Towards Autonomous Strategy Decisions in the RoboCup Four-Legged League

Michael J. Quinlan, Oliver Obst and **Stephan K. Chalup**
School of Electrical Engineering and Computer Science,
The University of Newcastle, Callaghan 2308, Australia
{michael.quinlan, oliver.obst, stephan.chalup}@newcastle.edu.au

Abstract

Each of the soccer leagues at RoboCup addresses a different aspect of the complex soccer task. The Four-Legged League is the only robot soccer league where more than two real legged robots play in a team. High levels of noise hamper vision and localisation, and therefore deliberate passing occurs rarely in normal game play. The present study highlights some connections to the simulation leagues where eleven agents play in a team and successful passes occur frequently. In the simulation leagues the development of successful global team strategies is at the centre of interest. The experiments in the present study evaluated the impact of varying global team strategies in the Four-Legged League. The NUbots 2006 system was tested against more aggressive and more defensive strategies. The results indicate that global team tactics should be considered in conjunction with a team's style of play. A set of metrics was developed which may enable a future robot soccer team to observe, reason, and modify its global strategy to suit that of an opposing team.

1 Introduction

In human soccer a primary object of discussion is tactics for team play and how to implement successful high-level team strategies in order to beat the opponent team. Several "systems" for team play have been developed by coaches in the international football/soccer community and are described in the relevant literature [Bauer, 2002; Allison, 1967; Wade, 1988]. Another feature of human soccer is that each player has different skills and a different character. Although the players of a particular team follow a common strategy, each player makes his own decisions. How this type of behaviour can be modelled in robot soccer is only one of the major challenges when attempting to "build a team of robot soccer players, which can beat a human World Cup champion team [Kitano, 1998]". Other substantial challenges include, for example, developing lightweight hardware, energy efficient motors, improving computer vision and robust legged locomotion.

Currently the RoboCup Four-Legged League is the only legged league which contains more than two agents per team. The only other legged league is the Humanoid League, which was founded in 2002 and games consist of either one-on-one or two-on-two matches. In the Four-Legged League there are four players per team so it is possible to implement a vast array of formations and other tactical decisions.

At RoboCup [Kitano *et al.*, 1997a; 1997b] strategic team play is primarily implemented in the Simulation League (Section 2) and the Small Size League (f-180) where games are played with 11 vs. 11 and 5 vs. 5 players, respectively. However, these two leagues circumvent major problems associated with vision and localisation by avoiding an individual agent based vision system and running either everything in simulation or on a mechanically precise wheeled platform with an overhead camera. These advantages allow teams in these leagues to focus increased attention on the implementation of global team strategies as opposed to low-level skills such as legged walking.

It could be argued that the result of games in the Four-Legged League was more dependent on low-level skills (such as vision, walking speed etc) than on high-level team strategies. In 2004 it was claimed that passing (and hence complex strategies) would not be necessary in Four-Legged League due to the small field size [Dylla *et al.*, 2005]. In 2005 the field size was increased from $2.7m \times 4.2m$ to $3.6m \times 5.4m$ [Rob, 2004; 2006] and some research on team strategies was performed [McMillen and Veloso, 2006]. However, it is still not clear if variations of a successful global team strategy (e.g. the NUbots system used at RoboCup 2006) would have significant impact on the performance of a team or whether performance in the Four-Legged League is still primarily dependent on low-level skills of the individual agents such as vision, agility, and ball handling which may be better characterized as "style of play" rather than team tactics.

In real football each coach, manager, club etc has an underlying style that tends to guide the approach taken by the team. For example, it is commonly accepted that Brazil plays a free-spirited attacking style of football, and as a result their players have developed extremely strong individual ball skills. In contrast, many European countries play a more physical defensive-minded game resulting in players developing a different and more defensive set of underlying skills. For this reason any discussion on strategy selection should include a discussion on the general style of play and the skill set that has been developed for this style.

