Game Strategies for Physical Robot Soccer Players: A Survey

Emanuele Antonioni, Vincenzo Suriani, Francesco Riccio, and Daniele Nardi

***Abstract*—Effective team strategies and joint decision-making processes are fundamental in modern robotic applications, where multiple units have to cooperate to achieve a common goal. The research community in artificial intelligence and robotics has launched robotic competitions to promote research and validate new approaches, by providing robust benchmarks to evaluate all the components of a multiagent system—ranging from hardware to high-level strategy learning. Among these competitions *RoboCup* has a prominent role, running one of the first worldwide multi-robot competition (in the late 1990s), challenging researchers to develop robotic systems able to compete in the game of soccer. Robotic soccer teams are complex multirobot systems, where each unit shows individual skills, and solid teamwork by exchanging information about their local perceptions and intentions. In this survey, we dive into the techniques developed within the *RoboCup* framework by analyzing and commenting on them in detail. We highlight significant trends in the research conducted in the field and to provide commentaries and insights, about challenges and achievements in generating decision-making processes for multi-robot adversarial scenarios. As an outcome, we provide an overview a body of work that lies at the intersection of three disciplines: Artificial intelligence, robotics, and games.**

***Index Terms*—Robotic competition, soccer robots, strategies in robotic games.**

Fig. 1. *RoboCup* competitions are organized in different leagues deploying various robotic platforms.

I. Introduction

ARTIFICIAL intelligence (AI) and games always had a very tight connection. Games, in fact, are one of the most creative and demanding activity for the human brain as they require reasoning, planning (PL), and often, intuition [1], [2]. For this reason, developing artificial agents able to compete in games—and eventually challenge humans—has always been one of the most appealing objectives for AI research. A recent breakthrough in this area has been made by AlphaGO [2], a deep-learning (DL) algorithm that won four out of five games against the world champion in the game of *GO*. In a different context, the virtual environment of the strategic videogame *DOTA2*,[1] the authors of [3] present an approach able to strategically operate a team of simulated agents, winning 99.4% of the 7000 matches played against professionals. The team of virtual agents showed emergent coordinated behaviors within a complex action-space and a continuous state-space.

While research in AI and games on virtual agents is loudly making progress, in this article, we focus on games of physical agents (i.e., robot soccer players). Acting in a physical world adds complexity, since perceptions are limited and error-prone, and actions can fail in unpredictable ways. Specifically, we investigate how physical interactions with the environment influence game strategies.

The development of a fully functional robot able to understand and interact with the real environment is a very difficult task regardless of the application. To promote the challenge and motivate researchers to attack such a problem holistically, different word-wide robotic competitions have been started during the years. Among those, *RoboCup* yearly organizes competitions to challenge researchers to develop robotic systems able to compete in the game of soccer [4].

Competitions are organized in different leagues, each of which features a specific kind of robotic platform and has its own research challenges (see Fig. 1). Such a framework allows

Manuscript received December 31, 2019; revised January 21, 2021; accepted April 9, 2021. Date of publication April 27, 2021; date of current version December 15, 2021. *(Emanuele Antonioni and Vincenzo Suriani contributed equally to this work.) (Corresponding author: Vincenzo Suriani.)*

The authors are with the Department of Computer, Control, and Management Engineering Antonio Ruberti, Sapienza University of Rome, 00185 Rome, Italy (e-mail: antonioni@diag.uniroma1.it; suriani@diag.uniroma1.it; riccio@diag.uniroma1.it; nardi@diag.uniroma1.it).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TG.2021.3075065.

Digital Object Identifier 10.1109/TG.2021.3075065

[1] https://openai.com/projects/five/

researchers to attack the problem of robot soccer from different perspectives, by splitting it into easier subtasks. Sports, in general, are excellent testbeds to validate and improve robotic systems. In soccer games, players continuously coordinate their plays, change tactics, and adapt to the opponent. In order to play soccer, in *RoboCup*, the same capabilities should be achieved by robots that act autonomously by:

1) collecting sensor observations;
2) maintaining a representation of the state of the game;
3) reasoning and PL their actions individually and collectively.

In this work, we survey recent developments in the context of *RoboCup* soccer, by collecting research contributions from 2015 to 2019. Our main sources are the *RoboCup* Symposium Proceedings [5], yearly published by Springer-Verlag. We focus on different approaches proposed to tackle the problem of individual and collective game strategies. We peruse the literature by exposing different trends in the last five years, highlighting how decision-making techniques are influenced by hardware limitations, the size of the team, and the task to address. By targeting recent research in robot soccer, our aim is to assess the current state of a research body lying at the intersection among AI, robotics, and games.

The remainder of this article is as follows. Section II recalls the *RoboCup* origins and its structure in subleagues. Section III formalizes the problem of robot behavior generation. Section IV presents our survey of the existing literature categorizing it and highlighting each step needed to achieve strategic team play in robot soccer. Section V discusses the surveyed papers relating proposed technologies and the hardware support. Finally, Section VI summarizes this article, recapitulates its key findings, and concludes by point to open research questions and future directions.

II. Soccer Game in *RoboCup*

The idea of using soccer as testbed for AI and robotics application has its origin in the paper by Alan Mackworth, 1992, who was already very active in the field publishing several contributions about Dynamo robot soccer [6]. Right after, a group of researchers started shaping the structure of *RoboCup*, whose first official competition was held in 1997. Over 40 teams participated with more than 5000 spectators. Nowadays, *RoboCup*, which is run by the *RoboCup* Federation, is a large event with more than 1500 senior participants (researchers and university students) and 200 teams (not including *RoboCup* Junior, which is devoted to foster AI and robotics among the young generations). Besides robot soccer, which is the main challenge, and is detailed as follows, *RoboCup* includes other robotic competitions inspired by search and rescue, home, and industrial robots, aiming at the transfer of novel approaches and techniques into application domains. However, in this article, we focus on robot soccer only.

Like other AI challenges, also *RoboCup* has a vision of competing against human, in fact “By the year 2050, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official FIFA rules, against the winner of the most recent World Cup of Human Soccer.”

However, since the beginning, *RoboCup* realized that the problem of developing fully autonomous robots playing sports cannot be solved at once, and it needs to be split into subproblems. To this end, the competitions have been organized in different leagues attacking different research aspects, ranging from hardware design and actuation to complex collective behaviors and opponent analysis (OA). Moreover, the setup of each league is constantly updated to reflect the progress and introduce new challenges incrementally.

In this section, we briefly recall the structure of *RoboCup* and its scientific challenges, providing an overview of *RoboCup* soccer leagues by describing their environment setup and main research aims. Details can be found in the *RoboCup* website[2] and in many publications (see, for example, [7] and [8]). Fig. 1 shows different robotic platforms competing in *RoboCup*.

During the years, several subleagues with distinct research challenges, have evolved to attack the problem of developing robot soccer players from multiple perspectives. Such leagues deploy various platforms that differ in hardware and size, which are engineered to promote research in different fields—ranging from low-level system dynamics to high-level collective strategies (CSs).

In fact, *RoboCup* features simulated soccer leagues [Simulation2D (Sim2D) and Simulation3D (Sim3D)] that serve as a proxy to foster research in complex collective behaviors—alleviating hardware, perception, and actuation problems. Research in these leagues is focused on formalizing novel approaches designed to be ultimately transferred to real robots. More in detail, the Sim3D league features small unicycle robots and it is specifically designed to challenge researcher in developing efficient coordination strategies. Each team is composed of eleven autonomous agents playing in a 2-D virtual soccer field. The game is run on a central server, the SoccerServer that has complete observability on the game: Players poses, ball position, and a model of the environment. The Sim3D features simulated NAO robots adding the third dimension to the environment. Hence, participants are forced to design strategic soccer behaviors that also consider the much more complex robot kinematics structures (e.g., hyper-redundant humanoid robots). The overall goal of simulation leagues is to provide basis for the development of real robot systems.

In fact, the *RoboCup* competitions feature four other leagues that are configured with different hardware specifics. For instance, the small size league (SSL) and the middle size league (MSL) leagues employ wheeled robot platforms. Such leagues focus research on coordination strategies and shall represent the first step into physical robot players. These two leagues differ in the robot size and how information about the environment is provided to robots. In particular, in the SSL, a set of top-down cameras and a central computing unit are used to, respectively, provide ground-truth positions of the robots and send them action specifications. The omnidirectional wheels give them stability and very high speeds. These features allow concentrating the development efforts in other aspects such as robot coordination and strategic team-play. It is important to

[2][Online]. Available: https://www.RoboCup.org

highlight that, even though the SSL has complete observability on the environment and they demonstrate a clear and direct link to soccer videogames, they differ on many levels. For example, in videogames, the focus of the game is always on the playmaker, and teammates move accordingly without taking the initiative. In *RoboCup*, every agent is autonomous and, in most cases, the decision process is distributed. Similarly, but with more hardware-demanding platforms, in the middle size, the robots have both sensors and computation onboard—deploying true distributed agents.

The first hyper-redundant robot plays the standard platform league (SPL), which promotes the development of robust robot behaviors in real settings by also attacking perception, localization, and bipedal locomotion issues. Importantly, teams participating in this league must all use the same platform, which is periodically improved by a technical committee. The SPL, in fact, started as four-legged league deploying dog-like Sony Aibo robots. Then, it moved to a new platform in the 2008 by adopting the Aldebaran humanoid NAO robot. Such a league is really important in transferring high-level individual and CSs to complex robot platforms. A standardization of the hardware, in fact, allows teams to focus on software development neglecting hardware design and configuration. Each robot playing an SPL match is fully autonomous and operates in a distributive manner that allows participants to focus on emergent local behaviors as well as coordinated team strategies.

Finally, more complex platforms are in the humanoid leagues (HLs) that challenge participants to compete in soccer matches played with humanoid robots, which foster research on hardware improvements disregarding high-level cognitive skills. This league is further split into sub-leagues. During the time window that this work considers for the decision-making evaluation, there were three classes, based on the robot size: Kid size (HKSL), teen size (HTSL), and adult size (HASL). Such a classification allows us to address issues related to noisy perceptions and bipedal locomotion incrementally. In these sub-leagues, the size of the team depends on the size of the robot platform. In the HASL, for instance, games are set up with no more than two robots for each team. In the HTSL and HKSL, instead, a game is played by two teams of at most three and four robots, respectively. However, it is worth noticing that the HL is currently being reorganized in the following two subleagues: HKSL and HASL.

Fig. 2. Sim2D and Sim3D leagues. In both leagues, matches are disputed by 11 players for each team.

