Psi_{c\,i,task_1}(A_s)$ are here implemented by using neural networks that take into account the agent's domain knowledge $DK_{task_1}(A_s)$ to obtain the agent's capabilities $CC_{c_i,task_1}(A_s)$ according to the available controllers $C_{task_1}(A_s)$ such that:

$$CC_{c_i,task_1}(A_s) = \Psi_{c_i,task_1}(DK_{task_1}(A_s)), \quad i = 1,2,3$$

$$\{Time_A, \Delta D_A, \Delta\theta_A, CE_A\}_{ci} = \Psi_{c_i,task_1}(\{D_2, \theta_A, task_1, Time_T\})$$

Fig. 5.13 shows an example of these capabilities.

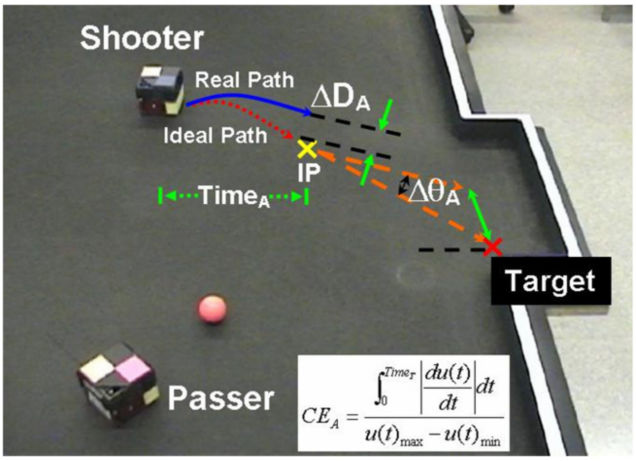

Shooter
Real Path
ΔD_A
Ideal Path
IP
Time$_A$
$\Delta\theta_A$
Target
Passer
$CE_A = \dfrac{\int_0^{Time_T} \left|\frac{du(t)}{dt}\right| dt}{u(t)_{max} - u(t)_{min}}$

Fig. 5.13. Example of the capabilities for passing a ball.

Where: u(t)→ velocity control signal, Time$_T$ → proposed execution time of the task.

The self-evaluation functions $\Phi_{ci,task_1}(A_s)$ are also implemented by using neural networks that calculate the certainty indexes $ci_{c_i,task_1}(A_s) \in [0,1]$ for each controller in the proposed task according to its capabilities {*ΔD$_A$*, *Δθ$_A$*} and the possibility of performing the task given by the condition ({*Time$_A$*}< *Time$_T$*) such that:

$$ci_{c_i,task_1}(A_s) = \Phi_{c_i,task_1}(CC_{c_i,task_1}(A_s)), \quad i = 1,2,3$$

$$ci_{c_i,task_1}(A_s) = \Phi_{c_i,task_1}(\{\Delta D_A, \Delta\theta_A, Time_A < Time_T\})$$

The resulting neural networks for each controller were feed-forward back-propagation networks. The networks were not found by an exhaustive search for the optimal configuration to suit this task, but rather were the quickest and most successful of some alternatives with different numbers of hidden units and different learning rates.

In addition, a fuzzy-based suitability function $\Theta_{ci,task_1}(A_s)$ helps physical agent to know the suitability $sr_{c_i,task_1}(A_s) \in [0,1]$ of each controller in the task according to the evaluation of its respective capabilities. For that, the certainty indexes are jointly used with the capability {*CEA*} in a fuzzy decision maker such that:

$$sr_{c_i,task_1}(A_s) = \Theta_{c_i,task_1}(ci_{c_i,task_1}(A_s), \{CE_A\}_{c_i,task_1}(A_s)), \quad i = 1,2,3$$

This decision maker selects the most suitable controller to execute the task by means of a comparative analysis $\zeta_{task_1}(A_s)$. The fuzzy decision maker is useful when more than one certainty index shows that the task can be correctly performed by more than one controller and the maximal certainty index is not enough to decide the most suitable controller. In this sense, it is necessary to optimize the decision-making structure (*increase the introspection degree*) for this selection problem. Fig. 5.14 shows the criterion to select the controller that best matches the requirements.

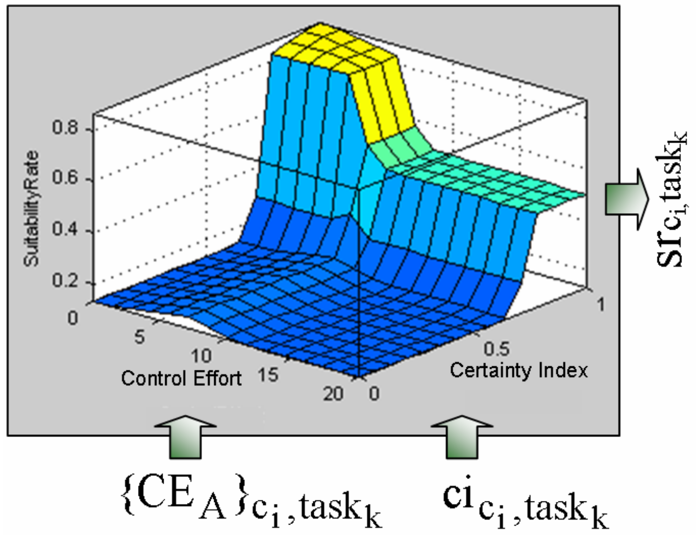


Fig. 5.14. Fuzzy decision maker.

The selection is based on suitability rates $SR_{task_1}(A_s)$. A high or medium $sr_{c_i,task_1}(A_s)$ indicates that the task can be performed, but with different performance levels. A low $sr_{c_i,task_1}(A_s)$ indicates that the task cannot be performed. Appropriate decision thresholds to implement this decision-making structure have been selected.

According to Fig. 5.14 is possible to conclude for each controller:

If (*Certainty Index* is high) and (*Control Effort* is low) then (*Suitability Rate* is high)
If (*Certainty Index* is high) and (*Control Effort* is high) then (*Suitability Rate* is medium)
If (*Certainty Index* is low) and (*Control Effort* is high) then (*Suitability Rate* is low)
If (*Certainty Index* is low) and (*Control Effort* is low) then (*Suitability Rate* is low)

5.1.6.1.1 Decisions Example in the Passing a Ball

The introspection approach based on neural networks (see Section 5.1.6.1) is illustrated for the passing a ball, where each physical agent has the set of controllers shown in Fig. 5.10. An example of the decision algorithm executed by the agents in this task is as follows:

1. *Proposition*: The *passer* proposes to the *shooter* that they perform the pass with a Time$_T$ = 0.39 s. The *shooter* is (D$_2$ = 41.23 cm, θ$_A$= 362.18°) away from the interception point.

$$\mathrm{REQUEST}(A_p, A_s, t_1, \mathrm{task}_1) \text{ where } \mathrm{DK}_{\mathrm{task}_1}(A_s) = \{41.23\mathrm{cm}, 362.18°, \mathrm{pas\,sin\,g}, 0.39\mathrm{s}\}$$

2. *Introspection*: The *shooter* looks for (*self-inspection*) and evaluates (*self-evaluation*) its capabilities to perform the task with its available controllers $C_{\mathrm{task}_1}(A_s)$ and domain knowledge $\mathrm{DK}_{\mathrm{task}_1}(A_s)$ (see Table 5.3).

$$\mathrm{KNOW}_{A_s}^{t_2}\left\{A_s, \mathrm{task}_1, \mathrm{SK}_{\mathrm{task}_1}(A_s)\right\} \text{ and } \mathrm{BELIEF}_{A_s}^{t_2}\left\{\mathrm{CAN}_{A_s}^{t_2}\left[\mathrm{task}_1, \zeta_{\mathrm{task}_1}(A_s)\right]\right\}$$

Table 5.3. Introspection of the shooter on the first passing opportunity.

$C_{\mathrm{task}_1}(A_s)$	$CC_{c_i,\mathrm{task}_1}(A_s)$				$ci_{c_i,\mathrm{task}_1}(A_s)$	$sr_{c_i,\mathrm{task}_1}(A_s)$
	Time$_A$(s)	ΔD$_A$(cm)	Δθ$_A$(°)	CE$_A$	(%)	(%)
c_1	0.58	7.63	3.61	3.98	1.5	15
c_2	0.46	3.90	6.55	11.51	0.04	13
c_3	0.65	7.39	9.80	2.57	0.0009	14

3. *Answer*: The *shooter* tells the *passer* that it can perform the task with the following maximal reliability $\zeta_{\text{task}_1}(A_s) = 15\%$.

$$\text{INFORM}\left(A_s, A_p, t_3, \text{BELIEF}_{A_s}^{t_2}\left\{\text{CAN}_{A_s}^{t_2}\left[\text{task}_1, \zeta_{\text{task}_1}(A_s)\right]\right\}\right)$$

4. *Decision*: The *passer* considers that this reliability is too low.

5. *New proposition*: The *passer* proposes to the *shooter* that they perform the pass with a Time$_T$ = 0.536 s.

$$\text{REQUEST}(A_p, A_s, t_5, \text{task}_1') \text{ where } \text{DK}_{\text{task}_1'}(A_s) = \{41.23\text{cm}, 362.18°, \text{pas sin g}, 0.536\text{s}\}$$

6. *Introspection*: The *shooter* looks for and evaluates its capabilities to perform the task with this $\text{DK}_{\text{task}_1'}(A_s)$ (see Table 5.4).

$$\text{KNOW}_{A_s}^{t_6}\left\{A_s, \text{task}_1', \text{SK}_{\text{task}_1'}(A_s)\right\} \text{ and } \text{BELIEF}_{A_s}^{t_6}\left\{\text{CAN}_{A_s}^{t_6}\left[\text{task}_1', \zeta_{\text{task}_1'}(A_s)\right]\right\}$$

Table 5.4. Introspection of the shooter on the second passing opportunity.

$C_{\text{task}_1'}(A_s)$	$CC_{c_i,\text{task}_1'}(A_s)$				$ci_{c_i,\text{task}_1'}(A_s)$	$sr_{c_i,\text{task}_1'}(A_s)$
	Time$_A$(s)	ΔD_A(cm)	$\Delta\theta_A$(°)	CE$_A$	(%)	(%)
c_1	0.58	5.28	2.12	3.49	4.22	14.84
c_2	0.46	3.95	5.79	9.36	99.9	50
c_3	0.65	5.55	8.78	2.61	0.001	14.13

7. *Answer*: The *shooter* tells the *passer* that it can perform the task with the following maximal reliability $\zeta_{\text{task}_1'}(A_s) = 50\%$.

$$\text{INFORM}\left(A_s, A_p, t_7, \text{BELIEF}_{A_s}^{t_6}\left\{\text{CAN}_{A_s}^{t_6}\left[\text{task}_1', \zeta_{\text{task}_1'}(A_s)\right]\right\}\right)$$

8. *Decision*: The *passer* considers that this reliability is too low.

9. *New proposition*: The *passer* proposes to the *shooter* that they perform the pass with a new Time$_T$ = 0.759 s.

$$\text{REQUEST}(A_p, A_s, t_9, \text{task}_1'') \text{ where } \text{DK}_{\text{task}_1''}(A_s) = \{41.23\text{cm}, 362.18°, \text{pas sin g}, 0.759\text{s}\}$$

10. Introspection: The *shooter* looks for and evaluates its capabilities with this new time (see Table 5.5).

$$\mathrm{KNOW}_{A_s}^{t_{10}}\left\{A_s, \mathrm{task}_1'', \mathrm{SK}_{\mathrm{task}_1''}(A_s)\right\} \text{ and } \mathrm{BELIEF}_{A_s}^{t_{10}}\left\{\mathrm{CAN}_{A_s}^{t_{10}}\left[\mathrm{task}_1'', \zeta_{\mathrm{task}_1''}(A_s)\right]\right\}$$

Table 5.5. Introspection of the shooter on the third passing opportunity.

$C_{\mathrm{task}_1''}(A_s)$	$CC_{c_i,\mathrm{task}_1''}(A_s)$				$ci_{c_i,\mathrm{task}_1''}(A_s)$	$sr_{c_i,\mathrm{task}_1''}(A_s)$
	$Time_A(s)$	$\Delta D_A(cm)$	$\Delta\theta_A(°)$	CE_A	(%)	(%)
c_1	0.58	4.29	1.37	3.25	97.83	85.36
c_2	0.46	4.52	4.94	5.53	99.99	83.15
c_3	0.65	4.35	6.82	2.70	97.18	85.80

11. Answer: The *shooter* tells the *passer* that it can perform the task with the following maximal reliability $\zeta_{\mathrm{task}_1''}(A_s) = 85.80\%$.

12. Decision: The *passer* agrees the reliability is large enough and they can perform the pass. The *shooter* selects then the most suitable controller (c_3) to perform its movement.

$$\mathrm{COMMIT}\left(A_p, A_s, t_{12}, \mathrm{task}_1''\right)$$

5.1.6.2 Implementation in the Offside Maneuver

In the offside maneuver described in Section 5.1.3.2, the introspective reasoning (see Section 4.2.3) is based on the Case-Based Reasoning *(CBR)* methodology. Here, the minimal time that needs the defenders to execute the task (*Time$_A$*) and the control effort (*CE$_A$*) in the execution of the task, have been used as capabilities such that:

$$CC_{c_i,\mathrm{task}_2}(A_{d1}) = \{\mathrm{Time}_A, \mathrm{CE}_A\}, \quad i = 1,2,3$$

$$CC_{c_i,\mathrm{task}_2}(A_{d2}) = \{\mathrm{Time}_A, \mathrm{CE}_A\}, \quad i = 1,2,3$$

The defenders must be able to select the most suitable controller of $C_{\mathrm{task}_2}(A_{d1})$ and $C_{\mathrm{task}_2}(A_{d2})$ to perform the maneuver. Thus, each defender performs introspective reasoning in relation to its body to enhance its decision ability and guarantee sure and trustworthy commitments in this coordinated task.

A case consists of spatial (D_A, θ_A), temporal ($Time_A$) and energy (CE_A) conditions under which each defender can perform the task using the controller *i* taking into account the information about the physical body's dynamics. The case database contains continuously updated representative data of the type: Case = {D_A, θ_A, $Time_A$, CE_A, (c_1 or c_2 or c_3)}. The environmental conditions are related to the defenders' initial states, represented by (D_3, θ_B) (D_4, θ_C). The task requirements are related to the time that the opposing passer strikes the ball ($Time_P$) such that:

$$EC(A_{d1}) = \{D_3, \theta_B\}, EC(A_{d2}) = \{D_4, \theta_C\}, TR = \{Time_P\} \mid$$

$$DK_{task_2}(A_{d1}) = \{D_3, \theta_B, task_2, Time_P\} \text{ and } DK_{task_2}(A_{d2}) = \{D_4, \theta_C, task_2, Time_P\}$$

The self-inspection functions $\Psi_{ci,task_2}(A_{d1})$ and $\Psi_{ci,task_2}(A_{d2})$, perform a progressive filtering (*retrieve: filter 1* and *filter 2*) on the case database. The filtering takes into account the agents' domain knowledge $DK_{task_2}(A_{d1})$ and $DK_{task_2}(A_{d2})$ to obtain the agent's capabilities $CC_{c_i,task_2}(A_{d1})$ and $CC_{c_i,task_2}(A_{d2})$ respectively from the cases most similar to the current situation of each defender such that:

$$CC_{c_i,task_2}(A_{d1}) = \Psi_{c_i,task_2}(DK_{task_2}(A_{d1})), \quad i = 1,2,3 \mid$$

$$\{Time_A, CE_A\}_{c_i} = \Psi_{c_i,task_2}(\{D_3, \theta_B, task_2, Time_P\})$$

$$CC_{c_i,task_2}(A_{d2}) = \Psi_{c_i,task_2}(DK_{task_2}(A_{d2})), \quad i = 1,2,3 \mid$$

$$\{Time_A, CE_A\}_{c_i} = \Psi_{c_i,task_2}(\{D_4, \theta_C, task_2, Time_P\})$$

Where $\Psi_{c_i,task_2}(A_{d1})$:

$$\text{Filter 1: } D_3 - 10\text{cm} \le D_A \le D_3 + 10\text{cm}?$$

$$\text{Filter 2: } \theta_B - 30° \le \theta_A \le \theta_B + 30°?$$

And $\Psi_{c_i,task_2}(A_{d2})$:

$$\text{Filter 1: } D_4 - 10\text{cm} \le D_A \le D_4 + 10\text{cm}?$$

$$\text{Filter 2: } \theta_C - 30° \le \theta_A \le \theta_C + 30°?$$

The self-evaluation functions $\Phi_{ci,task_2}(A_{d1})$ and $\Phi_{ci,task_2}(A_{d2})$ (*filter 3*) allow selection of the set of controllers with which the agents can perform the task. In addition, the self-evaluation calculates the certainty indexes $ci_{c_i,task_2}(A_{d1}) \in [0,1]$ and $ci_{c_i,task_2}(A_{d2}) \in [0,1]$ for each controller. The calculation takes into account the ratio between the retrieved cases (*RCases_ci*) for the controller *i* and all retrieved cases (*AllRCases*).

$$ci_{c_i,task_2}(A_{d1}) = \Phi_{c_i,task_2}(CC_{c_i,task_2}(A_{d1})), \quad i = 1,2,3$$

$$ci_{c_i,task_2}(A_{d2}) = \Phi_{c_i,task_2}(CC_{c_i,task_2}(A_{d2})), \quad i = 1,2,3$$

Where $\Phi_{c_i,task_2}(A_{d1})$: Filter 3: $Time_A < Time_P$, $ci_{c_i,task_2}(A_{d1}) = \dfrac{RCases\_c_i(A_{d1})}{AllRCases(A_{d1})}$

Where $\Phi_{c_i,task_2}(A_{d2})$: Filter 3: $Time_A < Time_P$, $ci_{c_i,task_2}(A_{d2}) = \dfrac{RCases\_c_i(A_{d2})}{AllRCases(A_{d2})}$

Similarly to the introspection approach for passing a ball, the certainty indexes can be jointly used with the capability {*CEA*} in the fuzzy decision maker shown in Fig. 5.14. In this sense, it has been selected as {*CEA*}, for the controller *i,* the minimum among all the retrieved cases for the same controller. Thus, the defenders select then the most suitable controllers to execute the offside.

A new solution (*reuse*) is generated from the retrieved cases according to the problem conditions. A revision of the proposed solution is done (*revise*) to evaluate the obtained results and verify that the solution is satisfactory. Finally, the problem conditions and the proposed solution are indexed (*retain*) to use in successive iterations of the *CBR* cycle if the results after the evaluation have been satisfactory.

The tasks performance improves with these new and effective approaches because the physical agents can manage their physical bodies according to their capabilities. Thus, physical agents have introspection on what they can and cannot do and how they are able to perform the tasks according to their physical limitations, mainly related to their dynamics.

5.1.6.2.1 Decisions Example in the Offside Maneuver

Similarly, an example of the decision algorithm executed by the defenders in the offside maneuver using the introspection approach based on *CBR* (see Section 5.1.6.2) is as follows:

1. *Proposition*: The *passer* of the opposite team has a time (Time$_P$ = 0.63 s) to pass the ball. Thus, *defender 1* proposes an offside maneuver to *defender 2*. *Defender 1* is (D$_3$ = 44.72 cm, θ$_B$ = 337.38°) away from the offside line. *Defender 2* is (D$_4$ = 63.23 cm, θ$_C$ = 355.14°) away from the same offside line.

$$\text{REQUEST}(A_{d1}, A_{d2}, t_1, \text{task}_2) \text{ where}$$

$$DK_{\text{task}_2}(A_{d1}) = \{44.72\text{cm}, 337.38°, \text{offside}, 0.63\text{s}\} \text{ and}$$

$$DK_{\text{task}_2}(A_{d2}) = \{63.23\text{cm}, 355.14°, \text{offside}, 0.63\text{s}\}$$

2. *Introspection*: The defenders look for and evaluate their capabilities to perform the task (see Table 5.6).

$$A_{d1} : \text{KNOW}_{A_{d1}}^{t_2}\left\{A_{d1}, \text{task}_2, SK_{\text{task}_2}(A_{d1})\right\} \text{ and BELIEF}_{A_{d1}}^{t_2}\left\{\text{CAN}_{A_{d1}}^{t_2}\left[\text{task}_2, \zeta_{\text{task}_2}(A_{d1})\right]\right\}$$

$$A_{d2} : \text{KNOW}_{A_{d2}}^{t_2}\left\{A_{d2}, \text{task}_2, SK_{\text{task}_2}(A_{d2})\right\} \text{ and BELIEF}_{A_{d2}}^{t_2}\left\{\text{CAN}_{A_{d2}}^{t_2}\left[\text{task}_2, \zeta_{\text{task}_2}(A_{d2})\right]\right\}$$

Table 5.6. Introspection of the defenders about the first offside opportunity.

$C_{task_2}(A_{d1})$	$CC_{c_i,task_2}(A_{d1})$		$ci_{c_i,task_2}(A_{d1})(\%)$	$sr_{c_i,task_2}(A_{d1})(\%)$
	Time$_A$(s)	CE$_A$		
c_1	0.57	3.37	97.80	85.33
c_2	0.46	7.57	99.00	82.33
c_3	0.62	2.65	99.91	88.21
$C_{task_2}(A_{d2})$	$CC_{c_i,task_2}(A_{d2})$		$ci_{c_i,task_2}(A_{d2})(\%)$	$sr_{c_i,task_2}(A_{d2})(\%)$
	Time$_A$(s)	CE$_A$		
c_1	0.60	3.43	1.15	14.79
c_2	0.47	8.85	99.00	50
c_3	0.95	2.66	0.0009	14.13

3. *Answer*: *Defender 2* tells *defender 1* that it can perform the task with the following maximal reliability $\zeta_{\text{task}_2}(A_{d2}) = 50\%$.

$$\text{INFORM}\left(A_{d2}, A_{d1}, t_3, \text{BELIEF}^{t_2}_{A_{d2}}\left\{\text{CAN}^{t_2}_{A_{d2}}\left[\text{task}_2, \zeta_{\text{task}_2}(A_{d2})\right]\right\}\right)$$

4. *Decision*: *Defender 1* considers that this reliability is too low, so the defenders do not commit to perform the task and instead wait for a new opportunity to avoid a pass between two opposing physical agents.

5. *New proposition*: At another time the task requirement is (Time_P = 0.74 s), and the environmental conditions are (D_3 = 41.23 cm, θ_B = 362.18°), (D_4 = 60.83 cm, θ_C = 382.26°) for a new offside line.

$$\text{REQUEST}(A_{d1}, A_{d2}, t_5, \text{task}'_2) \text{ where}$$

$$DK_{\text{task}'_2}(A_{d1}) = \{41.23\text{cm}, 362.18°, \text{offside}, 0.74\text{s}\} \text{ and}$$

$$DK_{\text{task}'_2}(A_{d2}) = \{60.83\text{cm}, 382.26°, \text{offside}, 0.74\text{s}\}$$

6. *Introspection*: Each defender looks for and evaluates the controllers with which they can perform the offside maneuver in this new opportunity (see Table 5.7).

$$A_{d1} : \text{KNOW}^{t_6}_{A_{d1}}\left\{A_{d1}, \text{task}'_2, SK_{\text{task}'_2}(A_{d1})\right\} \text{ and } \text{BELIEF}^{t_6}_{A_{d1}}\left\{\text{CAN}^{t_6}_{A_{d1}}\left[\text{task}'_2, \zeta_{\text{task}'_2}(A_{d1})\right]\right\}$$

$$A_{d2} : \text{KNOW}^{t_6}_{A_{d2}}\left\{A_{d2}, \text{task}'_2, SK_{\text{task}'_2}(A_{d2})\right\} \text{ and } \text{BELIEF}^{t_6}_{A_{d2}}\left\{\text{CAN}^{t_6}_{A_{d2}}\left[\text{task}'_2, \zeta_{\text{task}'_2}(A_{d2})\right]\right\}$$

Table 5.7. Introspection of the defenders about the second offside opportunity.

$C_{\text{task}'_2}(A_{d1})$	$CC_{c_i,\text{task}'_2}(A_{d1})$		$ci_{c_i,\text{task}'_2}(A_{d1})(\%)$	$sr_{c_i,\text{task}'_2}(A_{d1})(\%)$
	Time_A(s)	CE_A		
c_1	0.58	3.25	97.83	85.36
c_2	0.46	5.53	99.90	83.15
c_3	0.65	2.70	97.18	85.80
$C_{\text{task}'_2}(A_{d2})$	$CC_{c_i,\text{task}'_2}(A_{d2})$		$ci_{c_i,\text{task}'_2}(A_{d2})(\%)$	$sr_{c_i,\text{task}'_2}(A_{d2})(\%)$
	Time_A(s)	CE_A		
c_1	0.61	3.35	96.66	85.28
c_2	0.47	8.08	99.90	50
c_3	0.93	2.66	91.27	85.84

7. *Answer*: *Defender 2* tells *defender 1* that it can perform the task with the following maximal reliability $\zeta_{task'_2}(A_{d2})$ = 85.84%. Likewise, *defender1* has the following reliability $\zeta_{task'_2}(A_{d1})$ = 85.80%.

$$\mathrm{INFORM}\left(A_{d2}, A_{d1}, t_7, \mathrm{BELIEF}_{A_{d2}}^{t_6}\left\{\mathrm{CAN}_{A_{d2}}^{t_6}\left[task'_2, \zeta_{task'_2}(A_{d2})\right]\right\}\right)$$

8. *Decision*: *Defender 1* agrees the reliability is large enough and they can perform the task. Defenders 1 and 2 identify the most suitable controller (c_3) to perform the coordinated task and commit to executing it. In this example, the defenders selected the same controller to perform the task, but this not always happens.

$$\mathrm{COMMIT}\left(A_{d1}, A_{d2}, t_8, task'_2\right)$$

5.1.6.3 Implementation in the Team-work Coordination Case Study

In the literature, there are several coordination parameters to take into account in the utility/cost functions for allocating tasks. However, in this implementation, the *introspection* has been considered as one of them jointly with two more: *proximity* and *trust*.

Here, each physical agent has a set of movement controllers to execute the tasks described in Section 5.1.4 such that:

$$C(A_1) = C(A_2) = C(A_3) = C(A_4) = C(A_5) = \{c_1, c_2, c_3\} \quad \text{and}$$
$$C_{task_k}(A_1) = C_{task_k}(A_2) = C_{task_k}(A_3) = C_{task_k}(A_4) = C_{task_k}(A_5) = \{c_1, c_2, c_3\}$$

In addition, the following set of capabilities has been used to represent the physical agents' dynamics: the distance error (ΔD_A) and the orientation error ($\Delta\theta_A$) with respect to the desired final target such that:

$$CC_{c_i, task_k}(A_j) = \{\Delta D_A, \Delta\theta_A\}, \quad i = 1,2,3 \quad \text{and} \quad j = 1,2,3,4,5$$

The introspection (see Section 4.2.3) is implemented using feed-forward back-propagation neural networks. The environmental conditions *EC* are related to the agents' initial state, represented by the distance $\mathrm{D_j}$ and orientation θ_j of each agent with respect to the final target. The task requirements *TR* are related to the current scene in the environment $(\mathrm{s_l})$.

$$\mathrm{EC(A_j)=\{D_j,\theta_j\},\quad TR=\{s_l\}\ \mid\ DK_{task_k}(A_j)=\{D_j,\theta_j,task_k,s_l\}}$$

The self-inspection function $\Psi_{\mathrm{c\,i,task_k}}(\mathrm{A_j})$ (*set of neural networks 1*) takes into account the agent's domain knowledge $\mathrm{DK_{task_k}(A_j)}$ to obtain the agent's capabilities $\mathrm{CC_{c_i,task_k}(A_j)}$ according to the available controllers $\mathrm{C_{task_k}(A_j)}$ such that:

$$\mathrm{CC_{c_i,task_k}(A_j)=\Psi_{c_i,task_k}(DK_{task_k}(A_j)),\quad i=1,2,3\quad and\quad j=1,2,3,4,5}$$

$$\mathrm{\{\Delta D_A,\Delta\theta_A\}_{c\,i}=\Psi_{c_i,task_k}(\{D_j,\theta_j,task_k,s_l\})\quad i=1,2,3\quad and\quad j=1,2,3,4,5\quad l=1,2,3}$$

The self-evaluation function $\Phi_{\mathrm{c\,i,task_k}}(\mathrm{A_j})$ (*set of neural networks 2*) calculates the certainty index $\mathrm{ci_{c_i,task_k}(A_j)\in[0,1]}$ for each controller in the proposed task according to its capabilities $\mathrm{\{\Delta D_A,\Delta\theta_A\}}$ such that:

$$\mathrm{ci_{c_i,task_k}(A_j)=\Phi_{c_i,task_k}(CC_{c_i,task_k}(A_j)),\quad i=1,2,3\quad and\quad j=1,2,3,4,5}$$

$$\mathrm{ci_{c_i,task_k}(A_j)=\Phi_{c_i,task_k}(\{\Delta D_A,\Delta\theta_A\})}$$

The set of all certainty indexes $\mathrm{CI_{task_k}(A_j)}$ for a specific task *k* is constituted by all $\mathrm{ci_{c_i,task_k}}$ of the possible controllers of the agent $\mathrm{A_j}$ in this task.

