Multi-Robot Dynamic Role Assignment and Coordination Through Shared Potential Fields

Douglas Vail Manuela Veloso
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
{dvail2, mmv}@cs.cmu.edu

***Abstract*—Role assignment and coordination are difficult issues for multi-robot systems, especially in highly dynamic tasks. Robot soccer is one such task and it provides a unique challenge for multi-robot research. In this paper, we contribute the approach that we successfully developed for CMPACK'02, our team for the RoboCup-2002 Sony legged league. The RoboCup-2002 Sony robots were equipped with wireless communication for the first time this year. We developed an approach for sharing sensed information and effective coordination through the introduction of shared potential fields. The potential fields were based on the positions of the other robots on the team and the ball. The robots positioned themselves on the field by following the gradient to a minimum of the potential field. In principle, our potential functions can be applied to any multi-robot domain. We present the results of the RoboCup-2002 competition, which we won, and we show a post-competition, controlled empirical evaluation to analyze the impact of our algorithm. The results demonstrate that our team using our communication-dependent coordination outperforms a team of individually skilled robots without coordination.**

I. INTRODUCTION

Many open questions remain in the areas of multi-robot coordination and task assignment. How should a group of robots divide tasks among its members? Once roles have been assigned to the robots, how should they position themselves to fulfill their roles without interfering with their teammates. What happens if an agent fails or if the environment changes so that a different robot is more suitable for the task?

In this paper we present a framework for task assignment and coordination for a group of robots in a soccer domain. We show how heuristic bidding functions that use globally shared information may be used to determine which robot is the most suitable for each task. We also describe how obstacle avoidance may be combined with coordination through the use of artificial potential fields. We provide a robust experimental evaluation of our framework in a penalty shot domain showing that communication is important for coordination.

This paper is organized as follows: In the following section, we give a brief overview of the Robocup domain and review related work. Section III is the technical contribution of this paper. It describes the information sharing between robots and presents the framework used for task assignment and coordination. Both empirical results from Robocup-2002 and results from controlled experiments are presented in section IV. Section V concludes the paper.

This research was sponsored by Grant No. DABT63-99-1-0013, by generous support by Sony, Inc., and by a National Physical Science Consortium Fellowship with stipend support from HRL Laboratories. The content of this publication does not necessarily reflect the position of the funding agencies and no official endorsement should be inferred.

II. BACKGROUND

This section provides background information on Robocup and reviews related work in mutli-robot coordination and task assignment.

A. The Robocup Domain

The legged league of Robocup [6], the robot soccer world championship, provides a challenging testbed for multi-agent research. Two teams of quadruped robots compete for two ten minute halves on a small soccer field. The hardware is the same for each team: the commercially available Sony Aibo ERS210 Entertainment Robot. The rules specify that each team must be fully autonomous; no off board computation or human intervention is allowed. For Robocup-2002, the domain was extended to create additional research opportunities. The number of agents on each team was increased from three to four. Wireless communication, in the form of 802.11b wireless network cards, was added. Also, the size of the field was increased by 50% in both directions. This was a major change; formerly robots could detect the ball from across the length of the field. After the size increase this was no longer possible.

The challenges that arise during the game can be divided into two categories: challenges that may be addressed from a single robot perspective and those that arise due to the multi-agent nature of the domain. Single agent tasks include localization, detecting other robots, detecting the ball, as well as motion control.

Robots rely entirely on vision for sensing in these single agent tasks [3]. To simplify matters, the world is color coded. Each robot is dressed in either a red or a blue uniform. The ball and goals are also color coded. To aid localization, six distinct, brightly colored markers are placed around the edge of the field. Despite these aids, soccer is still a difficult task from a single robot perspective; visual processing, the behavior system, and motion control must all run in real time on the same processor [7], [10]. And, as with any physical system, sensor readings and motions commands are rife with noise and uncertainty. The presence of other agents compounds the difficulty of these tasks.

Including other agents fundamentally changes the domain. With other agents, the world is no longer static; even if the robot does not act, the world will continue to change around it. In addition to changing the environment, other agents can also interact directly with the robot. To cite a few examples of this, two robots may become entangled, causing motion commands to have unanticipated effects; the referee may pick up the robot and move it across the field to enforce a penalty; other robots may obscure the ball or the markers that are used for localization. The addition of other robots makes the world dynamic and increases the amount of uncertainty, but perhaps more importantly, the other agents have their own sets of goals. In the case of the agents on the other team, these goals are diametrically opposite to the goals of the agent.