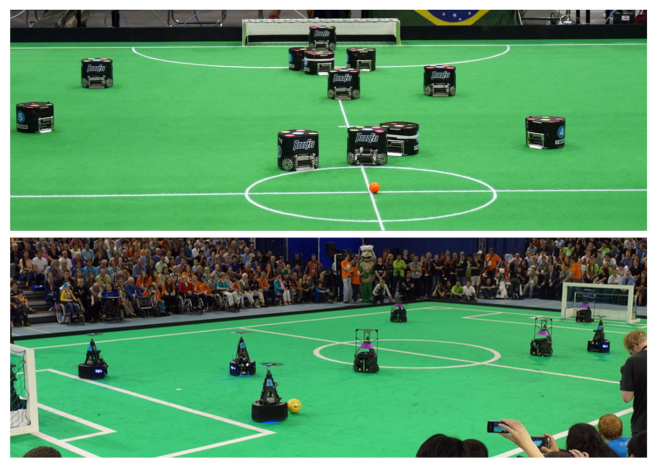


Fig. 3. SSL on top, and MSL during *RoboCup* matches. In MSL, robots must fit a box of $52 \times 52 \times 80$ cm. The ball used is the FIFA standard size 5 and the playing area of the field is 22×14 m. SSL robots, instead, are cylindrical and must be 18 cm in diameter and 15 cm in height. The playing area is of 12×9 m and the used ball is the standard orange golf ball with a diameter of 4.3 cm.

III. Intelligent Soccer Players

The notion of "intelligent agent" has been modeled in several ways. For this discussion, we are going to adopt the architecture shown in Fig. 6. In this model, the environment is analyzed via the perceptions of the robot. Sensing is particularly critical when dealing with real robots: Each agent has to create a representation, which is consistent with the environment state. The interaction with the environment is performed by actuators that execute the *actions* planned by the agent. Each *RoboCup* agent can, therefore, be modeled by using a variation of the sense-plan-act paradigm [9]. This formalization conceives the control of an agent as a cycle.

1) *Perception:* Gathers information through the sensors and turns it into the agent's world model. In the *RoboCup* agent, we have a massive acquisition of data through cameras, onboard sensors of the robot, and all the teammate agents.
2) *Decision making:* Chooses the next action based on the environment model.
3) *Action:* Executes the action it by coordinating all the hierarchical layers that bring the commands to actuators.

Hence, the architecture of a robotic agent links the perceptions of the environment to the action execution. In the agent, there is the whole decision-making process that the agent executes to succeed in its goal. Usually, for a single-agent system, the robot strategy that implements the decision-making process, is decomposed into skills (in the low level) and behaviors. Conversely, for a multiagent system, the CS controls coordination among the agents.

Fig. 4. SPL. The platform is the humanoid NAO robot which is 57.4 cm × 31 cm × 27.5 cm in size. In this league, each team is composed by not more than five players. The field size is 9 times 6 m and the ball used is black and white with 100 mm of diameter.

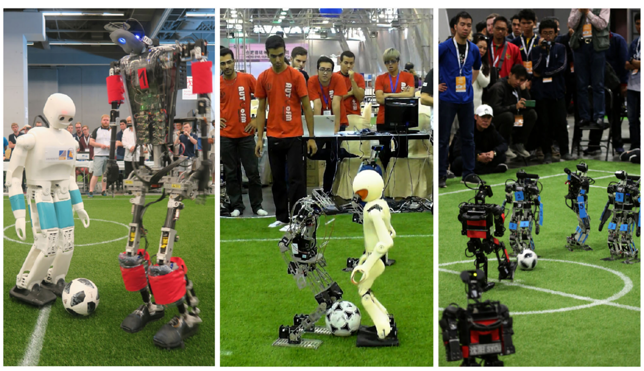


Fig. 5. HLs: HASL, HTSL, and HKSL. The number of robots per team goes from no more than two in the HASL to at most three and four robots in the HTSL and HKSL, respectively. The dimensions of the robots are 130–180 cm in HASL, 80–140 cm in HTSL, and 40–90 cm height in HKSL. The ball used is the FIFA size 1 for HKSL, size 3 for HTSL, and size 5 for HASL. The playing area of the field is 14 × 9 m in the HASL and 9 × 6 m in HTSL and HKSL.

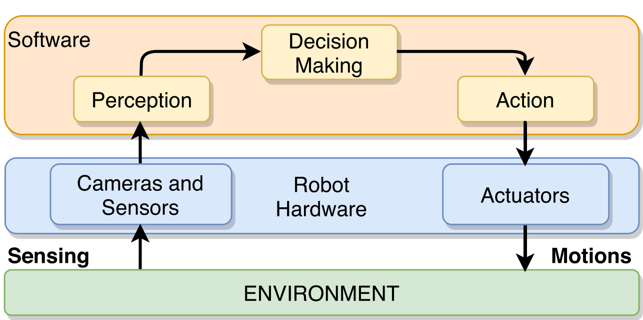


Fig. 6. Agent operation architecture and environment interaction.

Furthermore, the *RoboCup* soccer players act in competitive settings where the game strategy needs to consider also the presence of opponents. Hence, we structure our presentation of the game strategies defined for the implementation of robot soccer players as: Individual strategies (ISs) [10], CSs [11], and finally, in opponent strategy analysis [12].

In multiagent adversarial environments, ISs are an important discriminator to achieve successful game-play. As previously mentioned, the robot has to use its local perceptions to reconstruct the model of the environment. On top of this model, the decision-making system establishes the next action to be taken. As mentioned earlier, ISs include low-level skills and behaviors.

Low-level skills are usually defined as predefined commands for robot actuators to implement action primitives such as kicking, passing, dribbling, diving, and getting-up [13]. These skills are usually represented as routines that can result from a model whose parameters can be tuned through a learning process or obtained through a model-free approach. On top of the skills, individual behaviors determine the action of the agent. Depending on the size, the structure and, most important, on the quality of the perceptions, behaviors are developed to face the events that might occur during a soccer match. The execution of behaviors activate the set of skills associated with them. For example, the behavior of the striker robot can activate different primitive skills such as kick the ball, standing-up, and turn the head to search for the ball. Several technologies have been adopted to develop the behaviors. The single-agent decision making is, usual, carried out by using state machines (SMs) [14], planners [15], and various learning techniques, as evolutionary learning (EL) [16], statistical learning (SL), DL [17], deep reinforcement learning (DRL) [18]. For example, an SM can describe the behavior of a defender that has to stop the ball from reaching the goal by activating the low lever skills to intercept and stop the ball. Specific decision-making procedures can also be defined to isolated game situations when the robot has to behave properly as can happen, for example, in a corner kick situation.

In *RoboCup*, single-agent behaviors are fundamental, however, soccer is by definition a team effort and CSs are key to improve the team performance. When developing a distributed robot system, the things become more challenging. In fact, there are different nontrivial issues that have to be tackled, such as communication-based coordination. The agents in *RoboCup* teams are usually connected via Wi-Fi networks and share the computed local knowledge to make decisions. Coordination is typically distributed (with the exception of the SSL where computation is centralized), even though through communication it is possible to share meaningful information among the team, and reconstruct a distributed common model. While this makes it possible to do a coordination without an explicit protocol, the delays and failures in the network make a distributed approach more reliable.

In these leagues, in order to be effective, robots have to reason upon local and/or global perception. Global perceptions can be produced by external cameras over the field, or by merging the local information acquired by each robot. External cameras (e.g., SSL) allow for accurate perceptions that eliminate all the uncertainty associated with perception, by recreating a scenario similar to the one of soccer video games. Conversely, when information is not gathered from an external source, robots are required to act in a collaborative setting by relying and sharing their local perceptions. In such context, probabilistic modeling helps in reasoning on noisy local information. Moreover, since the robot environment is partially observable, a more comprehensive belief of the world state can be reconstructed only through communication among agents. As a result of a collective team perception each robot can adjust its perception considering others inputs and, thus, better act in the environment. It is important to remark that the reliability of such a collective perception significantly changes throughout the different leagues. Especially in legged leagues, where the robot cameras are not stable, the world representation of a single agent can be narrow and very noisy. In fact, the native partial-observability of such environments intuitively leads to an incomplete and noisy estimation of belief states of the world and, as a result, collective team knowledge is affected by it. In SPL, for example, knowledge among teammates can be significantly different, which promotes research in robust coordination protocols (and individual behaviors) that can operate in extreme conditions characterize by failures in the communication channels and noisy information. However, in this survey, we do not address the perception processes, and assume that the agent takes strategic decision based on a world model created by observation. In fact, direct connection (without an explicit world representation) can be effectively implemented for skills, but not yet for more strategic decisions.

For the CSs, we distinguish between collective behaviors, positioning on the field, and role assignment. Under the category of collective behaviors, we group the specific approaches developed to coordinate specific plays, for example, the pass, corner kick, and kick in. Petri Net Plans [19] have been effectively used in *RoboCup* competitions for teamwork and cooperative behaviors. Positioning in the field is the task of determining the position of players. Specifically, this problem resembles the situation of video games, where all the players that are under the control of the software must position in order to implement the best collective action to support the human controlled player (that we call striker). Historically, the positioning problem has been strictly related to the role assignment, as in [20]. Role assignment is the further step that has to be taken to have an effective CS. Role assignment is a special case of the general task assignment for the multirobot system. Starting in [21], utility-based role assignment and coordination allowed us to coordinate heterogeneous robots, with different hardware characteristic, for playing soccer. Many others technologies have been deployed in coordination and collective behavior problems. Markov decision process has also been applied to the multiagent role assignment [22] and, as in the individual behavior formalization, also learning techniques have been adopted.

In multirobot adversarial scenarios (as the one we are surveying), analyzing the opponent behaviors and strategies enables the team of robots to adapt their strategy to maximize performances. In leagues, where perception is efficiently solved, it is possible to robustly analyze the opponent team [23]. To this end, different technologies have been deployed such as sample clusterization and pattern recognition. In particular, in leagues with simple hardware and complex strategies, learning approaches are widely used.

