Distributed Robotics Project: Robot Soccer Simulation

Hayden Bader, Bill Sun, Gao Tang, Branch Vincent, and Yue Wang

Abstract— **In this paper, we discuss the design and implementation of a robotic soccer team that would compete in RoboCup's annual 3D Simulated Soccer League. First, we briefly introduce the motivation behind this competition. Second, we investigate previous approaches to this challenging problem, focusing on the progression of UT Austin's system which our implementation is based on. Third, we outline the entire design of our system and elaborate on the parts that were successfully implemented. Lastly, we evaluate our system's performance and suggest areas for further improvement.**

I. INTRODUCTION

A. Robot Soccer

Artificial intelligence has made large strides in recent years. In 1997, IBM's Deep Blue defeated the reigning world champion chess player Garry Kasparov. In 2011, IBM's Watson beat former winners at Jeopardy. In 2016, Google's AlphaGo defeated 18-time world champion Lee Sedol in Go. Each of these accomplishments were landmark achievements in AI. However, the problem domain of board and quiz games is relatively simplistic compared to human sports. Robot soccer is one such difficult problem because it involves integrating a wide range of technologies and achieving a number of technological breakthroughs [1]. The key differences between these two problem domains are summarized in Table I.

TABLE I
DIFFERENCES BETWEEN BOARD GAMES AND SOCCER

	Board Game	Soccer
Environment	Static	Dynamic
State Change	Turn taking	Real time
Info Accessibility	Complete	Incomplete
Sensor Readings	Symbolic	Non-symbolic
Control	Central	Distributed

The robot soccer problem is part of a broader category of multi-agent collaboration. Each team has its own goal and these goals are incompatible. Conversely, the problem can be divided into two subproblems: physical and mental. Physically, the robots need an understanding of mechanics and excellent motor control to dribble, kick, and perform more complicated actions. Mentally, the robots need to cooperate and intelligently react to different situations. This paper focuses on the later problem, which is arguably in need of the most development. Recent developments suggest the promising idea of learning behaviors, since it is infeasible to consider all situations.

B. RoboCup

RoboCup was founded in 1997 with the goal of promoting AI research through a fun, yet challenging, annual competition. The competition provides a common task to evaluate various algorithms and architectures, thus facilitating growth. Its ultimate objective is to have an autonomous robotic team surpass human-level performance by beating the most recent human world champion team. Since robot soccer is far from reaching this level of play, RoboCup attempts to motivate and enable progress in a problem domain that is still very young and rich with unanswered questions. At the same time, relevant breakthroughs would have much broader implications for not only AI and but also for society.

There are 5 different leagues within RoboCup soccer: (1) Simulation League, (2) Small Size League, (3) Middle Size League, (4) Standard Platform League, and (5) Humanoid League. This paper addresses the 3D Simulation League. The University of Texas at Austin traditionally participates in this league and has been named world champion 6 of the last 7 times. Thus, our design and implementation is based on theirs, as they have made significant contributions in terms of both control and distributed behaviors [2], [3].

II. PREVIOUS WORK

Two key papers [4], [5] from early RoboCup teams introduced algorithms for team formations, role allocation, and periodic team synchronization. These initial results formed a solid basis, on which subsequent teams developed their systems. Notably, UT Austin's history of success warrants a detailed review of its techniques, as well as those from other successful teams.

A. UT Austin

In 2011, the UT Austin team developed a learning-based omni-directional walk engine [6]. As a result, the team could walk and dribble the ball faster than other teams, likely contributing to it winning the championship. This was achieved by formulating the problem as an optimization problem with 40 real-valued parameters. Then, they applied a hierarchical approach and sequentially optimized 3 subsets of parameters, as shown in Figure 1. For formation and role assignment, the team used a 3-step approach. First, the formation is determined. Then, each player computes the best role assignments based on its world view. Finally, a coordination step is applied to choose among the players' suggestions.