The introspection parameter $\mathrm{i_{task_k}(A_j)\in[0,1]}$ is then calculated by means a comparative analysis between all possible certainty indexes $\mathrm{CI_{task_k}(A_j)}$ that allows physical agent, if it is possible, to select a controller for the execution of this task as is described in (5.10).

$$\mathrm{i_{task_k}(A_j)=max(CI_{task_k}(A_j))}\quad (5.10)$$

A high $i_{task_k}(A_j)$ value represents that the agent A_j can correctly perform the task *k*. Likewise, a low $i_{task_k}(A_j)$ value indicates that the agent cannot perform the task. Fig 5.15 depicts a scheme of the agents in the environment.

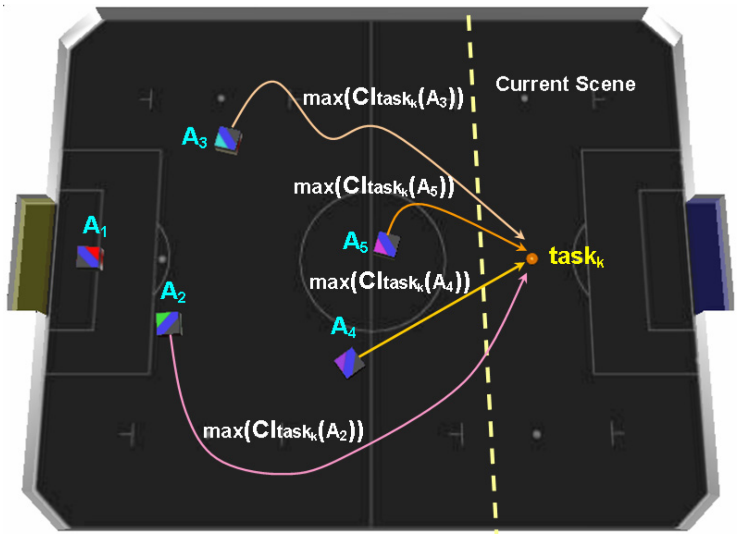


Fig. 5.15. General scheme of the robot soccer environment for the introspection calculation.

On the other hand, proximity represents the physical situation of each agent within an environment. The proximity parameter $p_{task_k}(A_j) \in [0,1]$ is related to the distance between the current location of the agent A_j and the location of the target as is described in (5.11).

$$p_{task_k}(A_j) = \left(1 - \frac{d_{task_k}(A_j)}{d_{max}}\right) \quad (5.11)$$

Where $d_{task_k}(A_j)$ is the distance between the physical agent A_j and the target $task_k$ and d_{max} establishes empirically a fixed maximal radius limit according to the target's location. Here, a radius limit of d_{max} = 110 cm has been used. Physical agents outside of such limit have a $p_{task_k}(A_j)$ = 0. Fig. 5.16 depicts a scheme of the physical agents' state for the proximity calculation.

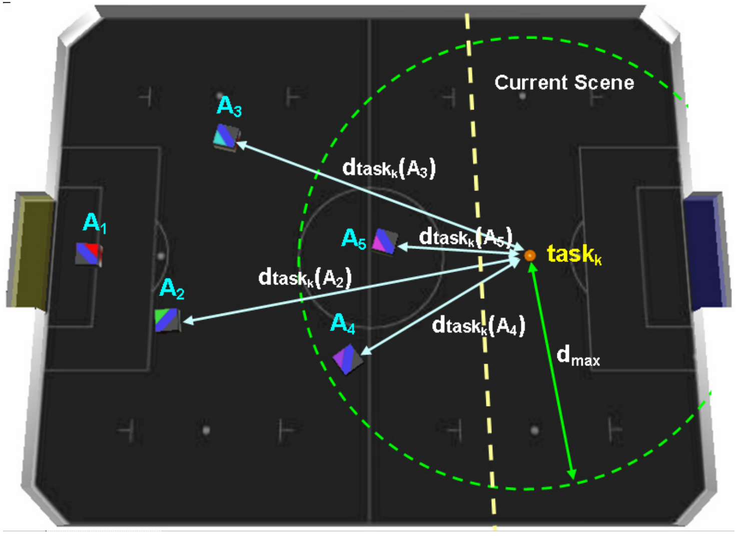


Fig. 5.16. General scheme of the robot soccer environment for the proximity calculation.

Trust represents the social relationship among physical agents that rule the interaction and behavior of them. The trust parameter $t_{task_k}(A_j) \in [0,1]$ takes into account the result of the past interactions of a physical agent with others. The performance of the proposed task is then evaluated based on $t_{task_k}(A_j)$. Equation (5.12) shows the reinforcement calculus if goals are correctly reached by the agent. Otherwise, the agent is penalized if goals are not reached using (5.13).

$$t_{task_k}(A_j) = t_{task_k}(A_j) + \Delta a_{task_k}(A_j) \quad (5.12)$$

$$t_{task_k}(A_j) = t_{task_k}(A_j) - \Delta p_{task_k}(A_j) \quad (5.13)$$

Where $\Delta a_{task_k}(A_j)$ and $\Delta p_{task_k}(A_j)$ are the awards and punishments given to A_j in the task *k* respectively. For the sake of simplicity, a task is correctly performed by an agent when it kicks the ball and the ball goes toward the opposite goal. The agent is then awarded. Otherwise, a task is not well performed when the agent kicks the ball and the ball goes toward its own goal. The agent is then penalized.

A high $t_{task_k}(A_j)$ value represents a more trusted physical agent in the task. Specifically, different trust values have been established for each agent depending on the scene as in (5.14) and (5.15):

$$t_{task_k,s_l}(A_j) = t_{task_k,s_l}(A_j) + \Delta a_{task_k,s_l}(A_j) \quad (5.14)$$

$$t_{task_k,s_l}(A_j) = t_{task_k,s_l}(A_j) - \Delta p_{task_k,s_l}(A_j) \quad (5.15)$$

Where appropriate values for the awards and punishments have been empirically selected for each scene. In particular, if s1 = defense, s2 = midfield and s3 = attack then $\Delta a_{task_k,s_1}(A_j) = 0.1$, $\Delta a_{task_k,s_2}(A_j) = 0.04$, $\Delta a_{task_k,s_3}(A_j) = 0.05$, $\Delta p_{task_k,s_1}(A_j) = 0.1$, $\Delta p_{task_k,s_2}(A_j) = 0.08$, $\Delta p_{task_k,s_3}(A_j) = 0.1$. Fig 5.17 depicts a scheme of the robot soccer environment for the trust calculation.

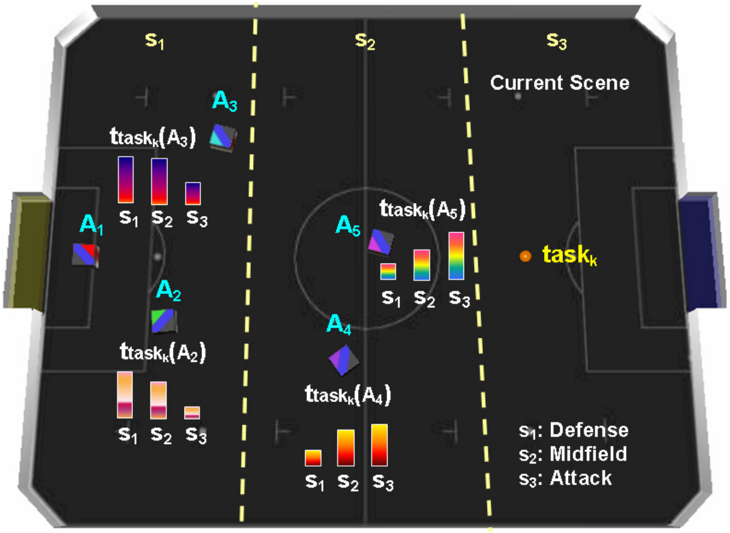


Fig. 5.17. General scheme of the robot soccer environment for the trust calculation.

The utility/cost function $u_{task_k}(A_j) \in [0,1]$ is therefore constituted as a proper average of the element-by-element multiplication of the tuples as in (5.16).

$$u_{task_k}(A_j) = \frac{\sum \left(Th_{task_k} \cdot Ok_{task_k} \cdot Pa_{task_k}(A_j)\right)}{\sum Ok_{task_k}} \quad (5.16)$$

Where $Pa_{task_k}(A_j)$ is a tuple formed by the coordination parameters such that:

$$Pa_{task_k}(A_j) = [i_{task_k}(A_j) \; p_{task_k}(A_j) \; t_{task_k}(A_j)]$$

$\mathrm{Th}_{\mathrm{task}_k}$ is a set of flags (1 or 0) that establishes if the above coordination parameters fulfill their respective decision thresholds such that:

$$\mathrm{Th}_{\mathrm{task}_k} = [\mathrm{i\_th}_{\mathrm{task}_k} \ \mathrm{p\_th}_{\mathrm{task}_k} \ \mathrm{t\_th}_{\mathrm{task}_k}]$$

Appropriate decision thresholds to set or not the above flags have been empirically selected. For this implementation, the thresholds to overcome for each coordination parameter are: $\mathrm{i\_lim}_{\mathrm{task}_k} = 0.65, \quad \mathrm{p\_lim}_{\mathrm{task}_k} = 0.5, \quad \mathrm{t\_lim}_{\mathrm{task}_k} = 0.5$.

$\mathrm{Ok}_{\mathrm{task}_k}$ is a set of flags (1 or 0) that establishes if the above coordination parameters are currently taking into account in the utility/cost function calculation such that:

$$\mathrm{Ok}_{\mathrm{task}_k} = [\mathrm{i\_ok}_{\mathrm{task}_k} \ \mathrm{p\_ok}_{\mathrm{task}_k} \ \mathrm{t\_ok}_{\mathrm{task}_k}]$$

Thus, a task is allocated to the most suitable physical agent according to the value of its utility/cost function (see Equation 5.16) (*1st level of decision*). However, if more than one agent can perform the task correctly, taking into account all the above-mentioned conditions for such functions, then a calculus $\mathrm{ur}_{\mathrm{task}_k}(\mathrm{A}_j) \in [0,1]$ based on the raw values of the coordination parameters is made (*2nd level of decision*). Such calculus does not consider the decisions thresholds represented in (5.16) by the flags $\mathrm{Th}_{\mathrm{task}_k}$. If the raw calculus does not allow selecting an agent, in this implementation, the agent is then selected taking into account some previous knowledge of its physical capabilities (*3rd level of decision*).

In summary, to achieve sure and trustworthy task allocations, each physical agent must use the above coordination parameters in the utility/costs functions before performing such tasks. Specifically, introspection contributes to improve the efficiency of the multi-agent system. Without introspection, physical agents would try to perform actions with no sense, decreasing the number of successful tasks performed.

5.1.6.3.1 Task Allocation Example in the Team-work Coordination Case Study

The following tables show real data related to the proposed approach and the case study described in Section 5.1.6.3. The *SimuroSOT* simulator has been used in this empirical testing. An example of the decision algorithm executed by the agents to allocate the tasks is as follows:

Ten (10) trials have been arbitrarily selected from the logs of a simulation where our team with Ok_{task_k} = [1 1 1] play versus a default opponent robotic team provided by the simulator. Table 5.8 presents the distances of each agent A_j to the target $task_k$ of the scene s_1.

Table 5.8. Distances between agents and target.

Trial	$d_{task_k}(A_2)$	$d_{task_k}(A_3)$	$d_{task_k}(A_4)$	$d_{task_k}(A_5)$
1	38.4795	62.5956	111.472	102.185
2	61.8435	63.3838	116.423	54.0086
3	125.015	84.5454	54.9207	182.168
4	105.922	59.6833	119.725	157.021
5	92.8696	114.692	90.1078	27.8692
6	18.9654	16.1307	86.4507	75.7643
7	49.7483	134.757	61.4014	28.4479
8	30.3353	99.6993	8.97904	149.18
9	54.386	17.7493	55.6383	44.1821
10	55.8887	19.1409	59.1438	49.2439

It is possible the calculation of the proximity parameter by using the above distances. Thus, Table 5.9 shows the values of the proximity parameter for each agent in each trial, highlighting those that overcome the established decision threshold $p\_lim_{task_k}$.

Likewise, Table 5.10 and Table 5.11 show the values of the introspection and trust parameters for each agent in each trial, highlighting those that overcome the established decision thresholds $i\_lim_{task_k}$ and $t\_lim_{task_k}$ respectively.

The flags $i\_th_{task_k}$, $p\_th_{task_k}$ and $t\_th_{task_k}$ are then set to 1 for the highlighted values while they are set to 0 for the remaining cases.

Table 5.9. Values of the proximity parameter.

Trial	$p_{task_k}(A_2)$	$p_{task_k}(A_3)$	$p_{task_k}(A_4)$	$p_{task_k}(A_5)$
1	0.650187	0.430949	0.00	0.0710424
2	0.437787	0.423784	0.00	0.509013
3	0.00	0.231405	0.500721	0.00
4	0.0370688	0.457424	0.00	0.00
5	0.155731	0.00	0.180838	0.746644
6	0.827588	0.853358	0.214085	0.311234
7	0.547742	0.00	0.441805	0.741383
8	0.724224	0.0936423	0.918372	0.00
9	0.505582	0.838643	0.494197	0.598345
10	0.49192	0.825992	0.462329	0.552328

Table 5.10. Values of the introspection parameter.

Trial	$i_{task_k}(A_2)$	$i_{task_k}(A_3)$	$i_{task_k}(A_4)$	$i_{task_k}(A_5)$
1	0.6844	0.833842	0.364194	0.352195
2	0.6844	0.920914	0.365322	0.374756
3	0.0324417	0.00423809	0.367697	0.04798
4	0.148058	0.0439432	0.364	0.32911
5	0.6844	0.920919	0.367667	0.398985
6	0.148074	0.0896823	0.367696	0.392799
7	0.106341	0.000333113	0.0778713	0.89794
8	0.105948	0.182187	0.347556	0.662322
9	0.6844	0.0441111	0.364409	0.403456
10	0.6844	0.0440691	0.364392	0.402255

Table 5.11. Values of the trust parameter.

Trial	$t_{task_k}(A_2)$	$t_{task_k}(A_3)$	$t_{task_k}(A_4)$	$t_{task_k}(A_5)$
1	0.7	0.7	0.6	0.5
2	0.36	0.44	0.6	0.56
3	0.6	0.5	0.6	0.8
4	0.6	0.6	0.6	0.6
5	0.55	0.5	0.5	0.9
6	0.6	0.6	0.6	0.52
7	0.7	0.7	0.6	0.5
8	0.6	0.5	0.6	0.8
9	0.6	0.6	0.6	0.6
10	0.6	0.6	0.6	0.6

The values of the utility/cost functions are calculated by using (5.16) according to the information provided in Tables 5.9, 5.10 and 5.11 and the tuples Ok_{task_k} and Th_{task_k}. Table 5.12 presents the results for these calculations. In this table, it is highlighted the highest value in the trials where it is possible to select easily the most suitable agent to perform the task. However, in the trials 4, 6 and 8 is necessary the raw values calculation. In these cases, it is more difficult to discern, with a certainty greater than 98%, the most suitable agent between the agents with the highest values of the utility/cost function.

Table 5.12. Values of the utility/cost functions.

Trial	$u_{task_k}(A_2)$	$u_{task_k}(A_3)$	$u_{task_k}(A_4)$	$u_{task_k}(A_5)$
1	0.678196	0.511281	0.2	0.166667
2	0.228133	0.306971	0.2	0.356338
3	0.2	0.166667	0.366907	0.266667
4	0.2	0.2	0.2	0.2
5	0.411467	0.47364	0.166667	0.548881
6	0.475863	0.484453	0.2	0.173333
7	0.415914	0.233333	0.2	0.713108
8	0.441408	0.166667	0.506124	0.487441
9	0.596661	0.479548	0.2	0.399448
10	0.428133	0.475331	0.2	0.384109

Therefore, Table 5.13 shows mainly the raw values calculation for the trial 4, 6 and 8.

Table 5.13. Raw values of the utility/cost functions.

Trial	$ur_{task_k}(A_2)$	$ur_{task_k}(A_3)$	$ur_{task_k}(A_4)$	$ur_{task_k}(A_5)$
1	0.699701	0.709376	0	0
2	0.511718	0.546565	0	0
3	0.554111	0	0	0.666779
4	0.261709	0.367122	0.321333	0.309703
5	0.511718	0.546565	0	0
6	0.52522	0.514347	0	0
7	0.476139	0	0	0.563211
8	0	0	0.621976	0.487441
9	0	0.497658	0	0.627652
10	0	0.497658	0	0.627652

However, it is difficult again to select for the case 6, with a high certainty, an agent between the most possible. In this sense, the selection is based on some previous knowledge of their physical capabilities. A task is then allocated to A_2, A_5, A_3 and A_4 respectively, where A_2 is the quickest agent and A_4 is the slowest agent.

In summary, the example shows in Table 5.14 the final results of our task allocation algorithm. The table presents the agent (A_1 or A_2 or A_3 or A_4 or A_5) selected for the execution of the proposed task in the current scene (s_1 or s_2 or s_3) and the level of decision (1 or 2 or 3) used for the allocation.

Table 5.14. Selected agent A_j in the scene s_i.

Trial	Scene	Level	Agent
1	s_1	1	A_2
2	s_2	1	A_5
3	s_3	1	A_4
4	s_1	2	A_3
5	s_3	1	A_5
6	s_2	3	A_2
7	s_1	1	A_5
8	s_3	2	A_4
9	s_2	1	A_2
10	s_2	1	A_3

5.1.6.4 Implementation in the Linear Control System Case Study

Different dynamics can be designed by using the linearized model provide in (5.6) and (5.7) depending on the control engineer's criteria. The following couples $\{\zeta, \omega_n\}$ = {0.4, 6}, {0.6, 10}, {0.8, 4} have been selected to design three movement controllers (c_1, c_2, and c_3). Such controllers generate different dynamics as it is shown in the Fig. 5.18 according to the paths described by the robot. Fig. 5.19 shows the step responses of this model for the designed controllers.

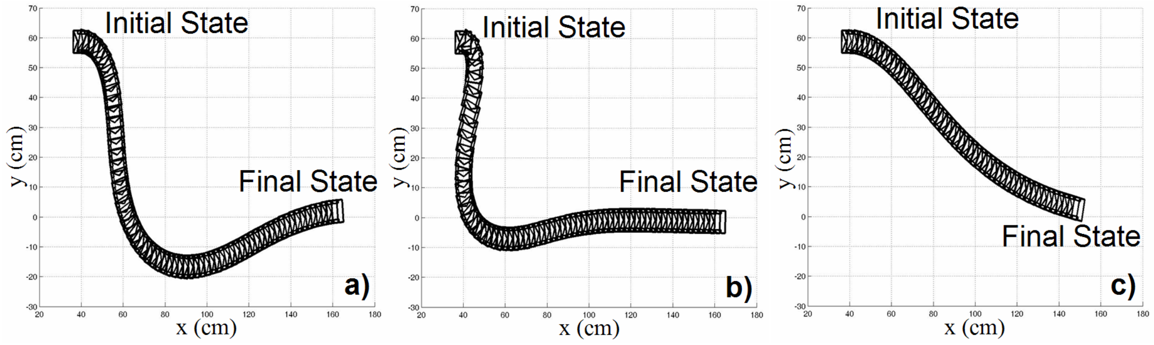


Fig. 5.18. Different dynamics generated with three different controllers. a). c_1 *; b).* c_2 *; c).* c_3 .

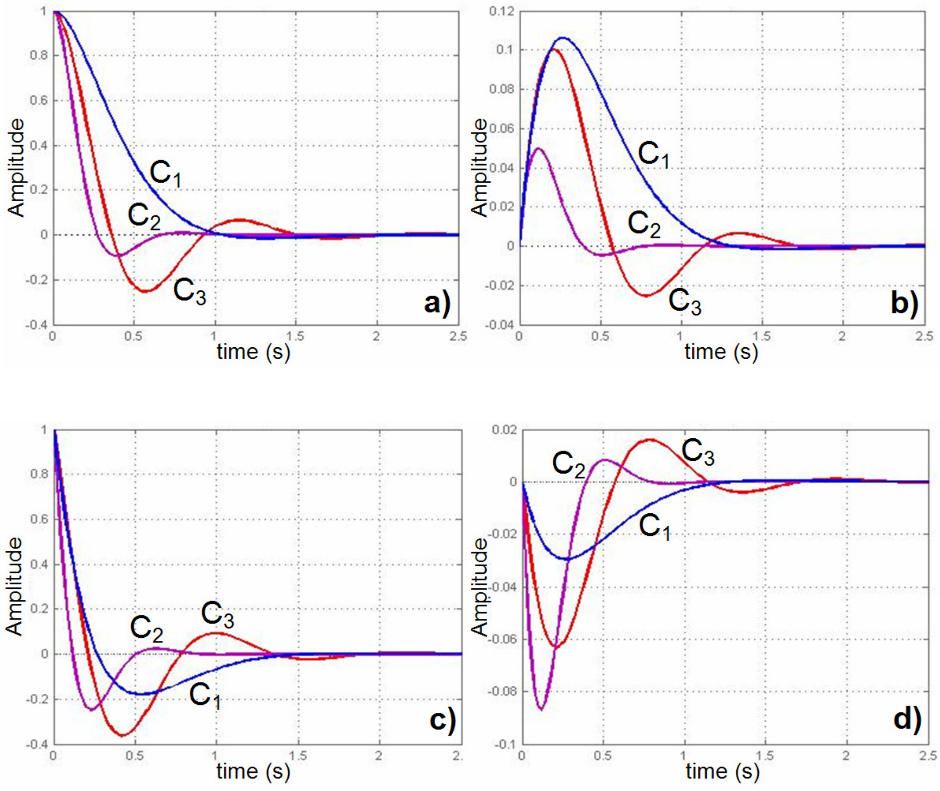


Fig. 5.19. Step responses components: a). Y(s)/y(0); b). Y(s)/θ(0); c). θ(s)/θ(0); d). θ(s)/y(0).

Essentially, this work focuses on the relevance of control-grounded capabilities in the decision-making of physical agents. Some control-grounded capabilities were then calculated by using the definitions described in Section 4.2.6 and the step responses in Fig. 5.19. In this particular implementation, all the weight coefficients are fixed to 1 to give the maximum relevance degree for each capability component. Moreover, it has been supposed that 100% of disturbances that affect the system are steps with amplitude of 3.75 cm and duration of 1 s (e.g., a probable crash with other mobile robot.). Table 5.15 shows the control-grounded capabilities calculated for each movement controller.

Table 5.15. Control-grounded capabilities of the movement controllers.

Control	μ(%)	σ(%)	γ(%)	α(%)	ε(%)	κ
C_1	77.83	-42.57	57.54	93.14	81.62	1
C_2	86.25	39.10	59.46	88.93	75.20	1
C_3	87.48	-4.03	57.08	80.30	100	1

For instance, Table 5.16 shows the information about the rise time t_r and the settling time t_s using the step responses in Fig. 5.19. These are useful to calculate the aggressiveness of each controller as it is shown for c_1 in (5.17) using (4.15).

Table 5.16. Rise time and settling time for each controller.

Fig. 5.19	t_r (s)			t_s (s)		
	C_1	C_2	C_3	C_1	C_2	C_3
a	0.245	0.186	0.617	1.40	0.595	0.939
b	0	0	0	1.61	0.711	1.21
c	0.165	0.089	0.165	1.66	0.687	1.26
d	0	0	0	1.61	0.711	1.21

$$\{\alpha(\%)\}_{c_1}=\frac{\dfrac{100\times\left(1-\dfrac{0.245}{1.40}\right)+100\times\left(1-\dfrac{0}{1.61}\right)}{2}+\dfrac{100\times\left(1-\dfrac{0.165}{1.66}\right)+100\times\left(1-\dfrac{0}{1.61}\right)}{2}}{2}\,\% \tag{5.17}$$

$$\{\alpha(\%)\}_{c_1}=93.14\,\%$$

Likewise, the aggressiveness can be calculated for the other controllers using the respective rise times and the settling times based on (4.15).

The following set of capabilities has been then used for the execution of the coordinated tasks (*passing a ball and offside maneuver*): the overshoot (μ), the speediness (σ), the persistence (γ), the aggressiveness (α), the control effort (ε) and the control kind (κ) such that:

For the passing a ball:

$$CC_{c_i,task_1}(A_s) = \{\mu, \sigma, \gamma, \alpha, \varepsilon, \kappa\}, \quad i = 1,2,3$$

For the offside maneuver:

$$CC_{c_i,task_2}(A_{d1}) = \{\mu, \sigma, \gamma, \alpha, \varepsilon, \kappa\}, \quad i = 1,2,3$$

$$CC_{c_i,task_2}(A_{d2}) = \{\mu, \sigma, \gamma, \alpha, \varepsilon, \kappa\}, \quad i = 1,2,3$$

The self-inspection functions $\Psi_{ci,task_1}(A_s)$, $\Psi_{ci,task_2}(A_{d1})$ and $\Psi_{ci,task_2}(A_{d2})$ allow the calculation of the capabilities according to the considerations established by the corresponding equations presented in Section 4.2.6.

Specifically, the introspection takes into account the environmental conditions and the task requirements in order to obtain the agent's certainty indexes $CI_{task_1}(A_s)$, $CI_{task_2}(A_{d1})$ and $CI_{task_2}(A_{d2})$ in the execution of the proposed tasks with each available controller of the sets $C_{task_1}(A_s)$, $C_{task_2}(A_{d1})$ and $C_{task_2}(A_{d2})$ respectively.

For passing a ball, the environment conditions are related to the *shooter* initial state represented by (*D*D_2, θ_A). The task requirements are related to the proposed execution time of the task ($Time_T$). A combination of the situations mentioned before with their influence degrees (ID_{task_1}) must be also taken into account in the task requirements such that:

$$EC(A_s) = \{D_2, \theta_A\}, \quad TR = \{Time_T, ID_{task_1}\} \mid$$

$$DK_{task_1}(A_s) = \{D_2, \theta_A, task_1, Time_T, ID_{task_1}\}$$

Likewise, for the offside maneuver the environment conditions are related to the initial state of the defenders (D_3, θ_B) and (D_4, θ_C) and the task requirements correspond to ($Time_P$), in the analysed combination of situations with a set of influence degrees (ID_{task_2}) such that:

$$EC(A_{d1}) = \{D_3, \theta_B\}, EC(A_{d2}) = \{D_4, \theta_C\}, TR = \{Time_P, ID_{task_2}\} \mid$$

$$DK_{task_2}(A_{d1}) = \{D_3, \theta_B, task_2, Time_P, ID_{task_2}\} \text{ and}$$

$$DK_{task_2}(A_{d2}) = \{D_4, \theta_C, task_2, Time_P, ID_{task_2}\}$$

The self-evaluation functions $\Phi_{c\,i,task_1}(A_s)$, $\Phi_{c\,i,task_2}(A_{d1})$ and $\Phi_{c\,i,task_2}(A_{d2})$ (*set of neural networks*) calculate the certainty indexes $ci_{c_i,task_1}(A_s) \in [0,1]$, $ci_{c_i,task_2}(A_{d1}) \in [0,1]$ and $ci_{c_i,task_2}(A_{d2}) \in [0,1]$ for each controller in the respective tasks.

