Modeling and Exploiting Behavior Patterns in Dynamic Environments

David Ball and Gordon Wyeth
School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, Australia
{dball, wyeth}@itee.uq.edu.au

***Abstract* — This paper presents a new approach to improving the effectiveness of autonomous systems that deal with dynamic environments. The basis of the approach is to find repeating patterns of behavior in the dynamic elements of the system, and then to use predictions of the repeating elements to better plan goal directed behavior. It is a layered approach involving classifying, modeling, predicting and exploiting. Classifying involves using observations to place the moving elements into previously defined classes. Modeling involves recording features of the behavior on a coarse grained grid. Exploitation is achieved by integrating predictions from the model into the behavior selection module to improve the utility of the robot's actions. This is in contrast to typical approaches that use the model to select between different strategies or plays. Three methods of adaptation to the dynamic features of the environment are explored. The effectiveness of each method is determined using statistical tests over a number of repeated experiments. The work is presented in the context of predicting opponent behavior in the highly dynamic and multi-agent robot soccer domain (RoboCup).**

Keywords – learning and adaptation; behavior classification; behavior modelling; prediction; behavior selection; robot soccer.

I. INTRODUCTION

Autonomous systems have the potential to improve their effectiveness by modifying their behavior to adapt to their environment. There has been significant successful research in modeling the static features of the environment and using those models to perform tasks such as navigation. This paper explores the more difficult problem of modeling the dynamic features of the environment in order to improve performance over time. This research could be applied to a service robot seeking to choose the best path through a busy office environment, or to a team of field robots working to control a herd of animals.

The key in these applications is to look for patterns and tendencies in the behavior of the dynamic obstacles, and then to make plans that exploit the understanding of the patterns.

The context of our research is robot soccer, where the static features of the environment are relatively simple, but the dynamic features of the environment (the opposition) form the greatest obstacles to success. The robot soccer context also introduces the element of coordinating multiple robots; although the methods that are described are equally applicable to single robots.

This paper explores three methods of adapting to the dynamic features of the environment, all aligned by a common approach. The approach is to:

1. Classify the behavior of the moving elements of the environment; using observations of the motion or the results of the motion to place each feature into a previously defined class.
2. Produce models that are used to predict the behavior of the moving elements for each class.
3. Exploit the predictions of movement in order to improve the effectiveness of the autonomous system.

Each of the three methods of adaptation has been implemented as a new module for the existing Multi-Agent Planning System (MAPS) [1]. MAPS has shown itself to be effective for multi-robot tasks such as team foraging, team pursuit / evasion and robot soccer. The three adaptive methods have been tested against the baseline non-adaptive MAPS system. The results show that adaptation is sometimes significantly beneficial, but can be harmful to the system's performance if applied without caution.

The rest of the paper is structured as follows. Section two describes the application domain and the existing robot system with particular emphasis on the multi-agent planning system. Section three details the methods used for adaptation. Section four describes the experimental setup and section four presents the results. Lastly section six concludes the paper.

II. CURRENT INTELLIGENCE SYSTEM

This section has three parts; an overview of the testing domain, a brief overview of the entire RoboRoos robot soccer system as it was at RoboCup 2003 and an explanation of the existing multi-agent planning system.

A. *Testing Environemnt*

The testing environment is the RoboCup F180 league [4]. The research platform, the RoboRoos, competes in the F180 league of the annual RoboCup competitions. In the this league both teams have five robots that each must physically fit inside a cylinder with a diameter of 180mm and a height of 150mm. Devices to dribble and kick the ball are permitted as long as they do not hold the ball and 80% of the ball is kept outside of the convex hull of the robot. The field is approximately 2.3 × 2.8 meters, with an orange golf ball acting as the soccer ball. Teams use global overhead vision as the primary sensor. The rules are similar to the human version of the game (FIFA), with exceptions such as the elimination of the offside rule and changes required to make sense for wheeled robots. There are two 10 minute halves. The robots are fully autonomous in the sense that no strategy or control input is allowed by the human operators during play. Humans referee the matches.