Robocup is an adversarial domain; the environment [in the form of the other team] actively works against the agents. We treat the other team as a part of the environment because we cannot control them directly. The adversarial nature of the domain changes the way agents must approach action selection. In addition to considering the expected payoff of actions, agents must also consider the worst possible outcome; the environment will attempt to steer the game in the direction of that worst case scenario. Agents must choose actions that minimize risk, even if choosing those particular actions reduces their expected payoff.

Next, we review related work before providing an overview of the coordination framework.

B. Related work

Artificial potential fields have long been used for obstacle avoidance [5]. They have a low computational overhead in comparison to higher level approaches such as path planning; they require simple, local knowledge about the environment; and, because they do not require computationally expensive repair, such as replanning, when the environment changes, they are robust in dynamic situations. On the other hand, potential fields have a tendency to guide robots to local rather than global minima. However, in highly dynamic environments such as soccer, this is not a major problem as the world quickly changes and jogs the robot from the local minimum.

In addition to static obstacle avoidance, potential fields may also be used for multi-agent formations and coordination. In [1], [2], Balch *et al* describe how robots can form and maintain formations using only local information to calculate potential fields. They name their approach “social potentials” because the potential functions are calculated using the distances between teammates. In [4], [11], [12], potential fields are used to position robots for particular roles. The potentials encode heuristic information about the environment. This information takes the form of attractive potentials that guide robots to desirable areas of the field. For example, potentials may guide robots to locations near an opponent’s goal to place the robot in a good position to receive a pass.

Before robots can take up positions based on their roles, those roles must be assigned. [4], [8] describe how robots may perform distributed task allocation by calculating their suitability for a task and broadcasting this suitability as a bid to their teammates. This approach resembles a continuous auction where the robot with the highest bid wins the task. If that robot becomes unavailable for some reason, its bid is no longer seen by the other robots and the robot with the next highest bid addresses the task.

III. Task Assignment and Coordination

As described in the introduction, each team consists of four robots with identical capabilities; we are solving a homogeneous agent task assignment problem. One of these robots serves as the goalie. It is the only robot with a fixed role. The other three robots play offense, but the rules do not specify fixed positions for them. We allow these three robots to dynamically switch between predefined, mutually exclusive roles.

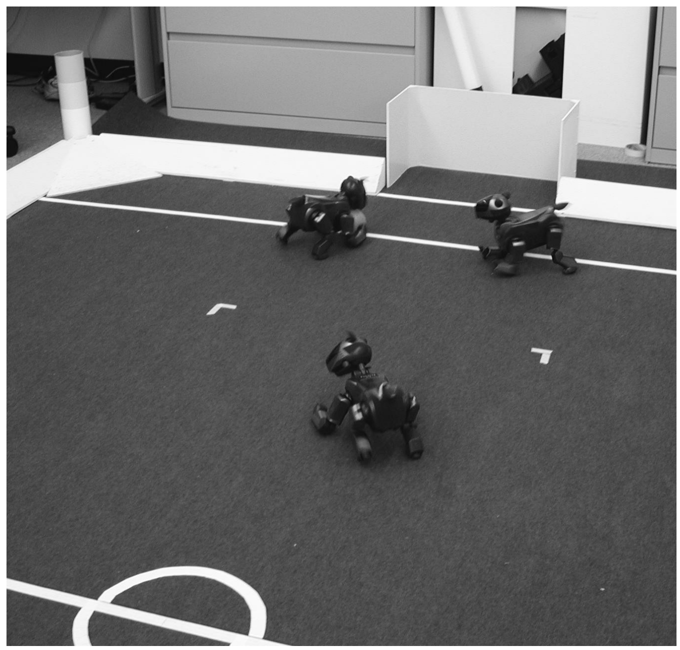

Fig. 1. The primary attacker holds the ball slightly to the left of the other team’s goal. The offensive supporter waits by the right edge of the goal to recover the ball if the primary attacker misses. The defensive supporter positions itself down field of the ball to recover it if the ball moves behind the primary attacker. The cylinder standing at the left corner of the field is one of the markers used for localization.