In 2012, a more sophisticated software architecture was developed to achieve three goals: (1) consistency, (2) flexibility, and (3) debugability. Consistency was desired to keep

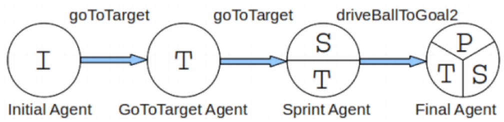

Fig. 1. UT Austin's optimization procedure for its walk engine. Parameter sets are **I**nitial, goTo**T**arget, **S**print, and **P**ositioning. [7]

the core system identical, regardless of the platform (robot, simulator, and debug tool). Flexibility was designed for so that plug and play modules only used local memory while common memory existed for communication. Debugability was needed to save the states at each time step. In addition, the walk engine was modified in two ways: (1) keep each robot's arms behind its back for collision avoidance and (2) lower the height of each robot for stability. The kick engine was also upgraded so static, standing kicks required less interpolation time and walking kicks achieved more types of kicking movements. Lastly, the game strategy was updated so the robot behavior behavior adapted to the distance from the opponents.

In 2014, new parameter sets were added to the walk engine for robots to stop before the ball to avoid overshooting [8]. Cooperative behaviors were also added to enable indirect kickoffs. More importantly, indirect passing was accomplished by broadcasting the predicted position of a pass and then assigning a teammate to meet the pass.

In 2015, UT Austin developed a set of optimized kicks based on different distances [9]. The team increased the precision of passing and shooting while avoiding overshooting the goal, which occurred frequently the year before. New set plays, including kickoffs and corners, were also added, illustrated by Figures 2 and 3. In the kickoff, the robot tried to pass sideways for an opportunity to immediately shoot. If both left and right side agents were surrounded by opponents, then the robot would simply pass backwards first before proceeding. In the offensive corner kick, the ball was passed to one of three teammates positioned at midfield. Lastly, in order to make the agents behave more intelligently, a logistic regression classifier was trained to predict the probability of a successful kick. The team collected training data by playing many games against a common opponent, in which agents were instructed to always attempt to kick the ball.

Fig. 2. Kickoff set plays

B. Other Work

There are many unique approaches to the competition each year. For example, in [10], the CMDragons team present a core coordination algorithm in offense and defense. With this

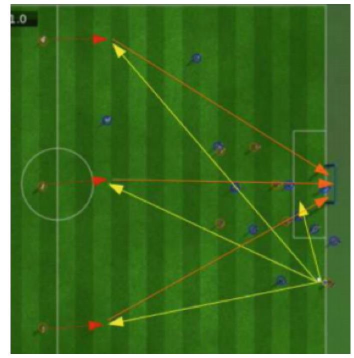

Fig. 3. Offensive corner kick set plays

algorithm, the team won the championship in 2015 for the small size league. The algorithm has a hierarchical structure. In the first layer, the number of robots assigned to defense and offense subteams is based on the state of the game. In the second layer, coordinated plans are created for each subteam. In the third layer, each robot selects actions consistent with the coordinated plan and tries to maximize the probability of success. In the last layer, each robot executes these actions through various reusable skills.

In [11], layered learning strategies are applied in robot soccer. Layered learning is a hierarchical machine learning paradigm where a complex behavior is learned through a series of incrementally trained subtasks. Compared with decentralized reinforced learning, the layered learning methods have higher computational efficiency.

COrrective Advice Communicated by Humans (COACH) is a technique that allows non-expert humans to shape a policy through corrective advice [12]. Compared with computational and machine learning methods, this method has lower cost.

III. SYSTEM DESIGN

Our system builds upon an understanding of modern soccer that is adjusted to account for the realistic capabilities of standard robotic soccer agents. In order to do this, we utilize previous works to reproduce motions and form a strategy that utilizes the robotic agents' abilities. This strategy aims to put the ball in a position where it is unlikely for the opponent's team to score while allowing our team the flexibility to maneuver around the opposing defense and score. This strategy relies heavily on solid goalkeeping and defense to shutdown opponents' strategies and a mobile offense that attempts to overwhelm structured defense and outrun clustered ball-chasers.