IV. Strategic Levels in Robot Soccer

RoboCup has represented an excellent testbed for research in AI and robotics. By embracing several leagues with real and simulated agents, *RoboCup* allowed us to explore the boundaries of AI. To this end, the game settings have been constantly upgraded, to make them more challenging and, to some extent, more realistic. Moreover, the developments of the research in AI and robotics, have led to a significant evolution in the proposed approaches: from ad-hoc modeling to various types of learning techniques. It is worth emphasizing that learning approaches must face the difficulties arising from training on real robots. To this end, typically learning is supported by a simulation environment, but the transfer of the results of the training on the real robot is not trivial. Consequently, the most sophisticated learning approaches are developed in the simulation leagues. This section focuses on how the problem of behavior creation for ISs, CSs, and OA, have been addressed using different approaches. Fig. 7 shows the correlation between different behavior, such as OA, CSs, and ISs (on the vertical axis), and used methodologies such as PL, learning, and others (on the horizontal axis). A discussion on the developed approaches, their characteristics, and their use in the *RoboCup* competition, is addressed later on in the survey.

A. Individual Strategies

The design of effective behaviors for the single-agent is still one of the core issues in robot soccer players development. Single-agent behaviors are divided into two categories based on the abstraction level that is required for completing the task: skills and behaviors. Skills execute primitive actions and behaviors determine how to select them to achieve a specific goal.

1) Skills: The idea of skill concerns the control of the execution of complex physical actions, which require the combination of multiple motion commands. For example, skills can involve the control of actuators for performing a kick, or the gait generation for a particular kind of walk system. Even though the approaches used by competing teams range all the spectrum from ad-hoc, to model-free learning, the research is currently focused on proving the performances of learning techniques. We first address the approaches developed in SPL. In [36], an imitation learning [70] system is developed. The agent has to learn the dribble and searching skills, imitating the motion of the Striker agent (model based) of the former world champions. Specifically a convolutional neural network is used for this work, to learn to control speed commands for robot joints using camera images as inputs. The learning to dribble with the ball problem is also addressed in [63], using reinforcement learning techniques. In this case, a hierarchical task decomposition is used for this problem. This article describes and compares several hierarchical learning strategies for designing robot skills. The evaluation metrics proposed are averaged between skill performances and sample efficiency. Interactive machine learning (IML) is another approach for training a dribbling engine for SPL [48]. The proposed method solves the dribbling problem by dividing it into two subproblems: determining the required dribbling direction and calculating walking velocities for pushing the ball. A predictive model of ball movements is used for improving dribbling performances, combined with a batch incremental learning implementation of the algorithm. Actually, IML approach can overcome typical shortcomings

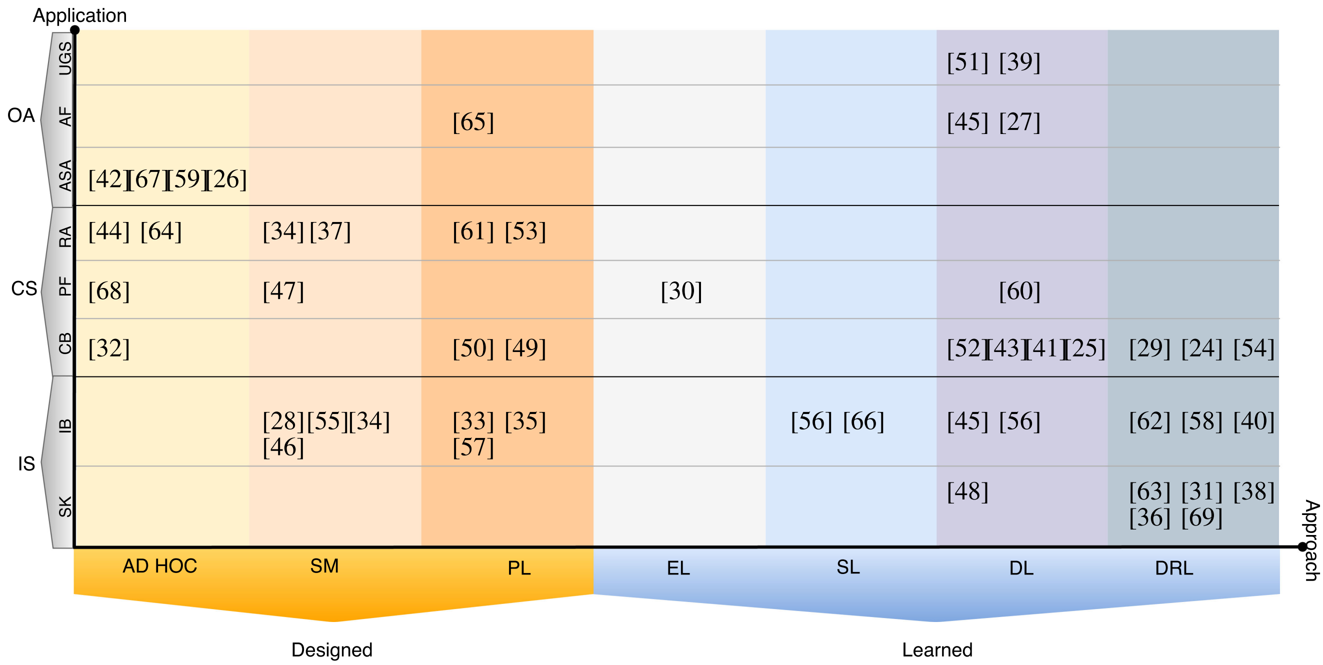


Fig. 7. Paper classification for application and approach for the time window 2015–2019. The designed approaches are subdivided in SM approach and PL approach. Learning-based approaches are subclassified as EL, SL, DL, and DRL. The application is divided in IS, CS, and OA. IS is subdivided in skills (SK) and individual behaviors (IB). CS is subdivided in collective behaviors (CB), positioning in the field (PF), role assignment (RA). Finally, OA is subdivided in action sequence analysis (ASA), forecasting future opponent action (AF), utility game strategy (UGS).

of ML, such as data reliability and sample efficiency, but it requires a human intervention in order to perform. Other HLs, such as simulated ones, use learning for improving modeled skills such as running and dribbling. For this purpose in [31], the PPO algorithm is used. The aim is to obtain natural gaits, even sacrificing performances. SimSpark [71] is used to simulate a NAO humanoid robot, using visual and force sensors to control actuators. The results are natural running and dribbling skill. However, some of those skills still have problems in switching to a real robot implementation due to hardware instability. We have seen that much work has been done for learning skills in humanoid robots. The complexity in bipedal walk-engine modeling encouraged the research in Learning methods for such a topic. However, ground skills are fundamental also for wheeled robots. In [69], an extension of the decision-making module, a tool used for handling the decision-making process in *RoboCup* SSL is proposed. Their aim is to develop a temporal-difference reinforcement learning system based on a multilayer perceptron as a function approximator. They develop different skills using this system: Shooting skills and passing skills. The authors conduct nine experiments to develop and evaluate these skills in various playing situations. Reinforcement learning in SSL is also addressed in [38]. In this work, an implementation of deep RL into the skills tactics and plays architecture is shown. The implementation relies on a deep deterministic policy gradient algorithm for learning a go-to-ball skill and aim-and-shoot skill. They evaluate performances in comparison with their previous modeled ones, using a physically realistic simulator.

2) Individual Behaviors: Behaviors determine what the robot should do depending on the game state and the current goal. Deciding when to shoot, pass, dribble, or perform a contrast fall under this concept. In recent past, behavior modeling has mainly relied on SMs and PL systems, both in wheeled, and humanoid robots. In [46], CABSL, a widely used system for behavior implementation, is presented. CABSL is an extension of the previously introduced XABSL [72] language and stands for C-based agent behavior specification language. CABSL has been created entirely in C++. The agent behavior is constructed as a hierarchical set of SMs. Its structure is composed of basic building blocks: options, states, transitions, and actions. Options are the implementations of the SM. A behavior starts from a single option, the root, then are called other options. The options are organized hierarchically. Each option describes a skill or a simple motion of the robot. Every state can call another option or execute the associated action. In the first part of each state, there is a transition part which may cause the switch to another state (before the execution of the action). Furthermore, to keep the code clean, CABSL supports also the use of libraries as collections of functions and variables that may be called inside transition and action boxes.

Other HLs rely on SMs for behavior implementation, as described in [55]. HASL also uses this kind of system. Interesting extensions to SMs have been addressed in works like [28] and [34]. The first is about the implementation of fuzzy SMs for the behavior of agents in a multirobot environment. The model includes elements of training to increase the agility of the behavior of the agent. The second paper presents active self-deciding stack, a lightweight variant of hierarchical SMs. It is used in the *RoboCup* HKSL.

If SMs only reason about the current state, PL can perform inferences on future outcomes of actions. In *RoboCup* SPL, after several years of use of CABSL, some teams are switching to PL-based systems. Röfer *et al.* [33] present the skills and card system. A skill is a behavior component that executes a task; it roughly corresponds to the notion of primitive actions presented in this survey. A card associates actions with the conditions under which they should be executed. Cards are characterized in the PL fashion by using pre- and postconditions. The execution consists in computing which cards satisfy the precondition and in selecting one of them in accordance with the given goal. Even if the skills and cards system adopts pre- and postconditions to model actions, it still lacks forward state exploration using a prediction model. The work done in [57] is based on the exploration of future states. They introduce a method for fast decision making. The outcome of each possible action is simulated based on the estimated state of the situation. The simulation of a single action is split into several simple deterministic simulations, considering the uncertainties of the estimated state and of the action model. Each of the samples is then evaluated separately, then combined in comparison with other actions for the overall decision. PL is a suitable solution also for sophisticated platforms like the HASL robots, the work in [35] shows the performances of hierarchical PL methods for the NimbRo robot.

Learning approaches have been proposed also for behaviors, however, the good performances of modeling approaches have slowed down the development of these methods. In [56] and [66], two different SL methods are used for solving behavioral problems. In the first, a learning to rank algorithm is used for determining state evaluation for a decision-making process. In the second, a linear regression approach is used for determining the position of the goalkeeper agent in an MSL game. Reinforcement learning methods have been widely used for behavior creation. In [40], a simulated 3-D striker agent is trained to score a goal without previous knowledge, using a transfer learning approach instead of the classical reward shaping. In [58], the problem is addressed using a method based on a combination of Monte Carlo search and data aggregation to adapt discrete-action soccer policies for a defender robot to the strategy of the opponent team. By exploiting a simple representation of the domain, a supervised learning algorithm is trained over a first collection of data consisting of several simulations of human expert policies. Monte Carlo policy rollouts are then generated and aggregated to previous data to improve the learned policy over multiple epochs and games. Finally, in [62], the classical bandit approach is exploited for solving the task of static free-kicks for agents.