These roles are a *primary attacker*, which approaches the ball and attempts to move it upfield; an *offensive supporter*, which moves up the field from the primary attacker and positions itself to recover the ball if the primary attacker misses its shot on goal; and a *defensive supporter*, which positions itself down the field from the primary attacker to recover the ball if the other team captures it. Figure 1 shows the robots positioning themselves in these roles.

The three agents negotiate among themselves using a predefined protocol so that a single robot fills each role. In addition, they coordinate with the goalie to avoid approaching the ball while the goalie is clearing it from the defense zone and they avoid collisions with their teammates.

Before providing the details of how the different roles are assigned and how the robots fill those roles, we briefly describe information sharing between teammates.

A. Shared information

In our framework, the robots must communicate in order to coordinate effectively. Coordination methods that rely on local information alone are not feasible in this domain since there are many cases where a robot cannot observe the ball or its teammates. Since a known, small number of robots are collaborating, we chose to use a system of broadcast messages to share information. This approach does not scale to large numbers of robots, but it is very simple to implement and understand.

Twice a second, each robot broadcasts a message to its teammates. This message contains the current position of the robot, according to its localization system, as well as an estimate of the uncertainty in that position. The message also contains the robot's estimation of the ball's position and the uncertainty associated with that measurement. The final pieces of information included are flags indicating whether or not the robot is the goalie and if the robot currently sees the ball. The goalie flag is needed for role assignment since the goalie can never play a different position and is the only robot allowed to clear the ball from the defense zone. The flag indicating whether or not the robot currently sees the ball is used when building a shared world model to avoid incorporating evidence about ball location from robots that do not see it.

A detailed explanation of the shared information and how this information is combined may be found in [9]. Next we describe how this shared information is used to assign roles to different agents and how the agents fill those roles.

B. Role assignment

The three robots playing offense need to be assigned to the roles of primary attacker, offensive supporter, and defensive supporter. Role assignment is done in a fixed, total order. The primary attacker is chosen first, followed by the defensive supporter, and finally the offensive supporter is picked. This order is designed to make the system more robust; if one or two of the robots fails, the remaining member(s) of the team can carry on playing.

All of the robots use a common set of functions to calculate real valued bids for each task. These functions encode heuristic information about the world to return an estimate of how suitable the robot is for a particular task. For example, the bid function for the primary attacker activation takes ball proximity and the relative orientation of the opponents' goal into account. Robots first calculate their own suitability using local information from their world models and then they use the same function to calculate the bids of their teammates using only the shared information provided by each teammate.

It is important to note that only the reported information is used for calculating teammates' bids; in effect the agent doing the calculation is putting itself into the shoes of the agent whose bid is being calculated. If the agent performing the calculation used its own information, it could erroneously assign another agent a different potential than it calculated for itself.

A concrete example of this would be when the agent performing the calculation sees the ball next to the other robot, but the agent whose bid is being calculated does not see the ball (perhaps the ball is behind it or occluded by an opponent). Since the robot that cannot detect the ball is not confident about the ball location, it will assign itself a low bid for the primary attacker role. The agent that sees the ball should not assign its teammate a higher bid than the teammate would pick for itself. And the teammate should not use the shared information from the robot that sees the ball to assign itself a higher value. In the case where an opponent is occluding the ball, the robot does not have a clear path to the ball; although it would be reasonable for it to use the shared information to turn to the ball.

Once each robot calculates the bids for itself and each of its teammates, it compares them. If it has the highest bid for the role being assigned, it assumes that role. If it was not the winner, it assumes that the winning robot will take up the role and performs calculations for the next role in the list. The winners of previous auctions are not considered in subsequent auctions for different roles; they have already been assigned a task. In principle, all of the robots are performing the same calculation on the same shared data, so they should arrive at the same result. In practice, no synchronization is provided, so it is possible for teammates to calculate different bids for each other due to factors such as network delays and transmission errors.

To address this, hysteresis is added to the system. Once a robot takes a particular role, it does not relinquish that role for a short time - on the order of seconds. Since the bid functions are self-reinforcing, that is to say once a robot takes up a role its actions increase its suitability to fill that role, this hysteresis is enough to overcome the lack of synchronization in practice.