A. Goalie

One of the simplifying assumptions our system makes is that the best defensive strategy is one in which the goalie never has to interact with the ball. Barring this, though, the goalie serves as the last line of defense to prevent the ball from crossing its team's goal line. Since it has such a critical role, its positioning needs to be handled precisely. Whenever the goalie chooses to move to a position, the chosen position should allow the goalie to quickly interpose itself between the ball and the goal for both current and potential future ball locations. The simplest way of accomplishing this is to tie the goalie to a testing-defined region near the goal. The goalie is constrained to travel within this region and to seek a position that gives the ball the smallest "view" of the goal. In standard soccer, this strategy works up until a point. If an opposing player has a "breakaway" or enough unimpeded time, they can maneuver to a position outside of the goalie's predefined region and shoot the ball into an opening of the goal that the goalie is not covering.

As a result, the goalie also requires the ability to be more aggressive if need be. We attempt to express this in code by defining conditions in which a goalie should shift from a passive blocker to an aggressive protector. Our system purports that this transition should occur when the goalie is the nearest to the ball of the teammates that are behind the ball, and the ball is heading towards our goal. In other words, if the ball gets past both the offense and the defense, the goalie will attempt to run towards it. Once in this state, the goalie's objective is to clear the ball, ideally delaying the attackers. If enough time can be regained, our team should be able to reset its positioning and defend even more effectively. Notably, this is a strategy that professional soccer teams also employ when under great pressure. Ultimately, most of this functionality was present in our final design. However, the region in which the goalie was allowed to roam was limited to the goal line and testing of multiple regions did not occur.

B. Defense

The defense's job is to keep the ball from reaching the goalie. However, this can be accomplished by following similar principles to those that the goalie follows. Namely, the defense should patrol a region in such a way that closes off the easiest and most common avenues for transporting the ball from its current position to our team's goal. Additionally, though, the defense should be cautious in its defensive tactics. In robot soccer, oftentimes only one robot is necessary to pressure an opposing robot enough to cause a turnover. As such, the defensive strategy we put forward has the defense assemble a formation that attempts to cut off attacking robots while closing off passing lanes. Specifically, when defending, our robots aim to dispatch one robot at a time to chase the ball while the remaining reassemble an adjusted defensive structure. Furthermore, if the chasing robot is able to separate the ball from the attacker, the nearest defensive robot attempts to pass the ball to the area where our offense is densest. This attempts to account for the case in which a chasing robot is able to knock the ball to another defensive player. However, if the chasing defender becomes incapacitated or begins to lag too far behind the ball, another defensive member will be dispatched to chase the ball, and the original chaser will be told to get back in formation. This idea takes advantage of the fact that the robots are most effective heading straight towards the ball and that even opposing robots often have difficulty redirecting the ball by a great margin. Finally, if at any time the ball is traveling away from the goal, defense members begin to reset to their initial configuration. Due to time constraints, our solution ended up implementing only parts of this design. Specifically, the defenders do not regroup if one aggressively attacks the ball, and instead of kicking the ball towards the majority of the offense, the defense attempts to clear the ball to midfield.

C. Offense

The design of the offense presents the greatest compromise between standard professional soccer considerations and expectations from other teams. In professional soccer, strategies often utilize the sides of the field to move the ball forward before attempting to "cross" the ball to a player in a better scoring position near the center of the field. This requires a large degree of coordination, cooperation, and prediction between agents which may be unnecessary considering the limited speed with which humanoid robot agents can actually move. Instead, we propose a system that combines some of the principles of the professionals with some principles for dealing with swarms. This can be summarized in three behaviors: (1) chase the ball, (2) prepare for ball rebounds that occur when opposing agents collide, and (3) capitalize on free space.