The suitability functions $\Theta_{c\,i,task_1}(A_s)$, $\Theta_{c\,i,task_2}(A_{d1})$ and $\Theta_{c\,i,task_2}(A_{d2})$ help physical agents to know the suitability of each controller to execute the tasks according to the evaluation of its respective capabilities. For that, the certainty indexes are jointly used with the control-grounded capabilities and the influence degrees of the involved situations such that:

For the passing a ball:

$$sr_{c_i,task_1}(A_s) = \Theta_{c_i,task_1}(ci_{c_i,task_1}(A_s), CC_{c_i,task_1}(A_s)), \quad i = 1,2,3$$

For the offside maneuver:

$$sr_{c_i,task_2}(A_{d1}) = \Theta_{c_i,task_2}(ci_{c_i,task_2}(A_{d1}), CC_{c_i,task_2}(A_{d1})), \quad i = 1,2,3$$

$$sr_{c_i,task_2}(A_{d2}) = \Theta_{c_i,task_2}(ci_{c_i,task_2}(A_{d2}), CC_{c_i,task_2}(A_{d2})), \quad i = 1,2,3$$

The control-grounded capabilities used in this particular application (see Section 5.1.5) have a direct relation with the studied situations, (e.g. μ_A with *SL*, σ_A with *QB*, γ_A with *MD*, α_A with *AB*, ε_A with *EP*). For instance, if the influence degrees *IDs* of *SL, QB, MD, AB, and EP* are taken into account, a suitability rate can be obtained for each controller such that:

For the passing a ball:

$$ID_{task_1} = [ID_{SL} \;\; ID_{QB} \;\; ID_{MD} \;\; ID_{AB} \;\; ID_{EP}]\%$$

$$sr_{c_i,task_1}(A_s) = \left(ci_{c_i,task_1}(A_s)\right) \cdot \left(ID_{SL} \cdot \mu + ID_{QB} \cdot \sigma + ID_{MD} \cdot \gamma + ID_{AB} \cdot \alpha + ID_{EP} \cdot \varepsilon\right)$$

For the offside maneuver:

$$ID_{task_2} = [ID_{SL} \;\; ID_{QB} \;\; ID_{MD} \;\; ID_{AB} \;\; ID_{EP}]\%$$

$$sr_{c_i,task_2}(A_{d1}) = \left(ci_{c_i,task_2}(A_{d1})\right) \cdot \left(ID_{SL} \cdot \mu + ID_{QB} \cdot \sigma + ID_{MD} \cdot \gamma + ID_{AB} \cdot \alpha + ID_{EP} \cdot \varepsilon\right)$$

$$sr_{c_i,task_2}(A_{d2}) = \left(ci_{c_i,task_2}(A_{d2})\right) \cdot \left(ID_{SL} \cdot \mu + ID_{QB} \cdot \sigma + ID_{MD} \cdot \gamma + ID_{AB} \cdot \alpha + ID_{EP} \cdot \varepsilon\right)$$

A high suitability rate indicates that the tasks can be performed. A low suitability rate indicates that the tasks cannot be performed. The controller with the highest suitability rate is the most suitable to be used in the task execution of all controller with a high certainty index. Appropriate decision thresholds have been empirically selected to implement this decision-making structure.

5.2 The Test bed 2: A Simulated Rescue Environment

A disaster environment is a dynamic environment with unpredictable situations. The set of rescue activities that take place depend on the kind of disaster. Such activities can range from searching and rescuing victims, extinguishing forest fires, re-establishing urban services, cleaning beaches, etc. Rescue resources must be assigned in a way that guarantees the achievement of the required tasks for the optimal recovery of the disaster zone. Technology must be able to make a contribution in this socially significant situation. To that end, several initiatives have been developed in order to promote research in such complex scenarios [Tadokoro et al., 00]. In particular, the designed simulated rescue environment provides a suitable test bed where several technologies can be examined and integrated. Such artificial scenario is restricted to specific rescue conditions making the problem easier to tackle.

In such rescue scenario, several heterogeneous rescue agents must interact with one common purpose: to maximize the number of rescued victims. The key issue in this environment is then that rescue agents must coordinate between them to perform the proposed tasks according to their capabilities.

5.2.1 Rescue Case Study

In the simulated rescue environment there are rescue agents RA = $\{A_1, A_2, A_3, \ldots, A_N\}$ and victims V = $\{V_1, V_2, V_3, \ldots, V_M\}$. Here, tasks allocation is related to allocating victims to save for each rescue agent. So, there are M tasks T = {task$_1$, task$_2$, task$_3$,…, task$_M$} to allocate. A victim has been saved when a rescue agent arrives to its location before its death time. In this sense, a saved victim constitutes a successful task performed by a rescue agent. Rescue agents are represented by non-holonomic mobile robots in this case study. The robots have just one controller for its movements in the environment. Therefore, these physical agents must coordinate their moves to save the greatest number of victims by means of a suitable task allocation. Fig. 5.20 shows an example of this case study.

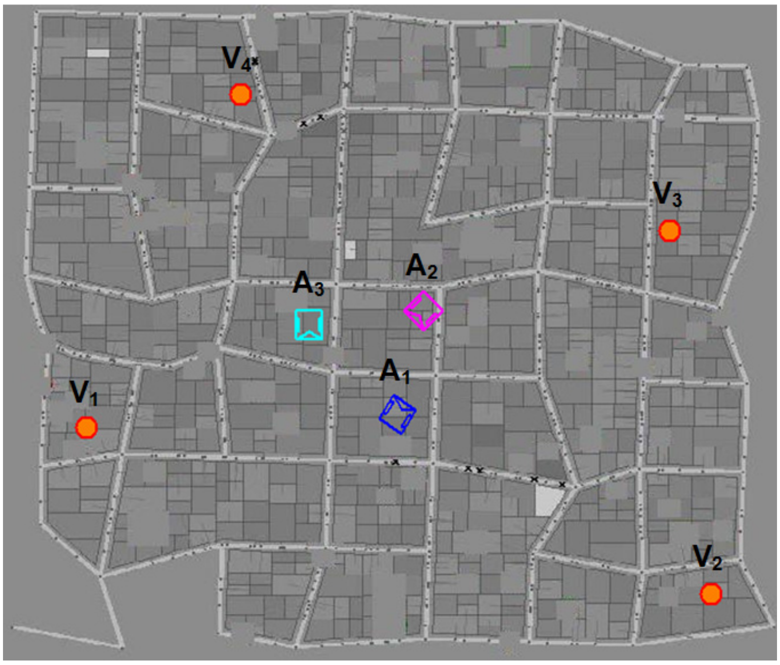


Fig. 5.20. Simulated rescue environment.

For the sake of simplicity, at the beginning of each simulation, the rescue agents are not moving. In addition, the rescue agents' locations, the victims' death time and their locations are arbitrarily selected in each simulation.

The rescue operation is mainly represented as follows: the initial distance $\mathrm{D}_{(\mathrm{A}_i,\mathrm{task}_j)}$ between each rescue agent A_i of RA and each victim task_j of T, the death time of each victim $\mathrm{Time}_{\mathrm{task}_j}$ and a compound orientation $\theta_{(\mathrm{A}_i,\mathrm{task}_j)}$ of each rescue agent with respect to each victim. The orientation of a rescue agent is described in (5.18).

$$\begin{aligned}\theta_{(A_i,task_j)} &= |\alpha_{1}(A_i,task_j)| + |\alpha_{2}(A_i,task_j)| \\ &= |\theta_{L}(A_i,task_j) - \theta_{I}(A_i,task_j)| + |\theta_{L}(A_i,task_j) - \theta_{F}(A_i,task_j)|\end{aligned} \qquad (5.18)$$

Fig. 5.21 shows an example of the variables that describe the rescue agent's state.

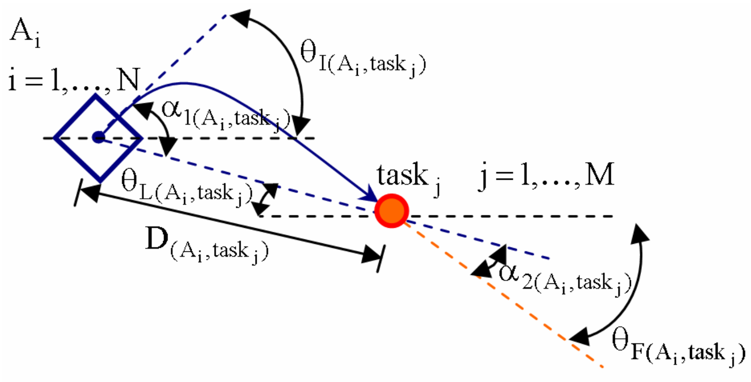


Fig. 5.21. Variables that describe the rescue agent's state.

5.2.2 Implementation in the Rescue Case Study

In this implementation, each rescue agent has a different movement controller to execute the above tasks. Three different physical agents RA = {A_1, A_2, A_3} have been designed with their specific movement controllers such that C(A_1) = {c_1}, C(A_2) = {c_2} and C(A_3) = {c_3}. There are four victims V = {V_1, V_2, V_3, V_4}, i.e., four tasks T = {$task_1$, $task_2$, $task_3$, $task_4$} to test this approach. Fig. 5.22 shows how the physical agents have different dynamics in the execution of the tasks.

It is therefore possible to obtain a set of control-grounded capabilities associated with each physical agent. These capabilities describe the dynamic features of the physical agents' bodies during the execution of the tasks. Fig. 5.23 shows an example of these capabilities.

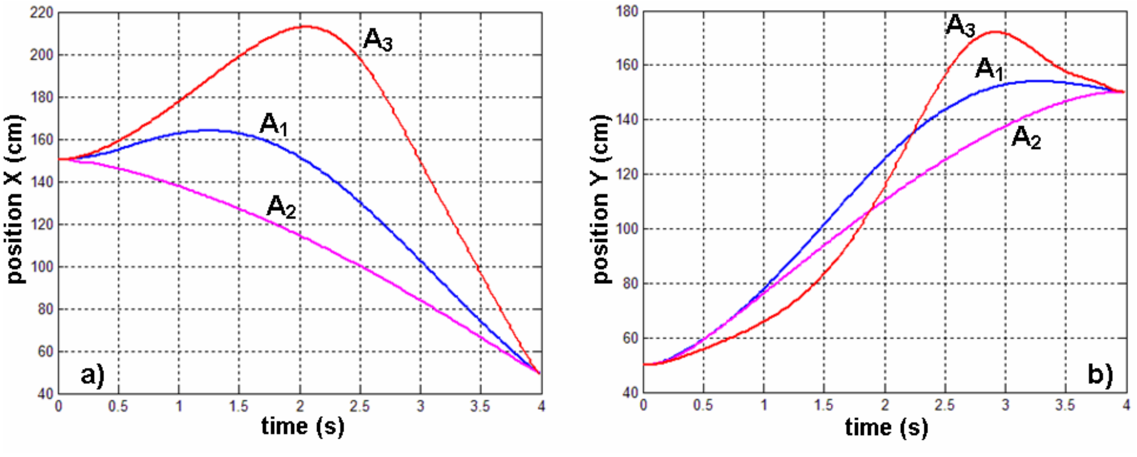
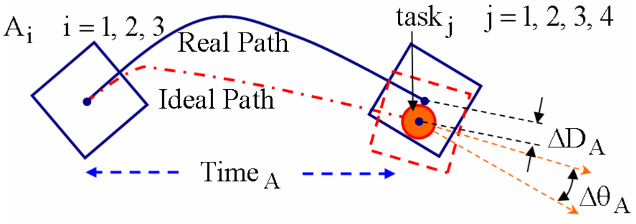


Fig. 5.22. Temporal evolution of the physical agents: a). In X-axis; b). In Y-axis.



$$CE_A = \frac{\int_0^{\text{Time}_{\text{task}_j}} \left|\frac{du(t)}{dt}\right| dt}{u(t)_{max} - u(t)_{min}}$$

Fig. 5.23. Example of the capabilities in the rescue case study.

Where: u(t)→velocity control signal, Time_{task_j}→proposed execution time of the task.

The following set of control-grounded capabilities has been used for the physical agents: the minimal time to perform the task (Time_A), the distance error (ΔD_A) and the orientation error ($\Delta\theta_A$) with respect to the *task$_j$*, and the control effort (CE_A) in the execution of the task such that:

$$CC_{c_i,\text{task}_j}(A_i) = \{\text{Time}_A, \Delta D_A, \Delta\theta_A, CE_A\}, \quad i = 1,2,3 \quad \text{and} \quad j = 1,2,3,4$$

The introspective reasoning (see Section 4.2.3) is implemented using feed-forward back-propagation neural networks. The environmental conditions are related to the physical agents' initial state, represented by $(D_{(A_i,\text{task}_j)}, \theta_{(A_i,\text{task}_j)})$. The task requirements are related to the proposed execution time of the task $(\text{Time}_{\text{task}_j})$ such that:

$$EC(A_i) = \{D_{(A_i,task_j)}, \theta_{(A_i,task_j)}\}, \quad TR = \{Time_{task_j}\} \mid$$

$$DK_{task_j}(A_i) = \{D_{(A_i,task_j)}, \theta_{(A_i,task_j)}, task_j, Time_{task_j}\}$$

The self-inspection function $\Psi_{ci,task_j}(A_i)$ (*set of neural networks 1*) takes into account the agent's domain knowledge $DK_{task_j}(A_i)$ to obtain the agent's capabilities $CC_{c_i,task_j}(A_i)$ according to the available controllers $C_{task_j}(A_i)$ such that:

$$CC_{c_i,task_j}(A_i) = \Psi_{c_i,task_j}(DK_{task_j}(A_i)), \quad i = 1,2,3 \quad \text{and} \quad j = 1,2,3,4$$

The self-evaluation function $\Phi_{ci,task_j}(A_i)$ (*set of neural networks 2*) calculates the certainty index $ci_{c_i,task_j}(A_i) \in [0,1]$ for each controller in the proposed task according to its capabilities {ΔDA, ΔθA} and the possibility of performing the task given by the condition ($\{Time_A\} < Time_{task_j}$) such that:

$$ci_{c_i,task_j}(A_i) = \Phi_{c_i,task_j}(CC_{c_i,task_j}(A_i)), \quad i = 1,2,3 \quad \text{and} \quad j = 1,2,3,4$$

$$ci_{c_i,task_j}(A_i) = \Phi_{c_i,task_j}(\{\Delta D_A, \Delta\theta_A, Time_A < Time_{task_j}\})$$

A fuzzy-based suitability function $\Theta_{c_{i},task_j}(A_i)$ helps physical agent to know its suitability to execute the tasks. To that end, the certainty indexes are jointly used with the capability {CEA} in a fuzzy decision maker (as the Fig. 5.14) such that:

$$sr_{c_i,task_j}(A_i) = \Theta_{c_i,task_j}(ci_{c_i,task_j}(A_i), \{CE_A\}_{c_i,task_j}(A_i)), \quad i = 1,2,3 \quad \text{and} \quad j = 1,2,3,4$$

This decision maker selects the most suitable task to execute by means of a comparative analysis $\zeta_{task_j}(A_i)$ (task1 or task2 or task3 or task4 or none).

The selection is then based on suitability rates $SR_{task_j}(A_i)$. A high or medium $sr_{c_i,task_j}(A_i)$ indicates that the task can be performed, but with different performance levels. A low $sr_{c_i,task_j}(A_i)$ indicates that the task cannot be performed. Appropriate decision thresholds have been empirically selected to implement this decision-making structure.

In addition, each physical agent sorts the tasks that it can perform in an increasing order based on $Time_{task_j}$. Therefore, in the simulated rescue environment, each rescue agent saves the victim with the lowest death time of all victims that it can rescue. However, if there is more than one rescue agent able to save the same victim, the selected rescue agent is the one with the highest $sr_{c_i,task_j}(A_i) \in [0,1]$ while the others go to save the next victim in their rescue scheduling. If there are rescue agents with no more victims in their scheduling, then these agents go to save the same victim as the agent with the highest $sr_{c_i,task_j}(A_i)$.

The task allocation performance improves with this new and effective approach because the physical agents can achieve a most suitable task allocation according to their capabilities. Thus, physical agents have introspection on what they can and cannot do and how they are able to perform the tasks according to their physical limitations, mainly related to their dynamics. Introspection on control-grounded capabilities is then an appropriate alternative to take into account in the current utility/cost functions for task/action selection.

5.2.2.1 Task Allocation Example in the Rescue Case Study

The task allocation case study is exemplified by comparing our proposal with other approaches. For comparative purposes, different task allocation approaches have been defined as follows:

Random: Rescue agents decide whether or not to save a victim in a random manner. That is to say, physical agents decide randomly which task to perform or not perform.

Proximity: Rescue agents decide to save their nearest victim. That is to say, physical agents decide to perform the nearest task according to their initial positions in the environment.

Deadline: Rescue agents decide to save the victims according to their death time. The victims are allocated to A_2, A_1 and A_3 respectively in an increasing death time order, where A_2 is the quickest rescue agent and A_3 is the slowest rescue agent. That is to say, physical agents decide which task to perform, taking into account deadline constraints and some minimal knowledge of their bodies.

Introspection: Rescue agents decide whether or not to save a victim based on introspection on their control-grounded capabilities in line with the considerations outlined in Section 5.2.2.

Let us then consider the application of the above approaches to the rescue case study described in Section 5.2.1. The following tables show real data related to the agents' domain knowledge and the task requirements for the four proposed tasks. The tables also show and highlight the allocated task for each agent and if the task was performed correctly by such agents (✓: *the allocated task was successfully performed by the selected agent or the agent made a right decision*, ✗: *the allocated task was not successfully performed or the agent made a wrong decision*). Specifically, Tables 5.17 and 5.18 show the data for the task allocation case study by using the *random* approach.

Table 5.17. Trial 1 for the random approach.

tasks	A_1		A_2		A_3		$\text{Time}_{\text{task}_j}$
	$D_{(A_1,\text{task}_j)}$	$\theta_{(A_1,\text{task}_j)}$	$D_{(A_2,\text{task}_j)}$	$\theta_{(A_2,\text{task}_j)}$	$D_{(A_3,\text{task}_j)}$	$\theta_{(A_3,\text{task}_j)}$	
task$_1$ (V$_1$)	93.54	103.10	122.58	303.15	87.57	135	2.619
task$_2$ (V$_2$)	83.66	87.28	83.45	180.33	115.11	250.19	0.714
task$_3$ (V$_3$)	111.44	225	77.10	445	103.94	55.27	1.937
task$_4$ (V$_4$)	120.10	149.83	115.26	355	85.42	45	2.901
Allocated task	task$_3$		task$_4$		none		
task OK?	✓		✓		✗		

Table 5.18. Trial 2 for the random approach.

tasks	A_1		A_2		A_3		$Time_{task_j}$
	$D_{(A_1,task_j)}$	$\theta_{(A_1,task_j)}$	$D_{(A_2,task_j)}$	$\theta_{(A_2,task_j)}$	$D_{(A_3,task_j)}$	$\theta_{(A_3,task_j)}$	
$task_1$ (V_1)	100.55	20.79	116.10	277.11	73.30	226.54	0.411
$task_2$ (V_2)	119.10	80	83.47	207.91	95.11	337.48	2.456
$task_3$ (V_3)	108.99	234.33	82.46	448.03	125.54	45	1.290
$task_4$ (V_4)	62.95	116.40	101.69	355	93.86	135	2.671
Allocated task	$task_3$		$task_3$		$task_1$		
task OK?	✓		✓		✗		

In these cases, the tasks are randomly allocated to each agent without any consideration. Such random decision does then possible that agents decide sometimes not to execute any task. In addition, there is not any guarantee that agents can correctly perform the allocated tasks.

On the other hand, Tables 5.19 and 5.20 show the data for the task allocation case study by using the *proximity* approach. In the tables is highlighted and allocated the nearest task to each agent. In this sense, agents use just domain knowledge to make decisions, i.e., information about the agents' locations. Again by using this approach, the agents can or cannot correctly perform the allocated tasks. It means that it is not enough the nearness of the agents to the tasks for guaranteeing the correct execution of them.

Table 5.19. Trial 1 for the proximity approach.

tasks	A_1		A_2		A_3		$Time_{task_j}$
	$D_{(A_1,task_j)}$	$\theta_{(A_1,task_j)}$	$D_{(A_2,task_j)}$	$\theta_{(A_2,task_j)}$	$D_{(A_3,task_j)}$	$\theta_{(A_3,task_j)}$	
$task_1$ (V_1)	67.49	10	104.89	265	108.29	225	2.150
$task_2$ (V_2)	95.58	124.44	112.03	175	64.48	315	1.534
$task_3$ (V_3)	123.69	196.31	94.18	482.59	86.68	67.95	2.329
$task_4$ (V_4)	94.02	120.39	71.22	355	118.62	156.26	1.468
Allocated task	$task_1$		$task_4$		$task_2$		
task OK?	✓		✓		✓		

Table 5.20. Trial 2 for the proximity approach.

tasks	A$_1$		A$_2$		A$_3$		$\text{Time}_{\text{task}_j}$
	$D_{(A_1,\text{task}_j)}$	$\theta_{(A_1,\text{task}_j)}$	$D_{(A_2,\text{task}_j)}$	$\theta_{(A_2,\text{task}_j)}$	$D_{(A_3,\text{task}_j)}$	$\theta_{(A_3,\text{task}_j)}$	
task$_1$ (V$_1$)	105.05	40.90	117.30	265	116.86	225	2.409
task$_2$ (V$_2$)	58.73	84.60	91.59	192.12	73.43	315	0.252
task$_3$ (V$_3$)	92.58	190	74.22	458.76	75.52	59.14	2.836
task$_4$ (V$_4$)	123.53	100	96.33	355	113.22	167.70	2.748
Allocated task	task$_2$		task$_3$		task$_2$		
task OK?	✗		✓		✗		

In addition, Tables 5.21 and 5.22 show the data for the task allocation case study by using the *deadline* approach. The tasks are allocated according to their death time requirements from the quickest agent to the slowest agent respectively how it is highlighted in the tables.

Table 5.21. Trial 1 for the deadline approach.

tasks	A$_1$		A$_2$		A$_3$		$\text{Time}_{\text{task}_j}$
	$D_{(A_1,\text{task}_j)}$	$\theta_{(A_1,\text{task}_j)}$	$D_{(A_2,\text{task}_j)}$	$\theta_{(A_2,\text{task}_j)}$	$D_{(A_3,\text{task}_j)}$	$\theta_{(A_3,\text{task}_j)}$	
task$_1$ (V$_1$)	122.25	225	80.45	45	115.60	125.95	1.518
task$_2$ (V$_2$)	111.04	315	101.53	146.62	81.42	28.98	1.394
task$_3$ (V$_3$)	79.12	45	117.37	164.79	83.16	281.05	1.624
task$_4$ (V$_4$)	103.07	173.52	102.87	54.66	126.60	180	2.827
Allocated task	task$_1$		task$_2$		task$_3$		
task OK?	✓		✓		✓		

Table 5.22. Trial 2 for the deadline approach.

tasks	A$_1$		A$_2$		A$_3$		$\text{Time}_{\text{task}_j}$
	$D_{(A_1,\text{task}_j)}$	$\theta_{(A_1,\text{task}_j)}$	$D_{(A_2,\text{task}_j)}$	$\theta_{(A_2,\text{task}_j)}$	$D_{(A_3,\text{task}_j)}$	$\theta_{(A_3,\text{task}_j)}$	
task$_1$ (V$_1$)	81.92	65.97	106.04	45	134.88	90	0.628
task$_2$ (V$_2$)	57.28	133.41	86.20	135	116.40	35.09	1.365
task$_3$ (V$_3$)	119.17	190	84.66	153.10	68.27	338.57	0.243
task$_4$ (V$_4$)	118.29	116.58	84.04	45	72.07	180	2.553
Allocated task	task$_1$		task$_3$		task$_2$		
task OK?	✗		✗		✓		

However, there are cases where agents can not correctly perform the task. Again there is not enough information for a suitable task allocation.

Finally, Tables 5.23, 525, and 5.27 show the data for the task allocation case study by using the *introspection* approach. In addition, Tables 5.24, 5.26 and 5.28 show the certainty indexes and the suitability rates for each agent in each task, highlighting those that overcome the established decision threshold.

Table 5.23. Trial 1 for the introspection approach.

tasks	A_1		A_2		A_3		$\text{Time}_{\text{task}_j}$
	$D_{(A_1,\text{task}_j)}$	$\theta_{(A_1,\text{task}_j)}$	$D_{(A_2,\text{task}_j)}$	$\theta_{(A_2,\text{task}_j)}$	$D_{(A_3,\text{task}_j)}$	$\theta_{(A_3,\text{task}_j)}$	
task_1 (V_1)	126.51	65.31	90.13	45	107.49	113.58	1.495
task_2 (V_2)	80.25	80	77.89	137.36	112.12	31.11	0.871
task_3 (V_3)	62.26	190	100.66	137.77	76.27	320.57	2.018
task_4 (V_4)	114.24	100	107.10	63.77	76.85	180	2.874
Allocated task	{task_1, task_3, task_4}		{task_2, task_1, task_3, task_4}		A_3 cannot perform any task		
task OK?	✓		✓		✓		

Table 5.24. Introspection of the agents on the first trial.

	$ci_{c_i,\text{task}_1}(A_i)$	$ci_{c_i,\text{task}_2}(A_i)$	$ci_{c_i,\text{task}_3}(A_i)$	$ci_{c_i,\text{task}_4}(A_i)$	$sr_{c_i,\text{task}_1}(A_i)$	$sr_{c_i,\text{task}_2}(A_i)$	$sr_{c_i,\text{task}_3}(A_i)$	$sr_{c_i,\text{task}_4}(A_i)$
A_1	0.9735	0.6548	0.8673	0.9734	0.8554	0.1473	0.8512	0.8597
A_2	0.9900	0.9900	0.9900	0.9900	0.5000	0.5000	0.5000	0.5000
A_3	0.5390	0.0000	0.0001	0.0001	0.1403	0.1408	0.1393	0.1431

Table 5.25. Trial 2 for the introspection approach.

tasks	A_1		A_2		A_3		$\text{Time}_{\text{task}_j}$
	$D_{(A_1,\text{task}_j)}$	$\theta_{(A_1,\text{task}_j)}$	$D_{(A_2,\text{task}_j)}$	$\theta_{(A_2,\text{task}_j)}$	$D_{(A_3,\text{task}_j)}$	$\theta_{(A_3,\text{task}_j)}$	
task_1 (V_1)	77.60	180	92.68	180	107.21	180	2.552
task_2 (V_2)	108.70	270.56	97.83	270	89.34	270	1.021
task_3 (V_3)	104.97	36.95	90.16	27.86	77.01	16.33	1.398
task_4 (V_4)	76.61	90	86.50	90	97.41	98.17	2.741
Allocated task	{task_1, task_4}		{task_2, task_3, task_1, task_4}		{task_3}		
task OK?	✓		✓		✓		

Table 5.26. Introspection of the agents on the second trial.

	$ci_{c_i,task_1}(A_i)$	$ci_{c_i,task_2}(A_i)$	$ci_{c_i,task_3}(A_i)$	$ci_{c_i,task_4}(A_i)$	$sr_{c_i,task_1}(A_i)$	$sr_{c_i,task_2}(A_i)$	$sr_{c_i,task_3}(A_i)$	$sr_{c_i,task_4}(A_i)$
A_1	0.8671	0.1287	0.9739	0.8670	0.8512	0.1619	0.8556	0.8511
A_2	0.9900	0.9900	0.9900	0.9900	0.5000	0.5000	0.5000	0.5000
A_3	0.3792	0.0020	0.9789	0.6538	0.1406	0.1441	0.8577	0.1514

Table 5.27. Trial 3 for the introspection approach.

tasks	A_1		A_2		A_3		$Time_{task_j}$
	$D_{(A_1,task_j)}$	$\theta_{(A_1,task_j)}$	$D_{(A_2,task_j)}$	$\theta_{(A_2,task_j)}$	$D_{(A_3,task_j)}$	$\theta_{(A_3,task_j)}$	
$task_1$ (V_1)	96.57	135	71.08	45	102.13	76.74	2.627
$task_2$ (V_2)	78.26	225	96.82	150.12	104.63	45	2.212
$task_3$ (V_3)	92.84	48.18	113.64	155.82	80.48	263.61	0.410
$task_4$ (V_4)	105.97	45	92.12	81.17	81.81	135	0.035
Allocated task	{$task_2$, $task_1$}		{$task_1$}		A_3 cannot perform any task		
task OK?	✓		✓		✓		

Table 5.28. Introspection of the agents on the third trial.

	$ci_{c_i,task_1}(A_i)$	$ci_{c_i,task_2}(A_i)$	$ci_{c_i,task_3}(A_i)$	$ci_{c_i,task_4}(A_i)$	$sr_{c_i,task_1}(A_i)$	$sr_{c_i,task_2}(A_i)$	$sr_{c_i,task_3}(A_i)$	$sr_{c_i,task_4}(A_i)$
A_1	0.8671	0.8672	0.0147	0.0147	0.8512	0.8512	0.1545	0.1669
A_2	0.9900	0.9900	0.0003	0.0003	0.5000	0.5000	0.1507	0.1456
A_3	0.6549	0.7285	0.0001	0.0012	0.1516	0.4378	0.1519	0.1492

The tables show that each physical agent sorts the tasks that it can perform in an increasing order based on $Time_{task_j}$. Therefore, each agent performs the task with the lowest $Time_{task_j}$ of all tasks that it can perform. However, if there is more than one agent able to perform the same task, the selected agent is the one with the highest $sr_{c_i,task_j}(A_i)$ while the others go to perform the next tasks, deleting the allocated task for the selected agent in their scheduling. If there are agents with no more tasks in their scheduling, then these agents go to perform the same task as the agent with the highest $sr_{c_i,task_j}(A_i)$. Agents with a low $sr_{c_i,task_j}(A_i)$ know that they cannot perform the respective task. So introspection guarantees the agents sure decisions, i.e., independently if agents can perform or not the proposed tasks, the agents be aware of their decisions. Agents can discriminate between the tasks in which they have a chance of correct performing and those in which they have no chance.