In the Four-Legged League there are a few distinct styles of play, but in recent years the dominating styles have been the *Australian* and *German* styles. In fact the past four RoboCup champions[1] and eight of the last nine place holders have played one of these two styles[2].

The two styles differ in a few key areas, but the most evident difference is when a robot is in possession of the ball. The *Australian* style can be roughly defined as "grab, turn (maybe dodge) then kick". In contrast the *German* style tends to favour an approach of 'quick and powerful kicks but with less precision'. As a consequence the teams that play the *Australian* style (i.e. NUbots [Quinlan *et al.*, 2005], rUNSWift[3], UTS Unleashed[4]) have spent substantial time refining their code base to be good at "chasing and grabbing" the ball. This requires a lot of specific vision, localisation, behaviour and locomotion development that other teams may not have needed. Therefore the NUbot set of strategies (and the selection policy) tends to favour an aggressive 'must-control the ball' philosophy that suits the developed skill set.

In order to investigate whether "style of play" or team tactics is the more dominant factor, we take the system of the current RoboCup Four-Legged League world-champion (the NUbots) and experimentally investigate the impact of varying the global team strategy.

From this study, we not only address the effect of modifying the global strategy but take the opportunity to gather information required to further automate the process of global strategy generation and modification.

This article begins by reviewing the state of the Simulated Soccer Leagues, this is followed by an overview on the behaviour system of the NUbots in Section 3. Section 4 presents experiments and results when changing the team strategy. A discussion on the further automating the strategy generation and selection is contained in Section 5. A conclusion follows in Section 6.

2 Implementation of Global Team Strategies in the Simulation Leagues at RoboCup

Among all RoboCup leagues the simulation leagues are the most advanced in the implementation of global human-like soccer team strategies. In the Soccer Simulation League each team plays with 11 agents. Agents possess, compared to robots in the real robotic leagues, an already advanced repertoire of low level skills. The first simulated league, a two-dimensional (2D) soccer simulation league with simple physics, has been in use for over 10 years. A second league implementing a three-dimensional (3D) soccer simulation with a commercial-grade physics engine has existed since 2004. The vision of the agents is handled by the simulator: each agent gets a lists of seen objects and their relative coordinates several times a second. Currently both simulations handle low level locomotion in an abstract way: behaviours like running, turning, and kicking against the ball are executed by sending commands to the simulator. Developers in Soccer Simulation League need to use the vision output and these low level commands to build up world models and the behaviours responsible for navigation. Compared to real robotic leagues, both sensing and acting in the Soccer Simulation League are subject to only a low amount of noise. Communication between agents is fairly restricted, but works reliable if a team uses a good communication strategy.

Some of the successful teams released their code bases containing the navigation level and sometimes the strategic positioning of players. This sharing of code helped others to concentrate on robot team strategies, without the need to re-implement lower lever code.

In the 2D Soccer Simulation League, agents can receive a ball and kick a close ball from any angle. Good positioning and passing skills are key factors for building a successful team. For an example, the 2005 final between Wright Eagle, China and Brainstormers, Germany, contains over 50 successful passes for each team within the first 5 minutes of regular playing time. The offside rule and the team size of 11 players per team further contribute to the necessity of good teamwork and positioning. In many teams, agents play according to different positions and a given home position. Dependent on the current location of the ball and a position specific attraction to the ball, the home position of an agent is adapted during a match. A complete description of a positioning system like this was published by the FC Portugal team which became champion in 2000 [Reis *et al.*, 2001].

There are different ways in which strategies are realised in the Simulation Leagues. The Brainstormer teams successfully implemented an approach making use of reinforcement learning to learn the attack of the team [Riedmiller and Merke, 2002]. A (machine learning) training scenario for the part of attack, when a team tries to just keep the ball possession was defined and researched by Stone et. al. [Stone *et al.*, 2006; Kalyanakrishnan *et al.*, 2007]. Three players have to stay in a rectangular area and try to keep away the ball from two opponent players.