B. The RoboRoos

The University of Queensland's robot soccer team, the RoboRoos [5], [6], [7] and [8] is one of the longest standing teams in the small-size league of RoboCup having competed annually since 1998. During these years many research areas have been explored especially in the areas of multi-robot coordination and navigation in dynamic environments. The RoboRoos came second at RoboCup 2003, beaten 1-0 in the final by Big Red from Cornell University.

The RoboRoos system is a layered set of subsystems, where each subsystem performs a different task. To give an overview of the system the flow of the information from the camera to the robot's actuators is presented. An overhead camera captures global images of the field. The vision system processes the images to identify and locate the robots and the ball. This state of the field is sent to the Multi-Agent Planning System [1]. MAPS coordinates the RoboRoos by selecting a behavior for each robot. The MAPS behaviors are interpreted by the AES system. Each behavior has a set of appropriate parameters and a notion of the overall desired robot motion. The Navigation [9] module attempts to achieve the desired motion behavior while avoiding obstacles. The Navigation module determines the immediate desired heading and distance for the Motion System. The Motion system accelerates and decelerates the robot to the desired heading and distance by creating force limited trajectories.

C. Multi-Agent Planning System (MAPS)

MAPS is the highest level planner in the system, responsible for distributing the overall goal of the team amongst the individual robots. MAPS is responsible for the multi-robot coordination and cooperation by selecting an action and an action location for each robot. MAPS determines the team's actions based on the current world model, the team goal and the currently available actions.

MAPS uses potential fields as the mechanism for determining action selection and action location. The potential fields can model the suitability of an action for the different agents, or be used to find a suitable action location. MAPS has a library of potential field functions and abstractions, where each field is a two dimensional array of values. A more positive value represents a more desirable action for an agent, or, in the case of determining an action location, a more desirable location for that action. The following are four examples of the types of potential fields used by MAPS:

- **Basefield:** This field represents favorable regions of the physical environment. The regions of the field that will always be favorable to the goals of the team are the most positive.
- **Object Regions:** These fields model physical objects on the field by representing an area of effect around an object. Object regions can be used to bias the positions of other team members to ensure they don't attempt to occupy the same location.
- **Clear Path:** This is an abstract feature that represents clear paths to objects or locations. It biases regions that offer a line-of-sight path to the point in question. Instead of producing a true/false field it determines the relative clearness of the path to the object or location in question.
- **Distance:** This is another abstract feature that represents the distance from objects, thus favoring action locations close to the robot or to a goal location.

It is the overlaying of multiple fields that gives an abstract goal-biased terrain map that provides the information to determine the robots' sub-goals. By weighting the strengths and shapes of the component fields, MAPS can be tuned to give peak performance for a specific goal, or to act against specific opponent strategies. MAPS uses potential fields to determine the location to dribble and kick the ball, where to place defending and attacking robots and which player should be kicking or dribbling the ball. Figure 1 shows an example of a potential field that determines where to dribble the ball.

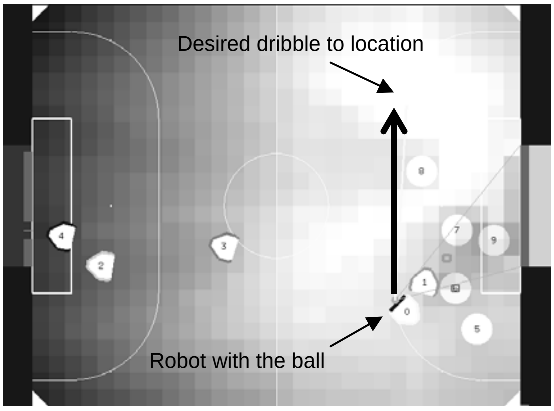

Figure 1: An example MAPS potential field that determines where a robot should dribble the ball. Note the effect of the Basefield, opponent object regions and clear path from the goal potential fields. Lighter cells represent more positive regions. The opponent's goal is on the right.