Chasing the ball and preparing for ball rebounds can be summarized by our consideration of an attacker and strong supporters. In youth soccer, where swarms tend to form, one of the most common strategies is to have a few players wait and collect the ball when it inevitably escapes from the group attempting to control the ball. From there, this pattern can repeat itself until something changes.

As such, the weak supporter is meant to mimic a player stretching the field and ensuring that the ball will travel around clusters of opposing players. Though this is not fully implemented in our initial submission, weak supporters are directed to find empty space, biased towards the opponents' goal while remaining within passing range. In soccer, the ball tends to travel faster than the players, so assuming play proceeds as predicted in the aforementioned swarm example, the weak supporters should provide a way to both quickly move the ball away from a swarm and upfield. Additionally, if swarming does not occur, it still allows the weak supporters to move upfield and receive passes that can thwart a team relying on zoning or positioning.

Ultimately, both weak and strong supporters were not implemented to the extent described above. Instead, weak supporters followed a potential field described further below. Strong supporters did assemble into a formation but were unable to intelligently pass further upfield. Instead, they used simple checks to determine who could be passed to. These checks did not consider opponents.

IV. SYSTEM IMPLEMENTATION

Our implementation was based on three main components: (1) goalie, (2) defense, and (3) offense. Each of these components are described in more detail below.

A. Goalie

The goalie is a single player who has two states: `ClearBall` and `DefendGoal`. The `ClearBall` state occurs when the goalie is within `GOALIE_CLEARANCE_DISTANCE` $\equiv$ 1 m of the ball. In this state, the goalie will go to the ball and kick it to midfield. If not in this state, the goalie is in the `DefendGoal` state. In this state, the goalie moves along the goal line to face and align with the ball. More specifically, the position is the intersection of the goal line with the line from the back of the goal to the ball. This way, the goalie positions itself to intersect a potential shot to the center of the goal.

B. Defense

The defense, which consists of 3 players, has the same two states as the goalie: `ClearBall` and `DefendGoal`. The `ClearBall` state occurs when the defender is within `DEFENDER_CLEARANCE_DISTANCE` $\equiv$ 5 m of the ball. In this state, the defender will go to the ball and kick it to midfield. If not in this state, the defender is in the `DefendGoal` state. In this state, the defender forms part of an arc around its goal while facing the ball. The arc's midpoint intersects the line from the goal line's midpoint to the ball. This way, the defenders form a protective barrier around our goal, positioned to intersect a potential shot to the goalie. See Figure 4.

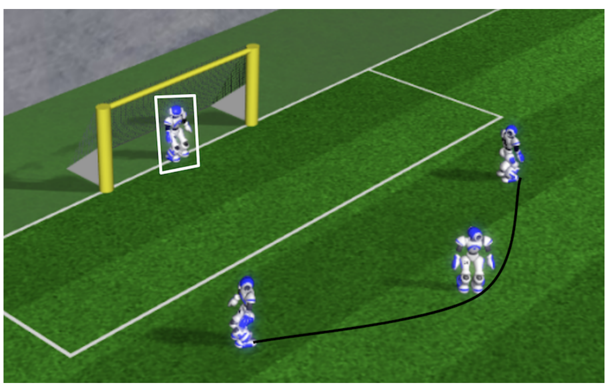

Fig. 4. Goalie (white) and defense (black)

C. Offense

The offense, which consists of the remaining 7 players, has three different roles: (1) attacker, (2) strong supporter, and (3) weak supporter. These roles are assigned dynamically, as described below and illustrated in Figure 5.