5.3 The Test bed 3: A Simulated Convoys Environment

Automobile has become the dominant transport mode in the world in the last century. In order to meet a continuously growing demand for transport, one solution is to change the control approach for vehicle to full driving automation, which removes the driver from the control loop to improve efficiency and reduce accidents [Parent and de la Fortelle, 05]. Recent work shows that there are several realistic paths towards this deployment. Cars of the future are vehicles with fully automated driving capabilities in order to have a real door-to-door service (cars can be called at any location and can be left anywhere because of their autonomous driving capability) [Baber et al., 05]. Such cars will crash if they do not consider their dynamics in both braking and accelerating movements when vehicle spacing is tight. In this sense, such autonomous vehicles must know, inspect and communicate their physical features to avoid collisions and traffic flow congestion. In particular, the simulated convoys environment provides a suitable test bed where the above problem can be examined. For illustrative purposes, such artificial scenario is restricted to specific conditions making the problem easier to tackle.

In the convoys scenario, several heterogeneous agent-controlled vehicles must interact by pairs with one common purpose: to minimize the number of collisions between them. Each vehicle is then a closed loop dynamical system governed by an agent who makes decisions. The key issue in this environment is that the agent-controlled vehicles must coordinate between them to achieve an autonomous driving free of collisions taking into account their dynamics.

5.3.1 Convoys Case Study

In the simulated convoys environment there are agent-controlled vehicles AV = {A_1, A_2, A_3,…, A_N} that form N/2 convoys. A problem arises when the agents in a convoy try to keep a constant security distance *D* between them. The distance between the vehicles continually changes because each vehicle has an own different dynamics, even though their static behaviour is equally achievable. Therefore, the possibility of collisions increases with velocity changes of the guiding vehicles. The guiding vehicles decide commonly to accelerate or decelerate without considering the dynamics of the other vehicles. In this case, the rear vehicles have the responsibility of keeping the distance constant and avoiding collisions in their respective convoy. For that, the traditional solution is to keep some minimum static distance and design the best possible control system [Horowitz and Varaiya, 00]. This alternative facilitates steady-state control solutions [Belkhouche and Belkhouche, 05]. However, a decision to change the velocity of the convoys is not correct under some conditions (e.g., if the security distances are changed), even though the control of the vehicles has been well designed. As a result, the vehicles sometimes cannot avoid colliding.

It is also possible that the designed distance control for a convoy will not work when a new vehicle with different dynamics is used in this convoy, such that the vehicles cannot avoid colliding. The control system must therefore be redesigned to satisfy the new dynamics of the vehicles. The above alternative is not a good solution. Such redesign induces generally more complexity than does not necessarily impact in a better performance.

In particular, these physical agents must then coordinate their moves by pairs to reduce the number of possible collisions in their deceleration maneuvers from an initial velocity V_H to a final velocity V_L, $(V_H > V_L)$ at a given time $Time_d$ by means of a suitable coordination based on their dynamics. Fig 5.24 shows a general scheme of the test bed.

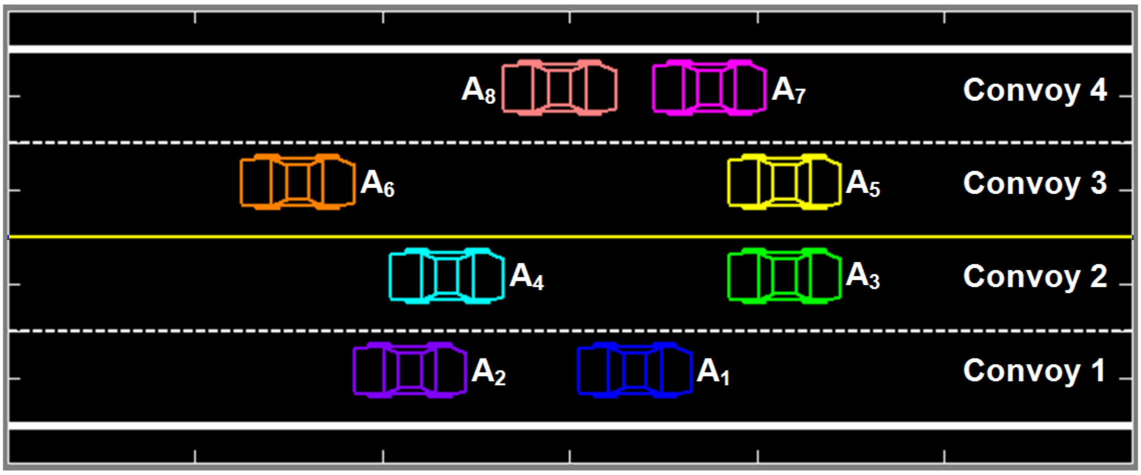


Fig. 5.24. Simulated convoys environment.

In this sense, autonomous driving free of collisions constitutes a successful task performed by the agents. Agent-controlled vehicles develop a straight-line and one-dimensional movements in this case study. The vehicles have just one distance controller in their movements within the environment. For the sake of simplicity, at the beginning of each simulation, the vehicles are not moving. In addition, the agents' initial locations, the initial and final velocities, the deceleration time and the security distance between the vehicles are selected arbitrarily in each simulation.

In this case study, there are then two possible behaviours for the agents: reactive (without introspection) and deliberative (with introspection). In the first, the guiding agents decide to decelerate without taking into account the dynamics of their respective rear vehicles. However, the rear agents are responsible for not colliding. In the second, the agents can negotiate and make coordinated decisions by exchanging information about these dynamics. In this case, the guiding vehicles modify their actions according to the information provided by the other vehicles about their own dynamics and vice versa. Thus, agents' decisions will be inhibited whenever the vehicles' dynamics do not allow the execution of the proposed actions and will be renegotiated until the agents agree. Thus, the reliability of sure decisions based on dynamics will be of both of them. In particular, the guiding agents communicate to the rear agents their decisions of decelerating at a given time Time_d. The rear agents introspect about their behaviour and answer to their respective guiding agents the certainty indexes $ci_{c_j,task_k}(A_j) \in [0,1]$ associated to these actions. These certainty indexes let guiding agents to decide about the actions. If there is no agreement, then the guiding agents propose to perform more reliable actions in a different execution time $Time'_d$ and a different deceleration velocity V'_L. These steps can be repeated until to obtain an agreement (see Section 4.2.4: the decision algorithm for coordinated tasks).

5.3.2 Implementation in the Convoys Case Study

In this implementation, transfer functions are used for analysing dynamics of each vehicle. Only very ideal systems will be analysed in this case study, then first order transfer functions are the proper way to represent dynamics in linear speed for one-dimensional movements. Other higher order transfer functions, non-linearities and other variables (like angular orientation, etc.) will be analysed in future work. Eight different physical agents AV = {A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8} that form four convoys with their specific distance controllers have been designed, i.e., four tasks T = {$task_1$, $task_2$, $task_3$, $task_4$}.

The real linear velocity of each vehicle is then modelled as is described in (5.19)

$$v_{real(A_j)} = G_j(s) v_{linear(A_j)}, \quad j = 1,\ldots,8 \quad (5.19)$$

Where $v_{linear(A_j)}$ is the speed step input and $G_j(s)$ is the corresponding first order transfer function as is described in (5.20).

$$G_j(s) = \frac{K_j}{\tau_j s + 1}, \quad j = 1,\ldots,8 \quad (5.20)$$

Where K_j is a static gain and τ_j is the time constant for each vehicle.

More specifically, the guiding agents have their internal controls that are apparently independent of rear agents. The rear agents have inherently more complicated structures since they have the responsibility for convoying and avoiding possible collisions.

Let us assume that a guiding agent's dynamics shows a first order transfer function $G_1(s)$, and its corresponding rear agent has a similar behaviour $G_2(s)$, then the convoying behaviour of the rear agent is indeed a high order transfer function. This is due to the dependency of the speed set points of the rear agent on the controller designed to keep the security distance *D* between vehicles, i.e., $D-(x_1-x_2)\approx 0$. Fig 5.25 shows a general scheme to describe the agents' state in a convoy.

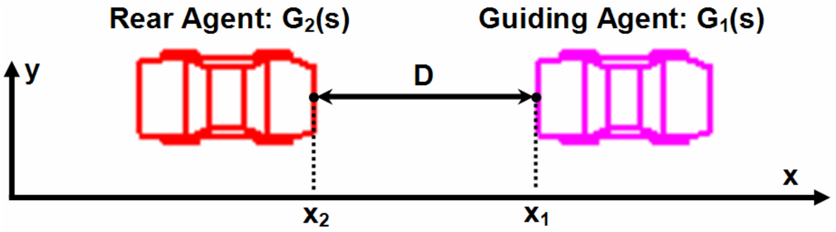


Fig. 5.25. Variables that describe the agents' state in a convoy.

The distance controller is accomplished by the use of a PI controller (*Kp*, *Ki*) whose output signal controls the velocity $v_{real(A_2)}$ of the rear vehicle. Under the above assumptions, if $G_1(s)$ and $G_2(s)$ are first order systems, then the convoying behaviour of the rear agent is indeed a high order system response as it is here deduced:

$$v_{real(A_2)} = G_2(s)v_{linear(A_2)}$$

$$x_2 = \left(\frac{1}{s}\right)v_{real(A_2)} + x_{i_2}$$

$$x_1 = \left(\frac{1}{s}\right)v_{real(A_1)} + x_{i_1}$$

Where x_1 and x_2 are the current position of each vehicle and x_{i_1} and x_{i_2} are the initial positions (t=0) of the same ones.

$$v_{linear(A_2)} = -\left(K_p + \frac{K_i}{s}\right)[D-(x_1-x_2)]$$

$$v_{real(A_2)} = -\left(K_p + \frac{K_i}{s}\right)G_2(s)\left[D - (x_1 - x_2)\right]$$

$$v_{real(A_2)} = -\left(K_p + \frac{K_i}{s}\right)G_2(s)\left[D - \left(\left(\frac{1}{s}\right)v_{real(A_1)} + x_{i_1} - \left(\frac{1}{s}\right)v_{real(A_2)} - x_{i_2}\right)\right]$$

$$v_{real(A_2)} = -\left(K_p + \frac{K_i}{s}\right)G_2(s)\left[D - \left(\frac{1}{s}\right)v_{real(A_1)} - x_{i_1} + \left(\frac{1}{s}\right)v_{real(A_2)} + x_{i_2}\right]$$

$$v_{real(A_2)}\left(1 + \left(K_p + \frac{K_i}{s}\right)\frac{G_2(s)}{s}\right) = \left(K_p + \frac{K_i}{s}\right)G_2(s)\left[\left(\frac{1}{s}\right)v_{real(A_1)} + x_{i_1} - x_{i_2} - D\right]$$

$$v_{real(A_2)} = \frac{\left(K_p + \frac{K_i}{s}\right)G_2(s)}{\left(1 + \left(K_p + \frac{K_i}{s}\right)\frac{G_2(s)}{s}\right)}\left[\left(\frac{1}{s}\right)v_{real(A_1)} + x_{i_1} - x_{i_2} - D\right]$$

$$v_{real(A_2)} = \frac{\left(K_p + \frac{K_i}{s}\right)G_2(s)}{\left(s + \left(K_p + \frac{K_i}{s}\right)G_2(s)\right)}v_{real(A_1)} + \frac{s\left(K_p + \frac{K_i}{s}\right)G_2(s)}{\left(s + \left(K_p + \frac{K_i}{s}\right)G_2(s)\right)}D'$$

Where $D' = x_{i_1} - x_{i_2} - D$ and

$$\frac{\left(K_p + \frac{K_i}{s}\right)G_2(s)}{\left(s + \left(K_p + \frac{K_i}{s}\right)G_2(s)\right)} = \frac{\left(K_p + \frac{K_i}{s}\right)\left(\frac{K_2}{\tau_2 s + 1}\right)}{\left(s + \left(K_p + \frac{K_i}{s}\right)\left(\frac{K_2}{\tau_2 s + 1}\right)\right)} = \frac{\left(\frac{K_p s + K_i}{s}\right)\left(\frac{K_2}{\tau_2 s + 1}\right)}{\left(s + \left(\frac{K_p s + K_i}{s}\right)\left(\frac{K_2}{\tau_2 s + 1}\right)\right)}$$

$$= \frac{K_p K_2 s + K_i K_2}{s^2(\tau_2 s + 1) + K_p K_2 s + K_i K_2} = \frac{K_p K_2 s + K_i K_2}{\tau_2 s^3 + s^2 + K_p K_2 s + K_i K_2}$$

Similar considerations can be established for the other convoys. Thus, the deliberative cooperative decisions based on dynamics must provide the controllers with safer set points. The fact is that not only feedback control is necessary for keeping the distance, but also the cooperative aspects of *AI* must be integrated as it will be shown later.

Here, the environmental conditions are related to the physical agents' initial state, represented by (x_{i_1}, x_{i_2}) and the initial and final velocities V_H and V_L in the deceleration maneuver. The task requirements are related to the proposed deceleration time Time_d and the desired security distance *D* such that:

$$\mathrm{EC}(A_1)=\{x_{i_1},V_H,V_L\},\quad \mathrm{EC}(A_2)=\{x_{i_2},V_H,V_L\},\quad \mathrm{TR}=\{\mathrm{Time}_d,D\}\mid$$

$$\mathrm{DK}_{\mathrm{task}_k}(A_j)=\{x_{i_j},V_H,V_L,\mathrm{task}_k,\mathrm{Time}_d,D\}$$

The self-evaluation functions $\Phi_{c_{j},\mathrm{task}_k}(A_j)$ calculate the certainty indexes $ci_{c_j,\mathrm{task}_k}(A_j)\in[0,1]$, for each rear agent in each convoy.

The decision is then based on the certainty. The guiding agent determines whether any action has to be done or not. The innovation is that the decision is also based on the dynamics of rear agent. For the rear agent, an action is possible if it can move from an initial state V_H to a final state V_L within the specified time for action Time_d that the guiding agent proposes. Once both agents agree, that is, the certainty about the decision is high enough, they trigger their respective decisions.

5.3.2.1 Decisions Example in the Convoys Case Study

The approach is illustrated for a convoy of two different agent-controlled vehicles A_1 and A_2, where each one has a different dynamics such that:

$$G_1(s)=\frac{1}{0.185s+1}\ \text{and}\ G_2(s)=\frac{1}{0.810s+1}$$

The designed distance controller for this convoy (task_1) has the following parameters $K_p = 3.5$ and $K_i = 0.5$. In this example, A_1 decides to decelerate from $V_H = 70$cm/s to $V_L = 30$cm/s at the $\mathrm{Time}_d = 10$s. The agents' initial positions are given by $x_{i_1} = 67.5$cm and $x_{i_2} = 32.5$cm. The security distance between agents must be D = 20cm such that:

$$\mathrm{EC}(A_1)=\{67.5\mathrm{cm},70\mathrm{cm/s},30\mathrm{cm/s}\},\quad \mathrm{EC}(A_2)=\{32.5\mathrm{cm},70\mathrm{cm/s},30\mathrm{cm/s}\},\quad \mathrm{TR}=\{10\mathrm{s},20\mathrm{cm}\}$$

$$DK_{task_1}(A_1) = \{67.5cm, 70cm/s, 30cm/s, task_1, 10s, 20cm\} \text{ and}$$

$$DK_{task_1}(A_2) = \{32.5cm, 70cm/s, 30cm/s, task_1, 10s, 20cm\}$$

Without introspection, A1 decides to decelerate without taking into account the dynamics of A2. However, A2 is responsible for not colliding. Fig 5.26 shows the velocity responses of the agents and the evolution of the distance between them for this example. The distance between vehicles continually changes. Fortunately, there is not collision in this case.

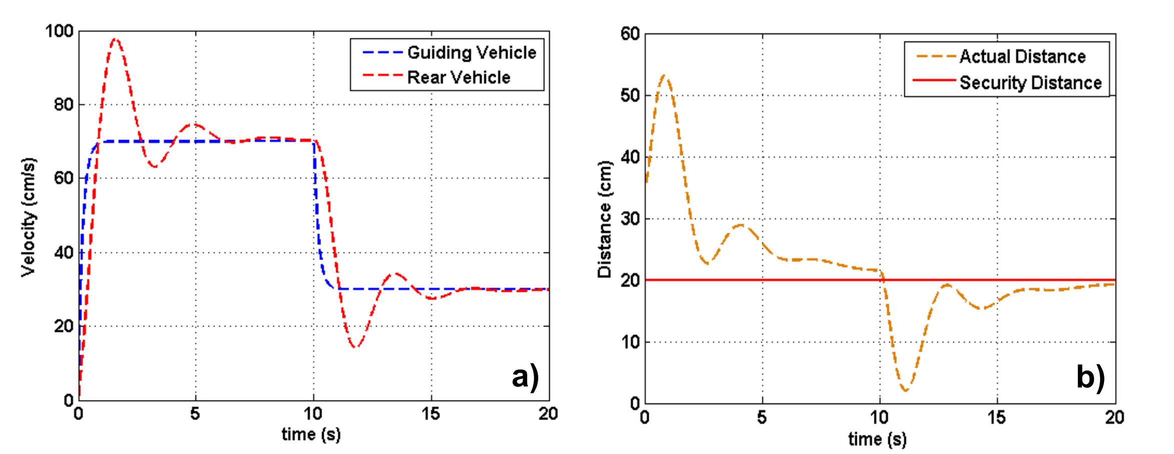


Fig. 5.26. a). Velocity responses of the guiding vehicle A1 and the rear vehicle A2 for a security distance D=20cm; b). Evolution of the distance between A1 and A2 without introspection.

However, a decision to change the velocity of the convoy is not correct under some conditions (e.g., if the security distances is changed to D=10cm), with the same control of the vehicles. As a result, the vehicles sometimes cannot avoid colliding as it is shown in Fig. 5.27.

It is also possible that the designed distance control for a convoy will not work when a new rear vehicle A3 with different dynamics $G_3(s) = \frac{1}{1.10s+1}$ is used in this convoy, such that the vehicles cannot avoid colliding as it is shown in Fig. 5.28.

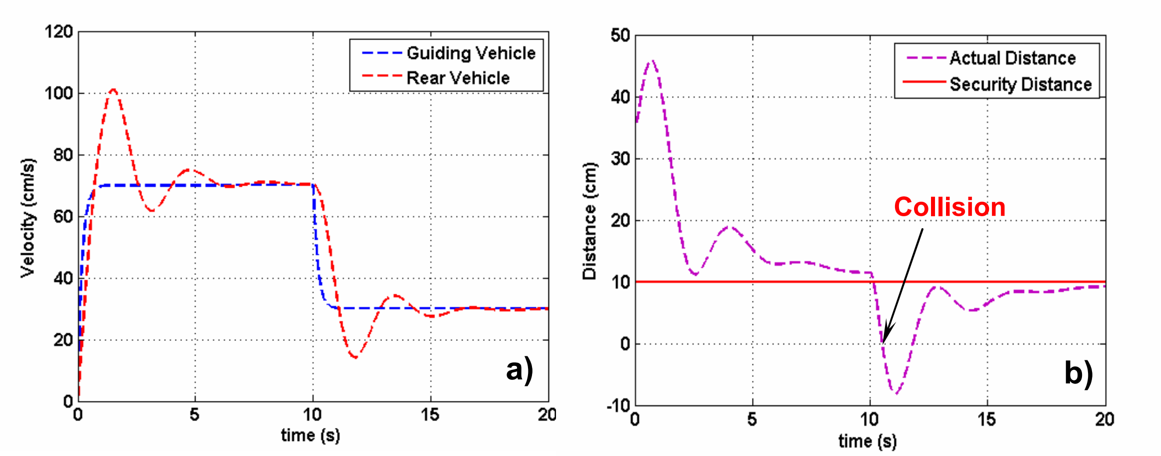


Fig. 5.27. a). Velocity responses of the guiding vehicle A_1 and the rear vehicle A_2 for a security distance D=10cm; b). Evolution of the distance between A_1 and A_2 without introspection.

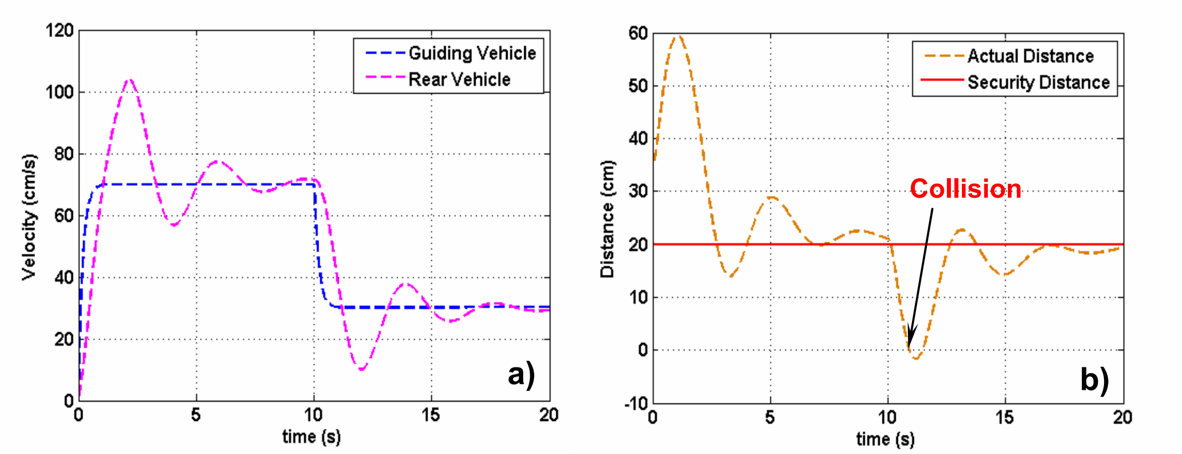


Fig. 5.28. a). Velocity responses of the guiding vehicle A_1 and the rear vehicle A_3 for a security distance D=20cm; b). Evolution of the distance between A_1 and A_3 without introspection.

On the other hand, the decision algorithm executed by the agents by using introspection in this task allows avoiding the undesirable situations shown in Fig. 5.27 and 5.28 as it is shown in Fig 5.29 and 5.30 respectively. Here, the new deceleration time and velocity are given by $\mathrm{Time}'_d = \mathrm{Time}_d + 5\tau_j$ and $V'_L = \frac{V_H + V_L}{2}$, where τ_j corresponds to the time constant of the first order transfer functions of the rear vehicles A_2 and A_3.

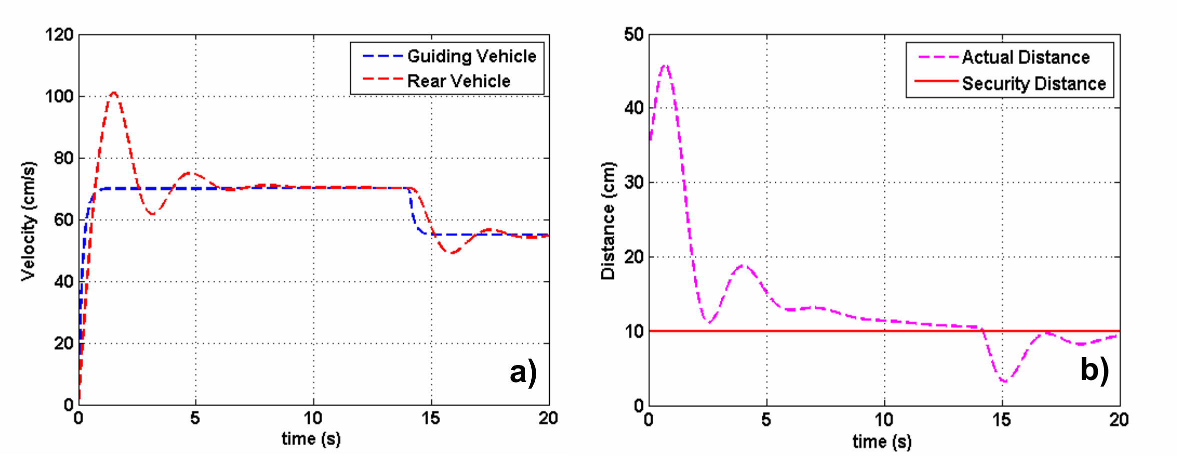


Fig. 5.29. a). Velocity responses of the guiding vehicle A1 and the rear vehicle A2 for a security distance D=10cm; b). Evolution of the distance between A1 and A2 with introspection.

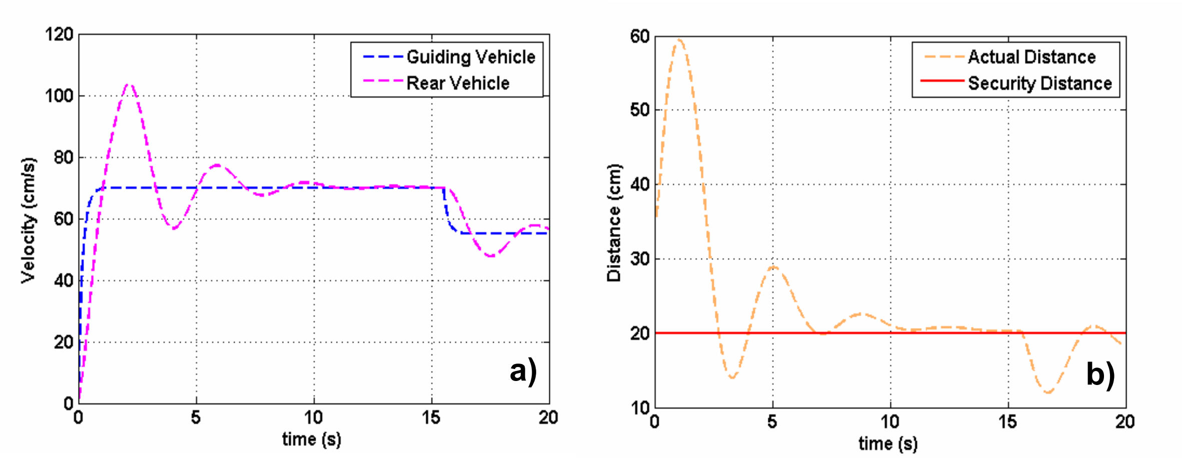


Fig. 5.30. a). Velocity responses of the guiding vehicle A1 and the rear vehicle A3 for a security distance D=20cm; b). Evolution of the distance between A1 and A3 with introspection.

Due to the high order nature of the convoying system, it can be clearly asserted that static information for deceleration decision is not enough at all. Then, decisions based only on static are dangerous in case of dealing with physical agents with dynamics.

PART III
RESULTS AND CONCLUSIONS

PART III
RESULTS AND CONCLUSIONS

Chapter 6

Experimental Results

This chapter presents the empirical experiments and testing that have been carried out for the proposed test beds. The results depicted in this chapter demonstrate the utility, feasibility and reliability of the overall proposed approach presented in the previous chapters.

6.1 Results in Robot Soccer

Insofar as the main goal of any test bed is to facilitate the trial and evaluation of ideas that have promise in the real world, robot soccer proved to be an excellent test bed for this thesis. Robot soccer has drawn a lot of attention over the past years as a platform to conduct research in the field of multi-agent systems. One of the main features of this test bed is the need for cooperation with other agents, in a changing and adversarial environment. These features make it extremely attractive for researchers interested in the deployment of multi-agent systems in scenarios where coordination is relevant and necessary. Research on *MAS* in the soccer domain can be conducted either on real robots or on simulators. On the one hand, real robots allow working on *MAS* on real systems, coping with the limitations of real systems and problems associated to them. On the other hand, the type of algorithms and strategies that can be tested must be simple, as malfunctions, communication problems, physical limitations, etc. affect real robots. Simulators overcome the limitations of real robots allowing researchers to develop and test more complex algorithms and strategies.

All of the thesis contributions were originally developed in a simulated robot soccer environment. However, in order to test the generality of the simulation results, the proposed techniques have been also transferred to the real robot system.

6.1.1 Results in the Passing a Ball

Table 6.1 presents the proposed empirical experiments using the *passing a ball* for testing the system performance when the introspection approach is used.

The tests in simulation used the robot models of the *SimuroSOT* simulator available from http://www.fira.net/soccer/simurosot/overview.html. The simulator facilitates extensive training and testing of this proposal. The selected simulation experiments for each case consist of a predefined number of trials that satisfy the restrictions established in Table 6.1.

Table 6.1. Proposed empirical experiments with passing a ball.