III. ADAPTIVE APPROACH DESCRIPTION

The adaptive approach aims to allow the existing planning system to autonomously improve its performance in real time. The effectiveness of the current RoboRoos intelligence system is potentially limited by its inability to predict and exploit the likely behavior of the opposition. The module that deals with the effects of the opposition – MAPS – currently treats the predicted effect of the opposition as a probability distribution around the opposition robot's current location. In order to allow more powerful multi-agent planning, a more accurate model of the effect of the opposition agents is required. A layered and modular approach is taken to predicting the behavior of the opponent team that breaks the problem into classification, modeling and prediction, and exploitation. The approach is shown in Figure 2. These layers are now described in detail.

- **Classification:** This layer determines relevant and important features based on the global motion information received from the visual sensor. The classifications are based on predefined or previously learnt models of general motion or the environment.
- **Modeling and Prediction:** This layer consists of two similar but separate sub modules. The model of the opponent is a 2D distribution that is incrementally built during the course of a game based on input from the classification layer. It can represent the relative strength of features in the environment or the behavior of agents. The prediction module can use the model to predict the future state of the dynamic environment and the agents. In the methods presented in this paper it predicts the overall behavior of the opponent's defense.
- **Exploitation:** This layer takes the model and the predictions and integrates them into the existing MAPS system. By overlaying them onto the existing potential fields they influence the choice of action locations towards those that will have a predicted higher utility.

All layers use a coarse grained grid with the same scale as the MAPS potential fields.

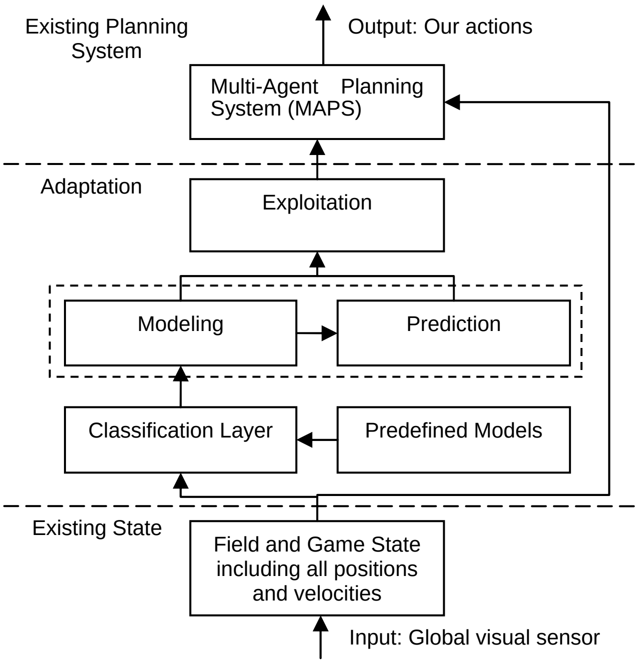

Figure 2: The adaptation approach shown integrated into the current planning system. The field state represents the observations that are available from the visual sensor.

Three methods of using the proposed adaptation approach have been developed. These consist of one direct and two indirect implementations. The direct approach involves explicit modeling of the behavior of the opponent's defensive formation by classifying their motion. Indirect modeling instead uses the resultant effect of the opponent's defensive behavior as the weight for model learning.

A. *Direct Model Implementation*

In the direct model method a particular characteristic of a behavior relative to a feature is modeled. In this implementation the behavior of the opponent's defense is modeled. The classification layer must therefore be able to distinguish which robots are parts of the opponent's defensive formation and whether they are at their desired final location. The inputs to this layer are the opponent's velocity and acceleration. Threshold values are predetermined for classification. To achieve this 10 000 samples of the opponent's motion were logged with the simulator determining whether the robot should be classified as part of the opponent's defense. This is because internally the simulator has knowledge about the actions of each of the robots and can also determine whether a robot is in its desired pose. Figure 3 shows the graph of the 10,000 samples that was used to determine the threshold values.