In contrast to learned approaches, rule based approaches have the advantage that strategies can be changed easily during a tournament. In the RoboLog simulation league team [Murray *et al.*, 2001], teamwork has been modeled explicitly using statecharts. Here, the teamwork has to be described at specification time.

In [Kok *et al.*, 2005], teamwork is also described by rules, in this solution only a few rules need to be given and the necessary coordination rules are computed online using coordination graphs. The approach requires more computational power, dependent on the number of applicable rules, but only a small number of rules needs to be specified in order to generate a rich set of coordination rules. In a comparison with the same team using no explicit coordination, the effect of explicit coordination using coordination graphs seems to be a slight improvement in terms of goals scored or ball possession for the price of the higher computation time.

[1] rUNSWift 2003, German Team 2004 & 2005, NUbots 2006.

[2] UPenn finished second in 2003.

[3] http://www.cse.unsw.edu.au/∼robocup/

[4] http://www.unleashed.it.uts.edu.au/

3 The NUbot's System for Team Behaviour

When analysing the effect of team strategies in the RoboCup Four-Legged League it is important to understand where the players are positioned and how the robots perform role, position and strategy switching.

3.1 Positions

The NUbot system defines eight basic positions that the robots can play. These being, *Goal Keeper (GK), Sweeper (SW), Left Back (LB), Centre Back (CB), Right Back (RB), Left Forward (LF), Centre Forward (CF)* and *Right Forward (RF)*.

The rules define that only one robot can be a goal keeper (this robot must be identified at the beginning of the game), while the other three robots (to be referred to as *field players*) are free to roam the field. Unlike other implementations [Veloso *et al.*, 2004] the NUbots positions **do not** define roles, for example the robot chasing the ball can still be a sweeper. It is important to keep a distinction between positions and roles as they are not the same in our system. Each position $\mathcal{P}$ is assigned a set of coordindates on the field (x, y) in cm and a heading θ. These define the general area on the field in which the robot should be positioned:

$$\mathcal{P} = \langle x, y, \theta \rangle \in [-300, 300] \times [-200, 200] \times [-\pi, \pi]$$

For example the positions of Goal Keeper and Left Forward are defined as:

$$\mathcal{P}_{GK} = \langle -265.0, 0.0, 0.0 \rangle\ , \quad \mathcal{P}_{LF} = \langle 150.0, 80.0, 0.0 \rangle$$

3.2 Formations and High-level strategies

A formation $\mathcal{F}$ is a vector that contains four positions. When playing a game each robot in the team is instructed to play one of these four positions. A high-level strategy $\mathcal{S}$ contains two formations, in general these are a defensive $\mathcal{D}$ and an offensive $\mathcal{O}$ formation:

$$\begin{aligned} \mathcal{F} &= \langle \mathcal{P}_a, \mathcal{P}_b, \mathcal{P}_c, \mathcal{P}_d \rangle\ , \\ \mathcal{S} &= \langle \mathcal{F}_{off}, \mathcal{F}_{def} \rangle \equiv \langle \mathcal{D}, \mathcal{O} \rangle \end{aligned}$$

The current NUbot system contains six standard formations that combine to form six high-level strategies:

$$\begin{aligned} \mathcal{S}_{normal} &= \langle \mathcal{D}_{normal}, \mathcal{O}_{normal} \rangle \\ \mathcal{S}_{sweeper} &= \langle \mathcal{D}_{sweeper}, \mathcal{O}_{sweeper} \rangle \\ \mathcal{S}_{aggressive} &= \langle \mathcal{D}_{aggressive}, \mathcal{O}_{aggressive} \rangle \\ \mathcal{S}_{offensive} &= \langle \mathcal{D}_{sweeper}, \mathcal{O}_{aggressive} \rangle \\ \mathcal{S}_{defensive} &= \langle \mathcal{D}_{aggressive}, \mathcal{O}_{normal} \rangle \\ \mathcal{S}_{hold} &= \langle \mathcal{D}_{aggressive}, \mathcal{D}_{aggressive} \rangle \end{aligned}$$

