Robotic Soccer as a Multi-Agent System: Literature Review

Yanli Yang and Marios M. Polycarpou

Department of Electrical and Computer Engineering and Computer Science
University of Cincinnati, Cincinnati, OH 45221-0030, USA

September 15, 2000

1 Introduction

Multi-agent system is a sub-field of Artificial Intelligence (AI) that aims to provide both principles for construction of complex systems involving multiple agents and mechanisms for coordination of agents' behaviors. An agent can be considered to be an entity, such as robot, with goals, actions, and domain knowledge, situated in an environment. The way it acts is called its *behavior* [18]. Although the ability to consider coordinating behavior of autonomous agents is a new research direction, the field is advancing quickly by building upon preexisting work in the field of Distributed Artificial Intelligence (DAI) [5].

Increasingly, multi-agent systems are being designed for a variety of complex, dynamic domains including autonomous vehicles and even some human agents. Effective agent interactions in such domains raise some of the most fundamental research challenges for agent-based systems. An agent in such domains must model other agents behaviors, learn/adapt from its interactions, form teams and act effectively in a team, negotiate with other agents, and so on. For each of these research problems, the uncertainty and the presence of multiple cooperative and non-cooperative agents, conspires to exacerbate the difficulty.

Consider the challenge of designing multi-agent teamwork, which has become a critical requirement across a wide range of multi-agent domains. In this case, an agent team must address the challenge of designing roles for individuals (i.e., dividing up team responsibilities based on individuals capabilities), doing so with fairness, and reorganizing roles based on new information. Furthermore, agents must also flexibly coordinate and communicate, so as to be robust despite individual members incomplete and inconsistent view of the environment, and despite unexpected individual failures. Learning in a team context also remains a difficult challenge.

Robotic Soccer was proposed to be a general test-bed for studying multi-agent system techniques. Originated by Mackworth [13], it has been gaining popularity in recent years, with several international competitions in several different leagues [15, 8, 2]. The simulation league is of particular interest and attracts the largest number of participants. The stated research goals of the simulation league are to investigate the areas of multi-agent teamwork, agent modeling, and multi-agent learning [10]. It can be used to evaluate different multi-agent system techniques in a direct manner: teams implemented with different techniques can play against each other.

2 Robotic Soccer System Overview

The first robotic soccer system was the Dynamo system [4]. Barman et al. built a 1 vs. 1 version of the game. Now there are mainly three kinds of competitions in Robotic Soccer:

- **Real Robot League:** Using physical robots to play soccer games.
- **Software Agent League:** Using software or synthetic agents to play soccer games on an official soccer server over the network.
- **Expert Skill Competition:** Competition of robots which have special skills, but are not able to play a game.

As discussed above, robotic soccer can be played either with real robots or in a simulator. Of particular interest in this review is the simulation robotic soccer. A particularly good simulator is the soccer server developed by Noda [14]. The simulator was first used for a competition among twenty-nine teams from around the world in 1997 [10] and continues to be used for this purpose currently.

This simulator server simulates the players bodies, the ball and the environment (e.g., the soccer field, flags, etc). Software agents provide the *brains* for the simulated bodies. Thus, 22 agents, who do not share memory, are needed for a full game. Visual and audio information as the sensory perception *sensed* by the player body is sent to the player agent (*brain*), which can then send action commands as actuators to control the simulated body (e.g., kick, dash, turn, say, etc.). The server constrains the actions an agent can take and the sensory information it receives. For instance, with the server used in the 1997 competition, a player could only send one action every 100 milliseconds and receive perceptual updates every 300 milliseconds. The server also simulates stamina: If a player has been running too hard, it gets *tired*, and can no longer dash as effectively. Both actions and sensors contain a noise factor, and so are not perfectly reliable. The quality of perceptual information depends on several factors, such as distance, view angle, and view mode (approximating visual focus). All communications between players are done via the server, and are subject to limitations such as bandwidth, range and latencies. So simulated robotic soccer with multiple agents on each team is a good representative test-bed for multi-agent domain with four characteristics: real-time, noisy, collaborative and adversarial.

3 Technical Challenges in Robotic Soccer

Robotic Soccer is an attempt to promote AI and robotics research by providing a common task, Soccer, for evaluation of various theories, algorithms, and agent architectures [10, 9]. Starting with the first competitions held in 1996 (Pre-Robotcup-96 and Mirosot-96) and continuing since then, there has been a great deal of robotic soccer-related research. It has been presented both at dedicated robotic soccer workshops held in conjunction with the competition and in other scientific forums.