B. *Collective Strategies*

Without coordination playing soccer is not successful. Hence, CSs play a key role for the performance in the *RoboCup* soccer game. Our discussion on CSs is structured into the following:

1) collective behaviors, which regards situations where a certain task needs the combined execution of actions from multiple robots to be accomplished;
2) positioning on field strategies that succeed the goal of maintaining the right position of robot soccer players on the field to maximize team performances;
3) role assignment, a process that determines which player has to take a specific role in a game situation to maximize the cumulative task utility of the entire team.

1) Collective Behaviors: The creation of specific behavior realized as part of the collaboration of multiple robots is, so, an essential field of player development. Collective behaviors discussed in this section are as follows: pass strategies, coordinated movements, and corner-kicks strategies. Ad-hoc methods can be quite effective for this purpose, as shown in [32], where the winner team of SSL assert that their passing strategy was fundamental to success. In this work, suitable passing points are extracted using search-based interception prediction (SBIP) as a bases for a dynamic passing points searching algorithm. After selecting all possible passing points, they use a value-based best pass strategy, getting the scores of each pass by calculating the weighted average of the pass point features. The features have been selected, taking into account game situations as the distance of the passing point from the goal or the interception time of teammates. Other structures, like PL with a deep action selection horizon, are used for modeling multirobot environments. In [50], an approach based on a modification of extended behavior networks is proposed. This approach is made with a situation-dependent utility function based on the effects of the executed actions. The use of PL-based systems like the previous one is frequent for the implementations of collective behaviors. In [49], the skills tactics and play coordination architecture is presented. This system is developed for centralized-multirobot teams. STP enables sophisticated team plays both in the MSL and the SSL. To adapt this practice to distributed systems, they use a voting system. In Sim3D league, throughout the years, the deployment of learning-based approaches has been very compelling, and usually resulting in the winning strategy [43]. Learning-based methods perform well also for the analysis of highly dynamic situations for determining the team strategy. In [41], a mimicking process for Sim2D agents is developed. It is proposed a method for improving the performances of a team by mimicking a stronger one. A neural network is employed to model the evaluation function. This network is trained by using positive and negative action sequences extracted from game logs. The experiment results in actual improvement of game performances. In [54], the team learning process is obtained through the mechanism of imitation learning combined with a semisupervised system for clustering teams' formations. If learning and mimicking opponents' strategies are essential, it is equally important to learn how to coordinate robots concerning the teammates' signals. In human interaction, gestures cover an essential role. For handling this kind of communication between robots, it is necessary to recognize and generalize enough the teammate's message will. Di Giambattista *et al.* [25] present a DL method for determining the pose of a teammate robot in the SPL. The system is built on a custom version of the OpenPose net. After the recognition of the other NAO robot's pose, the approach consists of a set of coordination strategies in peculiar game situations like corner-kicks and kick-offs. Using this gesture-based system, a robot can communicate the will of executing actions such as long passages, short passages, or faints.

All the learning methodologies seen so far are based on the presence of a model knowledge. To learn collective behaviors without relying on a knowledge model are often used reinforcement learning techniques. There are several applications of this approach in *RoboCup* both in multiagent strategies, as and in single-agent ones. In [24], they provide a comparison of the two main approaches to tackle this challenge, namely independent learners and joint-action learners (JAL). The methods are implemented and evaluated in a multirobot cooperative and adversarial scenario, the two versus two free-kick tasks, where scalability issues affecting JAL are less relevant given the small number of learners. They implement the systems using DRL approaches for both of them. The final results are convincing, with the robot managing to complete the task with good performances in both the methods, but with a better average reward obtained by the JAL approach. In reinforcement learning tasks, it is essential to determine rewards signals for agents. In [29], a methodology for obtaining reward signals for teams of robots directly rewarding the field and the configuration of teams on it is proposed.

2) Positioning in the Field: Another key part of the team strategy is selecting the correct formation on the field by choosing the position of the agents to optimize the performance in the game. A well placed formation allows having an advantage in a wide variety of game situations, both attacking or defending. Positioning on the field is achieved using several approaches, including SMs and learning algorithms. The work in [68] demonstrates the importance of team positioning in the wheeled leagues. They propose an approach for robot coordination based on the use of utility maps to improve the strategic positioning of a robotic soccer team. They design utility maps for adapting to several different situations, to make them adaptable to different opponents' strategies. Maps are used in a set of game contexts: passes in free-play, choosing the best position, shooting, and receiving the pass. Röfer *et al.* [47] describe a set of "advanced team tactics" that they implemented in order to win their matches. The authors, in fact, take advantage of adaptive coordination, and by changing the roles of robots during the active phases, they manage to organize sophisticated positioning strategies for the ball passing, dribbling, and cover weak points of the team. The implemented strategy is based on the presence of a single active agent called striker, which has the task of playing with the ball. Three inactive agents, two defenders, and a supporter have the task of positioning themselves in the best possible point on the field for supporting the striker play. The core of the presented strategy is to handle the chance of receiving a long-distance shoot from the opponent, by placing the defender robots. The supporter instead has the task of positioning in the field for receiving the striker pass or rebounding the ball in case of an opponent's goalkeeper save.

To overcome the limitations of the models developed for team positioning, teams also implement learning approaches. Several kinds of learning are applied here. One of them is the EL [73]. In [30], FEASO, a distributed framework for the *RoboCup* Sim3D based on EL, is proposed. It is used for the strategic optimization of robots by improving the placement of robots on the field. Their approach comprises the following three modules: evolutionary algorithm execution, parallel fitness evaluation, and fitness computation. The problem of positioning, taking into the opponent's team play, in Sim2D is addressed also in [60]. In this work, the authors use an approach based on a sequence of Bayes estimators. The implemented model is used for determining the association of player formation against the opponents during corner-kicks.

3) Role Assignment: Role assignment is the part of coordination, which allocates the right task to the right agent. In the *RoboCup* context, it can involve the choice of which robot should perform an offensive or a defensive role. A large number of ad-hoc modeled coordination systems have been implemented for the *RoboCup* environment, especially in the SPL. In this environment, much work has been done to develop the best communication and role switching between robots. Role assignment is essential to cover the soccer field well and to avoid dangerous situations coming from the opponents' strategies. Riccio *et al.* [64] introduce a novel approach for coordinating teams of robots. The key contribution consists in exploiting rules governing the scenario by identifying and using contexts. The method relies upon two well-known methods for coordination: distributed task assignment and distributed world modeling. Contexts could be considered as specific configurations of the operational environment. Situations where robots know where the ball is and situations where they do not are examples of contexts.

Given the complexity of the task, many different approaches have been tried during the last years. Some of these methodologies involve complex data structures as graphs or SMs. Many algorithms for coordination use SMs as a base instead of ad-hoc methods. For instance, in [37], they use a custom version of the Raft algorithm. Raft is a consensus algorithm. It is decomposed in several independent subproblems, and it then addresses all significant pieces needed for obtaining a sound system. The coordination presented in this article is based on the election of a leader, balancing the disadvantage of centralized and decentralized approaches. For instance, in [34], they implement a form of coordination inside the SM that controls the robot behavior. Switching between roles for performing a routine is, in fact, a type of implicit coordination. In 2015, in [61], a method based on the selectively reactive coordination algorithm is proposed. This method consists of two layers, one layer which allows the team to project the outcome of the chosen formation without taking into account the opponents. The other, called individual opponent-reactive layer, adjusts the result taking into account the opponents' actions. Finally, in [53], a centralized PL system is used as a base for coordination of Humanoid HTSL 3 versus 3. This system is based on the collection of common world modeling. The PL system uses the shared information from robots to build a global map. Using this map, it computes a utility function on the planner, based on assigned roles and their possible future actions.

To encourage alternative forms of coordination, *RoboCup*Soccer SPL offered for several years a complex challenge called the "drop-in" challenge. The trial consists of creating two teams of five robots composed of players from different teams. In this context, coordination can be complex; robots can have access just to a part of the teammates' information, because messages contain a standardized part and a team-specific one. The problem of developing strategies and coordination for such an environment is addressed in [44].

C. Opponent Analysis

In games, in order to have an holistic understanding on the environment, opponent players need to be modeled and integrated in the robots knowledge-base. Players in fact have to formalize opponents as an obstacle to their mission, and react to them strategically. To this end, several approaches have been proposed to analyze opponent behaviors, forecast future situations and evaluate the current game situations. Accordingly, we organize this section to categorize major contributions to the OA problem.

1) Action Sequence Analysis: In the *RoboCup* setting, opponents analysis represents very challenging task due to partial observability and a very unpredictable and dynamic environments. To several approaches propose domain-specific state representation and clusterize them to extract pattern in opponent state sequences. For example, Iglesias *et al.* [42] determine opponent strategies on simulated robots by analyzing previous game logs, and extracting a set of predefined actions, which are then pair with game states and organized in decision trees [74]. In the same context, Yasui *et al.* [67] propose instead an approach based on a dissimilarity function to classify opponent strategies on real robot platforms. In particular, they exploit such classification to determine action sequences using cluster analysis. The work has been later extended by Adachi *et al.* [59] and Adachi *et al.* [26]. The former improves classification of opponent actions by decreasing the computational demand of the approach, while the latter proposes an online formalism for the clustering algorithm. In all settings, the authors exploit their classification to adapt their formation to the opponents, guard them, and decrease their chances to score.

2) Forecasting Future Opponent Actions: Once opponent actions have been classified and their strategy formalized, a crucial step is to forecast their action and predict their intentions. Predicting the opponent intentions in fact can truly improve the performance by preventing strategic positioning and anticipate opponent attacking formations. To this end, Li and Chen [65] propose two approaches based on a fuzzy inference system (Mamdani-type fuzzy and Sugeno type-fuzzy) to determine players intentions in small corner-cases such as passing or not passing the ball to teammates. They conduct an exhaustive experimental evaluation that shows that by modeling opponents intentions they can double their chances of winning a match. Conversely, Suzuki and Nakashima [27] use a model-free approach to learn game states utilities in a simulated setting. They introduce forward simulation for situation evaluation (FOSSE), which is a two-step algorithm that forecasts future states and determines the strategic team positioning to respond to opponents attacks. FOSSE uses recursive neural networks for predicting future states and a simple forward model to conduct utility state evaluation.

On a different (more complex) platform, Rizzi *et al.* [45] propose an opponent forecasting algorithm on a humanoid robot within the SPL. The authors introduce situation-aware fear learning (SAFEL), a real-time algorithm to analyze and synthesize behavioral models based on LeDoux's architecture [75]. SAFEL allows an agent to learn behavior profiles and to respond accordingly.