Another question is: why broadcast so much information? Why not have agents send only their own bid values? It seems wasteful to broadcast position estimates instead of a real valued bid. However, robots must broadcast their position and their estimate of the ball position anyway. The utility of sharing ball information is obvious; robots frequently find themselves in situations where they cannot see the ball due to distance or occlusion. In these cases, teammates can help each other find the ball. The robot position information shared by teammates can be used for obstacle avoidance. The robots have a limited field of view so vision alone is not sufficient to detect neighboring robots.

We present a bidding function to calculate the robots' activation for the primary attacker role as a concrete example. Bid functions for other roles may be designed in a similar manner and there are many other possible functions that could be used for the primary attacker auction. For example, it might be desirable to take localization uncertainty into account in a principled way. This particular function is designed to produce high bids when robots are close to the ball and also to take into account how well lined up the robot is to kick the ball into the opponent's goal.

$$Bid = \underbrace{\frac{\theta_{goal}}{\pi}}_{angular\ component} + \underbrace{(1 - min(1, d_{ball}))}_{distance\ component} \tag{1}$$

In this equation, θ_{goal} is the angle formed between the line running from the robot to the ball and the line from the ball to the goal. When θ_{goal} equals π, the robot is perfectly lined up to kick the ball into the opponents' goal. The d_{ball} parameter is the distance from the robot to the ball in meters. This distance is capped at 1 meter.

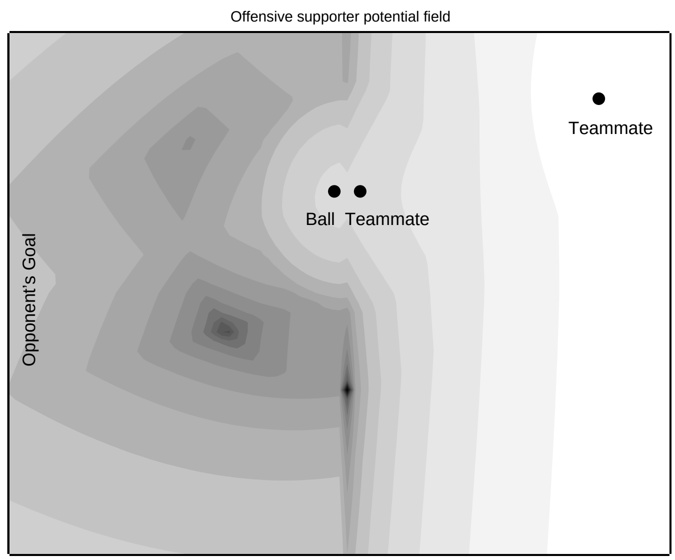

Fig. 2. A contour plot of the potential function used by the offensive supporter to position itself on the field. Darker shading corresponds to lower areas of the surface; the robot follows the gradient down to these minimum values. The opponent's goal lies on the left edge of the field and the goal being defended is on the right side.

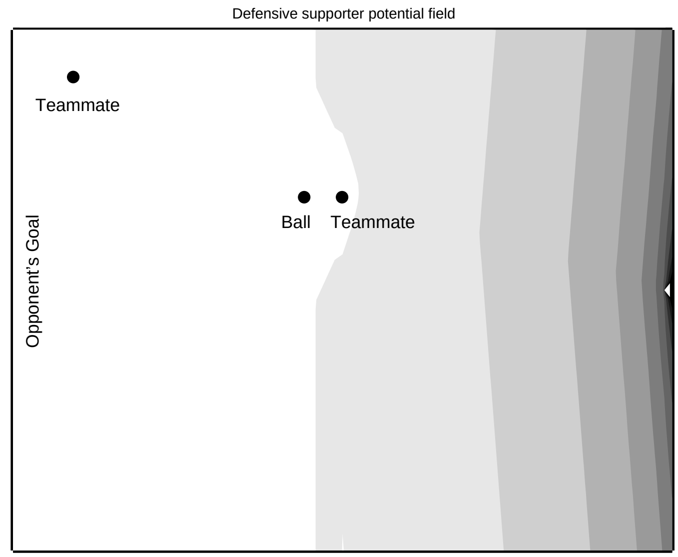

Fig. 3. A contour plot of the potential function used by the defensive supporter to position itself. Darker areas correspond to lower values and the agent navigates down the gradient. The opponents goal is on the right edge of the plot and the goal being defended is on the left edge.