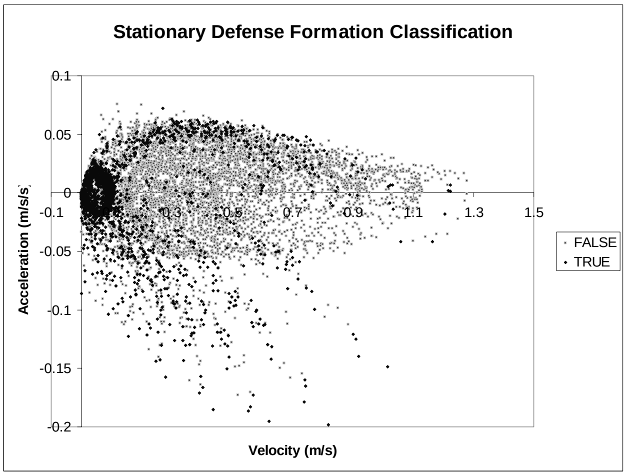

Figure 3: This graph was used to determine the threshold values for the classification system so that it could distinguish robots that are part of the defensive formation and are stationary. True points indicate that the robot was part of the opponent's defense and was stationary.

In the modeling layer the opponent's defensive behavior is recorded relative to each possible ball location. The ball position therefore acts the index into a lookup table for each set of positions of the opponent robots. The model is normalized by recording the number of frames that the ball is in each cell.

The prediction layer adds all of the normalized models together. This summation represents a prediction of the opponents overall defensive strategy. The strength of the defense is apparent from the intensity of the cells.

The exploitation layer takes this overall prediction of the opponent's defense and projects clear paths out from the opponent's goal cells. Stronger regions of the opponent's defense will stop the clear paths that are projected. This resultant potential field creates globally optimal favorable regions to kick the ball at the opponent's goal from.

B. Indirect Model Implementation

In the indirect method the behavioral effect of the dynamic features of the environment is used as the weighting for cells. In each frame, the derived behavioral effect at the robot's location is averaged with the current value at that cell.

In this application the behavioral effect is the percentage of the largest uncovered part of the opponent's goal. This is the amount of goal left uncovered by the opponent robots from the ball's perspective. To classify the amount of goal that is covered, the largest gap along the goal line is divided by the total length of the goal line. The sum of the uncovered gaps is not used as many smaller gaps do not indicate a region where a shot on goal can be taken.

1) Indirect Model – Location Model

This method builds a model that represents the strength of a feature from different locations. In this implementation the percentage of the goal currently covered is averaged at the current location of the ball. The result of this model is the predicted locations of weaknesses in the opponent's defense. The exploitation layer integrates this model into MAPS by overlaying this field onto the existing action location field. This increases the likelihood of selecting action locations where weaknesses in the behavior of the opponent's defense were previously seen.

2) Indirect Model – Action Model

This method builds a model that represents the strength of a feature relative to action locations. In this implementation the percentage of the goal currently covered is averaged at the current action location. The exploitation layer integrates this model into MAPS by overlaying this field onto an existing action location field. This increases the likelihood of selecting action locations that have previously experienced weaknesses in the behavior of the opponent's defense.

IV. EXPERIMENTAL SETUP

This section details the experimental setup to test the three methods that use the proposed adaptation approach.

The classification layer must be able to determine whether a robot is part of the defensive formation and is at its desired location. Threshold values need to be determined for the velocity and acceleration inputs. For this experiment, based on analysis of the graph in Figure 3, threshold values of 0.11m/s for velocity and ±0.02m/s^2 for acceleration were chosen. These were determined to give a detection accuracy of 76.3% based the 10,000 samples taken. These threshold values will still give significant error in the detection of robots that are part of the stationary defensive formation. A Type I error or false positive will be where robots are detected as part of the stationary defense but are not. In this application these errors do not cause problems as they tend to be caused by all robots and are evenly spread as noise across the field. In this application Type II errors mean that a particular robot is part of the defense but is not classified as such. As these are equally likely to happen across the whole defense it does not affect the normalized result.