3) Utility of Game Situations: Several approaches to strategic game-play in robot soccer matches formalize a methodology to evaluate game situations with a utility function—that can be either defined by experts or learned from experience. OA is extremely valuable in generating accurate utilities. Moreover, the majority of existing approaches are learned-based since the search-space is too big and cannot be embedded with predefined rules. In fact, Michael *et al.* [51] propose an approach to behaviors detection and identification. Their goal is to develop a domain-independent methodology to classify states of the environment based on recurrent neural networks that encode past experience. The approach is also used to learn conceptors that are lower dimensional manifolds that describe robot trajectories and, enable utility state predictions more easily. A different supervised approach is presented in [39] where the authors exploit a dataset of soccer matches to classifies game states and evaluate the likelihood of scoring for each playing team. In particular, the utility function derived measures the estimated time until a team scores.

V. Discussion

In this section, we dive more in depth, and discuss major constraints that the robot embodiment and environmental constraints forces on soccer competitions. We elaborate on how researchers drive their efforts in deploying holistic robotic systems that can be robust and effective.

A. Rules and Hardware Constraints

Different aspects influence the choice of an approach, or a game strategy, to implement. In fact, within the *RoboCup* competitions, researchers are challenged to attack the game of soccer from different angles and in different environmental configurations. When we closely observe the *RoboCup* leagues and the environments they recreate, we can classify them as continuous, dynamic, nonepisodic, and nondeterministic environments. Continuous because the perceptions and actions are not in a limited number. Dynamic since the environment can change while an agent is deliberating, for instance, due to the opponents. Nonepisodic and nondeterministic since there are no episodes, and, the next state of the environment is not entirely determined by the current state and the actions performed by the agents. However, one characterization of the environments does change among the different configurations. In fact, within the *RoboCup* leagues the level of observability does change, spanning from fully observable environments (e.g., SSL) to partially observable ones (e.g., ASL). In particular, each league provides different operating settings. For instance, in the SSL, all robots have unique identification patterns that are accurately tracked by a central vision system. Such a setup alleviates the

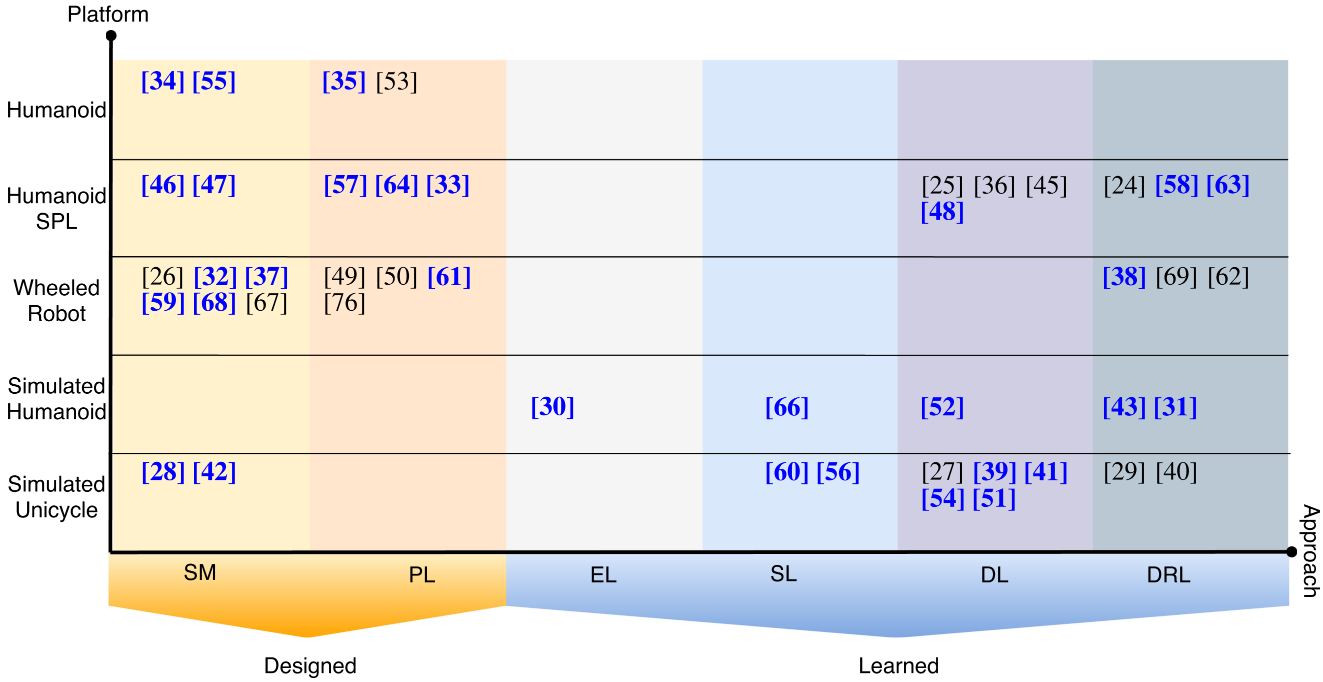


Fig. 8. Paper classification for approaches and platform for the time window 2015–2019. The model-based approaches are subdivided in SM approach and PL approach. Learning-based approaches are subclassified as EL, SL, DL, and DRL. In bold, we highlight contributions that report results of the proposed approaches obtained during the deployment in matches of official RoboCup competitions.

Fig. 9. Paper classification for application and platform for the time window 2015–2019. The applications are subdivided in OA, CS, SI.

partial-observability constraint and opens up to a wide range of research fields spanning from learned-based techniques to OA and high-level coordination strategies. Fig. 9 shows that stable platforms better supports OA and, more in particular, full observability simplifies the opponent tracking problem and promotes research in opponent behavior understandings, as in [59], [26], and [67]. In the simulation league, the environmental setting does not provide a central vision system, but virtual perception systems are generally more reliable and virtual agents do not break nor malfunction. This is a key to explore new methodology and techniques, and to attack the soccer game problem in a more controlled setting. This led to have learning approaches in official competitions, as shown in Fig. 8. The MSL is a wheeled league featuring relative big robots that have onboard an omnicamera providing an egocentric, but complete point of view of the soccer field. This forces developers to explicitly formalize errors and noise that characterize real vision system, but aids teams to safely improve on robot coordination and individual game strategies. Here, we can still find some attempt to have accurate [76]. Moving into an environment configuration similar to the one of human soccer, the SPL forces all teams to use the same robot platform. Such a constraint suggestively puts all teams on the same starting point, where the determining factor of the match is the skill of each team to implement effective software solutions maximizing the platform at disposal. Finally, the HL can be considered the most hardware-driven league, where teams need to build their own robots and can overlook on high-level behavior models both individual and collective.

Fig. 9 shows how the robot platform influences the application of the different approaches. In this analysis, the x-axis reports

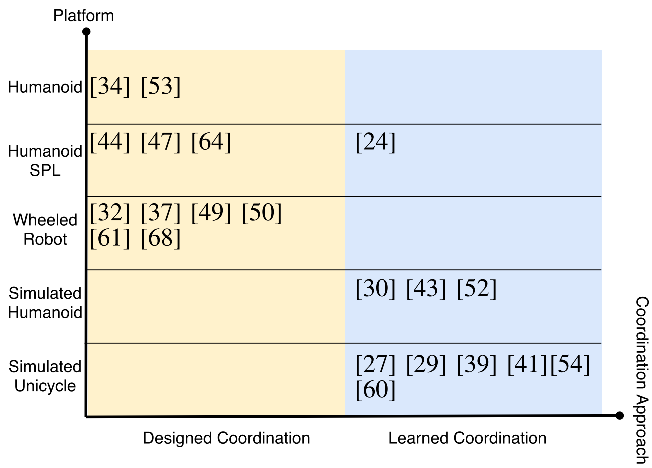


Fig. 10. Coordinated CSs compared to the deployed platform.

the platforms with increasing hardware complexity (as in the previous case), while the y-axis highlights three types of applications or use-cases, namely ISs, CSs, and OA. It is noticeable that there is not an emergent trend in the research of CS and IS as they are completely balance across platform categories. This hints to the fact that in multirobot, adversarial scenarios, in order to achieve the satisfactory performance both individual and CSs, must be addressed. In other words, in robot soccer competitions success is defined by both and not their individual contributions, despite the varying league constraints exposed earlier.

Nevertheless, more can be discussed on the OA row. Major contributions on the topic are dictated by the perception system. In fact, both for simulated and real platforms perceptions are provided, and robots operate in a fully observable environment. This is a key to analyze opponent behaviors, classify them, and react accordingly. As mentioned, the partial observability on a humanoid robot naturally affects the possibility to thoroughly analyze opponents, and as a consequence, it is not a mandatory skill on such platforms. However, as in the previous scenario, we can notice that the first attempts to transfer OA approaches to complex platforms are (again) in the humanoid SPL. Rizzi *et al.* [45] develop an approach to analyze opponent behaviors and to define strategic individual responses. This is another confirmation that approaches are scaling progressively toward more complex platforms even though, at the moment, OA remains an active research topic only for platforms with a reliable perception system.

Finally, Fig. 10 classifies collective coordination strategies with respect to different robot platforms. On the x-axis the platform categories divided in accordance with the *RoboCup* leagues, while on the y-axis, we divide in between designed and learned coordination approaches. Effective coordinated strategies represent the ultimate objectives for soccer robot teams and a determining factor that characterizes successful robotic systems. By analyzing the figure, we can clearly see one of the most sharp categorizations of the surveyed papers. In fact, all simulated platforms adopt a learning approach, whilst real platforms a predesigned one. In the SPL quadrant, the work by Catacora Ocana *et al.* [24] represent an outlier that shows a learning-based approach on a real platform. In this work, the authors exploit DRL to address robot coordination in particular situations such as penalty-kicks. Suggestively, coordination approaches heavily depend on the platform used and the most sophisticated solutions are deployed on simulated and wheeled robot platforms. However, the mission is to scale such methodologies to all physical platforms. The process gradually progressing as we notice that researchers in the SPL started to investigate learning coordination strategies also on humanoid robots.

B. Approaches and Deployment in Competition

As observed, in *RoboCup* competitions, proposed techniques can be coarsely divided in two major categories. Accordingly, Fig. 7 provides a visual representation of such a dichotomy by comparing predesigned and learning approaches. Here, we assume a different point of view that abstracts from the *RoboCup* leagues, and analyzes how the robot embodiment affects team strategies and can influence the approach implemented by researchers. To this end, Fig. 8 organizes the surveyed contributions by highlighting proposed approaches on the x-axis and used platforms on the y-axis. The x-axis shows a moderate range of options spanning from static SMs to learning-based approaches. Suggestively, presented approaches are divided in two macrosegments. The former is represented by predesigned approaches that are divided into two subclasses: SM and PL. Similarly, the latter is represented by learning-based approaches that are further divided into the following four subclasses: EL, SL, DL, and DRL. On the y-axis, instead, we show the robot platforms in five categories. By examining the platform axis bottom-up, we highlight the complexity of the robotic platform kinematic chain in increasing order.