where

$$\begin{aligned} \mathcal{D}_{normal} &= \langle \mathcal{P}_{GK}, \mathcal{P}_{LB}, \mathcal{P}_{RB}, \mathcal{P}_{CF} \rangle \\ \mathcal{O}_{normal} &= \langle \mathcal{P}_{GK}, \mathcal{P}_{CB}, \mathcal{P}_{LF}, \mathcal{P}_{RF} \rangle \end{aligned}$$

$$\begin{aligned} \mathcal{D}_{sweeper} &= \langle \mathcal{P}_{GK}, \mathcal{P}_{SW}, \mathcal{P}_{CB}, \mathcal{P}_{CF} \rangle \\ \mathcal{O}_{sweeper} &= \langle \mathcal{P}_{GK}, \mathcal{P}_{SW}, \mathcal{P}_{LF}, \mathcal{P}_{RF} \rangle \end{aligned}$$

$$\begin{aligned} \mathcal{D}_{aggressive} &= \langle \mathcal{P}_{GK}, \mathcal{P}_{CB}, \mathcal{P}_{LB}, \mathcal{P}_{RB} \rangle \\ \mathcal{O}_{aggressive} &= \langle \mathcal{P}_{GK}, \mathcal{P}_{CF}, \mathcal{P}_{LF}, \mathcal{P}_{RF} \rangle \end{aligned}$$

Using the described framework a new position, formation or strategy can be added in a few lines of code. It is also possible for a $\mathcal{S}$ to contain more then two formations, for example $\mathcal{S} \cong \langle \mathcal{F}_{off}, \mathcal{F}_{midfield}, \mathcal{F}_{def} \rangle$.

3.3 Roles and Role Allocation

In the NUbot system there are two distinct roles, `chasing` and `positioning`. If a robot is not `chasing` then it must be `positioning`. The ideal situation has exactly one robot per team in the `chasing` role.

$$\texttt{chasing} \cap \texttt{positioning} = \emptyset,\ |\texttt{chasing}| = 1$$

Role allocation is done without negotiation, that is, each robot makes a decision on the information it currently has and hopes that this is the same decision the other robots have made. A single variable, *bvrMessage*, is used to make the role decision, this variable contains a float that represents a modified distance from the robot to the ball [Quinlan *et al.*, 2005]. The `chasing` robot is defined as the robot with the lowest *bvrMessage*. The variable also encodes additional information such as attempting a grab, dribbling and kicking. This extra information allows the `positioning` robots to perform different actions depending on what the `chasing` robot is actually doing. For example, the robots should position differently when the `chasing` robot has gained control of the ball and is dribbling.

It is possible that none or multiple robots might momentarily believe they are `chasing`, but in practice the noise and the dynamic nature of the environment means the problem rarely occurs and it is fixed extremely quickly (fractions of a second). An additional problem occurs when a robot loses sight of the ball at close range. In this case it is likely that a second robot will rapidly enter the fray, this may cause collisions until the non-chasing robot is able to move back to its assigned position. This issue is something of a trade-off, in that a less aggressive strategy (i.e. introducing a delay of $\mathcal{S}$ seconds) would prevent the congestion but it could lead to extended periods of time where no robot is moving towards the ball. A general version of the rule would be:

if $delay > \mathcal{S}$ **then** `chasing` $= MIN($*bvrMessage*$)$

Historically the NUbots have preferred to have a `chasing` robot at all times, $\mathcal{S} < \frac{1}{10}$ second.

3.4 High-Level Strategy Switching

Currently the rules for automatic strategy switching are fairly simple. This is primarily due to the lack of information available to the robot. In 2006 the robots relied solely on the current score and the time remaining in the match; the assumption being that the score is indicative of the effect of a strategy.