C. Coordination

The robots use the same mechanism for both coordination and obstacle avoidance. They overlay a potential field over the environment and sample local points in the field to approximate its slope at their current location. They follow the gradient of the potential field until they reach a local minimum. The components of the field are designed such that local minimums arise at positions from which the robots can support the primary attacker. In the case of the offensive supporter, the field guides the robot to a good position to receive passes or recover the ball if the shot on goal goes wide. In the case of the defensive supporter, the gradient guides the robot to a position where it blocks its own goal and can recover the ball if it is intercepted by the opposing team. The primary attacker does not make use of the potential field; it always seeks out the ball and counts on its teammates to move out of its way instead of avoiding them.

The potential field is the sum of several linear components. Each of these components either represents heuristic information about the world, such as the offensive supporter should not block the primary attacker's shot on goal, or obstacle information, such as repulsion terms from the walls and other robots. Typically the components of the potential functions are bounded at zero. This makes the effect of the terms local and helps prevent undesirable interactions between terms.

Currently only teammates are included in the list of robots to avoid due to the difficulty of perceiving other robots. Teammates report their own positions via the wireless network; since opponents do not do this, high fidelity information about their locations is not available. However, this is a perceptual problem - the composite nature of the functions makes it trivial to add terms for opponents as soon as the perceptual system is able to provide that information.

Depending on their supporting role, the robots may use different subsets of the components. For example, the offensive supporter does not use the component that guides the robot to positions between the ball and its own goal - that heuristic information is not applicable when filling an offensive role.

Next we review the individual components of both the offensive and defensive supporters' potential fields. In the following equations c_n indicates a positive constant and k_n indicates a positive slope.

$$P_{wall} = max(0, c_1 - k_1 \cdot d_{wall}) \tag{2}$$

The wall potential term encodes a linear repulsion from the walls and the team's own defense zone; only the goalie on each team is allowed to be in the defense zone. c_1 is a positive base potential for when the robot is at the wall. The potential falls off linearly with the distance of the robot from the wall with a slope of k_1. This term is shared by both the offensive and defensive supporters.

$$P_{ball} = \|c_2 - d_{ball}\| \cdot k_2 \tag{3}$$

The ball potential term guides the offensive supporter to a position that is an equilibrium distance, c_2, away from the ball. The potential increases linearly with a slope of k_2 as the robot moves away from the equilibrium distance.

$$P_{teammate} = max(0, c_3 - k_3 \cdot d_{teammate}) \tag{4}$$

The teammate repulsion potential is a positive value that falls off linearly with distance. As with wall repulsion, this term is shared by both types of supporter.

$$P_{forward\,bias} = max(0, k_4 \cdot d_{behind\,ball}) \quad (5)$$

The forward bias potential guides the offensive supporter to a position parallel to or in front of the ball. The $d_{behind\,ball}$ parameter encodes how far the offensive supporter is down field from the ball.

$$P_{defensive\,bias} = k_5 \cdot d_{from\,goalline} \quad (6)$$

The defensive bias potential is analogous to the forward bias only it acts on the defensive supporter. It forces the robot to remain in a position close to its own goal; it increases in value linearly as the robot moves up the field away from the goal line.

$$P_{ball\,corridor} = \|c_6 - d_{shot\,path}\| \cdot k_6 \quad (7)$$

The ball corridor potential encodes the heuristic information that the offensive supporter should not block shots on the goal, but it should also position itself close to the path taken by the ball in order to recover the ball if it stops before reaching the goal. c_6 represents the equilibrium distance of the agent from the ball path. $d_{shot\,path}$ is the actual distance of the agent from the path. The shot path is defined as the line segment from the ball to the center of the opponent's goal line. The offensive supporter is the only robot that uses this potential.

$$P_{block\,goal} = d_{block\,path} \cdot k_7 \quad (8)$$

The block goal potential guides the defensive supporter to a position on the line between the ball and its own goal. $d_{block\,path}$ is the distance between the robot and the line segment running from the ball to the center of the robot's goal line.

$$P_{side\,bias} = max(0, k_8 \cdot offset_{robot} \cdot \frac{offset_{ball}}{\frac{1}{2} \cdot width_{field}}) \quad (9)$$

The side bias term applies only to the offensive supporter. It encodes the fact that the robot should position itself across the field from the primary attacker. The $offset$ terms represent the offset of either the ball or the robot from the line drawn between the centers of the two goals. Notice that this is not a distance - the offset has a negative value for one half of the field.