It is worth mentioning that simulated robots are considered as the most basic hardware embodiment and therefore occupy the first segment on the axis. In particular, Sim2D robots are modeled as simulated unicycles, while simulated humanoids resemble the NAO robot. Intuitively, in such a context, simulated platforms do not represent a real limitation, which opens up to a very rich scenario for exploring and evaluating novel techniques. As can be observed in the figure, approaches developed on simulated platforms have a clear bias in exploring learning-based techniques, whereas only two contributions are not learning behavioral models, but rather focus on single-agent strategies [28] and OA [42]. Moreover, simulated humanoids support a broad and balanced exploration of learning techniques, which suggests that there is no dominant trend in the research strategies, and hyper-redundant simulated robots do not represent a limit or bias for researchers. In contrast, simulated unicycle robots feature a very basic platform, and proposed contributions show a dominant pivot in DL. In such a scenario, robots are deployed in a very dynamic environment, and neural networks are usually preferred due to their generalization capabilities. It is important to highlight that also DRL approaches are recently receiving more attention as both reported papers are from 2019—achieving state-of-the-art results in transfer learning [40] and in learning effective strategies [29]. Proceeding on the platform axis, wheeled robots represent the first step into real platforms analysis, where we group small- and middle-sized wheeled. It is important to highlight that middle-sized robots have omni-directional cameras on board as well as their computation unit. Instead, small-sized robots are remotely controlled by a central processing unit that collects perception by exploiting a set of top-down looking cameras placed above the soccer field and commands the robots via Wi-Fi. Interestingly, the third row of Fig. 8 shows a clear cut in explored approaches, which are mainly focused on nonlearning techniques. Learning-based solutions are being investigated only recently; however, all papers in the DRL quadrant rely on small-sized platforms, suggesting a strong dependency between the hardware characteristics and deployed technique. Researchers relying on middle-sized platforms prefer more controlled solutions, which they can explain in the case a failure occurs. Moreover, in such a context, robots operate in a fully decentralized fashion, which seems to make the exploration of learning-based techniques more difficult. Then, standard humanoid platforms have a particular hardware configuration within *RoboCup* competitions. The platform deployed is the humanoid NAO robot. Teams are not allowed us to alter the platform, which enables researchers to focus on high-level cognitive tasks. Key issues in using such a robot are partial observability and bipedal locomotion. The tradeoff between a ready-to-use but hyper-redundant platform reflects in an unbiased exploration of the research field; research works equally employ existing techniques. It is important to notice that DL approaches are always investigated when perception and data gathering are robust and reliable (e.g., simulated robots, fully observable). This suggests that individual and CSs are not only limited by the hardware, but also the reliability of the perception system plays a role. In this setting, such a constraint is less strict and, although the perceptions are not much more reliable with respect to other humanoid platforms, the standardization of the platform allows researchers to exploit learning-based technologies in partial, noisy, and fully decentralized settings. The most hardware-demanding category is represented by large humanoid robots, which gathers custom robots with different sizes. In this context, strategic game-play does not find large support by the platform as researchers center their efforts on maneuvering the hyper-redundant platforms. Hence, implemented behaviors are basic, and the environments are more structured. As a consequence, there is not a widespread use of need to deploy learning-based techniques yet, and all contributed papers describe predefined static approaches. This further remarks our findings that hardware represents a challenge in behavior generation; thus, when developing game strategies, physical aspects must be taken into account to achieve good results.

Fig. 8 shows an interesting relationship between the approach adopted for strategic game-play and the physical platform. Learning-based techniques are preferred when complex individual and CSs are needed. However, existing methods assume a simple platform and do not scale to large and more hardware-demanding robots. Nevertheless, it is worth remarking that such a distinction is going to blur in the next years. We are already observing that techniques developed both in simulations and small-sized platforms are scaling up to more complex platforms such as the humanoid SPL. It is not a surprise that such a phenomenon begins within the SPL, where a more stable and controlled environment is configured. Additionally, Fig. 8 also shows approaches that have already been successfully deployed in *RoboCup* competition. The contributions marked in bold are, in fact, the ones that have been deployed and used in the real games, while in black, approaches that did not yet led to successful deployment in the competitions but that are important to report in order to provide a complete picture of the ongoing research within the *RoboCup* environment and its future developments. We can notice that SMs are almost always deployed in competitions. This is due to the fact that SMs tradeoff between computational efficiency and expressive power. On the other hand, learning-based methods and automated PL still struggle to achieve a complete deployment in this kind of situation. In particular, it is easy to see that simulated leagues, such as the Sim3D, always manage to effectively implement the behavior developed in research also in competition. This is given by the absence of strong hardware limitations that other leagues could have.

PL methods offer the possibility to reason about future actions, allowing the agent to perform plans that take into account the evolution of the environment. However, PL approaches can often require many resources for the exploration of the state space, particularly in complex environments like the ones that the *RoboCup* leagues propose. Moreover, the development and generation of complete offline plans are not rewarding for dynamic environments. In fact, PL techniques require precise modeling of both the environment and of its evolution and in the situations provided by the *RoboCup* competitions is nearly impossible to achieve. The aforementioned PL problems are often mitigated by using online and bounded PL techniques that allow the agent to plan on a limited subset of the state space. These systems allow reducing the needed precision environment modeling, given the chance of monitoring the game evolution. Also, the reduced state expansion led to an intensive implementation of these approaches also in hardware bounded leagues as the humanoid SPL one.

End-to-end DL approaches, on the other hand, do not require human modeling at all. By exploiting this category of approaches, it is possible to automatically learn the game behaviors using the agent experience or a precollected dataset. This kind of algorithms are extremely useful to tackle problems difficult to model; such as dynamic game situations and complex control problems. However, such techniques have been deployed less frequently onto real robots. This lack of real robot implementation is given by the huge computational cost [77] that this kind of methods require. Moreover, another problem related to the use of DL on physical robots is the “sample efficiency,” this category of approaches requires a large amount of training samples in order to make the algorithms converge to a competitive solution in the policy search-space. During the learning process, computational time and complexity of the algorithms might represent a prohibitive cost that in most situations, still, cannot be paid.

Additionally, DL methods often work as a black-box. This is not always suitable given the need of teams to evolve and refine the code as the competition takes place. Humanoid SPL is surely one of the leagues that contributes more on the topic, but still, the use of the DL for behavior generation struggles to be widely implemented in the game due to all the disadvantages of holistic learning solutions.

Summarizing, we have seen the implementation of the different approaches and their use in the several leagues of the *RoboCup* competition. At the current stage, as highlighted in Fig. 8, model-based methods represent the preferred in-game implementation due to the practical use, quick adaptability and intelligible solutions. The figure also shows that more complex approaches, are incrementally finding an application in scenarios as humanoid and SPL leagues that, given the complexity of the platform, can benefit from the chance of reasoning in advance of the possible outcomes of the agents' actions. Finally, it is important to note that DL techniques do find a widespread in game application only in leagues that are lightly bounded to the hardware constraints—like the simulated ones. The use of learning for decision in the physical robot leagues still struggle to be adopted given the computational cost required by this kind of approaches, sample inefficiency, and lack of intelligibility. However, this does stop researchers to experiment and explore these methodologies that conversely represent the most advanced research topics in these years. In fact, we do report several learning-based contributions that even if (in the majority of the cases) are not deployed directly in the real competitions, they do contribute to improve components of the robotics system and decision-making strategies being combined with PL-based approaches.

VI. Conclusion

Competitions in *RoboCup* have become more and more dynamic and complex through the years. The year 2050 is around the corner and researchers are working around the clock to improve the state-of-the-art and win the *RoboCup* challenge. Recently, research on behavior generation and decision making started to be a key research topic for the competition, shifting attention—especially in certain leagues—from hardware and low-level control to decision making. In this survey, we have addressed the issue of creating behaviors for physical agents in robotic soccer. Through an overview of research on the topic over the last five years. The main objective of this article is to illustrate the trends in research on behavior creation and, in addition, to shed light on the different characteristics that portray the decision-making processes in each league. We have shown how 1) OA, 2) CSs, and 3) ISs take advantage of the use of a wide range of AI solutions, such as PL systems, SMs, machine learning, and reinforcement learning. We have also seen how the hardware of robots can affect the effectiveness and complexity of these behaviors. The work carried out in leagues with high mobility and reliable perceptions (such as simulated ones) has its primary focus on developing sophisticated strategies for team play. On the other hand, critical hardware leagues like the humanoid focused effort on behavior modeling for individual agents. This article also shows the change in the research trend on behavior, which, over the years, is increasingly targeting learning algorithms and group methodologies. Behavior design in *RoboCup* offers multiple possibilities for future developments. The increasing hardware capacity of robots allows for the use of more sophisticated and heavy algorithms. Moreover, the evolution in predictive methods is allowing us to create solutions able to develop gaming strategies at multiple levels. More and more leagues are shifting their development standards toward PL and machine learning approaches, making ad-hoc modeling increasingly obsolete.

Another critical open track is the creation of heterogeneous teams for *RoboCup*. Until now, teams always played with homogeneous teams composed of identical robots. The aim is, like for real soccer games, to create teams with different robots with different skills. Some work has already been done in this direction. In humanoid and SPL leagues, the problem has been addressed with the drop-in competition. Also the MSL [78] has a technical challenge called cooperative mixed-team play in which teams need to demonstrate the team play between two or more robots from different teams. Also, in Sim2D and Sim3D, heterogeneous teams have been composed to perform the drop-in. In 3-D league [79], in the drop-in player challenge, each team contributed two drop-in field players to a game where both teams consisted of drop-in field players. Players were scored exclusively by their average goal difference across all of their games. To accurately measure their performance, every team played at least one game against opponents from each other team. A total of nine teams participated in the challenge. Game pairings were chosen by a greedy algorithm that tries to even out the number of times agents from different teams play with and against each other [79].