The adaptation methods were tested within the existing robot soccer system. The simulator simulates two teams of robots in line with the 2003 RoboCup Small Size League rules and regulations. This includes simulating some of the important human referee tasks such detection of when a goal is scored and restarting play. The simulator has a very high fidelity with a one millisecond resolution in the simulation of all robot and ball kinematics, dynamics and collisions. It closely simulates the performance of the real robots.

Each test was run for the length of an entire game which is 20 minutes. Both teams have an equal chance of winning the match as all bias in the simulator has been removed. For example the order in which collisions are detected is randomized. This has been confirmed by comparing the mean goals scored for 50 games.

In each simulation only one team uses the adaptation module. At the end of a game the number of goals scored by both teams is logged. It is important to note this means that the two sample results are not fully independent from each other.

When the exploitation layer integrates the model and predictions into the MAPS system it must weight these fields relative to the other potential fields. For these experiments the weights were only minimally tuned and were constant for an entire implementation testing. The minimal amount of tuning was to ensure that they were not too low as to have no effect on the decision but also not completely determine a decision.

The three different adaptation methods were each tested using two different offences against two different defenses (four tests using each method). This is designed to test the generality of the adaptation methods. The following offensive strategies (that dictate the behavior of the attacking robot) were tested:

- **Ball player Screening (BS):** If our other attacking robot doesn't have the ball then prevent their player that is closest to the ball from reaching it, else screen their closest player away from our robot with the ball to create more freedom of motion.
- **Cover Screening (CS):** Attempt to impede the motion of the opponent players that are preventing direct shots on the goal.

The following defensive strategies were tested

- **Double BallPlayer (DB):** Each side of our goal is assigned its own tackling player. This prevents the screening type of offences from leaving our defense completely held on one side.

- **Goal Side (GS):** Our defense player is assigned to stay level with and goal side of the opponent to prevent them from moving across the goal face.

The three modeling methods as described in the approach description were tested. They are summarized here.

- **Defense Model (DM):** Modeling the behavior of the opponents and forming clear paths from their goal.
- **Location Modeling (LM):** Modeling locations based on the percentage of the goal that is covered.
- **Action Modeling (AM):** Modeling the current action location based on the percentage of the goal that is covered.

For this experiment the null and alternate statistical hypothesizes are defined as:

H_0 – The proposed adaptation approach has no effect on the number of goals scored per game.

H_1 – The proposed adaptation approach increases the number of goals scored per game.

A heteroscedastic (unequal variance) Students T-test is used to determine the probability that the three methods are statistically significant. A confidence level of 5% is used to conclude a statistically significant change in the number of goals scored per game.

V. RESULTS

This section presents models created by each of the three methods and presents the results of the experiments.

Figure 4 shows the final 2D distributions for each of the three modeling methods. The top distribution shows the direct Defense Model prediction implementation, the middle distribution shows the indirect Action Model implementation and the bottom distribution shows the indirect Location Model implementation.

The Defense Model shows that the opponent's defensive is stronger on one side of the goal and that there are three separate across goal motions for the defensive players. The extremities of their motion are also clear from this distribution. The Action Model shows a bias towards action locations on either side of the opponent's goal. The Location Model shows areas of weakness in the behavior of the opponent's defense.

The results for the three methods are shown in Table 1. The first column shows the particular combination of adaptation method, offence and defense strategy. The second and third columns show the sample mean goals for both the team with the adaptation module and the team without it. The next column shows the results of the Student's one tailed heteroscedastic T-test. Positive results indicate that the mean goals scored increased significantly by using the adaptation module. Negative indicates a significant decrease in the goals scored with the adaptation module. Zero indicates an inconclusive result. The last two columns show the number of games won by both the team with adaptation and without. These may not necessarily add to 50 due to tied games.

