DYNAMIC MULTI-ROBOT COORDINATION

Douglas Vail
Computer Science Department
Carnegie Mellon University
*Pittsburgh, PA 15213, USA**
dvail2@cs.cmu.edu

Manuela Veloso
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA†
veloso@cs.cmu.edu

Abstract Communication among a group of robots should in principle improve the overall performance of the team of robots, as robots may share their world views and may negotiate task assignments. However, in practice, effectively handling in real-time multi-robot merge of information and coordination is a challenging task. In this paper, we present the approach that we have successfully developed for a team of communicating soccer robots acting in a highly dynamic environment. Our approach involves creating shared potential functions based on shared positions of relevant obstacles in the world. The biases introduced in the potential functions are general and they could in principle be provided by external sources, such as a human or robot coach. We provide controlled experiments to analyze the impact of our approach in the overall performance of a robot team.

Keywords: Teams of robots, robot soccer, multi-robot communication and coordination

*Partial funding provided by a National Physical Science Consortium Fellowship with stipend support from HRL Laboratories, a generous grant from Sony, and grant DABT63-99-1-0013.
†Funding provided by a generous grant from Sony and grant DABT63-99-1-0013.

1. 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 a robot 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 have put our algorithm to the test while participating in the Sony legged league at the RoboCup'02 competition. We came in first place in the competition out of 20 participating teams. This performance was certainly the result of many contributions of our complete team (Veloso et al., 2002). Although during the games we could clearly observe the successful multi-robot coordination, it was hard to quantify the value of this specific component. In this paper, we present a focused experimental evaluation of our framework in a penalty shot situation that demonstrates the effectiveness of communication and coordination.

Our approach is based on the use of shared potential functions. Artificial potential fields have long been used for obstacle avoidance (Khatib, 1985). Others have extended this idea to allow a group of robots to assemble and maintain formations using only local information in the potential calculation (Balch and Arkin, 1997; Balch and Hybinette, 2000). Domain specific heuristics may also be encoded in potential fields to position robots for particular roles (Castelpietra et al., 2001; Veloso et al., 1999; Weigel et al., 2001). For example, these heuristics may guide robots to locations near an opponent's goal or place them in a good position to receive a pass. We combine these ideas with distributed task allocation by continuous bidding (Castelpietra et al., 2001; Mataric and Sukhatme, 2001) to create a system where obstacle avoidance is combined with coordination through dynamic potential functions that change based on the role each robot is assigned though the distributed task allocation.

2. Background - The RoboCup Domain

The legged league of Robocup (Kitano et al., 1997), the robot soccer world championship, provides a challenging test bed 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 wireless Ethernet 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 (Bruce et al., 2002). 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 (Lenser et al., 2001; Uther et al., 2002). 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.

3. 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.

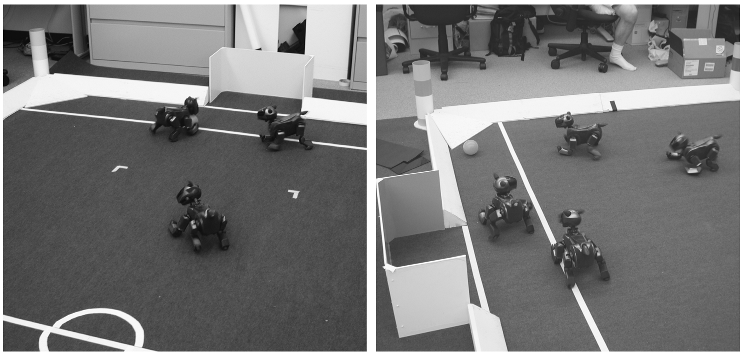

Figure 1. On the left, the primary attacker prepares to shoot while two other robots position themselves in supporting positions. On the right, three robots spread around the ball while the goalie clears it from the defense zone.

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.

3.1 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 robot's current position and ball location estimates as well as uncertainty estimates on those values. The messages also contain flags indicating if 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. 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 (Roth et al., 2003). Next we describe how this shared information is used to assign roles to different agents and how the agents fill those roles.

3.2 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.

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.

Another question is: why broadcast so much information? It seems wasteful to broadcast position estimates instead of a real valued bid. However, robots must broadcast their location and ball position estimates anyway. This allows obstacle avoidance and aids ball discovery when teammates or the ball are not visible.

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.