(Pos→Position, Vel→Velocity, Traj→Trajectory, f→fixed, v→variable).

Case	Ball						IP		Passer				Shooter			
	Pos		Vel		Traj		Pos		Pos		Vel		Pos		Vel	
	f	v	f	v	f	v	f	v	f	v	f	v	f	v	f	v
1		●		●		●	●			●		●		●		●
2	●			●	●			●		●		●		●		●
3	●		●			●		●	●		●			●		●
4		●		●	●		●			●		●		●		●
5		●		●		●		●		●		●		●		●

For instance, all the experiments labeled as case 5, the worst case, are those where the position, velocity and trajectory of the ball, the position of the interception point, the position and velocity of the passer and the position and velocity of the shooter are uniformly distributed in a pseudo-random way within delimited regions in every trial.

Similar considerations classify the other cases studied. Fig. 6.1 sketches the regions where the ball, the interception point, the passer and the shooter were located in case 5.

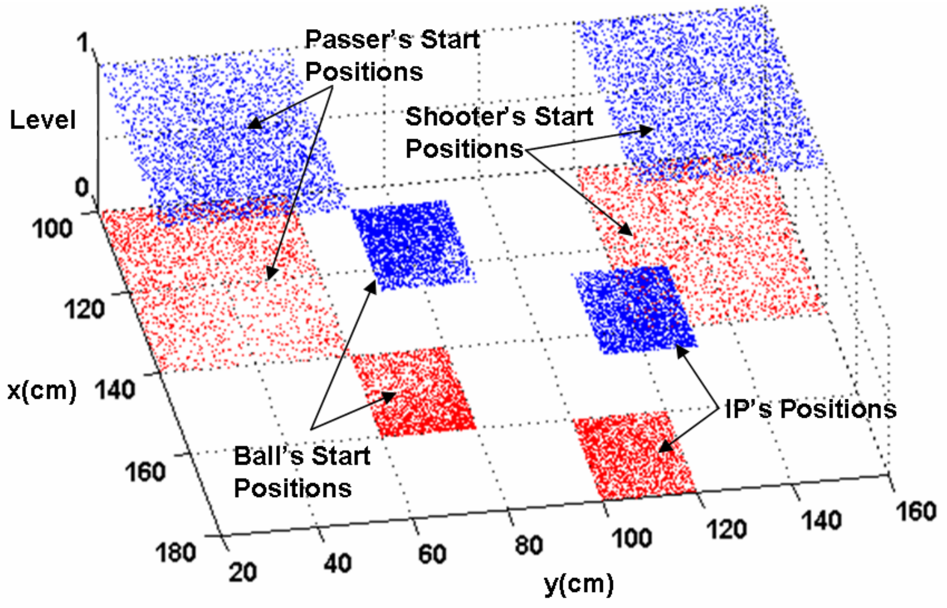


Fig. 6.1. Regions sketch to generate different trials in passing a ball for the case 5. Level 1: successful trials, level 0: unsuccessful trials.

The shooter performance has been tested with a large number of examples (5000 for each case). Fig. 6.2 displays the results for case 5 where in each curve is computed the arithmetic mean of successes (scores or right decisions) of the most recent trials using a sliding window up to the current trial. Several t-tests comparing the values of the curves have been performed, and the levels of significance for important comparisons using the p-value have been reported. Reasonable decision performance is achieved in all cases when the introspective skill is used. Specifically, Fig. 6.2 shows how the task performance (successful scores) of each controller with and without introspection does not make a significant difference (low confidence c_1: p = 0.0104, c_2: p = 0.4966, and c_3: p = 0.0283) and how significantly higher performance is achieved by selecting the most suitable controller using introspection. This curve remains ahead of all the others, with (p = 1.3466e-014 vs. c_1, p = 4.7589e-024 vs. c_2, and p = 1.0477e-022 vs. c_3. Comparison with c_1, c_2 and c_3 using introspection) (see Fig. 6.2a, 6.2c, 6.2e). But more importantly, the decision performance (successful decisions) of each controller is significantly better

(c_1: p = 5.9525e-027, c_2: p = 3.1106e-037 and c_3: p = 4.0583e-039) when the agent uses introspection than when the agent does not use it (see Fig. 6.2b, 6.2d, 6.2f).

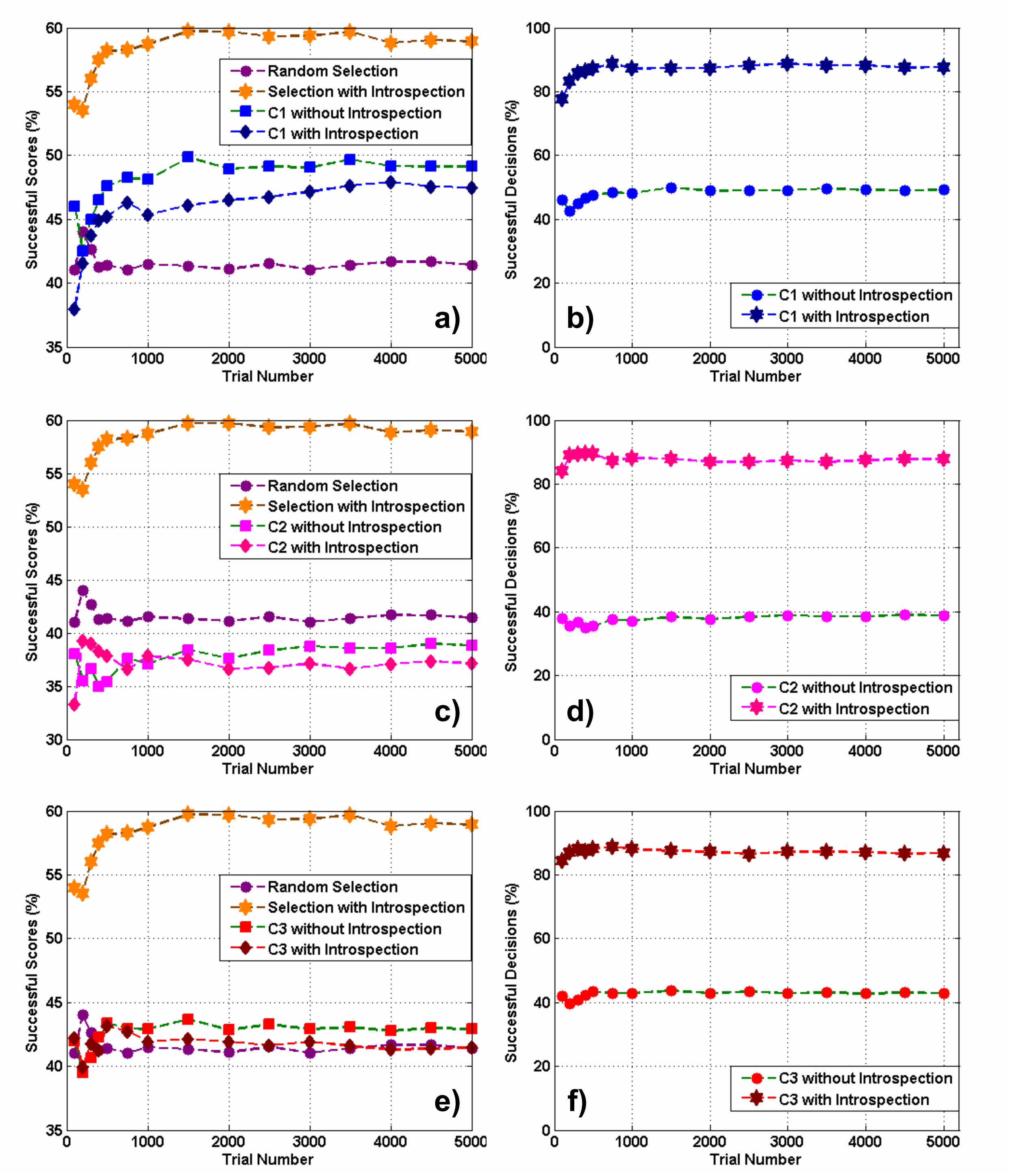


Fig. 6.2. Performance results for case 5. a). Task success scores for controller c_1 with and without introspection compared to random selection and selection with introspection; b). Successful decisions with controller c_1, with and without introspection; c)., d). as for a) and b) but using controller c_2; e)., f). as for a) and b) but using controller c_3.

In addition, Fig. 6.3 sketches the regions where the ball, the interception point, the passer and the shooter were located in cases 1, 2, 3 and 4.

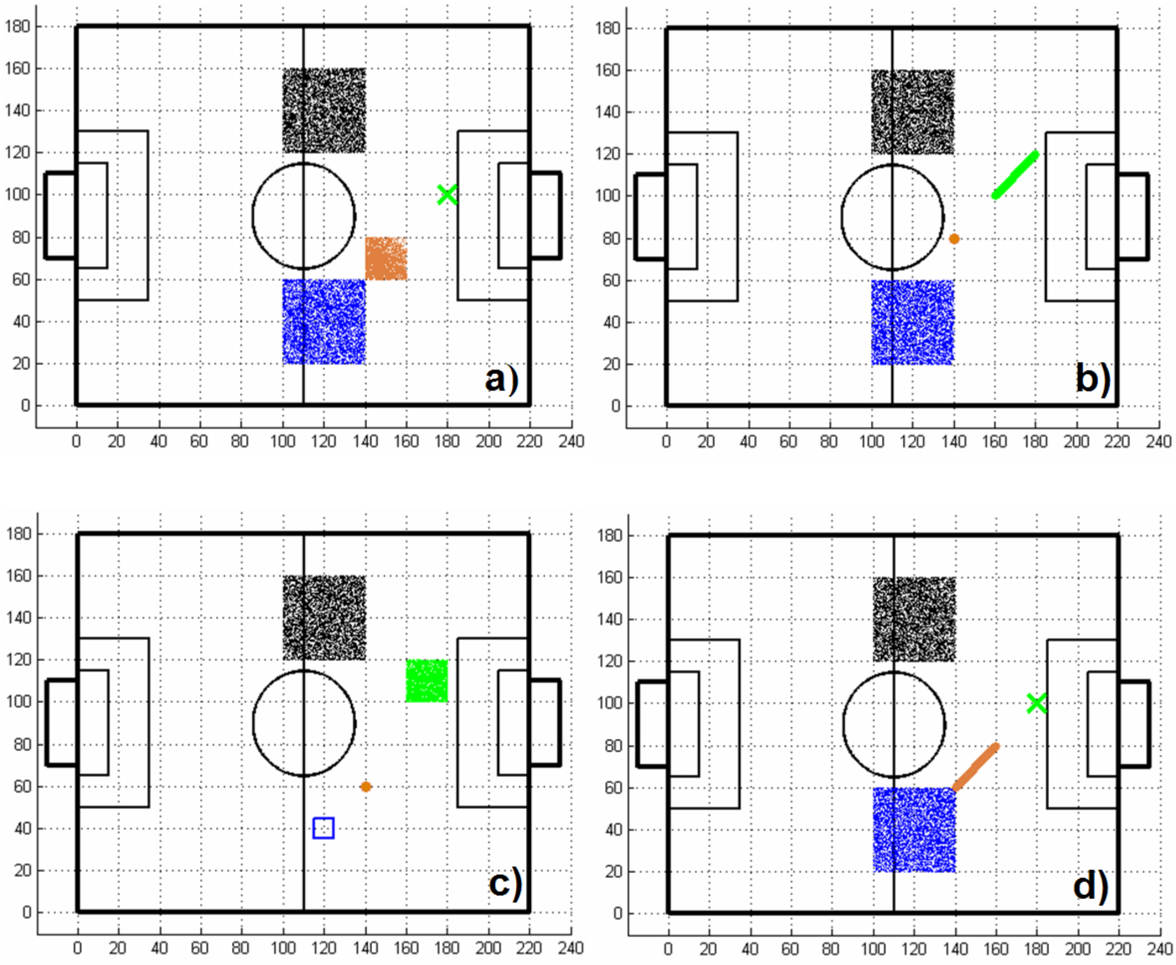


Fig. 6.3. Regions sketch of successful trials in passing a ball. a). Case 1; b). Case 2; c). Case 3; d). Case 4.

Fig, 6.4, 6.5, 6.6 and 6.7 show the results for theses cases. These figures confirm the preliminary conclusion disclosed for the case 5: with introspection increase the performance as the result of most suitable controller selection in the system. The system performance always improves when the physical agents take into account their physical capabilities based on introspection. The figures also confirm that successful decisions related to the task increase when agents use introspection.

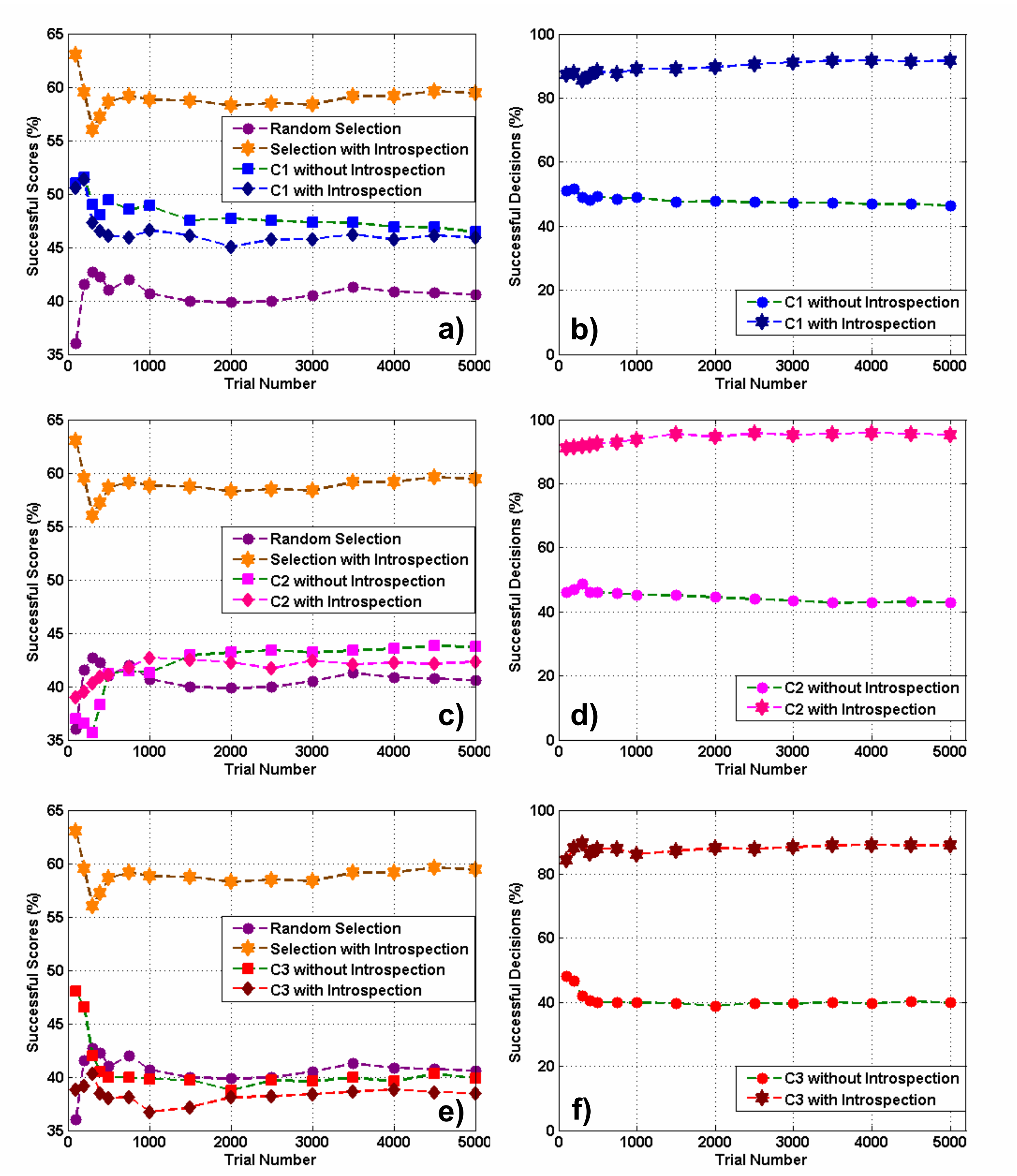


Fig. 6.4. Performance results for case 1. a). Task success scores for controller c1 with and without introspection compared to random selection and selection with introspection; b). Successful decisions with controller c1, with and without introspection; c)., d). as for a) and b) but using controller c2; e)., f). as for a) and b) but using controller c3.

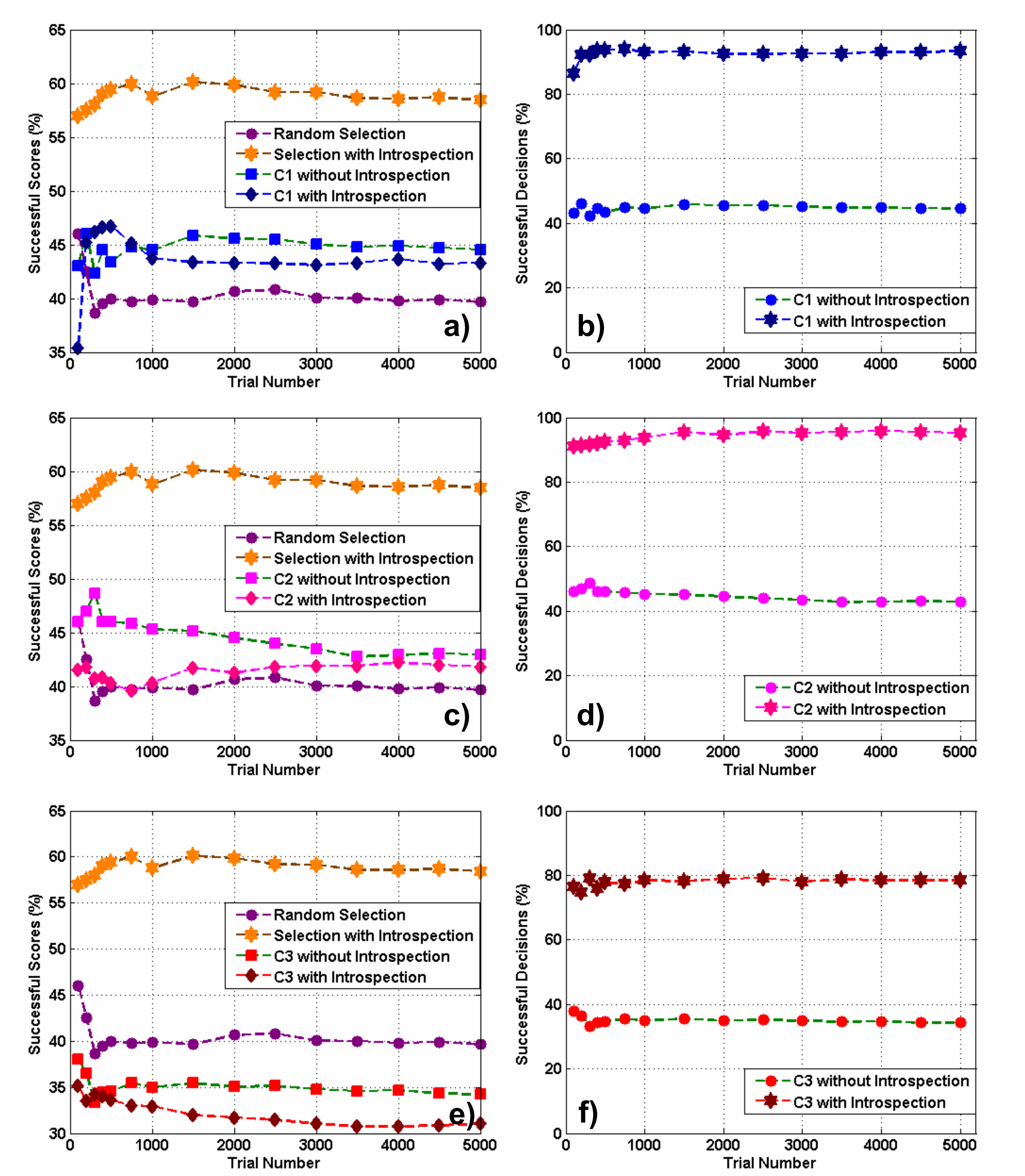


Fig. 6.5. Performance results for case 2. a). Task success scores for controller c1 with and without introspection compared to random selection and selection with introspection; b). Successful decisions with controller c1, with and without introspection; c)., d). as for a) and b) but using controller c2; e)., f). as for a) and b) but using controller c3.

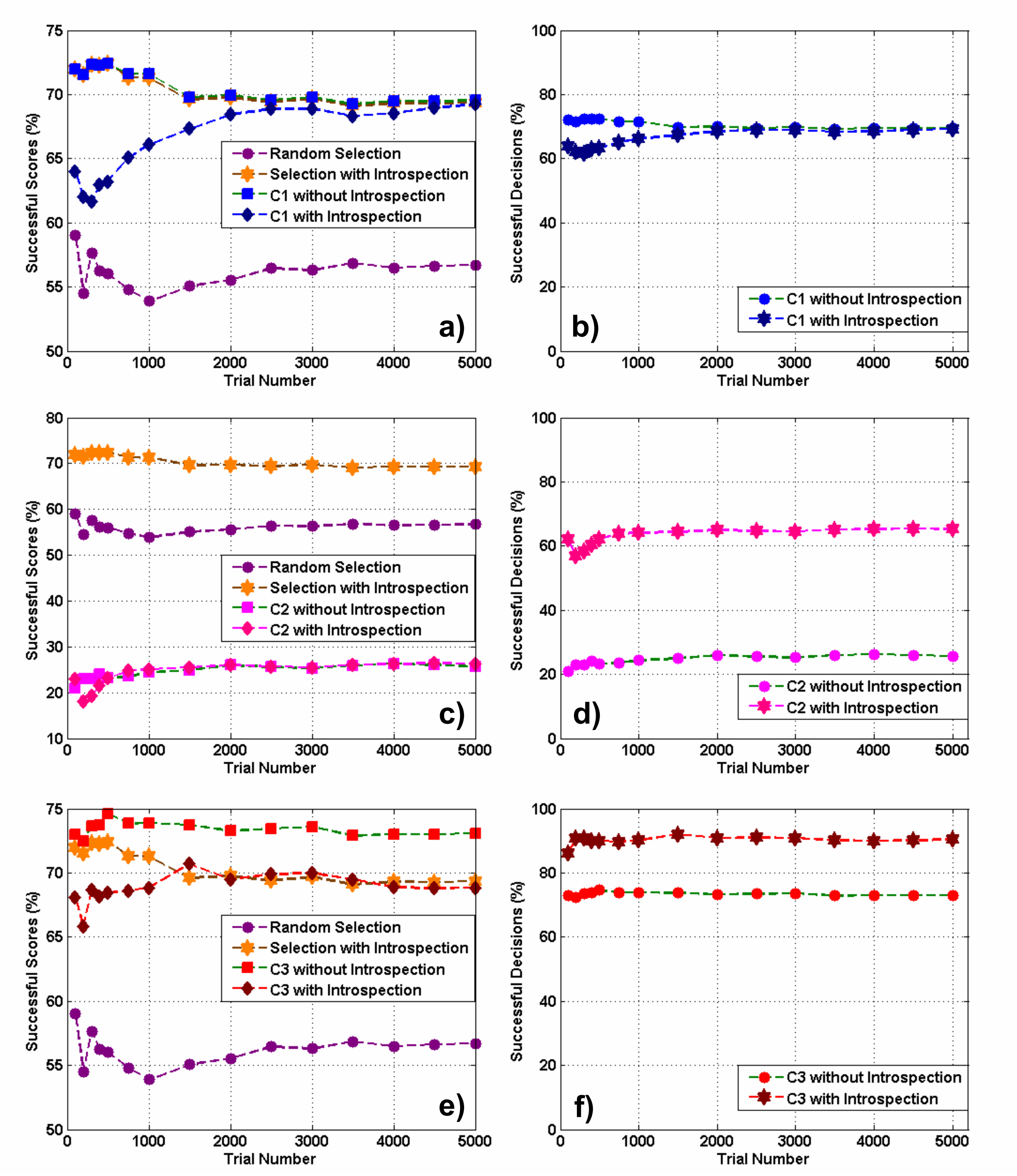


Fig. 6.6. Performance results for case 3. a). Task success scores for controller c1 with and without introspection compared to random selection and selection with introspection; b). Successful decisions with controller c1, with and without introspection; c)., d). as for a) and b) but using controller c2; e)., f). as for a) and b) but using controller c3.

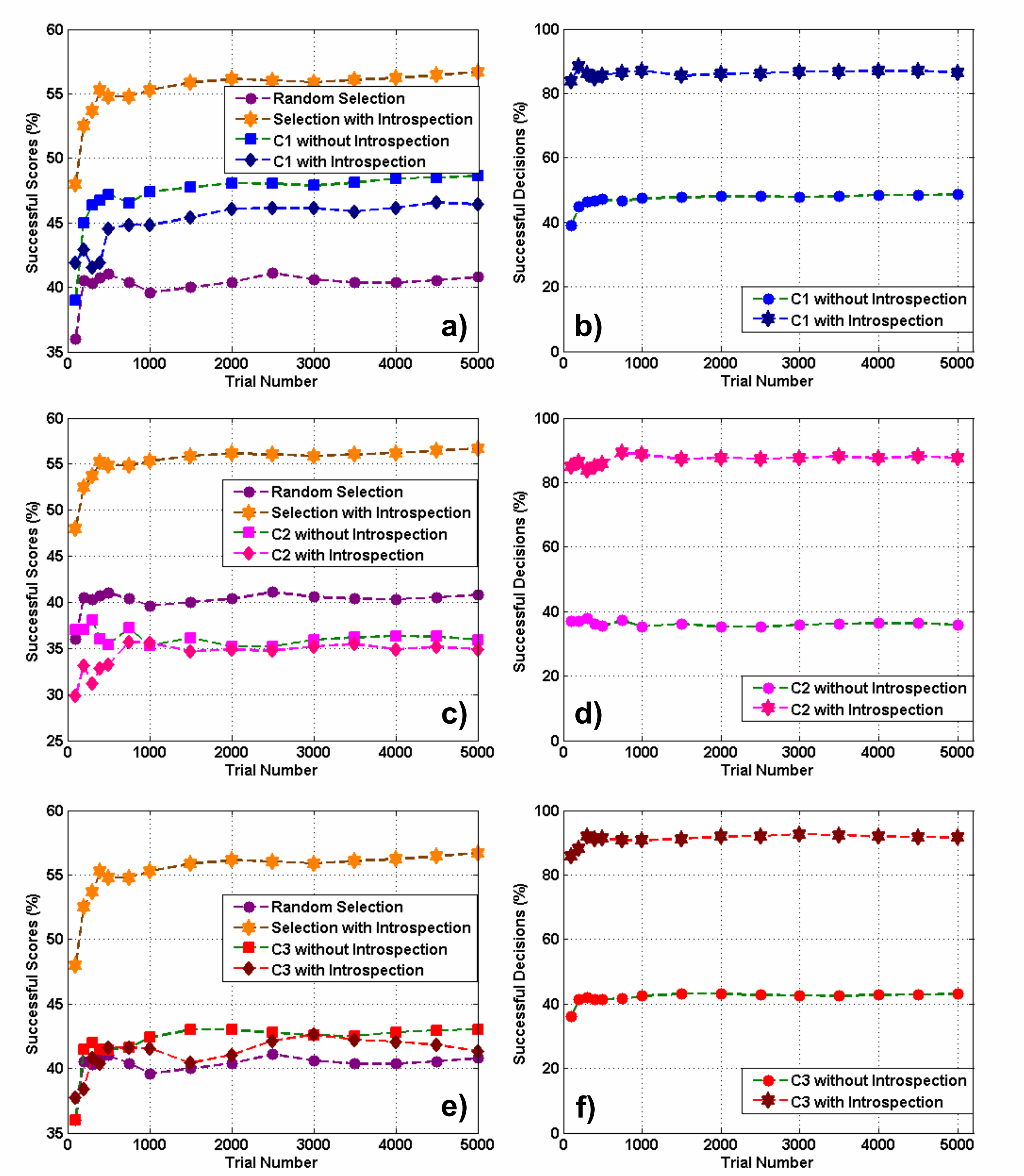


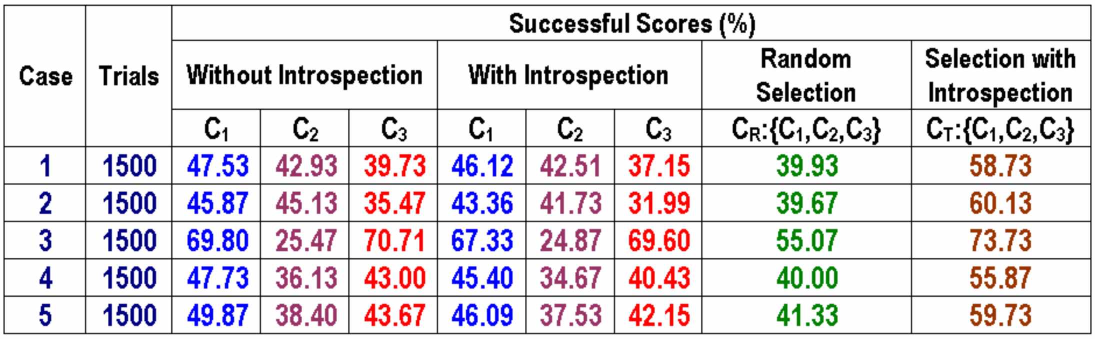
Fig. 6.7. Performance results for case 4. a). Task success scores for controller c1 with and without introspection compared to random selection and selection with introspection; b). Successful decisions with controller c1, with and without introspection; c)., d). as for a) and b) but using controller c2; e)., f). as for a) and b) but using controller c3.