In the SPL, the drop-in competition has been bench-marked and has been analyzed by Genter *et al.* [44]. In the HL, the strategy for handling robots from different teams playing together has been analyzed by Paetzel *et al.* [80], which proposes to revise the competition scheme, moving away from participating with a team of robots to participating with a single robot that preserves the competitive element of the ranking performance of individual robots and awarding trophies. They propose a set of rules, which shall be continuously judged during the game, composed of positive and negative actions suitable to rank the robot behavior. However, despite these previous efforts, the goal of achieving completely different teamwork between players remains far away. In real football, players have entirely different physical characteristics and specific skills. This diversity in playing style guides the development of the initial strategy and also the evolution of the decisions of all players. This high difference in the characteristics of individual agents still remains an objective for *RoboCup*.

Over the years, *RoboCup* competitions always evolved to improve developed approaches and continuously challenge research works. The goal is to raise the level of the competition until they match real-world settings. The research within the Robocup environment is fundamental for bridging AI and games to the real world. In fact, implementing AI algorithms in real-world environments puts researchers in front of several challenges that must be tackled in order to transfer AI onto physical agents, such as noise in perception, limited environment knowledge, and reduced computational power. All these issues are being addressed in the *RoboCup*, which nowadays represents excellent testbed for researchers that are focusing on the intersection of AI, games, and robotics to transfer state-of-the-art technologies in the real world.

References

[1] F.-H. Hsu, "IBM's deep blue chess grandmaster chips," *IEEE Micro*, vol. 19, no. 2, pp. 70–81, Mar. 1999.

[2] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016. [Online]. Available: http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

[3] C. Berner *et al.*, "Dota 2 with large scale deep reinforcement learning," 2019, *arXiv:1912.06680*.

[4] M. Consalvo, K. Mitgutsch, and A. Stein, *Sports Videogames*. Evanston, IL, USA: Routledge, 2013.

[5] S. K. Chalup, T. Niemüller, J. Suthakorn, and M. Williams, Eds., *RoboCup 2019: Robot World Cup XXIII*. Berlin, Germany: Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-35699-6

[6] A. K. Mackworth, "On seeing robots," Univ. British Columbia, Vancouver, BC, Canada, Tech. Rep., pp. 1–13, 1993.

[7] D. Nardi, I. Noda, F. Ribeiro, P. Stone, O. V. Stryk, and M. Veloso, "RoboCup soccer leagues," *AI Mag.*, vol. 35, no. 3, pp. 77–85, 2014. [Online]. Available: https://www.aaai.org/ojs/index.php/aimagazine/article/view/2549/2442

[8] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.

[9] L. De Silva and H. Ekanayake, "Behavior-based robotics and the reactive paradigm a survey," in *Proc. 11th Int. Conf. Comput. Inf. Technol.*, 2008, pp. 36–43.

[10] G. F. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. London, U.K.: Pearson Educ., 2005.

[11] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28593, Apr. 2018.

[12] S. C. Bakkes, P. H. Spronck, and H. J. V. D. Herik, "Opponent modelling for case-based adaptive game AI," *Entertainment Comput.*, vol. 1, no. 1, pp. 27–37, 2009.

[13] M. Asada *et al.*, "The RoboCup physical agent challenge: Goals and protocols for phase I," in *RoboCup-97: Robot Soccer World Cup I*, Berlin, Germany: Springer, 1998, pp. 42–61.

[14] M. Risler and O. V. Stryk, "Formal behavior specification of multi-robot systems using hierarchical state machines in XABSL," in *Proc. Workshop Formal Models Methods Multi-Robot Syst.*, 2008, pp. 12–16.

[15] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning and Acting*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[16] Z.-H. Zhou, Y. Yu, and C. Qian, *Evolutionary Learning: Advances in Theories and Algorithms*. Berlin, Germany: Springer, 2019.

[17] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: A review of recent research," *Adv. Robot.*, vol. 31, no. 16, pp. 821–835, 2017.

[18] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *CoRR*, 2016, *arXiv:1509.02971*.

[19] V. A. Ziparo, L. Iocchi, D. Nardi, P. F. Palamara, and H. Costelha, "Petri net plans: A formal model for representation and execution of multi-robot plans," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.-Volume 1*, 2008, pp. 79–86.

[20] P. MacAlpine, F. Barrera, and P. Stone, "Positioning to win: A dynamic role assignment and formation positioning system," in *Proc. Workshops 26th AAAI Conf. Artif. Intell.*, 2012, pp. 190–201.

[21] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa, "Communication and coordination among heterogeneous mid-size players: Art99," in *Robot Soccer World Cup*. Berlin, Germany: Springer, 2000, pp. 86–95.

[22] M. T. Spaan *et al.*, "High level coordination of agents based on multiagent Markov decision processes with roles," in *Proc. Int. Conf. Intell. Robots Syst.*, 2002, pp. 66–73.

[23] F. W. Trevizan and M. M. Veloso, "Learning opponent's strategies in the RoboCup small size league," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2010.

[24] J. M. C. Ocana, F. Riccio, R. Capobianco, and D. Nardi, "Cooperative multi-agent deep reinforcement learning in a 2 versus 2 free-kick task," in *Robo Cup 2019: Robot World Cup XXIII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 44–57.

[25] V. D. Giambattista, M. Fawakherji, V. Suriani, D. D. Bloisi, and D. Nardi, "On field gesture-based robot-to- robot communication with NAO soccer players," in *RoboCup 2019: Robot World Cup XXIII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 367–375.

[26] Y. Adachi, M. Ito, and T. Naruse, "Online strategy clustering based on action sequences in RoboCupSoccer small size league," *Robotics*, vol. 8, no. 3, pp. 58–76, 2019.

[27] Y. Suzuki and T. Nakashima, "On the use of simulated future information for evaluating game situations," in *Robo Cup 2019: Robot World Cup XXIII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 294–308.

[28] E. V. Postnikov, S. A. Belyaev, A. V. Ekalo, and A. A. Shkulev, "Application of fuzzy state machines to control players in virtual soccer simulation," in *Proc. IEEE Conf. Russian Young Researchers Elect. Electron. Eng.*, Jan. 2019, pp. 291–294.

[29] M. Abreu, L. P. Reis, and H. L. Cardoso, "Learning high-level robotic soccer strategies from scratch through reinforcement learning," in *Proc. IEEE Int. Conf. Auton. Robot Syst. Competitions*, Apr. 2019, pp. 1–7.

[30] A. Larik and S. Haider, "A framework based on evolutionary algorithm for strategy optimization in robot soccer," *Soft Comput.*, vol. 23, no. 16, pp. 7287–7302, Aug. 2019. [Online]. Available: https://doi.org/10.1007/s00500-018-3376-6

[31] T. R. F. da-Silva, "Humanoid low-level skills using machine learning for RoboCup," 2019, *repositorio-aberto.up.pt*.

[32] Z. Chen *et al.*, "Champion team paper: Dynamic passing-shooting algorithm of the RoboCup soccer SSL 2019 champion," in *Robo Cup 2019: Robot World Cup XXIII*. Cham, Switzerland: Springer Int. Publishing, 2019, pp. 479–490.

[33] T. Röfer *et al.*, "B-Human team report and code release 2019," 2019. [Online]. Available: http://www.b-human.de/downloads/publications/2019/CodeRelease2019.pdf

[34] M. Poppinga and M. Bestmann, "DSD-dynamic stack decider a lightweight decision making framework for robots and software agents," *Int. J. Social Robotics*, 2021.

[35] P. Allgeuer and S. Behnke, "Hierarchical and state-based architectures for robot behavior planning and control," *Proc. 8th Workshop Humanoid Soccer Robots, Int. Conf. Humanoid Robots (Humanoids)*, Atlanta, USA, 2013.

[36] O. Aşik, B. Görer, and H. L. Akin, "End-to-end deep imitation learning: Robot soccer case study," in *Robo Cup 2018: Robot World Cup XXII*. Cham, Switzerland: Springer Int. Publishing, 2019, pp. 137–149.

[37] R. Dias, B. Cunha, J. L. Azevedo, A. Pereira, and N. Lau, "Multi- robot fast-paced coordination with leader election," in *RoboCup 2018: Robot World Cup XXII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 19–31.

[38] Y. Schwab and M. Veloso, "Learning skills for small size league Robo Cup," in *RoboCup 2018: Robot World Cup XXII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 83–95.

[39] T. Pomas and T. Nakashima, "Evaluation of situations in Robocup 2d simulations using soccer field images," in *Robo Cup 2018: Robot World Cup XXII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 275–286.

[40] W. B. Watkinson and T. Camp, "Training a RoboCup striker agent via transferred reinforcement learning," in *Robo Cup 2018: Robot World Cup XXII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 109–121.

[41] T. Fukushima, T. Nakashima, and H. Akiyama, "Mimicking an expert team through the learning of evaluation functions from action sequences," in *Robo Cup 2018: Robot World Cup XXII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 170–180.

[42] J. A. Iglesias, A. Ledezma, and A. Sanchis, "Opponent modeling in RoboCup soccer simulation," in *Advances in Physical Agents*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 303–316.

[43] P. MacAlpine, F. Torabi, B. Pavse, J. Sigmon, and P. Stone, "UT Austin Villa: Robo Cup 2018 3D simulation league champions," in *RoboCup 2018: Robot World Cup XXII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 462–475.

[44] K. Genter, T. Laue, and P. Stone, "Three years of the RoboCup standard platform league drop-in player competition," *Auton. Agents Multi-Agent Syst.*, vol. 31, no. 4, pp. 790–820, 2017.

[45] C. Rizzi, C. G. Johnson, and P. A. Vargas, "Fear learning for flexible decision making in robocup: A discussion," in *RoboCup 2017: Robot World Cup H. XXI*, Cham, Switzerland: Springer Int. Publishing, 2018, pp. 59–70.

[46] T. Röfer, "Cabsl-c-based agent behavior specification language," in *Robot World Cup*. Berlin, Germany: Springer, 2017, pp. 135–142.

[47] T. Röfer, T. Laue, A. Hasselbring, J. Richter-Klug, and E. Röhrig, "B-Human 2017—Team tactics and robot skills in the standard platform league," in *RoboCup 2017: Robot World Cup XXI*, Cham, Switzerland: Springer Int. Publishing, 2018, pp. 461–472.

[48] C. Celemin, R. Perez, J. R.-D. Solar, and M. Veloso, "Interactive machine learning applied to dribble a ball in soccer with biped robots," in *RoboCup 2017: Robot World Cup XXI*, Cham, Switzerland: Springer Int. Publishing, 2018, pp. 363–375.