For an agent (a physical robot or a synthetic agent) to play soccer reasonably well, a wide range of technologies need to be integrated. The range of technologies spans both AI and robotics research, such as design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning and planning, intelligent robotics, sensor fusion, and so forth.

Robotic Soccer challenges can be organized into three major classes [10]:

1. Synthetic Agent Challenge;
2. Physical Agent Challenge;
3. Infrastructure Challenge.

The synthetic agent challenge deals with technologies involving multi-agent control and robotic soccer strategies, which can be developed using software simulator. The research issues in this aspect involve: learning challenge; teamwork challenge; opponent challenge.

The physical agent challenge intends to promote research using real robots. Details of this challenge were described in [3]. The infrastructure challenge is to facilitate research to establish infrastructure aspect of Robot Soccer, AI, and robotics in general. Such challenge includes education programs, common robot platforms and components, standard automatic commentary systems and intelligent studio systems for robot soccer games.

4 Multi-agent Control and Robotic Soccer Strategy

In this section, some of the multi-agent control and robotic soccer strategies are reviewed. Particular emphasis is placed on techniques that are closely related to cooperative control of multiple unmanned air vehicles (UAVs).

The fundamental issue for researchers who wish to build a team for robotic soccer is to design a multi-agent system that behaves in real-time, performing reasonable goal-directed behaviors. Goals and situations change dynamically and in real-time. This domain has inspired many different approaches to building and organizing teams of agents.

4.1 Layered Learning Method

Team CMUnited [22] won simulator league champions in yearly competitions. Their success depended on their novel multi-agent techniques to achieve adaptive coordination: layered learning [19] and a flexible teamwork structure (Locker-Room Agreement) [20].

Layered learning is a general-purpose machine learning paradigm for complex domains in which learning a mapping directly from agents' sensors to their actuators is intractable. Given a hierarchical task decomposition, layered learning allows for learning at each level of the hierarchy, with learning at each level directly affecting learning at the next higher level. For example, in the robotic soccer domain, they link the following three learned layers:

- Neural networks are used by individual players to learn how to intercept a moving ball.
- With the receivers and opponents using this first learned behavior to try to receive or intercept passes, a decision tree (C4.5) is used to learn the likelihood that a given pass would succeed.
- TPOT-RL (Team-Partitioned, Opaque-Transition Reinforcement Learning) algorithm [21], a new machine learning method used to train collaborative and adversarial team behavior, is used to learn pass selection, taking advantage of the learned pass-evaluation capability to construct the input representation for learning. However, due to the system complexity, the results haven't been applied directly.

This approach characterizes robotic soccer as an instance of a class of domains called Periodic Team Synchronization (PTS) domains. In this class of domains, a team of agents has periodic opportunities to communicate fully in a safe, off-line situation (i.e. in the *locker-room*). To deal with the challenges of PTS domains, they introduce the concept of a Locker-Room Agreement by which agents determine ahead of time their communication language, their sensory triggers for changes in team strategy, and some multi-agent plans for predictable situations.

In CMUnited, the locker-room agreement includes a flexible team structure that allows homogeneous agents to switch roles (positions such as defender or attacker) within a single formation. It also allows the entire team to switch formations (for instance from a defensive to an offensive formation) based on agreed-upon sensory triggers. For example, CMUnited began all of its games in a 4-3-3 formation (4 defenders, 3 midfielders, 3 forwards). However, if they had ever found themselves losing near the end of the game, they would have smoothly switched to a formation with fewer defenders and more forwards. In the actual competition, they often switched to a defensive formation with additional defenders and fewer forwards once they were safely in the lead.

Scerri [16] presents another multi-layered approach to robotic soccer. However, unlike Stone's hierarchical approach, it does not involve the learning of any behaviors. In this approach, the different abstraction layers deal with different granularities of sensory input. For example, a low-level move-to-ball behavior is given the ball's precise location, while a high-level defend behavior-which might call go-to-ball-knows only that the ball is in the defensive half of the field.

4.2 Teamwork Model-based Method

In terms of work within robot soccer, ISIS Team is the only team that investigated the use of a general, domain-independent teamwork model to guide agent's communication and coordination in teamwork.

The ISIS team [23] uses a role-based approach to robotic soccer based on STEAM, a general, explicit model of teamwork to enable teamwork among player agents. This general model is motivated by the need for flexibility in team activities, as well as reuse of teamwork capabilities across different domains. STEAM requires that individual team members explicitly represent their team's goals, plans and mutual beliefs. It then enables team members to autonomously reason about coordination and communication in teamwork, providing improved flexibility. With respect to multi-agent learning, ISIS used C4.5 to train ISIS players learned off-line to choose an intelligent kicking direction, avoiding areas of concentration of opponent players.