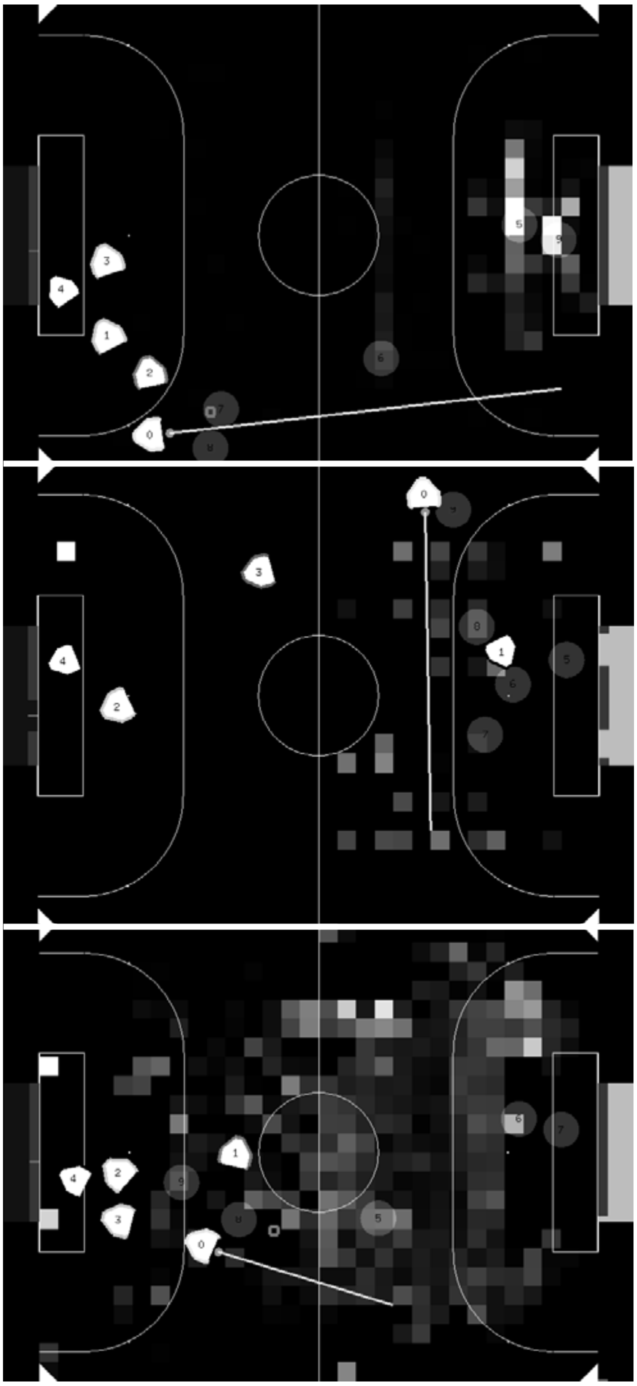

Figure 4: Final 2D model of the opponent for the three implementations. The implementations are from top to bottom, Defense Model, Action Model and Location Model. The Defense Model figure shows the overall predicted defensive behavior. The system with the adaptive module is going from left to right. In the direct model a lighter cell indicates a stronger stationary region for the defense. In the indirect models a lighter cell indicates the strength of the opponent's defense relative to the action location or the position of the ball.

TABLE I. RESULTS FOR THE TWELVE TESTING COMBINATIONS.

Testing Combination	$\overline{x}_{ADAPT}$	$\overline{x}_{NONE}$	T-test	Wins Adaptive	Wins None
DM, BS, DB	5.10	4.68	0	25	17
DM, BS, GS	5.40	4.62	+	26	18
DM, CS, DB	11.70	12.18	0	20	28
DM, CS, GS	15.00	14.96	0	19	27
LM, BS, DB	6.16	4.52	+	34	11
LM, BS, GS	6.06	4.32	+	35	8
LM, CS, DB	5.60	4.80	+	28	18
LM, CS, GS	14.52	14.54	0	23	22
AM, BS, DB	5.18	4.90	0	24	18
AM, BS, GS	4.64	4.54	0	21	22
AM, CS, DB	11.70	9.84	+	35	13
AM, CS, GS	13.96	15.30	-	20	26

The table shows that, based on the average goal mean for 50 games, for the twelve tests performed:

- five showed a statistically significant increase in performance,
- six showed no statistically significant result although four showed an increased mean,
- one showed a statistically significant decrease in performance.