Prior to the match the coaches (humans) select a strategy ($\mathcal{S}_{current}$). A hierarchy of rules is then used to change this strategy, for example a rule taken directly from our 2006 RoboCup code (in the Python language) was

```
if (secondHalf and (oppScore > ownScore)):
  scoreDiff = oppScore - ownScore
```

```
if (timeLeft/60.0 < scoreDiff):
  newStrategy = AGGRESSIVE
```

For the purpose of this paper we will not discuss the entire rule set, rather emphasise that rules should remain flexible. Our system was designed to make it easy to modify or create new rules for different situations and opponents. It was not designed to completely remove the human element from the strategy decision. It differs from the work of McMillen et al [McMillen and Veloso, 2006] in which the strategies are cycled until a good strategy is found. While this should result in the discovery of a good strategy there exists the possibility of conceding a goal (or two) while rotating through inferior strategies.

3.5 Position Switching

In real football each player is physically different and has different skills and for these reasons they are suited to play particular positions. For all practical purposes each AIBO is identical, and hence the position switching algorithm does not need to consider the attributes of each individual robot. This allows us to base our position switching purely on the physical locations of the robots and the position of the ball on the field.

Again no form of negotiation is used, instead the robots rely on the shared world model being consistent between each other. The system essentially relies on the fact that each robot is capable of replicating the decision of the other robots, and thus the team should function as one.

The first step of the position switching algorithm is to select the relevant formation from the current strategy. Currently this is done by determining a position on the field in which the team should be offensive or defensive. For RoboCup 2006 the following rule was used:

$$\mathcal{F}_{current} = \begin{cases} \mathcal{D}_{current} & \text{ball in defensive half} \\ \mathcal{O}_{current} & \text{ball in offensive half} \end{cases}$$

In this version of the rule only the location of the ball determines offensive or defensive. This could be replaced by a more complicated rule involving possession. For example, if the ball is the defensive half but in the possession of a robot from your team then you should begin a transition to offence.

The goal keeper can not be switched so the algorithm need only consider the positions for the three field players. This is done by using the following rules:

- The `chasing` robot is assigned the position closest to the ball. Note: The goal keeper will never switch positions.
- The `positioning` robots are then free to play any of the remaining positions. This decision is made by minimising the total distance that these robots would need to travel.

3.6 Potential Field Based Positioning

In the NUbot system only off ball positioning is controlled by the potential field. The potential field is not used for path planning, and as a result we are only interested in the potential at the robot at the present time. This allows us to save on computation as the gradient of the potential at this point gives the direction and size of the movement required.

The potential field system has two types of attractors/repulsors: those being points and lines. In both cases the attractiveness/repulsiveness is modelled under a normal distribution and controlled by two parameters, σ (spread) and α (height). A negative α implies a repulsive point or line.

Points

Points are common in most potential field systems [Laue and Röfer, 2005] and are used for object such as other robots, the ball, the goal etc. The current system models: *ball, homePosition, keeper, team1, team2, team3, ownGoal, oppGoal, closeball*. The home position is that decided upon in the formation and position subsections above. In general the robots are attracted to the ball, attracted to their home position, repulsed from their keeper and teammates. The opponent goal and close ball are only used when required, i.e. when dribbling a ball the opponent goal is an attractor and if the robot is a keeper the close ball is also attractive.

Lines and Line Segments

To our knowledge the NUbots are the first team to use potential lines. These are very powerful, and greatly assist in the generation of rules. The repulsive lines exist for all the edges of the field. By having a low σ and high α it can be made it impossible for the robots to drift off the field while positioning (barring a localisation mistake). This enables the construction of the other points of potential without giving thought to a robot leaving the field, i.e. if the robot is repulsed from the ball and the ball is nearing the sideline, then the robot will not walk off the field to avoid the ball rather it will move in some direction (forwards or backwards) to avoid both the ball and the line. Lines of repulsiveness were particularly useful for controlling the dribbling behaviour. The original line implementation [Quinlan *et al.*, 2005] only contained lines of infinite length, which were useful for the edge of field but proved inadequate for more complicated structures such as the penalty box. This limitation was overcome by introducing line segments [Knorn, 2006] see Figure 1.