Several other researchers investigating teamwork in robot soccer use explicit team plans and roles, but they rely on domain-dependent communication and coordination. A typical example includes work by Ch'ng and Padgham [6]. In this scheme, agents dynamically adopt and abandon roles in the predefined tactics. The responsibilities and actions of each agent are determined by its current role in the current plan.

4.3 Programming Methodologies-based Method

Some research is based on applying existing programming methodologies to the robotic soccer domain.

Team ROGI [11] is built using agent-oriented programming (AOP)[17] by means of Matlab/Simulink, a widely known computer aided control system design framework. AOP can be intuitively viewed as a specialization of object-oriented programming (OOP). On the other hand, AOP specializes the OOP framework by extending the state of the objects that are considered as agents with mental state (which consist of components such as beliefs, capabilities, and decisions). The agent-oriented paradigms formalize interactions between multiple agents in terms of changing their mental states by communication between agents. The reasoning procedure is developed in three steps:

1. Each agent decides its own reactive action depending on its position on the ground and the relative situation of the ball. Then they inform other agents of their decision.

2. Each agent decides its cognitive action. Agents get new information and take new decisions (cooperative-cognitive ones) that have higher degree of certainty than the reactive ones. Then they inform the coach-agent.

3. Individual decisions of each agent are criticized by the coach-agent and converted into actions by selecting from proposals of soccer agents.

This system could be used as a first step in using agent techeniques in automatic control and robotics by using tools (Matlab) that are common in the control area. The drawback is that because of the complexity of the problem, object-oriented paradigms are limited to simulated studies and are typically not suitable for practical implementation.

Other researchers introduced new multi-agent control methodologies and applied them to robotic soccer. For example, the MICROB robotic soccer team is an implementation of the Cassiopeia programming method [7] whose purpose is to provide a methodological framework to design multi-agent systems. It assumes that although the agents can have different aims, the goal of the designer is to make them behave cooperatively. Cassiopeia focuses on the organizational issues of multi-agent tasks. According to this method a multi-agent system should be designed in terms of agents provided with three levels of behavior: elementary behavior, relational behavior and organizational behavior. Based on analyzing the independence of the elementary behavior, facilitates the formation of groups by specifying the organizational behavior. For example, in the MICROB soccer robot system, the player with the ball is analyzing the inter-dependencies of low-level skills and facilitating the formation of groups based on these inter-dependencies. The player may contract with another player to place itself in a particular location to receive a pass. This approach differs from that of the CMUnited where the agents position themselves autonomously, and the agent with the ball decides autonomously where to pass: no negotiation is involved, enabling the players to act as quickly as possible.

The results in the robot soccer competition has proved that the Cassiopeia programming method is definitely useful as a method for designing multi-agent systems since it allows one to evaluated various types of agent architectures without revising the analysis choice. It also has shown its limits: has no reactive and learning ability.

4.4 Evolutionary Method

All of the learning approaches described above are used to learn portions of an agent's behavior. Other aspects are created manually. In contrast, a few entirely learned soccer behaviors have been created.

Luke et al. [12] use genetic programming to build agents that learn to use their basic individual skills on coordination. Their goal was to use genetic programming (GP) to evolve high-level decision behaviors for an entire team of robot soccer. Genetic programming is an evolutionary computational method which searches for the most fit program for a given problem (in this case, operating a soccer strategy in the robot soccer server). To do this, genetic programming system was supplied with a set of low-level atomic functions designed for the soccer environment. The GP system used this function set to build its programs. In this approach, program trees did not represent individual players but entire teams.

Darwin United [1] is a team that has been evolved as a team of coordinated agents in the robot soccer simulator. They use a modified genetic programming paradigm to solve the complex problem of designing teams for robot soccer. Different from Luke et al, they introduce a graduated fitness function that tests each individual for increasing levels of skill. For example, before evolving teams are allowed to compete with one another for representation in the next generation, they must first pass three competition filters, the first of which is simply the ability to score on an empty field.

In both cases, the goal was to learn entirely from agents sensors to actuators in the soccer server. The competition results show that the genetic programming can be used successfully as a technique for training a team using the basic percepts and actions of the simulator. The main problem is that GP is remarkably slow to learn generalizable routines.