The performance does not improve significantly beyond about 1500 cases for any example. This number of trials is therefore used initially to confirm the task performance.

Table 6.2 shows the successful scores obtained when the shooter uses each controller (c_1, c_2 and c_3) with and without introspection in all cases.

Table 6.2 also presents the performance with random selection of controllers and selection by using introspection. The approach without introspection in Table 6.2 takes into account only the proposed execution time of the task *Time$_T$* by using a supervised learning method described in [De la Rosa et al., 04].

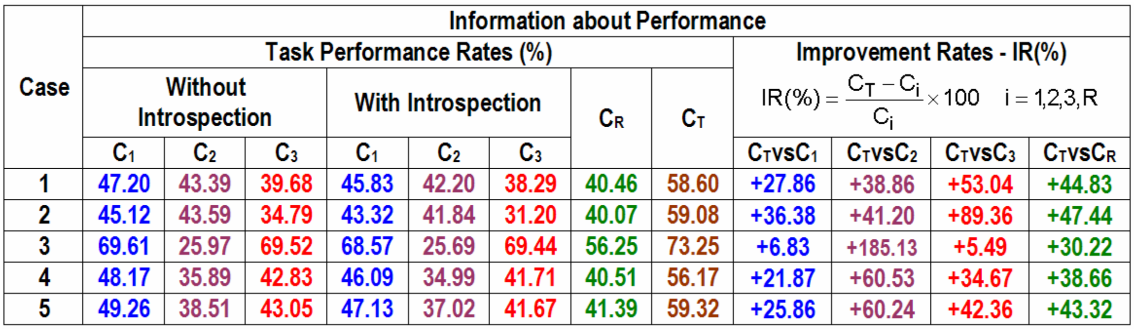
Table 6.2. Successful scores using the controllers in passing a ball.



Case	Trials	Successful Scores (%)							
		Without Introspection			With Introspection			Random Selection	Selection with Introspection
		C_1	C_2	C_3	C_1	C_2	C_3	C_R:{C_1,C_2,C_3}	C_T:{C_1,C_2,C_3}
1	1500	47.53	42.93	39.73	46.12	42.51	37.15	39.93	58.73
2	1500	45.87	45.13	35.47	43.36	41.73	31.99	39.67	60.13
3	1500	69.80	25.47	70.71	67.33	24.87	69.60	55.07	73.73
4	1500	47.73	36.13	43.00	45.40	34.67	40.43	40.00	55.87
5	1500	49.87	38.40	43.67	46.09	37.53	42.15	41.33	59.73

Table 6.3 summarizes the task performance rates and the improvement rates. The results shown in Table 6.3 take into account the average of the achieved successful scores for different simulation points (from 1500 to 5000 with a window of 500 trials).

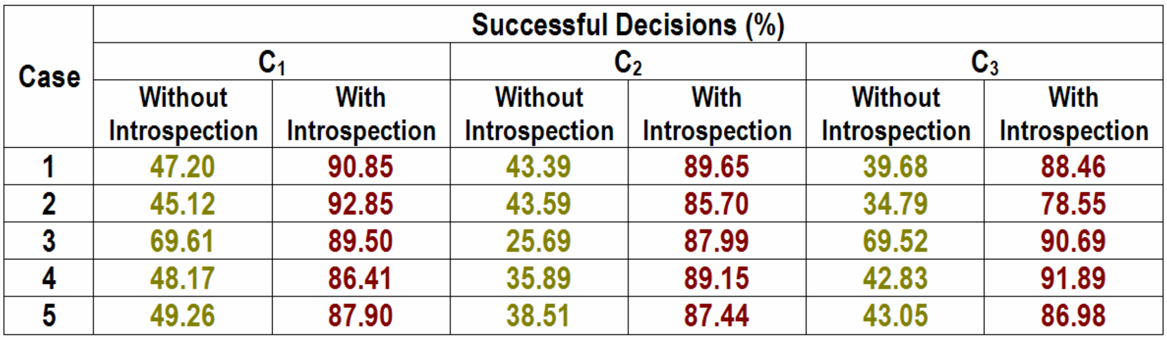
Table 6.3. Information about task performance in passing a ball.



Case	Information about Performance											
	Task Performance Rates (%)								Improvement Rates - IR(%)			
	Without Introspection			With Introspection			C_R	C_T	$IR(\%) = \frac{C_T - C_i}{C_i} \times 100 \quad i = 1,2,3,R$			
	C_1	C_2	C_3	C_1	C_2	C_3			$C_T$vsC_1	$C_T$vsC_2	$C_T$vsC_3	$C_T$vsC_R
1	47.20	43.39	39.68	45.83	42.20	38.29	40.46	58.60	+27.86	+38.86	+53.04	+44.83
2	45.12	43.59	34.79	43.32	41.84	31.20	40.07	59.08	+36.38	+41.20	+89.36	+47.44
3	69.61	25.97	69.52	68.57	25.69	69.44	56.25	73.25	+6.83	+185.13	+5.49	+30.22
4	48.17	35.89	42.83	46.09	34.99	41.71	40.51	56.17	+21.87	+60.53	+34.67	+38.66
5	49.26	38.51	43.05	47.13	37.02	41.67	41.39	59.32	+25.86	+60.24	+42.36	+43.32

Tables 6.2 and 6.3 show that the task performance always improves when the shooter selects its controller taking into account its physical capabilities based on introspection. These tables also corroborate how the task performance of each controller with and without introspection is similar for all cases studied. However, in spite of a similar task performance with and without introspection, Table 6.4 shows that the successful decisions related to the task are increased when agents use introspection: agents can make better decisions and can consequently make more sure and trustworthy commitments.

Table 6.4. Information about decision performance in passing a ball.



Case	Successful Decisions (%)					
	C_1		C_2		C_3	
	Without Introspection	With Introspection	Without Introspection	With Introspection	Without Introspection	With Introspection
1	47.20	90.85	43.39	89.65	39.68	88.46
2	45.12	92.85	43.59	85.70	34.79	78.55
3	69.61	89.50	25.69	87.99	69.52	90.69
4	48.17	86.41	35.89	89.15	42.83	91.89
5	49.26	87.90	38.51	87.44	43.05	86.98

Table 6.5 shows the controller selection rates of the shooter's movement controllers when this agent selects the most suitable controller using introspection. The selected automatic controller has the best performance in the execution of the proposed task. The management rates disclose how the physical agent really manages its controllers. In fact, some experiments are only correctly performed by a specific controller. Other experiments can be performed by more than one controller with different performance indicators. The remaining experiments cannot be performed with any of the available controllers. Thus, both self-examination and management of the physical agent's body based on introspection on its capabilities is in fact a proper alternative to solve this decision-making problem.

Table 6.5. Management rates of the controllers for passing a ball.

Case	Successful Scores	Management Rate of the Agent's Controllers (%)		
		C_1	C_2	C_3
1	881	20.20	46.77	33.03
2	902	24.72	46.90	28.38
3	1106	50.71	13.85	35.44
4	838	23.03	37.35	39.62
5	896	32.14	34.82	33.04

6.1.1.1 Comparison with other Approaches

Experimental results of the successful use of neural networks for learning low-level behavior are presented in [Stone, 00]. This learned behavior, namely shooting a moving ball, is crucial to successful action in the multi-agent domain and illustrates how learned individual skills can be used as a basis for higher level multi-agent learning. To simplify the problem, shooting a moving ball will be considered as intercepting a moving ball. According to the above constraints, a comparative analysis between the proposed approach based on neural networks and the CMU approach [Stone, 00] is shown in Table 6.6.

Table 6.6. Comparative numerical results in intercepting a moving ball.

Trials	Successful Decisions (%)					Improvement Rate IR(%)
	CMU Approach		Proposed Approach			
	Training Data	NN	Training Data	Without Introspection	With Introspection	
750	53	86	50.9	81.25	93.19	+8.36

The misses are not included in the CMU results because these shots are so wide that the agent does not have much chance of even reaching the ball before it goes past. The CMU approach and the approach without introspection have similar performance. In addition, the improvement rate (+8.36 %) of the introspection approach over the CMU approach is caused by the possibility of including the misses in the agents' decisions. In fact, this is an advantage of introspection. Agents can discriminate between the trials in which they have a chance of performing the task and those in which they have no chance.

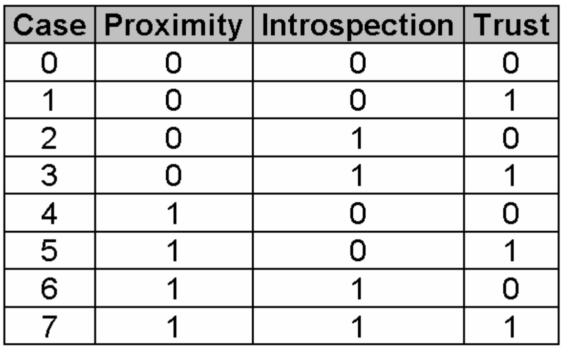
6.1.2 Results in a SimuroSOT Middle League

Empirical experiments featuring simulated robot soccer games have been established following the considerations established in Sections 5.1.4 and 5.1.6.3 to test the system performance when mainly introspection on physical agents' dynamics is taken into account.

The coordination between agents is based on the decision-making parameters described in Section 5.1.6.3. By modifying the set of flags $\mathrm{Ok}_{\mathrm{task}_k}$ is then defined the coordination parameters used in the utility/cost functions for allocating the proposed tasks. Table 6.7 summarizes the combinations of flags (*8 cases*) that have been used in this study.

Table 6.7. Combinations of flags for the coordination parameters.

(0: it is not taken into account, 1: it is taken into account)



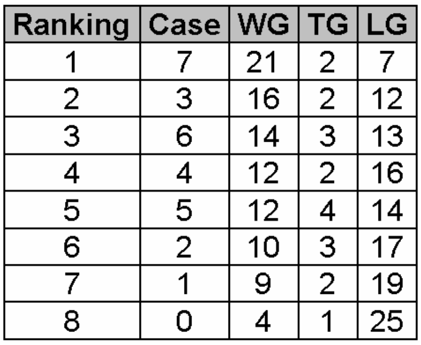
Case	Proximity	Introspection	Trust
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Thirty (30) games have been run for each case described in Table 6.7. Our team played versus a default opponent robotic team provided by the simulator where the initial state of each physical agent in the playground was randomly set at every game.

Table 6.8 shows the results of the experiments performed for each case. The ranking is sorted in a decreasing order taking into account the number of won games to highlight the case with higher performance.

Table 6.8. Empirical results for the cases defined in Table 6.7.

(WG: won games, TG: tied games, LG: lost games).



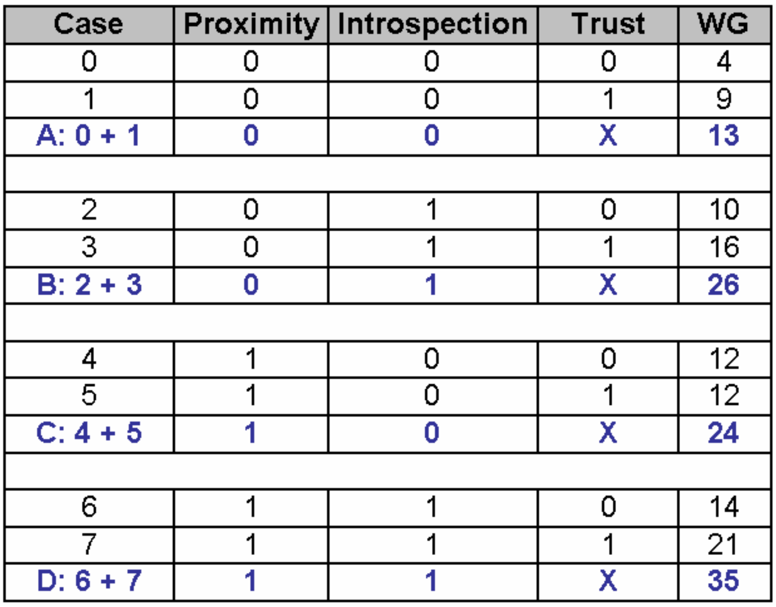
Ranking	Case	WG	TG	LG
1	7	21	2	7
2	3	16	2	12
3	6	14	3	13
4	4	12	2	16
5	5	12	4	14
6	2	10	3	17
7	1	9	2	19
8	0	4	1	25

With the aim of stressing the relevance of managing the diversity in dynamics of the physical agents from a control-oriented perspective, the analysis has been focused on the influence on the agents' decisions of the physical knowledge at the moment of executing the proposed tasks. So, a comparison used the obtained results has been performed in order to note how the system performance improves when agents take into account their physical body in their decision-making. Namely, the relevance of introspection on the multi-agent performance.

Table 6.9 presents a classification of the cases when considering or not the *proximity*, without taking into account the *trust*, it is observed how the introspection influence. In this sense, the system performance improves when introspection is taken into account (cases B and D), than when not (cases A and C).

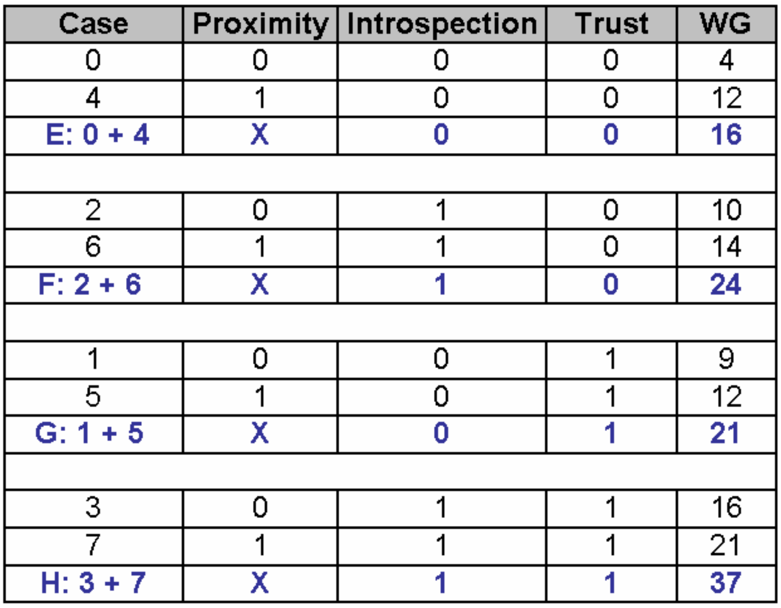
Likewise, Table 6.10 presents a classification of the cases when considering or not the *trust*, without taking into account the *proximity*, it is observed how the introspection influence. In the same way, the system performance is better when introspection is taken into account (cases F and H) than when not (cases E and G). With introspection, physical agents are able to decide based on a suitable knowledge of their capabilities and physical limitations when they can commit to perform a particular task.

Table 6.9. Introspection without taking into account the trust, (X: in any case).



Case	Proximity	Introspection	Trust	WG
0	0	0	0	4
1	0	0	1	9
A: 0 + 1	0	0	X	13
2	0	1	0	10
3	0	1	1	16
B: 2 + 3	0	1	X	26
4	1	0	0	12
5	1	0	1	12
C: 4 + 5	1	0	X	24
6	1	1	0	14
7	1	1	1	21
D: 6 + 7	1	1	X	35

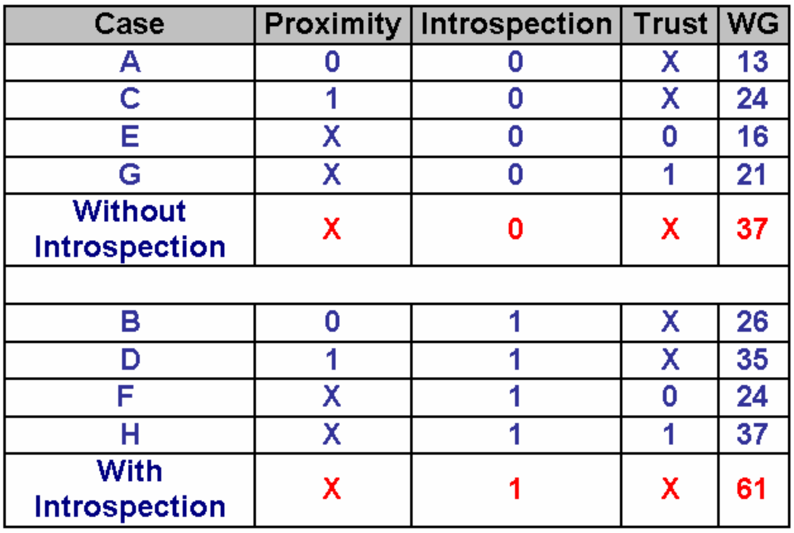
Table 6.10. Introspection without taking into account the proximity, (X: in any case).



Case	Proximity	Introspection	Trust	WG
0	0	0	0	4
4	1	0	0	12
E: 0 + 4	X	0	0	16
2	0	1	0	10
6	1	1	0	14
F: 2 + 6	X	1	0	24
1	0	0	1	9
5	1	0	1	12
G: 1 + 5	X	0	1	21
3	0	1	1	16
7	1	1	1	21
H: 3 + 7	X	1	1	37

The above results show clearly how physical agents can make decisions that carry a better system performance when they use knowledge related to their bodies (introspection) to perform the proposed tasks. Table 6.11 summarizes and concludes the obtained results.

Table 6.11. Classification of the cases with and without introspection, (X: in any case).



Case	Proximity	Introspection	Trust	WG
A	0	0	X	13
C	1	0	X	24
E	X	0	0	16
G	X	0	1	21
Without Introspection	X	0	X	37
B	0	1	X	26
D	1	1	X	35
F	X	1	0	24
H	X	1	1	37
With Introspection	X	1	X	61

Table 6.11 discloses the influence of the introspection in the system performance. In particular, there is an improvement of around a 65% in this case study.

In addition, empirical experiments featuring simulated robot soccer tournaments have been established to corroborate the relevance of the introspection on physical agents' dynamics in the system performance. The selected simulation experiments consist of a predefined number of championships (10), each one with a predefined number of games (10). The performance is measured as a ratio between the total points (*won game: 3 points, tied game: 1 point*) achieved by our team in each championship and the all possible points (30) in this championship where our team played versus a default opponent robotic team provided by the simulator. The initial state of each physical agent in the playground was randomly set at every game.

In particular, the following teams were compared: R vs. I (case 0 vs. case 2), T vs. T + I (case 1 vs. case 3), P vs. P + I (case 4 vs. case 6), and P + T vs. P + T + I (case 5 vs. case 7)(see Table 6.7), where R → random, I → introspection, P → proximity and T → trust.

Fig. 6.8 illustrates how the best system performance is achieved by using introspection in all cases.

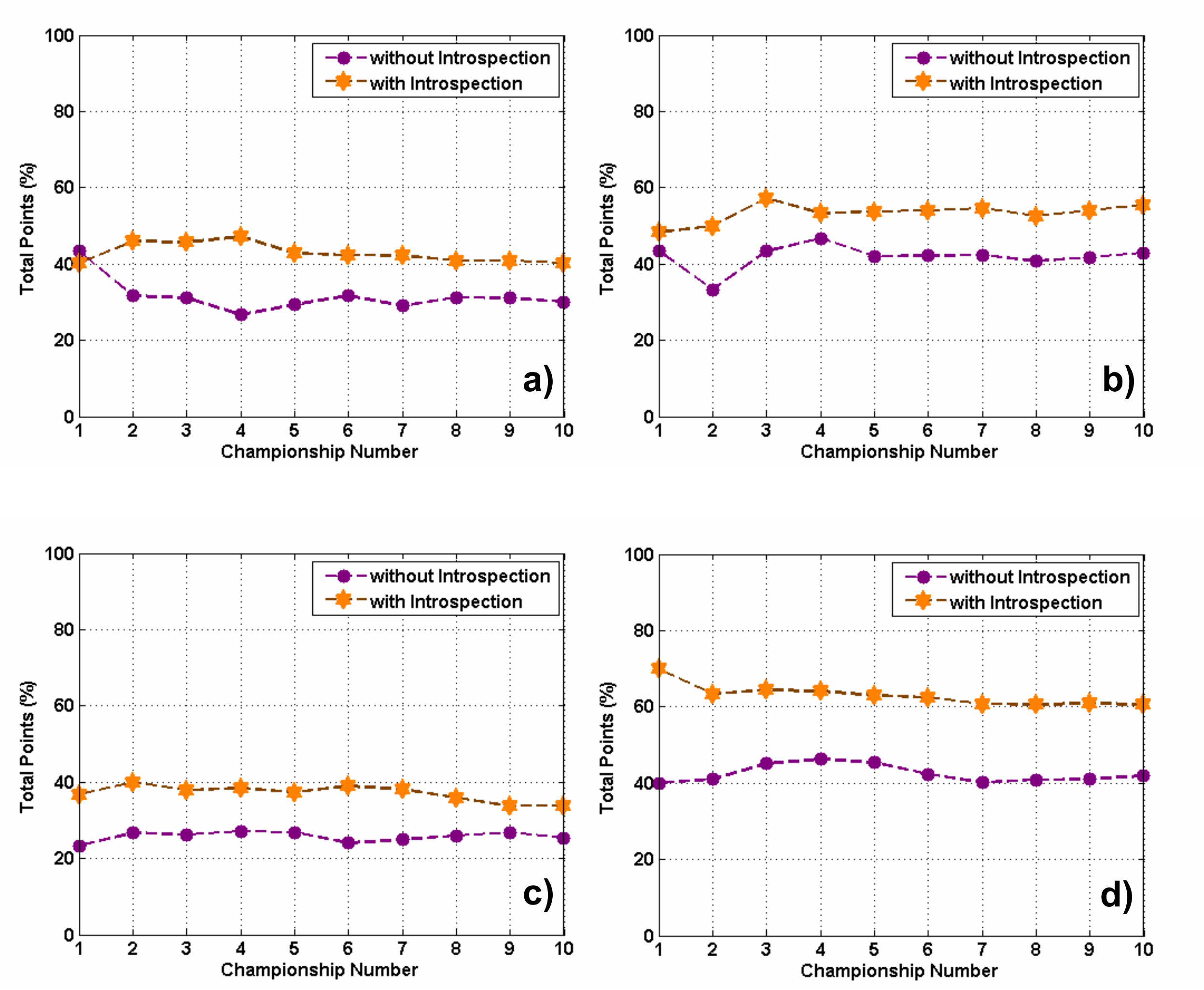


Fig. 6.8. Performance comparison. a) R vs. I; b). P vs. P + I; c). T vs. T + I; d). P + T vs. P + T + I.

Here follows a preliminary conclusion: the composition of any parameters with introspection increases the performance as the result of most suitable task allocation in the system. The system performance always improves when the physical agents take into account their physical capabilities based on introspection. The figure also confirms that successful decisions related to task allocation increase when agents use introspection: agents can make better decisions and can consequently make more sure and trustworthy task allocations. In addition, it should be noted that the improvement rate of the introspection approach over the other approaches is a result of the possibility of including the misses in the agents' decisions. In fact, this is an advantage of introspection. Agents can discriminate between the trials in which they have a chance of successfully performing the tasks and those in which they have no chance.

6.1.3 Results for the Linear Control System Case Study

Tables 6.12 and 6.13 present the proposed empirical experiments using the *passing a ball* and the *offside maneuver* respectively for testing the system performance for the linear control system case study described in Section 5.1.6.4.

Table 6.12. Proposed empirical experiments with passing a ball.

(Pos→Position, Vel→Velocity, Traj→Trajectory, f→fixed, v→variable).

Case	Ball						IP		Passer				Shooter			
	Pos		Vel		Traj		Pos		Pos		Vel		Pos		Vel	
	f	v	f	v	f	v	f	v	f	v	f	v	f	v	f	v
1		•		•		•	•			•		•		•		•
2	•		•			•		•	•		•			•		•
3		•		•		•		•		•		•		•		•

Table 6.13. Proposed empirical experiments with offside maneuver.

(Pos→Position, Vel→Velocity, f→fixed, v→variable).

Case	OL		Defender1				Defender2				Passer	
	Pos		Pos		Vel		Pos		Vel		Time	
	f	v	f	v	f	v	f	v	f	v	f	v
1	•			•		•		•		•		•
2		•		•		•		•		•	•	
3		•		•		•		•		•		•

The selected simulation experiments for each case consist again of a predefined number of trials that satisfy the restrictions established in Tables 6.12 and 6.13. For instance, all the experiments labeled as case 3 in Table 6.12, are those where the position, velocity and trajectory of the ball, the position of the interception point, the position and velocity of the passer and the position and velocity of the shooter can change randomly within delimited regions in every trial. Similar considerations classify the other cases studied.

All cases were tested using a typical scene that involved three situations mentioned in Section 5.1.6.4, *SL (space limitations)*, *MD* (*motion disturbances)* and *EP* (*energy performance*) selecting empirically to this set a $\mathrm{ID}_{\mathrm{task}_1} = \mathrm{ID}_{\mathrm{task}_2}$ = [40(*SL*) 0(*QB*) 20(*MD*)

0(AB) 40(EP)]%. Tables 6.14 and 6.15 show the successful actions obtained when the physical agents performed the proposed coordinated tasks in the empirical experiments.

Table 6.14. Successful scores in the passing a ball.

Case	Trials	Successful Scores (%)				
		Individual Performance			Random Selection	Selection with Our Approach
		C1	C2	C3	{C1,C2,C3}	{C1,C2,C3}
1	1296	26.62	29.48	33.72	26.16	34.65
2	1575	15.56	34.73	31.62	18.60	39.05
3	2304	36.68	38.88	55.51	37.50	57.42

Table 6.15. Successful offsides in the offside maneuver.

Case	Trials	Successful Offsides (%)	
		Collective Performance Random Selection	Collective Performance Selection with Our Approach
		{C1, C2, C3}	{C1,C2,C3}
1	1152	12.85	39.58
2	1296	19.60	40.28
3	2048	23.97	45.07

These tables show how the performance is improved when the introspection approach is used for individual and cooperative decisions in order to perform correctly the required tasks. Table 6.14 shows the successful scores obtained when the shooter uses each controller (c_1, c_2, and c_3) individually. Table 6.14 also presents the performance with random selection of controllers and selection using introspection.

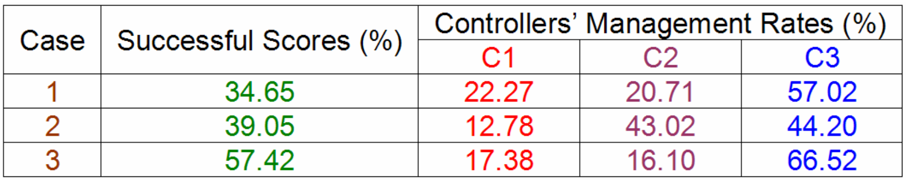
Tables 6.14 shows that the passing a ball performance always improves when the shooter selects its controller taking into account its control-grounded capabilities.

Table 6.15 shows a comparison about successful offsides obtained when the defender1 and the defender2 select their controllers randomly and when they do the selection using introspection. Tables 6.15 shows that the offside performance always improves when the defenders select their controller taking into account their control-grounded capabilities.

According to the above results, agents can make better decisions and can consequently make more sure and trustworthy commitments using their control-grounded capabilities.

Table 6.16 shows the controller selection rates of the shooter's movement controllers when this agent selects the most suitable controller for passing a ball. The selected automatic controller has the best performance in the execution of the proposed task. The management rates disclose how the physical agent really manages its controllers.

Table 6.16. Management rates of the controllers for passing a ball in the linear control system case study.



Case	Successful Scores (%)	Controllers' Management Rates (%)		
		C1	C2	C3
1	34.65	22.27	20.71	57.02
2	39.05	12.78	43.02	44.20
3	57.42	17.38	16.10	66.52

Fig. 6.9 shows the management rates of each controller (c_1, c_2, and c_3) using several successful actions of the experiment 3 described in Table 6.12 (609 successes scores in the opposite goal of 752 attempts) when the task involves the following situations, *SL*, *QB*, *MD*, *AB*, and *EP* with different influence degrees.