3.7 Low-Level Strategies

In addition to high-level strategies the same decision code allows for low-level strategy modifications. Currently the modifications allow us to adjust our underlying aggressiveness through the following variables: *Slap Defensive, Slap Offensive, Veer* or adjust to rule interpretations: *Illegal Defender Chase, Illegal Defender Position, Ram Over Line, Stop Dribble Over Line* and finally turn on/off some experimental code: *Chase Velocity, Chase Grab Adjust*.

The general concept of low-level strategies was to define a central place to control how an individual robot plays soccer as opposed to high-level strategies which are more concerned with the global team approach. By incorporating code for both low-level strategy decisions and high-level strategy decisions the robots have the ability to change many aspects of their play to suit the current state of the game.

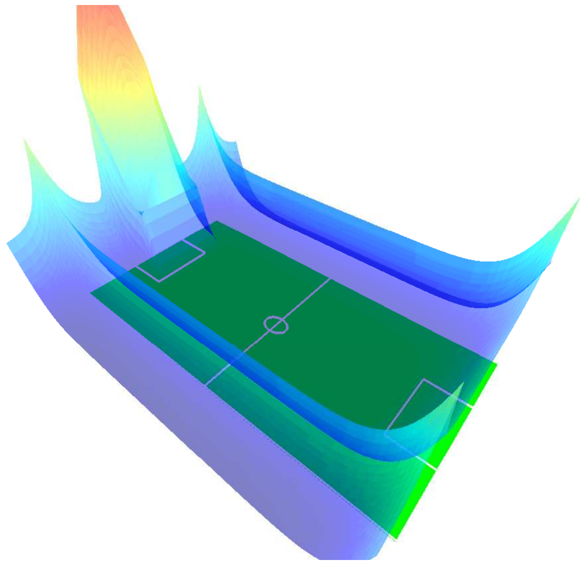


Figure 1: Potential Lines and Line Segments. The side-lines and goal-lines have repulsive lines, the defensive penalty box is incorporated using repulsive line segments.

4 Results of Manually Changing Strategies

RoboCup 2006 should have provided the ideal test bed for the strategy switching code, but unfortunately (for this paper) the NUbots never trailed in a game. As a consequence it was not possible to gain any real-world data as to the effect of strategy switching against an unknown opponent. To compensate a set of experiments was derived to test the effectiveness of different strategies.

The experiments took place in two phases. First, control games were run to gather key information on the performance of the standard strategy. For example, the average game score and the variance in this score. Secondly, the strategy of one team was varied, therefore measuring the impact of the strategy change on the result of the game.

4.1 Setup of the Experiment

- Two identical teams[5] play 10 games, each game will go for 5 minutes.
- The games will be 4 on 4, but the goal keeper on each side will remain stationary. The motionless aspect allows more goals and hence will provide more statistical information. Note that the keeper is identical in each strategy and therefore the stationary decision equally affects both teams.
- Calculate the variance in results when both teams play with our "preferred"-known strategy ($\mathcal{S}_{Sweeper}$).
- Change $\mathcal{S}$ on one of the teams, and play a number of games with this varied strategy.

Each match was video recorded and further statistics gathered, for example, *shots at goal* and *ball position*.

[5]The software is identical but the physical condition of the AIBOs may vary slightly between teams.

4.2 Results

The set of 10 control matches played with both teams using $\mathcal{S}_{sweeper}$ resulted in 4 wins for the *Red* team, 3 wins for the *Blue* team and 3 *draws*. The two teams scored 13 and 14 goals respectively (see Tables 1 and 2). The distribution of the *ball position* in an average match (a 1-1 draw) can be found in Figure 2a. It can be seen that the ball spent the majority of the match in the midfield ($-100 \leq x \leq 100$). The scores of the matches and the evenness of the ball distribution indicates that the two teams were playing at an equal level.