5 Conclusion

Robotic Soccer is a useful domain for the study of multi-agent systems. This review describes control strategies in the field of robotic soccer from various multi-agent system viewpoints. It provide an introduction to people unfamiliar with the field and gives an organized overview of the research in this area.

The goal of our project is to develop and evaluate the performance of strategies for Distributed Cooperation and Control for Autonomous Air Vehicles. Some of the common technical challenges between robotic soccer and control of autonomous air vehicles include the following:

1. Machine learning in a multi-agent, collaborative and adversarial environment;
2. Multi-agent architectures, enabling real-time multi-agent planning and plan execution;
3. Opponent modeling.

In general, understanding the progress in research for robotic soccer will also be helpful in designing distributed cooperation and control strategies for multiple autonomous air vehicles.

References

[1] D. Andre and A. Teller. Evolving team Darwin United. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, pages 373–388. Springer Verlag: Berlin, 1999.

[2] M. Asada and H. Kitano. Lecture notes in artificial intelligence. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, pages 373–388. Springer Verlag: Berlin, 1999.

[3] M. Asada, M. Kuniyoshi, A. Drogoul, M. Mataric H. Asama, D. Duhout, P. Stone, and H. Kitano. The robocup physical agent challenge: Phase-I. *Applied Artificial Intelligence(AAI)*, 12:251–263, 1998.

[4] R. A. Banman, S. J. Kingdon, J. J. Mackworthk, A. K. Pai, D. K. Sahota, M. Wilkinson, and Y. Zhang. Dynamo: Real-time experiments with multiple mobile robots. In *Intelligent Vehicles Symposium*, pages 261–266. Tokyo, 1993.

[5] A. H. Bond and L. Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–55. Morgan Kaufmann Publishers: San Mateo, CA, 1988.

[6] S. Ch'ng and L. Padgham. From roles to teamwork: A framework and architecture. *Applied Artificial Intelligence*, 12:211–232, 1998.

[7] A. Drogoul and A. Collinot. Applying an agent-oriented methodology to the design of artificial organizations: A case study in robotic soccer. *Autonomous Agents and Multi-Agent Systems*, 1:113–129, 1998.

[8] J. H. Kim. Path planning for multi-mobile robots in the dynamic enviroment. In *Proceedings of the Micro-Robot World Cup Soccer Tournament*, Taejon, Korea, 1996.

[9] H. Kitano. Robocup: A challenge problem for AI. *AI Magazine*, 18:340–347, 1997.

[10] H. Kitano, M. Asada, E. Osawa, I. Noda, Y. Kuniyoshi, and H. Matsubara. Robocup: The robot world cup initiative. In *Proceedings of The First International Conference on Autonomous Agent(Agent-97)*, 1997.

[11] J. Li, De La Rosa, A. Oller, J. Vehi, and J. Puyol. Soccer team based on agent-oriented programming. *Robotics and Autonomous Systems*, 21:161–176, 1997.

[12] S. Luke, C. Hohn, J. Farris, G. Jackson, and J. Hendler. Co-evolving soccer softbot team coordination with genetic programming. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 398–411. Springer Verlag: Berlin, 1998.

[13] A. K. Mackworth. *On seeing robots*. World Scientific Press: Singapore, Computer Vision: Systems, Theory, and Aapplications, 1993.

[14] I. Noda. Team gamma: Agent programming on goal. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 500–507. Springer Verlag: Berlin, 1998.

[15] RoboCup. Robocup web page. In *URL http://www.robocup.org*.

[16] P. Scerri. A multi-layered behavior based system for controlling robocup agents. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 467–474. Springer Verlag: Berlin, 1998.

[17] Y. Shoham. *Agent-oriented programming*. Computer Science Dept., Stanford University, Technical Report CS-1335-990, 1990.

[18] P. Stone. *Layered learning in multi-agent systems: A winning approach to robotic soccer*. MIT Press, Cambridge, MA, 2000.

[19] P. Stone and M. Veloso. Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, 48:83–104, 1998.

[20] P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence (AIJ)*, 110:241–273, 1999.

[21] P. Stone and M. Veloso. Team-partitioned, opaque-transition reinforcement learning. In *RoboCup-98: Robot Soccer World Cup II*, Springer Verlag, Berlin, 1999.

[22] P. Stone, M. Veloso, and P. Riley. The CMUnited-98 champion simulator team. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag: Berlin, 1999.

[23] M. Tambe, J. Adibi, Y. Al. Onaizan, A. Kaminka, S. C. Marsela, I. Musea, and M. Tallis. Using an explicit model of teamwork in robocup-97. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 123–131. Springer Verlag: Berlin, 1998.