IV. Experimental Results

We first present empirical results from Robocup-2002 to demonstrate that our system functions well against a wide variety of opponents. We then present results from controlled experiments in a penalty shot domain to provide a more quantitative view of the importance of coordination.

A. Empirical Results from Robocup-2002

We present the final scores from the games that CMPack'02 participated in during Robocup-2002 in table I. The opposing teams used different strategies for coordination and task assignment. For example, the German Team also used wireless communication for coordination. The University of New South Wales, the 2001 champions, used only local information from vision and sound to coordinate their robots. CMPack'02 performed well against all opponents and did not lose a single game. However, our communication framework was only one facet of the team. Vision, motion control, localization, and behaviors were equally important. Since it is impossible to separate out the contribution of only our coordination framework from Robocup, we also present results from controlled experiments designed to test the importance of coordination alone while the other systems are held constant.

TABLE I

Games Scores from Robocup-2002

Opponent	Final score (CMU:Opponent)
ARAIBO	5:1
German Team	3:1
SPQR	7:0
Team Sweden	9:0
Robomutts++	4:0
rUNSWift	3:3 (penalty kicks 2:1)

B. Controlled Experiments

We tested how coordination affects the performance of the robots in a penalty shot domain. In these experiments, a robot, or a team of robots, attempted to score on an empty goal. No opponent robots were used, which means that while the environment was dynamic and uncertain, it was not adversarial; the world did not actively work against the robots while they were performing the task. We did not use opponents to reduce the amount of noise in the data and the time required for each trial.

To test how long it took the robots to score, we marked 30 locations on the field. The 30 locations were divided evenly between each half of the field and within each half the locations were distributed in an approximately uniform fashion. Each marker was assigned a unique number so that the locations could be visited in a fixed order. The same order was used for all experiments.

Since the goal of the experiment was to test how quickly the robots scored in general, we did not want to specify their starting position. For this reason the robots were not moved after scoring a point; their starting position for each point was where they had scored from during the previous point. (Before scoring for the first time, they started on their own goal line) This means that samples are not independent, but it does mimic what happens in real games when the goalie of the opposing team clears the ball to an unknown place on the field.

We ran three separate experiments. The first was a single robot performing alone to provide a baseline. Next, three robots without coordination performed the task followed by three robots with coordination. Each experiment began with the robot(s) on their own goal line. The ball started on the first marker. The robots were unpaused and the length of time it took for them to score was recorded. As soon as the robots scored, they were paused, the ball was moved to the next marker in the sequence, and then the robots were restarted without being moved. This procedure was repeated until the ball had started

TABLE II
TIME TO SCORE FOR THE THREE TRIALS

	mean (sec)	std. dev.
Single Robot	93.48	62.38
Three Robots (No coordination)	156.40	125.01
Three Robots (Coordination)	78.89	52.58

TABLE III
P VALUES FROM THE WILCOXON SIGNED RANK TEST TO DETERMINE IF TWO DISTRIBUTIONS ARE THE SAME

Distributions		P
Single Robot	No coordination	0.043
Single Robot	Coordination	0.221
Coordination	No Coordination	0.006

from each of the 30 markers. If the ball left the field or entered the penalty region, it was immediately replaced in legal territory.

Figure 4 shows cumulative distributions of the time to score for each of the three trials. The minimum times to score for all three trials were very similar; for these points nothing went wrong. The robots approached the ball, captured it, and kicked it into the goal on their first attempt. On the other hand, there is a large difference between the maximum values for the single robot versus the team without coordination and again for the maximum values between the robots with coordination versus the robots without coordination. The means and standard deviations for the distributions are listed in table II.

We uses a Wilcoxon signed rank test to determine whether or not the distributions were the same. The results of these tests are shown in table IV-B. There was a significant difference between the case with coordination and the case without it. There was also a significant difference between the single robot case and the case without coordination. While the mean for the trial with three robots using the coordination framework was lower than the mean for the single robot case, there was not a statistically significant difference in the distributions from these trials.