The attacker is assigned to the player closest to the ball. Our offense is centered around this role, which is designed to score points. It has four states: `GainPossession`, `Attack`, `Pass`, and `Dribble`. The `GainPossession` state occurs when the attacker does not possess the ball, defined as when the distance to the ball is greater than `POSSESSION_DISTANCE` $\equiv$ 2 m. In this state, the attacker will go towards the ball in an attempt to gain possession. The `Attack` state occurs when the attacker possesses the ball and is within `ATTACK_DISTANCE` $\equiv$ 3 m to the opponent's goal. In this state, the attacker will shoot the ball at the center of the opponent's goal. The `Pass` state occurs when the attacker possesses the ball and another player, within `PASSING_DISTANCE` $\equiv$ 5 m, is closer to the opponent's goal. In this state, the attacker will pass to this player. Thus, the player potentially in the best position to score will have the ball. Lastly, the `Dribble` state occurs when the attacker possesses the ball and is not in the other states. In this state, the attacker will dribble the ball towards the opponent's goal, therefore making progress to eventually score.

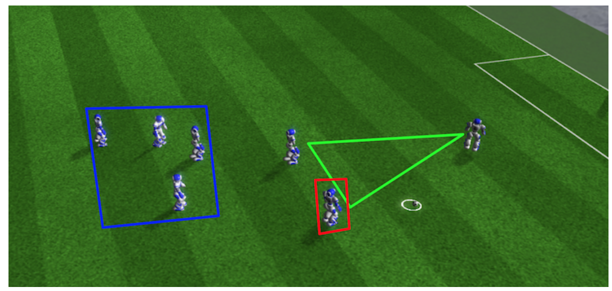

Fig. 5. Offensive team consisting of attacker (red), strong supporters (green), and weak supporters (blue)

The strong supporters are assigned to the second and third closest players to the ball. This role serves to support the attacker and aid in scoring points. Each strong supporter always follows behind the attacker, forming a triangle.

The weak supporters are assigned to the remaining offensive players. This role is designed to support the attacker by providing options to pass the ball. Each weak supporter is controlled by a potential field, which is repulsed by teammates and opponents while occasionally attracted to the ball and opponent's goal. This potential field is expressed more precisely below.

$$F_{\text{teammates}} = \begin{cases} -v_t & ||v_t|| \leq \texttt{AVOID\_DISTANCE1} \\ 0 & \text{otherwise} \end{cases}$$

$$F_{\text{opponents}} = \begin{cases} -1.5\, v_o & ||v_o|| \leq \texttt{AVOID\_DISTANCE2} \\ 0 & \text{otherwise} \end{cases}$$

$$F_{\text{ball}} = \begin{cases} 0 & ||v_b|| \leq \texttt{PASSING\_DISTANCE} \\ 1.5\, v_b & \text{otherwise} \end{cases}$$

$$F_{\text{goal}} = \begin{cases} -2\, v_g & ||v_g|| \leq \texttt{ATTACK\_DISTANCE} \\ v_g & \text{otherwise} \end{cases}$$

Note that v_t is the vector from the player to the teammate, and v_i is defined similarly for all other forces. This potential field was designed so that teammates will be relatively spread out from each other and, more strongly, from opponents. In addition, teammates will be drawn toward the opponent's goal, except when they are too close, and those outside the passing range will be drawn towards the ball.

V. SYSTEM PERFORMANCE

Our overall performance is quite good compared to other teams in the class. However, we also observed several problems from the simulation and class competition. In summary, there are many conditions that were not considered in the implementation, so the team did not perform as expected in many situations. In this section, we discuss the problems we observed and their potential solutions.

When attacking, our attacker frequently collided with opposing defenders. This is because we did not consider opponents. One solution for this problem would be adding a force vector to avoid opponents when attacking. Our attack team, consisting of the attacker and strong supporters, was not able to proceed if the main attacker fell down. One solution would be to require that the chosen attacker is standing. If the attacker loses the ball, the attack team should hover around the ball until clear to avoid collisions. Additionally, our current logic to shoot could be improved by optimizing the target to maximize distance from opponents.