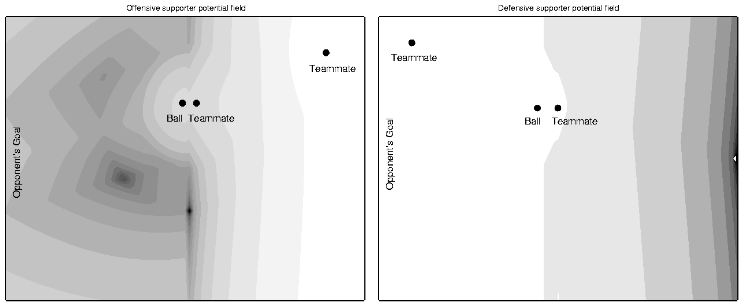

Figure 2. A contour plot of the potential functions used by the offensive and defensive supporters to position themselves. Darker shading corresponds to lower areas of the surfaces; the robots follow the gradient down to these minimum values.

3.3 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.

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

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.

$$P_{defensive\,bias} = k_5 \cdot d_{from\,goalline} \tag{6}$$

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.

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

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.

$$P_{block\,goal} = d_{block\,path} \cdot k_7 \tag{8}$$

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.

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

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.

4. Experimental Results

In order to quantify the difference that coordinations makes, 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 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 3 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 1.

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 2. 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.

5. 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

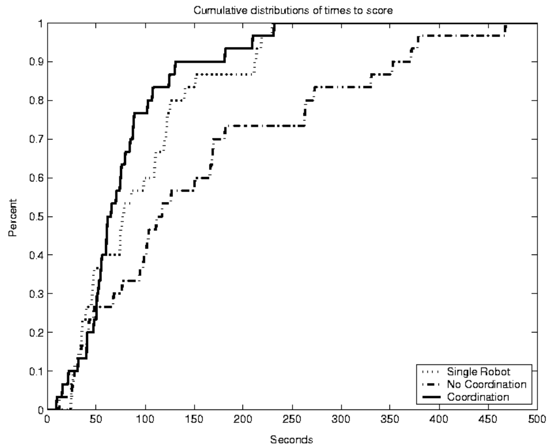

Figure 3. Cumulative distributions of the time between points for the three trials.

Table 1. 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 2. 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

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.

6. 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

Balch, T. and Arkin, R. (1997). Behavior-based formation control for multi-robot teams.

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

Bruce, J., Balch, T., and Veloso, M. (2002). CMVision (http://www.coral.cs.cmu.edu/∼jbruce/cmvision/).

Castelpietra, C., Iocchi, L., Nardi, D., Piaggio, M., Scalzo, A., and Sgorbissa, A. (2001). Communication and coordination among heterogeneous mid-size players: ART99. *Lecture Notes in Computer Science*, 2019:86–95.

Khatib, O. (1985). 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.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., and Osawa, E. (1997). RoboCup: The robot world cup initiative. In Johnson, W. L. and Hayes-Roth, B., editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 340–347, New York. ACM Press.

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

Mataric, M. J. and Sukhatme, G. S. (2001). Task-allocation and coordination of multiple robots for planetary exploration. In *Proceedings of the 10th International Conference on Advanced Robotics*.

Roth, M., Vail, D., and Veloso, M. (2003). A world model for multi-robot teams with communication. In *Proceedings of ICRA'03, the 2003 IEEE International Conference on Robotics and Automation*, Taiwan. under submission.

Uther, W., Lenser, S., Bruce, J., Hock, M., and Veloso, M. (2002). CM-Pack'01: Fast legged robot walking, robust localization, and team behaviors. In Birk, A., Coradeschi, S., and Tadokoro, S., editors, *RoboCup-2001: The Fifth RoboCup Competitions and Conferences*, Berlin. Springer Verlag.

Veloso, M., Lenser, S., Vail, D., Roth, M., Stroupe, A., and Chernova, S. (2002). CMPack-02: Cmu's legged robot soccer team (https://www.openr.org/page1_2003/index.html/).

Veloso, M., Stone, P., and Bowling, M. (1999). 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.*

Weigel, T., Auerbach, W., Dietl, M., Dümler, B., Gutmann, J.-S., Marko, K., Müller, K., Nebel, B., Szerbakowski, B., and Thiel, M. (2001). CS freiburg: Doing the right thing in a group. *Lecture Notes in Computer Science*, 2019:52–63.