Team	Control Matches	*Red* Off	*Red* Def
Red	4	0	1
Blue	3	3	2
Draw	3	2	2

Table 1: Number of wins by each team during the experiments.

Team	Control Matches	*Red* Off	*Red* Def
Red	1.3 ± 1.160	$\mathbf{0.6 \pm 0.894}$	1.0 ± 0.707
Blue	1.4 ± 0.9949	2.0 ± 1.225	1.4 ± 0.894
Margin	1.1 ± 0.994	1.4 ± 1.341	$\mathbf{0.6 \pm 0.548}$

Table 2: Average goals scored by each team during the experiments.

Team	Shots at Goal	Pass (catch)	Pass (near)
Control			
Red	2.2 ± 1.317	0.1 ± 0.317	0.7 ± 0.483
Blue	1.9 ± 1.287	0.2 ± 0.422	0.8 ± 0.634
Red Off			
Red	0.8 ± 0.837	0.0 ± 0	0.6 ± 0.578
Blue	$\mathbf{3.8 \pm 1.095}$	0.4 ± 0.578	1.0 ± 1.0
Red Def			
Red	1.6 ± 1.140	0.0 ± 0	0.0 ± 0.0
Blue	1.6 ± 0.894	0.2 ± 0.447	0.4 ± 0.5478

Table 3: Statistics gathered during the experiment. *Shots at goal* are any deliberate kicks at goal, both on target and off target shots are counted. *Pass (catch)* is any situation where a robot kicks the ball directly to a team-mate who then makes a clean catch on the ball. *Pass (near)* occurs when a robot kicks a ball in the direction of a team-mate, the team-mate then moves to gain control of the ball (through-pass).

The second set of matches involved the strategy for the *Red* being changed to $\mathcal{S}_{offensive}$. This strategy was designed to be played when the team is trailing in a match and required quick goals. The key difference between $\mathcal{S}_{sweeper}$ and $\mathcal{S}_{offensive}$ is on the offence aspect of the game, now all three field players will position inside the offensive half. Defensively $\mathcal{S}_{sweeper}$ and $\mathcal{S}_{offensive}$ are identical. It was expected that $\mathcal{S}_{offensive}$ would score more goals (with three attackers), but it would also be extremely susceptible to counter-attacks.

In practice $\mathcal{S}_{offensive}$ performed poorly in comparison to the standard strategy. The *Red* team failed to win a game (Table 1) and its goal average halved from 1.3 to 0.6 goals per game. In contrast the scoring of the *Blue* team increased from 1.4 to 2.0 goals per game (Table 2). A sample *ball position* distribution (Figure 2b) indicates that the possession was again in the midfield but the key difference was the lack of defensive support, therefore situations occurred where the $Blue$ team could quickly progress the ball from midfield to the area surrounding the yellow goal ($x \approx 300$). This fact is evident when comparing the *shots at goal*. In the control games the average *shots at goal* for the two teams was 2.05. When Red was playing offensively the $Blue$ team increased this to 3.8 *shots at goal* per game (Table 3).

The third set of experiments had the Red strategy changed to $\mathcal{S}_{hold}$. This is an extremely defensive strategy, at all stage of the games at least two field players will be in the defensive half. When the ball is in the defensive half all three field players will be playing defence, while on offence a loan attacker will be allowed to move into the offensive half. $\mathcal{S}_{hold}$ was designed to be used when ahead, and the intention was to 'play' out the game without conceding any further goals.

Somewhat surprisingly $\mathcal{S}_{hold}$ was quite competitive with our standard strategy. In the 5 games, Red won 1 game, $Blue$ won 2 games and the teams drew on 2 occasions (Table 1). The scoring in the matches was similar to that of the control matches, Red scored on average 1.2 goals (in comparison to 1.3) while $Blue$ scored an identical 1.4 goals per game. The key difference was in the margin (Table 2), that is, the average goal difference in each match. In the control matches games were won by an average of 1.1 ± 0.994 goals, this is basically saying that the while the results were even (4 wins, 3 wins and 3 draws) the actual matches were decided by a margin of anywhere between $0 \rightarrow 2$ goals. In the $\mathcal{S}_{hold}$ matches the margin was 0.6 ± 0.548 goals per game, this indicates that the matches were much closer (decided by 0 or 1 goals). The plot of the *ball position* (Figure 2c) indicates that $Blue$ had slightly better control of the ball ($x > 0$) but the score indicates that the increased presence of the Red defenders was enough to handle the additional pressure.