When our robots attempted passing, they did not consider a possible interception. A solution here would be passing only if the ball's trajectory is relatively clear. Also, more extensive testing is needed to determine the optimal passing range. Our passing strategy was quite naïve: pass the ball to the opponent closest to the goal but within passing range. This strategy is weak when the receiver is not in a good attack position. One solution would be rewarding quick passing. Another problem we observed is after passing, the receiver is no longer at the intended target. We would incorporate teammates' behavior when passing to solve this kind of problem.

When defending, robots went directly to the ball instead of attempting an interception, resulting in a long time before we regained the ball. Also, we observed that only our pre-assigned defenders actually defended when the ball was near our own goal. Midfielders and attackers should be more attracted to defense when our opponents are highly likely to shoot. We can address this problem by dynamically adjusting the potential field to better control positioning. If the ball is on our side, there should be a weaker force towards opponent's goal.

Our current goalie implementation is simple. Its performance could be improved by adding tracking and interception.

VI. FUTURE WORK

Based on our performance in the class competition, there is still much work to be done in order to achieve a truly adaptive and agile team. To improve our system, we believe that further mimicking human behaviors and formations as a general heuristic may be a promising approach.

One of the improvements may come from formulating and solving optimization problems. For example, we can use optimization techniques to calculate an optimal attacking trajectory. Opponents, rules, and physical limitations are regarded as constraints while the objective of optimization would be minimizing the time taken to score.

We can also try methods from machine learning such as reinforcement learning or deep learning to improve our team's performance. The former is promising when applied to learning how to recover after falling.

Finally, we can adopt adaptive strategies to cope with attackers falling, failing to intercept the ball, and other similar situations.

REFERENCES

[1] H. Kitano, *RoboCup-97: Robot Soccer World Cup I*. Springer Science & Business Media, 1998, vol. 1395.

[2] UT Austin. (2017). Austin villa robot soccer team, [Online]. Available: `https://www.cs.utexas.edu/~AustinVilla` (visited on 12/01/2017).

[3] ——, *Utaustinvilla3d*, version 1.0, Nov. 25, 2017. [Online]. Available: `https://github.com/LARG/utaustinvilla3d`.

[4] P. Stone and M. Veloso, "Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork," *Artificial Intelligence*, vol. 110, no. 2, pp. 241–273, 1999.

[5] D. Vail and M. Veloso, "Multi-robot dynamic role assignment and coordination through shared potential fields," *Multi-robot systems*, pp. 87–98, 2003.

[6] T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, *RoboCup 2011: Robot Soccer World Cup XV*. Springer, 2012, vol. 7416.

[7] P. MacAlpine, S. Barrett, D. Urieli, V. Vu, and P. Stone, "Design and optimization of an omnidirectional humanoid walk: A winning approach at the RoboCup 2011 3D simulation competition," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, Toronto, Ontario, Canada, 2012.

[8] R. A. Bianchi, H. L. Akin, S. Ramamoorthy, and K. Sugiura, *RoboCup 2014: Robot World Cup XVIII*. Springer, 2015, vol. 8992.

[9] L. Almeida, J. Ji, G. Steinbauer, and S. Luke, *RoboCup 2015: Robot World Cup XIX*. Springer, 2016, vol. 9513.

[10] P. Cooksey, S. Klee, and M. Veloso, "Cmdragons 2015: Coordinated offense and defense of the ssl champions," *RoboCup 2015: Robot World Cup XIX*, vol. 9513, p. 106, 2016.

[11] D. L. Leottau, J. Ruiz-del Solar, P. MacAlpine, and P. Stone, "A study of layered learning strategies applied to individual behaviors in robot soccer," in *Robot Soccer World Cup*, Springer, 2015, pp. 290–302.

[12] C. Celemin and J. Ruiz-del Solar, "Interactive learning of continuous actions from corrective advice communicated by humans," in *Robot Soccer World Cup*, Springer, 2015, pp. 16–27.