The results in Fig. 6.9 show how the control-grounded capabilities help to select the most skilled controller (aggressive, fast, persistent, with low control effort, etc) with the best performance for the execution of the proposed task according to each set of situations and the influence degrees in the task. Thus, the physical agent really manages its automatic controllers, according to the explicit information embedded in its control-grounded capabilities.

A better self-examination and management of the agent's physical body taking into account its capabilities is in fact a proper alternative to solve this decision problem.

This approach improves the performance in coordinated tasks based on a deep understanding of the physical features of the physical agents' structures from a control-oriented viewpoint.

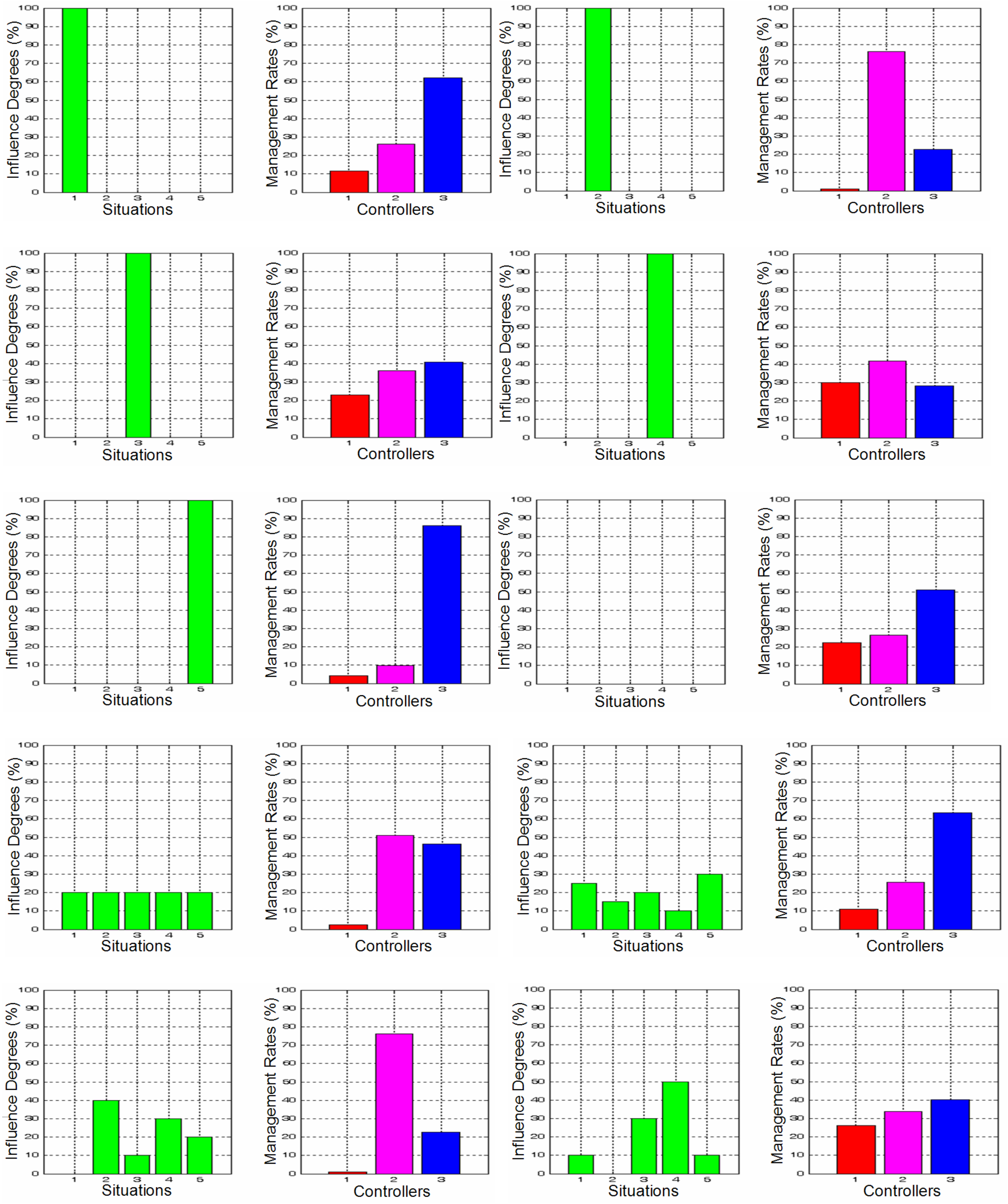


Fig. 6.9. Management rates of each controller (c_1, c_2, and c_3) for 10 different cases that involve different IDs of SL (1), QB (2), MD (3), AB (4), and EP (5).

6.2 Results in the Simulated Rescue Case Study

Here, several empirical experiments featuring a simulated rescue environment are proposed when introspection on control-grounded capabilities is used. The tests follow the specifications described in Section 5.2.2 for this case study.

A predefined number of trials constitute the selected simulation experiments. They satisfy the following restrictions: the rescue agents' locations, the victims' death times and their locations change randomly in every trial.

The system performance has been tested by comparing the proposal with other task allocation approaches: *random, proximity and deadline* described in Section 5.2.2.1. In these approaches, the arithmetic mean of successes (rescues or right decisions) of the most recent trials using a sliding window up to the current trial is computed in all curves.

The task allocation performance was tested with a large number of examples (5000). Fig. 6.10 illustrates how reasonable better performance (all successful rescues) is achieved by using introspection.

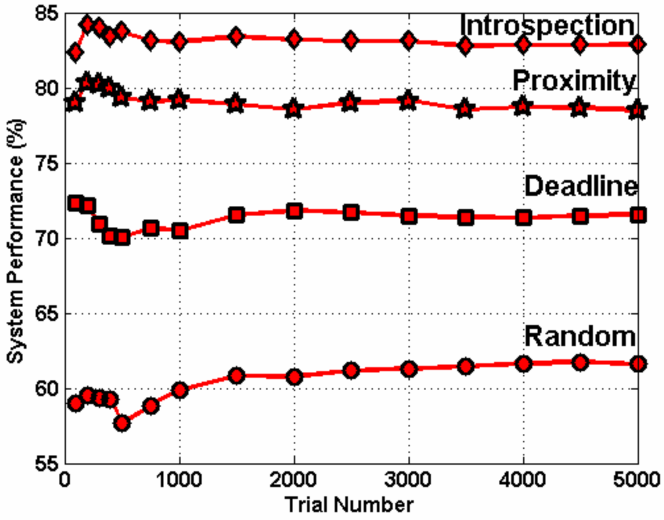


Fig. 6.10. Comparison of task allocation performance.

Fig. 6.11 shows for instance the successful number of tasks performed by the physical agents 1 and 2 respectively.

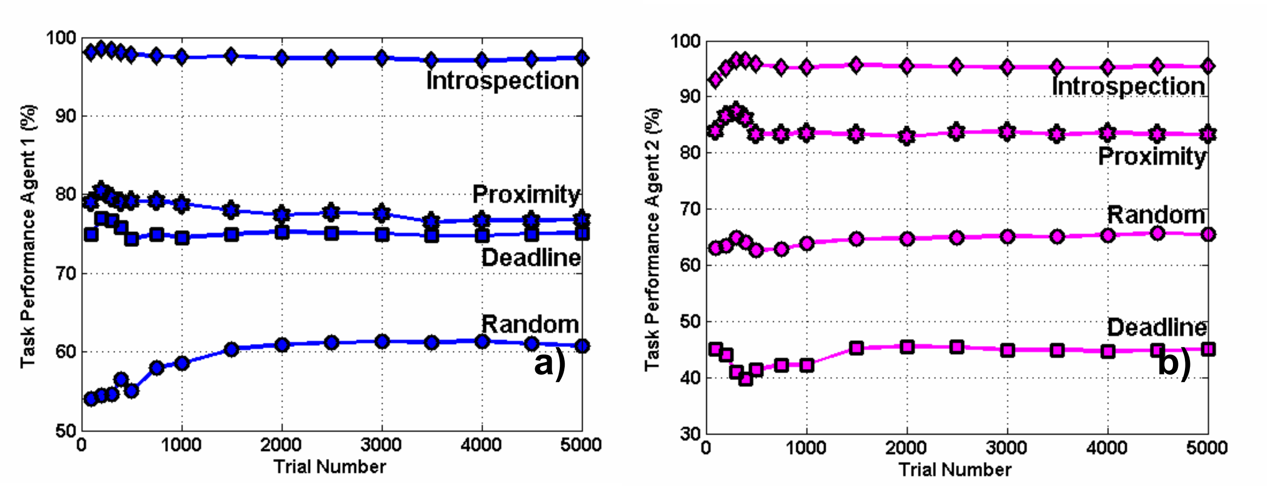


Fig. 6.11. a). Successful tasks of physical agent 1; b). Successful tasks of physical agent 2.

Fig. 6.12 shows a comparison of the successful performances of tasks 1, 2, 3 and 4, i.e. successful rescues of victims 1, 2, 3 and 4 respectively. The figures show how the best performance is achieved in all proposed tasks when the physical agents use their introspective skills.

The decision performance (successful decisions) of each physical agent is better when the agent uses introspection than when the agent does not use it (see Fig. 6.13). Successful decisions are related to the ratio between the number of successful tasks performed by the agent and the total number of decided tasks by the same agent. Performance does not improve significantly beyond about 1500 cases for any experiment.

The figures show that system performance always improves when the physical agents take into account their physical capabilities based on introspection. These figures also confirm that successful decisions related to task allocation increase when agents use introspection: agents can make better decisions and can consequently make more sure and trustworthy task allocations.

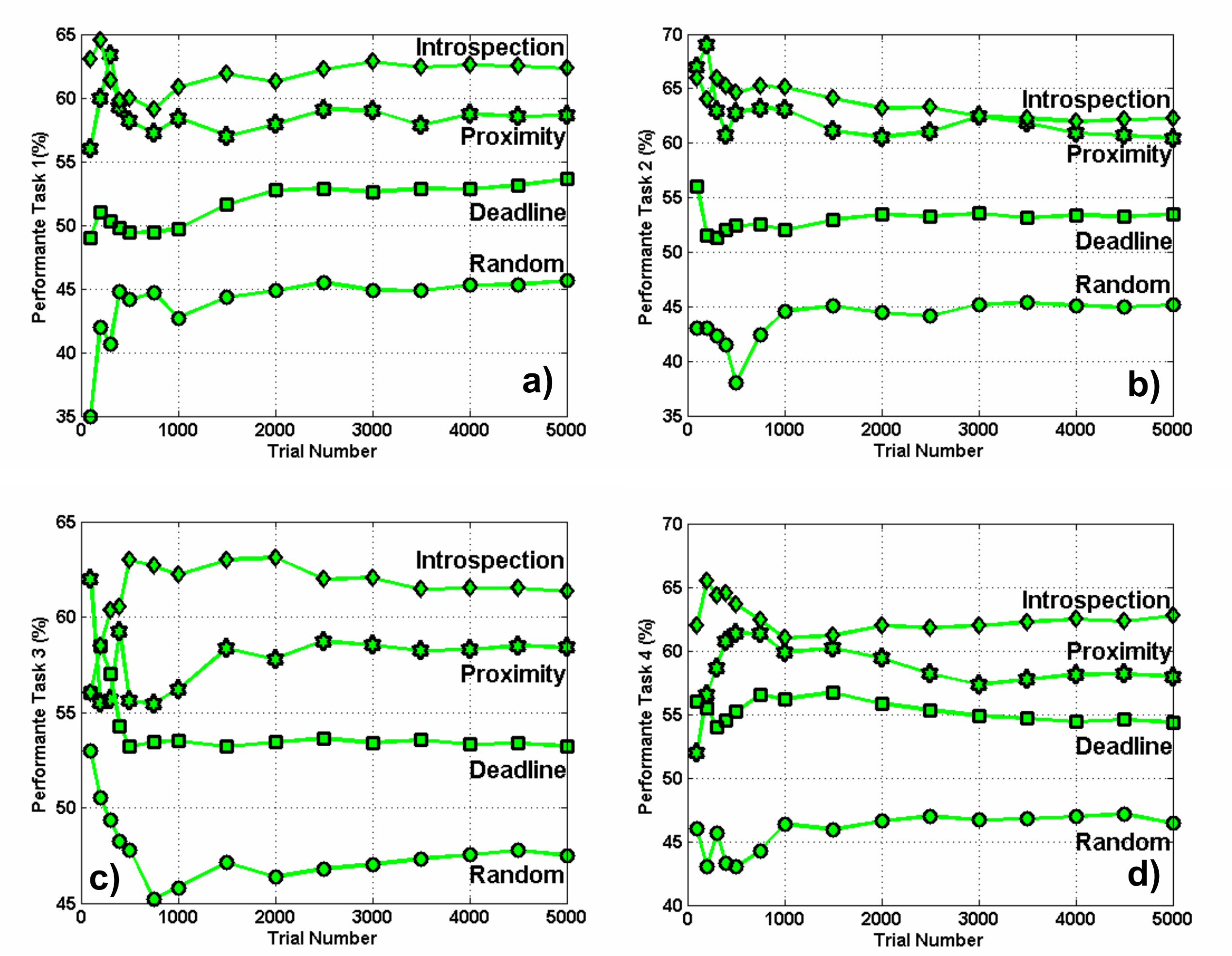


Fig. 6.12. a). Successful performance of task 1; b). Successful performance of task 2; c). Successful performance of task 3; d). Successful performance of task 4.

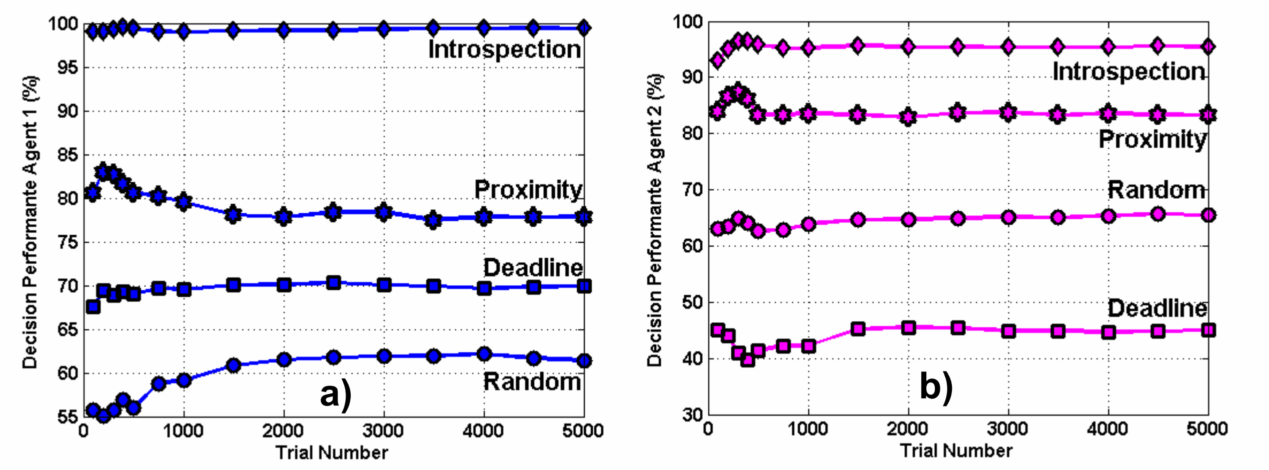


Fig. 6.13. a). Successful decisions of the physical agent 1; b). Successful decisions of the physical agent 2.

In addition, it should be noted that the improvement rate of the introspection approach over the other approaches is a result of the possibility of including the misses in the agents' decisions. In fact, this is an advantage of introspection. Agents can discriminate between the trials in which they have a chance of successfully performing the tasks and those in which they have no chance.

6.3 Results in the Simulated Convoys Case Study

Empirical experiments featuring four different simulated convoys are presented in this section to test the system performance related to avoid collisions. The tests follow the specifications described in Section 5.3.2 for this case study.

The selected simulation experiments consist of a predefined number of trials that satisfy the following restrictions: the agents' initial locations, the initial and final velocities, the deceleration time and the security distance between the vehicles can change randomly in every trial.

The first order transfer functions to represent the dynamics of the agent-controlled vehicles for the four convoys, as it is described in Section 5.3.2, are:

$$\text{Convoy 1:} \qquad G_1(s) = \frac{1}{0.485s+1} \text{ and } G_2(s) = \frac{1}{0.510s+1}$$

$$\text{Convoy 2:} \qquad G_3(s) = \frac{1}{0.385s+1} \text{ and } G_4(s) = \frac{1}{0.610s+1}$$

$$\text{Convoy 3:} \qquad G_5(s) = \frac{1}{0.285s+1} \text{ and } G_6(s) = \frac{1}{0.710s+1}$$

$$\text{Convoy 4:} \qquad G_7(s) = \frac{1}{0.185s+1} \text{ and } G_8(s) = \frac{1}{0.810s+1}$$

The designed distance controllers for the convoys have the following parameters *convoy 1*: $K_p = 5.5$ and $K_i = 0.5$, *convoy 2*: $K_p = 4.5$ and $K_i = 0.5$, *convoy 3*: $K_p = 6$ and $K_i = 0.5$, and *convoy 4*: $K_p = 3.5$ and $K_i = 0.5$.

A large number of experiments (2000) have been made. Fig. 6.14 illustrates how reasonable better performance (successful tasks: autonomous driving free of collisions) is achieved by the agents when they use introspection. As a strong result, a greater amount of collisions can be avoided using introspection than when it is not used.

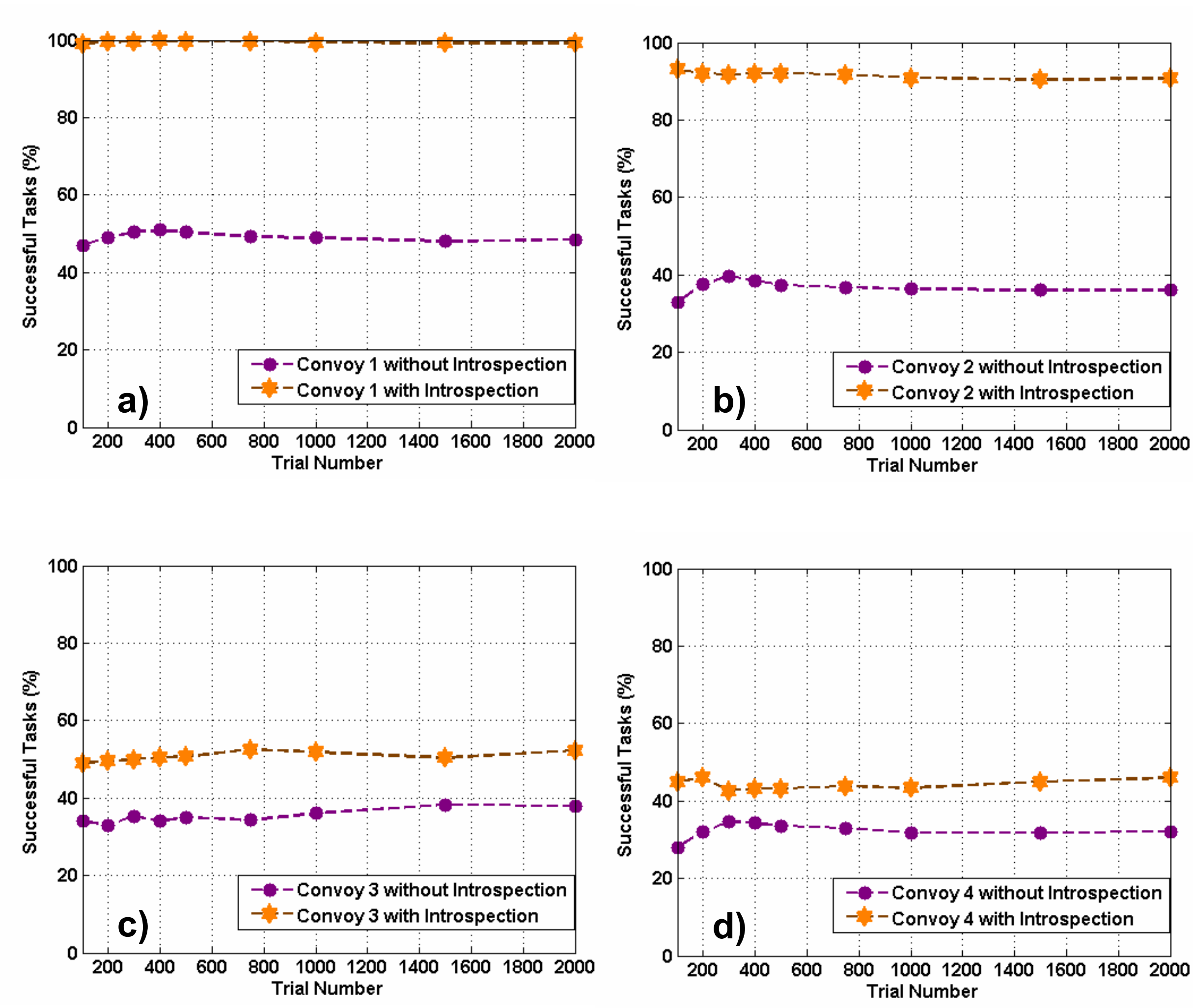


Fig. 6.14. Performance comparison related to autonomous driving free of collisions for a). Convoy 1; b). Convoy 2; c). Convoy 3; d). Convoy 4.

It should be noted that the distance controllers are the same in both reactive (without introspection) and deliberative (with introspection) decisions. However, better decisions are now obtained for better execution of the autonomous driving free of collisions by the implemented controllers with introspection.

In addition, Fig. 6.14 discloses how the diversity in dynamics impact in the system performance. In this sense, as a natural conclusion, a greater amount of collisions is avoided when the vehicles' dynamics in the convoy are similar (e.g., Fig 6.14a) than when they are progressively different (e.g., Fig. 6.14b → 6.14c → 6.14d).

Chapter 7

Conclusions and Future Work

This chapter summarizes the main conclusions arisen of the analysis and discussion of the results reported in this work. The chapter also reviews the dissertation's scientific contributions and then discusses promising directions for future research and application in certain topics in which the work of this thesis can continue. Finally, some concluding remarks are drawn.

7.1 Analysis and Discussion of Results

The work and results presented in Chapters 4, 5 and 6, show that a good decision tool based on introspective reasoning can increase the autonomy and self-control of agents in coordinated tasks and allows obtaining reliable utility/costs functions in the agents' decision-making for task allocation problems.

Introspection and decisions based on capabilities give a trustworthy indication of the real reliability with which each agent make commitments in cooperative systems and embody well-defined concepts to enable an effective agents' performance to meet high-level goals.

This new and effective approach contributes to improve multi-agent efficiency and performance in a cooperative scenario because the physical agents can know and inform opportunely if they can perform the required tasks. If they cannot, the agents can make another decision depending on the general interests of the multi-agent system. Thus, physical agents have some introspection on what they can or cannot do. Without introspection, physical agents would try to perform actions without no sense, decreasing generally the whole system performance.

7.2 Main Contributions

Motivated by the challenges mentioned in Chapter 1 and the problem statement detailed in Chapter 4, this thesis contributes with powerful tools, independent of particular implementation technologies, for building intelligent artificial agents with strong introspective reasoning mainly related to their dynamics, aiming at a correct execution of tasks in cooperative environments. More specifically, the thesis presents an appropriate alternative to include control-oriented knowledge in the physical agents' decision-making and represent such knowledge in a set of control-grounded capabilities. The conventional control techniques tend to either ignore or do implicit and naïve suppositions on the dynamics of the controllers, however this thesis shows that an explicit representation of dynamics based on control-grounded capabilities is possible and useful. The thesis has then shown how to bridge the gap between the high abstraction level of agents and the low abstraction level of the automatic control architectures.

The main contributions of this thesis are summarized as follows:

- *A formal design methodology based on introspective reasoning to use control-oriented knowledge in an agent-oriented manner.*

The thesis reported a way of taking advantage of control-oriented information related to the physical agents' dynamics and outlined the way of including such knowledge in the agents' decision-making. For that, an agent-inspired approach based on introspective reasoning that embraces concepts and techniques from automatic control, mobile robotics and artificial intelligence was proposed.

- *A formulation based on control-grounded capabilities to represent explicitly control-oriented information of agent-controlled systems.*

Here, control-grounded capabilities are seen as a proper alternative used by physical agents to include the knowledge of them related to their physical bodies obtained by means of introspection. More specifically, the capabilities consist in parameters that describe the dynamical behavior of the physical agents when they used an specific automatic controller in a proposed task, i.e., capabilities embed relevant control-oriented knowledge related to the physical agents' bodies (mainly about their automatic controllers) to obtain reliable low level information to use in the high level decision-making.

- *A decision-making tool based on introspection on control-grounded capabilities as a bridge to the gap between the high abstraction level of agents and the low abstraction level of the automatic control architectures.*

The requirements that the control-oriented knowledge must achieve to be a reliable agent-oriented representation and a useful decision tool were established, and the utility and feasibility of the overall proposed approach on several coordinated control examples were demonstrated.

7.3 Related Publications

The work developed for this thesis has led to several contributions presented and discussed in different conferences and congresses. The most relevant works are listed below.

De la Rosa J. Ll., Figueras A., Quintero M. Christian G., Ramón J. A., Ibarra S., and Esteva S. "Outline of Modification Systems". **Studies in Computational Intelligence**, vol. 57, pp. 55 – 69, Springer-Verlag, ISSN: 1860-949X, 2007.

> *New hints for engineers to design control systems are presented. We propose that control engineers may keep KISS design in the control dimension, by explicitly introducing awareness (introspection) and interaction (trust) that let improve the performance of a machine while keeping the design simplicity.*

Quintero M. Christian G., Busquets D., de la Rosa J., and Vehí J. "Introspection on Control-grounded Capabilities. Relevance in Task Allocation Problems". In Proceedings of the **European Control Conference ECC 2007**, vol. 1, pp. 2833 – 2840, ISBN: 978-960-89028-5-5, Kos – Greece, Jul 2, 2007.

> *An approach for reliable task allocation in cooperative agent-controlled systems by means of introspection on control-oriented features is presented. In particular, this proposal is demonstrated in the successful performing of rescue operations by cooperative mobile robots in a simulated environment.*

Quintero M. Christian G., Ibarra S., de la Rosa J., and Vehí J. "Introspection on Control-grounded Capabilities. A Task Allocation Case Study in Robot Soccer". In Proceedings of the 4th **International Conference on Informatics in Control, Automation and Robotics ICINCO 2007**, vol. 2, pp. 461 – 467, INSTICC Press, ISBN: 978-972-8865-87-0, Angers – France, May 9, 2007.

> *A proposal for task allocation in physical multi-agent systems by means of novel coordination parameters in the task utility/cost functions is presented. The composition of any parameters with introspection increases the performance as the result of most suitable task allocation. This proposal is demonstrated in the successful performing of tasks by cooperative mobile robots in a simulated robot soccer environment.*

Quintero M. Christian G., Ibarra S., de la Rosa J., and Vehí J. "A Coordination Approach for Task Allocation. Case Study in Robot Soccer". In Proceedings of the 2nd **Spanish Congress on Computer Science CEDI 2007**, vol. 1, pp. 35 – 42, Thomson-Paraninfo Press, ISBN: 978-84-9732-597-4, Zaragosa – Spain, Sep. 11, 2007.

> *An illustrative example in robot soccer of new coordination parameters to improve the coordination among physical agents in task allocation problems is shown. Our approach proposes introspection, proximity, and trust as key parameters in the utility/cost functions to achieve the above aim. These parameters were managed in a holistic manner to select the most suitable agent to perform the proposed tasks.*

Ibarra S., Quintero M. Christian G., Busquets D., Ramón J., de la Rosa J., and Castán J. "Improving the Team-work in Heterogeneous Multi-agent Systems. Situation Matching Approach". **Frontiers in Artificial Intelligence and Applications – AI Research & Development**, vol. 146, pp. 275 – 282, IOS Press, ISSN: 0922-6389, Amsterdam – Netherlands, Oct., 2006.

> *A "Situation Matching" method that aims at improving cooperative tasks in heterogeneous multi-agent systems is proposed. The situation matching represents a match between the system requirements and the agents' capabilities. We studied how the heterogeneous agents' performance improves by means of such "situation matching" in the robot soccer test bed.*

Quintero M. Christian G., Ibarra S., de la Rosa J., and Vehí J. "Dynamics Features on Robots Decisions. A Perspective based on Control-grounded Capabilities". In Proceedings of 5th **IEEE International Symposium on Robotics and Automation ISRA 2006**, vol. 1, pp. 199 – 206, ISBN: 970-769-070-4, Hidalgo – México, Ago. 25, 2006.