It should also be noted that in three of the tests the team with the adaptive module won twice as often as the team without.

The results show that this approach can significantly increase the number of goals scored and the number of games won, if applied in the right situation. However they also show that the adaptation module can have a negative impact on the rest of the system.

The direct Defense Model method performs significantly worse when the Cover Screening offence is used as apposed to the Ball player Screening offence. This offence involves determining the best location for shooting that is currently optimal as the opponent formation is broken up and screened to one side of the goal. The direct Defense Model is biasing regions that are determined to be globally optimal for shooting. This demonstrates a case where the goal of the exploitation module is in conflict with the goal of the offence. This shows that this method cannot be applied naively without considering the current overall strategy.

The overall lower performance of the Action Model method illustrates that a model based on long term predictions is flawed. For example it does not account for the current motion, nor does it account for where the robot has come from. This indicates that a prediction based on the destination locations for actions does not improve the ability of the team.

The Location Model method gave the best overall improvement in performance based on the average goal mean and the relative number of wins. This method's model represents an absolute and definite measure of the strength of the opponent's defensive behavior and is directly determining the most effective places to shoot the ball. As this is exactly what the MAPS action location potential field is attempting to determine it directly improves the selection of the action location. While the Direct Model approach is attempting to achieve a similar effect it relies on a layer of reasoning (clear paths) to connect predictions of the behavior of an opponent's defense with finding where the best locations to shoot.

Given that the RoboRoos system was already highly tuned (beaten by a single goal in the world cup final), it is especially significant that two of the methods were able to further improve performance.

VI. CONCLUSION

This paper has presented a novel approach for improving performance in a dynamic environment using a layered approach involving classification, modeling and prediction and exploitation. In five of the twelve tests a statistically significant increase in average goal performance was demonstrated. However in one test there was a statistically significant decrease in performance. While these results indicate that this approach can improve the performance against opponent teams more work is needed to ensure a consistently improved performance.

Areas for future work include improved coordination of our players and the accounting for the all of opponents' behaviors. Previous work involving classifying opponent behaviors into higher level behaviors will be integrated with this layered approach. This will increase the scope and ability of the adaptation module.

REFERENCES

[1] A. D. Tews, "Achieving Multi-Robot Cooperation in Highly Dynamic Environments," in School of Computer Science and Electrical Engineering. Brisbane: University of Queensland, 2002.

[2] M. J. Mataric, "Reinforcement Learning in the Multi-Robot Domain," Autonomous Robots, vol. 4, pp. 73-83, 1997.

[3] R. C. Arkin, Y. Endo, B. Lee, D. C. MacKenzie, and E. Martinson, "Multistrategy Learning Methods for Multirobot Systems," presented at In Multi-Robot Systems: From Swarms to Intelligent Automata (Proceedings Second International Workshop on Multi-Robot Systems), Washington D.C., A.C, 2003.

[4] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The Robot World Cup Initiative," presented at IJCAI-95 Workshop on Entertainment and AI/ALife, 1995.

[5] D. Ball, G. Wyeth, and D. Cusack, "Design of a Team of Soccer Playing Robots," presented at Autonomous Minirobots in Research and Edutainment (AMiRE), Brisbane, Australia, 2003.

[6] G. Wyeth, D. Ball, D. Cusack, and A. Ratnapala, "UQ RoboRoos: Achieving Power and Agility in a Small Size Robot," presented at RoboCup 2001, 2002.

[7] G. Wyeth, A. D. Tews, and B. Browning, "UQ RoboRoos: Kicking on to 2000," presented at RoboCup-2000: Robot Soccer World Cup IV, 2001.

[8] G. Wyeth, B. Browning, and A. D. Tews, "UQ RoboRoos: Preliminary Design of a Robot Soccer Team," presented at Lecture Notes in AI: RoboCup '98, 1604, 1999.

[9] B. Browning, "Biologically Plausible Spatial Navigation for a Mobile Robot," in Department of Computer Science and Electrical Engineering: University of Queensland, 2000.