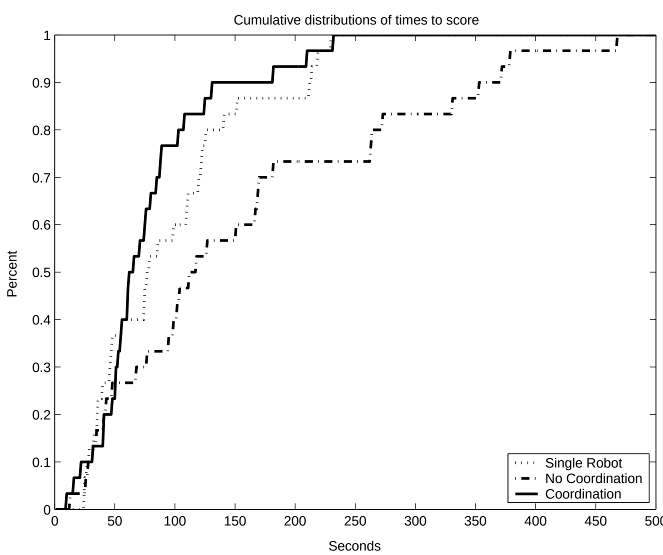

Fig. 4. Cumulative distributions of the time between points for the three trials.

V. DISCUSSION AND CONCLUSIONS

Our results show that coordination is vital for multi-agent systems. A stronger result would have shown the case with three coordinating robots out performing the single robot case, however, our results do show the extra robots do not decrease performance in the non-adversarial test domain. Even without increasing performance in the penalty shot domain, the extra robots do make the system more robust against failure; if a single robot fails, two other remain to complete the task.

In the future, we would like to investigate what happens in an adversarial domain by adding either a goalie or a single robot to the opposing team. We hypothesize that the difference between the single agent case and the three robots with coordination case would be widened. That is, three robots should be able to fare better against an opponent than a single robot.

VI. ACKNOWLEDGMENTS

The authors would like to thank the other team members of CMPACK'02 : Scott Lenser (team leader), Ashley Stroupe, Maayan Roth, and Sonia Chernova. The authors would also like to thank James Bruce and Brett Browning for many helpful discussions.

REFERENCES

[1] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams, 1997.

[2] T. Balch and M. Hybinette. Social potentials for scalable multirobot formations. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-2000)*, 2000.

[3] J. Bruce, T. Balch, and M. Veloso. CMVision (http://www.cs.cmu.edu/ jbruce/cmvision/).

[4] Claudio Castelpietra, Luca Iocchi, Daniele Nardi, Maurizio Piaggio, Alessandro Scalzo, and Antonio Sgorbissa. Communication and coordination among heterogeneous mid-size players: ART99. *Lecture Notes in Computer Science*, 2019:86–95, 2001.

[5] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation (ICRA-1985)*, pages 500–505, 1985.

[6] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. RoboCup: The robot world cup initiative. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 340–347, New York, 5–8, 1997. ACM Press.

[7] S. Lenser, J. Bruce, and M. Veloso. CMPack: A complete software system for autonomous legged soccer robots. In *Autonomous Agents*, 2001.

[8] Maja J. Mataric and Gaurav S. Sukhatme. Task-allocation and coordination of multiple robots for planetary exploration. In *Proceedings of the 10th International Conference on Advanced Robotics*.

[9] Maayan Roth, Douglas Vail, and Manuela Veloso. A world model for multi-robot teams with communication, 2002.

[10] William Uther, Scott Lenser, James Bruce, Martin Hock, and Manuela Veloso. CM-Pack'01: Fast legged robot walking, robust localization, and team behaviors. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: The Fifth RoboCup Competitions and Conferences*. Springer Verlag, Berlin, 2002, forthcoming.

[11] M. Veloso, P. Stone, and M. Bowling. Anticipation as a key for collaboration in a team of agents: A case study in robotic soccer. In *Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II*, 1999.

[12] Thilo Weigel, Willi Auerbach, Markus Dietl, Burkhard Dümler, Jens-Steffen Gutmann, Kornel Marko, Klaus Müller, Bernhard Nebel, Boris Szerbakowski, and Maximilian Thiel. CS freiburg: Doing the right thing in a group. *Lecture Notes in Computer Science*, 2019:52–63, 2001.