[49] L. de-Koning, J. P. Mendoza, M. Veloso, and R. V. de-Molengraft, "Skills, tactics, and plays for distributed multi-robot control in adversarial environments," in *RoboCup 2017: Robot World Cup XXI*, Cham, Switzerland: Springer Int. Publishing, 2018, pp. 277–289.

[50] M. Hofmann and T. Seeland, "Situation-dependent utility in extended behavior networks," in *Robo Cup 2017: Robot World Cup XXI*, Cham, Switzerland: Springer Int. Publishing, 2018, pp. 216–227.

[51] O. M. Obst, F. Schmidsberger, and F. Stolzenburg, "Analysing soccer games with clustering and conceptors," in *RoboCup 2017: Robot World Cup H. XXI*, Cham, Switzerland: Springer Int. Publishing, 2018, pp. 120–131.

[52] P. MacAlpine, F. Torabi, B. Pavse, and P. Stone, "UT Austin Villa: Robocup 2019 3D simulation league competition and technical challenge champions," in *Robo Cup 2019: Robot World Cup XXIII*, Cham, Switzerland: Springer Int. Publishing, 2019, pp. 540–552.

[53] N. Fang *et al.*, "Multi-robot coordination strategy for 3 vs. 3 teen-sized humanoid robot soccer game," in *Proc. Int. Autom. Control Conf.*, Nov. 2017, pp. 1–6.

[54] H. M. Le, Y. Yue, P. Carr, and P. Lucey, "Coordinated multi-agent imitation learning," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1995–2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=3305381.3305587

[55] H. Farazi *et al.*, "Robocup 2016 humanoid teensize winner nimbro: Robust visual perception and soccer behaviors," in *Robo Cup 2016: Robot World Cup XX*, Cham, Switzerland: Springer Int. Publishing, 2017, pp. 478–490.

[56] H. Akiyama, M. Tsuji, and S. Aramaki, "Learning evaluation function for decision making of soccer agents using learning to rank," in *Proc. Joint 8th Int. Conf. Soft Comput. Intell. Syst. 17th Int. Symp. Adv. Intell. Syst.*, 2016, pp. 239–242.

[57] H. Mellmann, B. Schlotter, and C. Blum, "Simulation based selection of actions for a humanoid soccer-robot," in *Robo Cup 2016: Robot World Cup XX*, Cham, Switzerland: Springer Int. Publishing, 2017, pp. 193–205.

[58] F. Riccio, R. Capobianco, and D. Nardi, "Using Monte Carlo search with data aggregation to improve robot soccer policies," in RoboCup 2016: Robot World Cup XX, Cham, Switzerland: Springer Int. Publishing, 2017, pp. 256–267.

[59] Y. Adachi, M. Ito, and T. Naruse, "Classifying the strategies of an opponent team based on a sequence of actions in the Robocup SSL," in *Robo Cup 2016: Robot World Cup XX*, Cham, Switzerland: Springer Int. Publishing, 2017, pp. 109–120.

[60] J. Henrio, T. Henn, T. Nakashima, and H. Akiyama, "Selecting the best player formation for corner-kick situations based on Bayes' estimation," in *Robo Cup 2016: Robot World Cup XX*, Cham, Switzerland: Springer Int. Publishing, 2017, pp. 428–439.

[61] J. P. Mendoza *et al.*, "Selectively reactive coordination for a team of robot soccer champions," in *Proc. 30th AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 3354–3360.

[62] J. P. Mendoza, R. Simmons, and M. Veloso, "Online learning of robot soccer free kick plans using a bandit approach," in *Proc. 26th Int. Conf. Automated Plan. Scheduling*, vol. 26, no. 1, 2016, pp. 504–508.

[63] D. L. Leottau, J. R.-D. Solar, P. MacAlpine, and P. Stone, "A study of layered learning strategies applied to individual behaviors in robot soccer," in *RoboCup 2015: Robot World Cup XIX*, Cham, Switzerland: Springer Int. Publishing, 2015, pp. 290–302.

[64] F. Riccio, E. Borzi, G. Gemignani, and D. Nardi, "Context-based coordination for a multi- robot soccer team," in *RoboCup 2015: Robot World Cup L. XIX*, Cham, Switzerland: Springer Int. Publishing, 2015, pp. 276–289.

[65] X. Li and X. Chen, "Fuzzy inference based forecasting in soccer simulation 2D, the Robocup 2015 soccer simulation 2D league champion team," in *Robo Cup 2015: Robot World Cup XIX*, Cham, Switzerland: Springer Int. Publishing, 2015, pp. 144–152.

[66] J. G. Masterjohn, M. Polceanu, J. Jarrett, A. Seekircher, C. Buche, and U. Visser, "Regression and mental models for decision making on robotic biped goalkeepers," in *RoboCup 2015: Robot World Cup XIX*, Cham, Switzerland: Springer Int. Publishing, 2015, pp. 177–189.

[67] K. Yasui, K. Kobayashi, K. Murakami, and T. Naruse, "Analyzing and Learning an opponent's strategies in the Robocup small size league," in *Robo Cup 2013: Robot World Cup XVII*, Berlin, Germany: Springer, 2014, pp. 159–170.

[68] A. J. R. Neves, F. Amaral, R. Dias, J. Silva, and N. Lau, "A new approach for dynamic strategic positioning in RoboCup middle-size league," in *Progress in Artificial Intelligence*, Cham, Switzerland: Springer Int. Publishing, 2015, pp. 433–444.

[69] M. Yoon, "Developing basic soccer skills using reinforcement learning for the RoboCup small size league," Ph.D. dissertation, Dept. Industrial Eng. Stellenbosch Univ., Stellenbosch, South Africa, 2015.

[70] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, 1999.

[71] Y. Xu and H. Vatankhah, "Simspark: An open source robot simulator developed by the RoboCup community," in *Robot Soccer World Cup*. Berlin, Germany: Springer, 2013, pp. 632–639.

[72] M. Lötzsch, M. Risler, and M. Jüngel, "XABSL—A pragmatic approach to behavior engineering," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5124–5129.

[73] M. Ponsen, H. Munoz-Avila, P. Spronck, and D. W. Aha, "Automatically generating game tactics through evolutionary learning," *AI Mag.*, vol. 27, no. 3, pp. 75–75, 2006.

[74] R. D. L. Briandais, "File searching using variable length keys," *in Proc. Western Joint Comput. Conf.*, 1959, pp. 295–298. [Online]. Available: http://doi.acm.org/10.1145/1457838.1457895

[75] J. Ledoux, "The emotional brain, fear, and the amygdala," *Cellular Mol. Neurobiol.*, vol. 23, pp. 727–38, 2003.

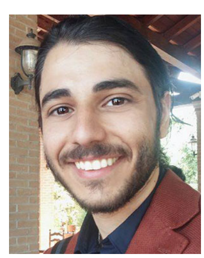
[76] X. Wang *et al.*, "The goalkeeper strategy of Robo Cup MSL based on dual image source," in *RoboCup 2015: Robot World Cup XIX*, Cham: Springer Int. Publishing, 2015, pp. 165–174.

[77] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016

[78] M. Asada, T. Balch, A. Bonarini, A. Bredenfeld, and S. Gutmann, "Middle size robot league rules and regulation," in *Proc. Int. MSL Workshop*, 2016.

[79] P. MacAlpine, K. Genter, S. Barrett, and P. Stone, "The RoboCup 2013 drop-in player challenges: Experiments in Ad Hoc teamwork," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 382–387.

[80] M. Paetzel, J. Baltes, and R. Gerndt, "Robots as individuals in the humanoid league," in *RoboCup 2016: Robot World Cup S. XX*, Cham, Switzerland: Springer Int. Publishing, 2017, pp. 339–346.



Emanuele Antonioni received the Graduate degree in computer science from the Faculty of Computer Science, University of Rome Tor Vergata, Rome, Italy, in 2016, and the master's degree in artificial intelligence and robotics from the Sapienza University of Rome, Rome, Italy, in 2019. He has been working toward the Ph.D. degree in computer engineering at Sapienza University of Rome, since 2019.

His main research interests include reinforcement learning, deep learning, representation learning, and automated planning.

Mr. Antonioni is a member of the SPQRTeam of RoboCup@Soccer since 2016.

Vincenzo Suriani received the B.Sc. degree in computer engineering and the M.Sc. degree in artificial intelligence and robotics from the Sapienza University of Rome, Rome, Italy, where he is currently working toward the Ph.D. degree with the Department of Computer, Control, and Management Engineering.

From 2016, he holds the position of the Software Development Leader of RoboCup SPL team, Sapienza University of Rome. His main research interests include multiagent systems, often represented by humanoid robots, and continuous and imitation learning processes.

Francesco Riccio received the B.Eng. degree in computer science from the University of Siena, Siena, Italy in 2012, and both the M.Sc. degree (*cum laude*) in artificial intelligence and robotics in 2014, and the Ph.D. degree in engineering in computer science, from the Department of Computer, Control, and Management Engineering "Antonio Ruberti" (DIAG), Sapienza University of Rome, Rome, Italy in 2018. He is currently a Research Scientists with RADiCAL-AI, New York, NY, USA, and a researcher affiliated with the Sapienza University of Rome. He was a visiting student with the University of Washington, Seattle, WA, USA, at the beginning of the academic year 2016–2017. His research revolves around the idea of creating autonomous robots which are able to dynamically and continuously adapt their behavior to the environment.

Daniele Nardi received the Graduated degree in electrical engineering at Politecnico di Torino in 1981. He has been a Full Professor with the Faculty of Ingegneria dell'Informazione, Informatica, Statistica, Department of Computer, Control, and Management Engineering "A. Ruberti," Sapienza University of Rome, Rome, Italy, since 2000. He is currently the referent for the Master curriculum in Artificial Intelligence and Robotics, and he is teaching the course Artificial Intelligence of the master, since several years. He leads the research laboratory "Cognitive Robot Teams," addressing different research topics: Cognitive Robotics, Localization, Navigation, Perception, Cooperation in multirobot systems, Human Robot Interaction, multimodal interfaces and speech. The scientific and technical achievements are deployed in manifold application domains: Ambient Intelligence and robots to support elderly people, Disaster Response robots to explore and gather information from the environment, Cultural Heritage, Precision Agriculture, Soccer Player robots for RoboCup competitions. He has a publication record with several contributions in Artificial Intelligence and Robotics (H-index 50). He has been the principal investigator of several collaborative projects funded by FP7, H2020, and several other research funding institutions.

Mr. Nardi is a EurAI Fellow, and was a President of RoboCup Federation from 2011 to 2014.