> *Theoretical and practical groundwork based on control-grounded capabilities to include dynamics features on the decision-making of cooperative mobile robots from a control-oriented viewpoint is presented. We stress the advantages of our approach in coordinated tasks of robot soccer.*

Ibarra S., Quintero M. Christian G., Ramón J. A., de la Rosa J., and Castán J. "Studies about Multi-agent Team Work Coordination in the Robot Soccer Environment". In Proceedings of the **FIRA Robot World Congress 2006**, vol. 1, pp. 63 – 67, ISBN: 3-00-019061-9, Dortmund – Germany, Jun. 30 - Jul. 1, 2006.

> *A mechanism based on a characteristic of physical agents named "degrees of situation" that aids to improve the coordination among heterogeneous agents is suggested. These systems can be represented by means of the "physical agent" paradigm. We studied how the team work can be improved by the "degrees of situation" management in robot soccer.*

Quintero M. Christian G., de la Rosa J., and Vehí J. "Self-Knowledge based on the Atomic Capabilities Concept. A Perspective to Achieve Sure Commitments among Physical Agents". In Proceedings of the 2nd **International Conference on Informatics in Control, Automation and Robotics ICINCO 2005**, vol. 3, pp. 425 – 430, INSTICC Press, ISBN: 972-8865-30-9, Barcelona – Spain, Sep. 14 - 17, 2005.

> *An example based on the CBR methodology to manage the atomic capabilities of physical agents executing an offside maneuver in robot soccer is established. This approach allows to each agent a reliable self-knowledge which concludes in achieving sure commitments and intelligent self-control in cooperative environments.*

Quintero M. Christian G., de la Rosa J., and Vehí J. "Studies about the Atomic Capabilities Concept for Linear Control Systems in Physical Multi-Agent Environments". In Proceedings of the **IEEE International Conference on Computational Intelligence in Robotics and Automation CIRA 2005**, vol. 1, pp. 727 – 732, IEEE Catalog Number: 05EX1153, ISBN: 0-7803-9355-4, Espoo – Finland, Jun. 27 - 30, 2005.

> *The impact of the atomic capabilities concept to include control-oriented knowledge of linear control systems in the decision-making of physical agents is highlighted. This approach is presented using an introspective reasoning approach and control theory based on the specific*

tasks of passing a ball and executing the offside maneuver between physical agents in the robot soccer test bed.

Quintero M. Christian G., de la Rosa J., and Vehí J. "Physical Intelligent Agents' Capabilities Management for Sure Commitments in a Collaborative World". **Frontiers in Artificial Intelligence and Applications – AI Research & Development**, vol. 113, pp. 251 – 258, IOS Press, ISSN: 0922- 6389, ISBN: I 58603 466 9, Amsterdam – Netherlands, 2004.

A way of improving the management of commitments among physical agents by exploiting all knowledge related to their physical bodies represented in their atomic capabilities is proposed. This approach is implemented using interconnected neural networks and fuzzy sets in the specific task of passing a ball among physical agents along with their results in a robotic soccer game.

Zubelzu J., de la Rosa J., Ramon J. A., and Quintero M. Christian G., "Managing Heterogeneity in a Robot Soccer Environment". **Frontiers in Artificial Intelligence and Applications – AI Research & Development**, vol. 113, pp. 317 – 322, IOS Press, ISSN: 0922- 6389, ISBN: I 58603 466 9, Amsterdam – Netherlands, 2004.

The development of a physical multi-agent system based on the concepts of heterogeneity, scenes or meetings among agents and introspection, using a robot soccer simulator is presented. The benefits of managing heterogeneous systems are also addressed. We claim diversity is desirable if it is managed by the agent introspection. The impact of managing heterogeneous multi-agent systems on performance is also analysed.

Wan J., Quintero M. Christian G., Luo N., and Vehí J., "Predictive Motion Control of a MiroSOT Mobile Robot". In Proceedings of the **World Automation Congress WAC 2004**, vol. 15, pp. 325 - 330, IEEE Catalog Number: 04EX832C, Seville – Spain, Jun. 28 - Jul. 1, 2004.

The dynamic model of a MiroSOT robot has been deduced with the consideration of the whole process including robot, vision, control and transmission systems. An integrated predictive control algorithm is proposed to control such complex dynamic system with either stationary or moving obstacle avoidance. Simulation results demonstrate the feasibility of such control strategy for the deduced dynamic model.

De la Rosa J., Quintero M. Christian G., and Vehí J., "About the Value of Introspection for Physically Grounded Commitments of Cooperative Intelligent Agents". In Proceedings of the V **Workshop on Physical Agents WAF 2004**, vol. 1, pp. 131 – 137, ISBN: 84-933619-6-8, Girona – Spain, Mar. 25 - 27, 2004.

How relevant is introspection for individual and cooperative agents' decisions in order to be aware of "to be or not to be able" to perform correctly a mission, task or set point is addressed. Interconnected neural networks sets are used as an implemented machine learning technique in the specific task of passing a ball among physical agents in the game of robotic soccer.

7.4 Future Research and Directions

In this thesis, introspection on control-grounded capabilities has been extensively studied in the robot soccer domain. The obtained results present some features that could extend to other domains and applications that may also benefit from the explicit knowledge on dynamics mentioned in this work. In this sense, preliminary results in other simulated test beds as teams of agent-controlled mobile robots in rescue environments and agent-controlled vehicles for automation driving in future highways are provided. However, extensive deepening in these case studies and more complex scenarios is necessary, principally through real experimentation in real cooperative environments.

The current efforts and results of this thesis show that the control-grounded capabilities notion and the agent introspection concept are very useful for cooperative control systems. However, it is still difficult to choose the necessary information to be included in the capabilities as well as the most suitable particular implementation technologies for imitating conscious-level reasoning in physical agents. As preliminary steps, this thesis reports some results obtained by using quite efficient machine learning and soft-computing techniques for building introspective agents. Such agents managed successfully their behaviour controllers by using the information included in their control-grounded capabilities associated with them. In spite of this, more extensive studies on these topics should be carried out to guarantee a better agent-oriented representation of the agents' inertial dynamics and the specifications, structure and other relevant details encapsulated in their automatic controllers.

Introspection on control-grounded capabilities has been studied in this thesis mainly in the context of the dynamics resulting from the actions of automatic controllers. Although it has been shown its relevance in the multi-agent performance especially in cooperative mobile robots, introspection on other interesting types of capabilities should be explored more in depth to enrich the agents' decision-making with self-knowledge related to other aspects of their physical bodies and control architectures.

In addition, studies on self-modeling, self-reference and proprioception could be carried out to increase and improve the physical agents' self-knowledge.

The thesis has also presented how introspection can contribute in a holistic manner when is integrated with other coordination parameters in the utility/cost functions for a more reliable agents' decision-making design. Nevertheless, one of the most interesting aspects to study in the future is how introspection on control-grounded capabilities will give new hints to engineers to design control systems. In the future, control engineers may keep KISS design in control systems of physical agents [De la Rosa et al., 07], by explicitly introducing introspection on control-grounded capabilities in such design in slightly natural way.

7.5 Concluding Remarks

This thesis argues the need for introspective reasoning on control-grounded capabilities in physically grounded agents to improve the agents' decision-making performance in both individual and cooperative decisions and close the gap between agents and the low abstraction level of automatic control architectures. Introspection on control-grounded capabilities allows agents to achieve sure and trustworthy commitments in cooperative systems, improving the performance of agents in coordinated tasks and task allocation problems. The thesis has shown how introspective skills help to prevent undesirable situations, to make safer decisions, to drive a better coordinated control and to obtain enhanced levels of performance and autonomy in any group of cooperating agents. In both individual and cooperative decisions about commitments between physical agents, the introspection allows an agent to know about its physical body's actual ability to perform all proposed coordinated tasks. Therefore, physical agents can behave intelligently when they negotiate commitments with other agents or humans. Here, intelligence is understood as the appropriate exploitation of knowledge about dynamics to perform better and achieve enhanced levels of performance and autonomy. In this sense, the thesis claims that introspection contributes to increasing the level of intelligence in physical agents by means of a suitable self-examination capability. In particular, with introspection, physical agents have a great deal of flexibility and self-control that make them more intelligent. Likewise, introspection is closely related to self-awareness. Self-aware control systems research aims at building systems that exhibit flexible, autonomous and goal-directed behaviors. The emergence of such behaviors is based on a deep understanding of the world and the self. Since introspection is related to self-knowing, it helps to fulfill the aims of self-aware control systems.

A representation based on capabilities related to the agent body's dynamics has been considered. These capabilities were managed in an introspective manner when agents were required to make a decision or to commit to the fulfillment of a task. Nevertheless, it is still difficult to choose the necessary information to include in the capabilities to represent control-oriented knowledge mainly related to dynamics. In spite of this, the experimental results have shown that introspection on control-grounded capabilities helps agents to make physically feasible decisions and to form sure, reachable and physically grounded commitments. Here, control-grounded capabilities were closely related to automatic controllers of physical agents. From the controllers, suitable information was extracted to obtain reliable control-oriented knowledge of the agent body's dynamics. There is still much to explore about how to take advantage of this approach. In particular, in the future, it is necessary to extend the contributions to other controlled systems. Furthermore, selection of a paradigm for the implementation of these concepts is not trivial at all, and its development is still an open question.

References

[Arai et al., 02] T. Arai, E. Pagello, and L. E. Parker. "Editorial: Advances in Multi-robot Systems". *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 665-661, 2002.

[Arkin, 98] R. C. Arkin. "Behavior-based Robotics". MIT Press, ISBN: 0262011654, Cambridge, MA, USA, 1998.

[Asada et al., 97] M. Asada, Y. Kuniyoshi, et al. "The Robocup Physical Agent Challenge". *First RoboCup Workshop in the XV International Joint Conference on Artificial Intelligence*, pp. 51-56, 1997.

[Balakirsky and Lacaze, 02] S. Balakirsky and A. Lacaze. "Value-Driven Behavior Generation for an Autonomous Mobile Ground Robot". In *Proc. of the SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, 2002.

[Belkhouche and Belkhouche, 05] F. Belkhouche and B. Belkhouche, "Modeling and Controlling a Robotic Convoy Using Guidance Laws Strategies". *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, no. 4, pp. 813–825, Aug. 2005.

[Beer et al., 99] M. Beer, M. d'Inverno, M. Luck, N. R. Jennings, C. Preist, and M. Schroeder. "Negotiations in Multi-agent Systems". *Knowledge Engineering Review*, vol. 14, no. 3, pp. 285-289, 1999.

[Beer, 00] R. D. Beer. "Dynamical Approaches to Cognitive Science". *Trends in Cognitive Sciences*, vol. 4, no. 3, pp. 91-99, 2000.

[Bolander, 03] T. Bolander. "Logical Theories for Agent Introspection". Ph.D. thesis, Informatics and Mathematical Modelling (IMM), Technical University of Denmark, 2003.

[Breemen, 01] A.J.N. van Breemen. "Agent-Based Multi-Controller Systems - A design framework for complex control problems". PhD thesis, Twente University Press, University of Twente, Enschede, The Netherlands, ISBN: 90 365 15955, 2001 (Supervised by J. van Amerongen with T.J.A. de Vries)

[Brooks, 91] R. A. Brooks. "Intelligence without Representation". *Artificial Intelligence,* vol. 47, pp.139-159, 1991.

[Burkhard et al., 02] H-D. Burkhard, D. Duhaut, M. Fujita, P. Lima, R. Murphy and R. Rojas. "The Road to RoboCup 2050". *IEEE Robotics and Automation Magazine*, vol. 9, no. 2, pp. 31-38, June, 2002.

[Busi et al., 01] N. Busi, P. Ciancarini, R. Gorrieri, G. Zavattaro. "Coordination Models: A Guided Tour". *Coordination of Internet Agents: Models, Technologies and Applications*, (Ominici, Zambonelli, Klusch and Tolksdorf, Editors), Springer-Verlag, 2001.

[Cao et al., 97] Y.U. Cao, A.S. Fukunaga, A.B. Kahng. "Cooperative Mobile Robotics: Antecedents and Directions". *Autonomous Robots*, vol.4, pp. 1-23, Kluwer Academic Publishers, Boston, 1997.

[Dahl et al., 04] T. S. Dahl, M. J. Matarić and G. S. Sukhatme. "Emergent Robot Differentiation for Distributed Multi-Robot Task Allocation". In *Proc. of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pp. 191-200, 2004.

[Davids, 02] A. Davids. "Urban Search and Rescue Robots: from Tragedy to Technology". *IEEE Intelligent Systems*, vol. 17, no. 2, pp. 81–83, Mar/Apr 2002.

[Dias and Stentz, 00] M. B. Dias and A. Stentz. "A Free Market Architecture for Distributed Control of a Multi-robot System". *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*, pp. 115 – 122, 2000.

[De la Rosa et al., 97] J. Ll. de la Rosa, A. Oller, et al., "Soccer Team Based on Agent-Oriented Programming", *Robotics and Autonomous Systems*, vol. 21, pp. 167–176, Oct 1997.

[De la Rosa et al., 00] J. Ll. de la Rosa, B. Innocenti, A. Oller, A. Figueras, J. A. Ramon, I. Muñoz and M. Montaner. "An Example of Dynamical Physical Agents". *1st European RoboCup Workshop*, Amsterdam, Holland, 2000.

[De la Rosa et al., 01] J. Ll. de la Rosa, I. Muñoz, B. Innocenti, M. Montaner, J. A. Ramon, A. Figueras. "RoGi Team Description", *RoboCup-2000*. LNAI 2019. Springer Verlag, pp. 551-554, Berlin, Germany, 2001.

[De la Rosa et al., 04] J. Ll. de la Rosa, M. C. Quintero and J. Vehí. "About the Value of Introspection for Physically Grounded Commitments of Cooperative Intelligent Agents". *Proc. of V Workshop on Physical Agents*, *edited by* J. de la Rosa and J. A. Ramon, pp. 131–137, 2004.

[De la Rosa et al., 07] J. Ll. de la Rosa, A. Figueras, C. Quintero, J. A. Ramon, S. Ibarra and S. Esteva. "Outline of Modification Systems". *Studies in Computational Intelligence,* Springer-Verlag, ISSN: 1860-949X, vol. 57, pp. 55-69, 2007.

[D'Inverno and Luck, 04] M. D'Inverno and M. Luck. "Understanding Agent Systems". 2nd Edition, Springer-Verlag, 2004.

[Duffy, 04] B. Duffy. "Robots Social Embodiment in Autonomous Mobile Robotics". *International Journal of Advanced Robotic Systems*, ISSN: 1729-8806, vol. 1, no. 3, pp. 155–170, 2004.

[Ferber, 99] J. Ferber. "Multi-Agent System: An introduction to Distributed Artificial Intelligence". Addison –Wesley, 1999.

[Ferguson, 92] I. A. Ferguson. "Touring Machines: An Architecture for Dynamic, Rational, Mobile Agents". Ph.D. Thesis, Clare Hall, University of Cambridge, UK, 1992.

[Gerkey and Matarić, 02] B. P. Gerkey and M. J. Matarić. "Sold!: Auction Methods for Multi-robot Coordination". *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758-768, 2002.

[Gerkey and Matarić, 04] B. P. Gerkey and M. J. Matarić. "A Formal Analysis and Taxonomy of Task Allocation in Multi-robot Systems". *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939-954, 2004.

[Goldberg and Matarić, 00] D. Goldberg and M. J. Matarić. "Robust Behavior-Based Control for Distributed Multi-Robot Collection Tasks". Technical Report IRIS-00-387, USC Institute for Robotics and Intelligent Systems. 2000.

[Goldberg et al., 03] D. Goldberg, V. Cicirello, M. B. Dias, R. Simmons, S. Smith, and A. T. Stentz. "Market-based Multi-robot Planning in a Distributed Layered Architecture". In *Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings from the 2003 International Workshop on Multi-Robot Systems*, Kluwer Academic Publishers, vol. 2, pp. 27-38, 2003.

[Halang et al., 05] W. Halang, R. Sanz, R. Babuska, and H. Roth. "Information and Communication Technology Embraces Control". *Status Report by the IFAC Coordinating Committee on Computers, Cognition and Communication, IFAC*, 2005.

[Horowitz and Varaiya, 00] R. Horowitz and P. Varaiya. "Control Design of an Automated Highway System". *Proc. of the IEEE,* vol. 88, no. 7, pp. 913-925, Feb. 2000.

[Innocenti et al., 01] B. Innocenti, J. Ll. de la Rosa, J. Vehí, I. Muñoz, M. Montaner, M. Fàbregas, A. Figueras and J. A. Ramon. "Examples of Dynamical Physical Agents for Multi Robot Control". *2do Workshop Hispano-Luso de Agentes Físicos*, Madrid, Spain, 2001.

[Jennings et al., 98] N. R. Jennings, K. Sycara and M. Wooldridge. "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems Journal*, N.R. Jennings, K. Sycara and M. Georgeff (Eds.), Kluwer Academic Publishers, Boston, vol. 1, no. 1, pp. 7-38, 1998.

[Jennings, 00] N. R. Jennings. "On Agent-Based Software Engineering". Artificial Intelligence, vol. 117, no. 2, pp. 277-296, 2000.

[Jennings, 01] N. R. Jennings. "An Agent-based Approach for Building Complex Software Systems". *Communications of the ACM*, vol. 44, no. 4, pp. 35-41, 2001.

[Jennings and Bussmann, 03] N. R. Jennings and S. Bussmann. "Agent-Based Control Systems. Why Are They Suited to Engineering Complex Systems?". *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 61-73, Jun. 2003.

[Kim and Vadakkepat, 00] J. H. Kim and P. Vadakkepat. "Multi-Agent Systems: A Survey from the Robot-soccer Perspective". *International Journal of Intelligent Automation and Soft-computing,* vol. 6, no. 1, pp. 3–17, 2000.

[Koenig et al., 06] S. Koenig, C. Tovey, M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson and S. Jain. "The Power of Sequential Single-Item Auctions for Agent Coordination". In *Proc. of the Twenty-First National Conference on Artificial Intelligence,* 2006.

[Lagoudakis et al., 05] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson and S. Jain. "Auction-based Multi-robot Routing". In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, 2005.

[Lesser, 99] V. R., Lesser. "Cooperative Multi-agent Systems: a Personal View of the State of the Art". *IEEE Transactions on Knowledge and Data Engineering,* vol. 11, no. 1, pp. 133–142, Jan/Feb 1999.

[Low et al., 02] K. H. Low, W. K. Leow and M. H. Ang Jr. "A Hybrid Mobile Robot Architecture with Integrated Planning and Control". In *Proceedings of 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, pp. 219-226, Bologna, Italy, Jul 15-19, 2002.

[Luck et al., 03] M. Luck, P. McBurney and C. Preist. "Agent Technology: Enabling Next Generation Computing". *A Roadmap for Agent-Based Computing,* ISBN: 0854327886, ver. 1.0. Southampton: AgentLink 2003.

[Luck et al., 05] M. Luck, P. McBurne, O. Shehory and S. Willmott. "Agent Technology: Computing as Interaction". *A Roadmap for Agent Based Computing, Compiled, written and edited by* M. Luck, P. McBurney, O. Shehory, S. Willmott and the AgentLink Community, pp. 11-12, 2005.

[Mallya et al., 03] A. U. Mallya, P. Yolum, and M. P. Singh. "Resolving Commitments among Autonomous Agents". *Proceedings of the International Workshop on Agent Communication Languages and Conversation Policies (ACL)*, Melbourne, July 2003.

[Malone and Crowston, 94] T. Malone and K. Crowston. "The Interdisciplinary Study of Coordination". *Computing Surveys,* vol. 26, no. 1, pp. 87-119, 1994.

[Mas et al., 05] A. Mas. "Agentes Software y Sistemas Multi-agente: Conceptos, Arquitecturas y Aplicaciones". ISBN: 84-205-4367-5, Pearson Educación S. A., Madrid, España, 2005.

[Matarić, 99] M. J. Matarić. "Behavior-Based Robotics". *The MIT Encyclopedia of the Cognitive Sciences,* MIT Press, pp.74-77, 1999.

[Matellán and Borrajo, 01] V. Matellán and D. Borrajo. "An Agenda Based Multi-Agent Model for Robots Control and Cooperation". *Journal of Intelligent Robotic Systems,* ISSN: 0921-0296, vol. 32, no. 1, pp. 93-114, 2001.

[Mayoh, 02] B. Mayoh. "Evolution of Cooperation in Multi-agent Systems". *EurAsia-ICT 2002: Information and Communication Technology: First EurAsian Conference,* LNCS, Ed. Springer-Verlag, ISSN: 0302-9743, Shiraz, Iran, pp. 701-710, October 29-31, 2002.

[McCarthy, 99] J. McCarthy. "Making Robots Conscious of Their Mental States". *Machine Intelligence 15, Intelligent Agents*, St. Catherine's College, Oxford, July 1995, pp.3-17, January 1999.

[Molina et al., 04] J. Molina, J. García and A. Bernardos. "Agentes y Sistemas Multi-agente". Departamento de Informática, Universidad Carlos III de Madrid, Centro de Difusión de Tecnologías, Universidad Politécnica de Madrid, CEDITEC 2004.

[Müller, 97] J.P. Muller. "The Design of Intelligence Agents. A Layered Approach". *Lectures Notes in Artificial Intelligence*, vol. 1177, Ed. Springer-Verlag, 1997.

[Murphy, 04] R.R. Murphy. "Human–robot Interaction in Rescue Robotics". *IEEE Transactions on Systems, Man and Cybernetics, Part C,* vol. 34, no. 2, pp. 138–153, May 2004.

[Murray et al., 03] R. Murray, K.J. Astrom, S. Boyd, R.W. Brockett and G. Stein. "Future Directions in Control in an Information-Rich World". *IEEE Control Systems Magazine*, vol. 23, no. 2, pp. 20-33, Apr. 2003.

[Novak, 02] G. Novak. "Robot Soccer - An Example for Autonomous Mobile Cooperating Robots". *Proceedings of the First Workshop on Intelligent Solutions in Embedded Systems* (WISES), University of Technology, Austria, June 27th, 2003.

[Oller et al., 97] A. Oller, J. de la Rosa, R. García, J.A. Ramon, and A. Figueras, "Micro-Robots Playing Soccer Games: A Real Implementation Based on Multi-Agent Decision-Making Structure", *Int. Journal of Intelligent Automation and Soft Computing, Special Issue on Soccer Robotics*, vol. 6, no. 1, pp. 65-74, Jan. 1997.

[Oller et al., 99] A. Oller, J. Ll. de la Rosa and E. Del Acebo. "DPA^2: Architecture for Cooperative Dynamical Physical Agents". *9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW '99.* Valencia, Spain, 1999.

[Oller and Garcia, 02] A. Oller and M.A. Garcia. "Using a Linear Model to Evaluate Performance of Non-holonomic Paths: the Shooting-ball as a Case Study". *FIRA Robot World Congress,* ISBN: 89-86522-47-0-9356, pp. 643-647, Seoul, Korea, 2002.

[Oller, 02] A. Oller. "Disseny d'Agents Físics: Inclusió de Capacitats Específiques per a l'Avaluació de l'Eficiència d'Accions", Ph.D. Thesis, 2002.

[Panait and Luke, 05] L. Panait and S. Luke. "Cooperative Multi-Agent Learning: The State of the Art". In *Autonomous Agents and Multi-Agent Systems,* Ed. Springer-Verlag, vol. 11, no. 3, pp. 387-434, 2005.

[Parent and de la Fortelle, 05] M. Parent, A. de La Fortelle. "Cybercars: Past, Present and Future of the Technology". *World Congress on Intelligent Transport Systems 2005*, San Francisco, USA, 2005.

[Parker, 00] L. E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics". *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen eds., Springer-Verlag, pp. 3-12, 2000.

[Ossowski, 99] S. Ossowski. "Coordination in Artificial Agent Societies". LNAI1535, Ed. Springer-Verlag, 1999.

[Quintero et al., 04] C. Quintero, J. de la Rosa and J. Vehí. "Physical Intelligent Agents' Capabilities Management for Sure Commitments in a Collaborative World". *Frontiers in Artificial Intelligence and Applications*, IOS Press, ISBN: I 58603 466 9, ISSN: 0922-6389, pp. 251-258, 2004.

[Rao and Georgeff, 95] A. S. Rao and M. P. Georgeff. "BDI Agents from Theory to Practice". *Proceedings of the First International Conference on Multi-agent Systems ICMAS95*, 1995.

[Ramos et al., 06] N. Ramos, P. Lima and J. M. Sousa. "Robot Behavior Coordination Based on Fuzzy Decision-Making". In *Actas do Encontro Cientifico do Robotica*, 2006.

[Rocher and Duhaut, 98] S. Rocher, D. Duhaut. "MARCH: A Flexible Multi-Agent Architecture, Applied to Autonomous Robots Playing Football". *In Proceedings of the First Collective Robotics Workshop*, ISBN: 3-540-64768-6, pp. 98 – 109, July 04 – 05, 1998.

[Sanz et al, 00] R. Sanz, F. Matía and S. Galán. "Fridges, Elephants, and the Meaning of Autonomy and Intelligence". *IEEE International Symposium on Intelligent Control (ISIC)*, pp. 217-222, Jul. 2000.

[Sanz et al., 01] R. Sanz, J. Escasany and I. López. "Systems and Consciousness". *Proc. Conf. Towards a Science of Consciousness (TSC)*, 2001.

[Sanz et al., 03] R. Sanz, A. Yela and R. Chinchilla. "A Pattern Schema for Complex Controllers". *IEEE Emergent Technologies for Factory Automation (ETFA)*, pp. 101-105, Sept. 2003.

[Sanz et al., 04] R. Sanz, O. Holland, A. Sloman, A. Kirilyuk, W. Edmondson and S. Torrance. "Self-aware Control Systems". *Research Whitepaper for the Bio-inspired Intelligent Information Systems Call*, 2004.

[Sariel and Balch, 06] S. Sariel and T. Balch. "Efficient Bids on Task Allocation for Multi Robot Exploration". In *Proc. of the 19th International FLAIRS Conference*, 2006.

[Scerri et al., 04] P. Scerri, R. Vincent and R. Mailler. "Comparing three Approaches to Large Scale Coordination". In *AAMAS'04 Workshop on Challenges in the Coordination of Large Scale Multi-Agent Systems*, 2004.

[Scheutz, 02] M. Scheutz. "Affective Action Selection and Behavior Arbitration for Autonomous Robots". In *Proc. of IC-AI 02*, 2002.

[Shoham, 93] Y. Shoham. "Agent-oriented programming". *Artificial Intelligence*, vol. 60, no. 1, pp. 51-92, 1993.

[Stone, 00] P. Stone. *Layered Learning in Multi-agent Systems: A Winning Approach to Robotic Soccer*. MIT Press, pp. 105–115, 2000.

[Stone and Veloso, 00] P. Stone and M. Veloso, "Multi-agent Systems: A Survey from a Machine Learning Perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, July 2000.

[Tadokoro et al., 00] S. Tadokoro, H. Kitano, T. Takahashi, I. Noda, H. Matsubara, A. Shinjoh, T. Koto, I. Takeuchi, H. Takahashi, F. Matsuno, M. Hatayama, J. Nobe, S. Shimada. "The RoboCup-Rescue Project: A Robotic Approach to the Disaster Mitigation Problem". *Proceedings of 2000 IEEE International Conference on Robotics and Automation (ICRA 2000)*, April, 2000.

[Watson et al., 02] R.A. Watson, S.G. Filici and J.B. Pollack. "Embodied Evolution: Distributing an Evolutionary Algorithm in a Population of Robots". *Robotics and Autonomous Systems,* vol. 39 pp. 1–18, 2002.

[Wegner, 97] P. Wegner. "Why Interaction is more Powerful than Computing". *Communications of the ACM,* vol. 5, no. 40, pp. 80-91, 1997.

[Weiss, 99] G. Weiss. "Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence". Edited by Gerard Weiss. ISBN: 0-262-23203-0, 1999.

[Wilson and Keil, 01] R. A. Wilson and F. C. Keil (Eds.) "The MIT Encyclopedia of the Cognitive Sciences". Cambridge, MA: The MIT Press, ISBN: 0-262-73144-4, 2001.

[Wooldridge, 02] M. Wooldridge. "An Introduction to Multi-agent Systems". Published in February 2002 by John Wiley & Sons (Chichester, England), ISBN: 0 47149691X, 2002.

[Zubelzu et al., 04] J. Zubelzu, J. de la Rosa, J. A. Ramon and C. Quintero. "Managing Heterogeneity in a Robot Soccer Environment". *Frontiers in Artificial Intelligence and Applications,* IOS Press, ISBN: I 58603 466 9, ISSN: 0922-6389, pp. 317-322, 2004.