4.3 Discussion on Results

The results indicated that changing the global strategy of a team *can* affect the result. Often the true effect cannot be measured by statistics alone, but rather it requires human observation.

In contrast to the Simulation Leagues the Four-Legged League has **no** offside rule, therefore attacking robots can position as far up the field as they choose. The robots can not move as quickly or as fluidly as players in simulation or human players, and therefore the off-ball positioning of the robots becomes more critical. Additionally the ball does not travel in the air (due to the physical properties of the robots); the effect being that robots can not simply kick the ball over the head of an opponent to gain field position, rather they have to move the ball along the ground at all times.

As a result it appears that the key aspect of any strategy is to keep players behind the ball. While this is true in all forms of soccer, it appears to be even more true in this form. This fact became evident by observing how poorly the $\mathcal{S}_{offensive}$ strategy performed.

This leaves two options: Should all the robots attempt to position behind the ball (i.e. $\mathcal{S}_{hold}$) or should an attacker be positioned further up field (i.e. $\mathcal{S}_{sweeper}$)? The results indicate that both strategies have merit and this decision may actually depend on the "style of play" (discussed in the introduction). The NUbots have chosen the more offensive strategy (an upfield attacker) since we believe two defenders are capable of controlling the ball. Other teams such as the German Team [Röfer *et al.*, 2004] have been extremely successful in employing the more defensive strategy. These experiments have confirmed that successful strategies (and teams) are built on a solid defence, a fact supported by observing that the top RoboCup teams concede the fewest goals.

5 Automatically Generating and Modifying the Strategy

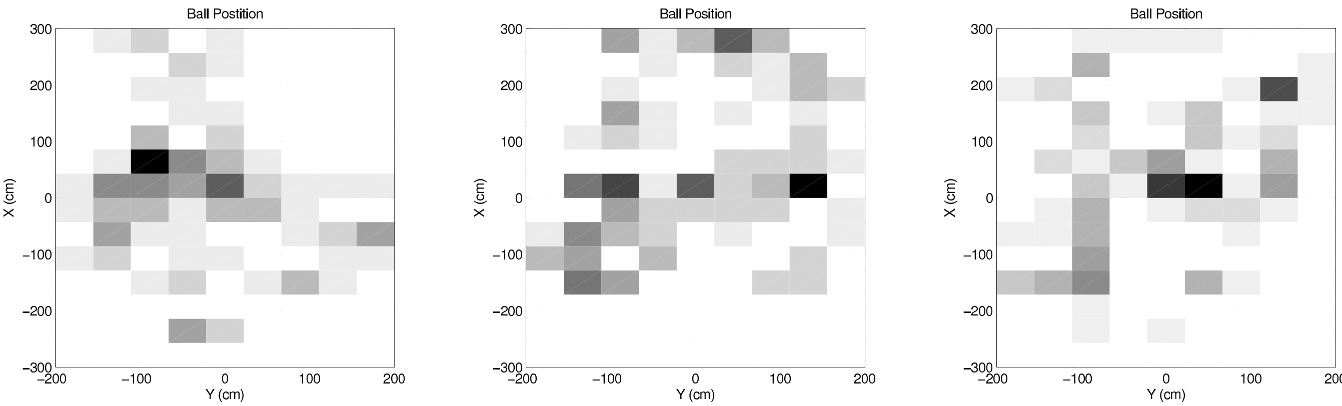
The above study into manually changing the strategies, discovered metrics that can easily be programmed into the robots' strategy selection code. For example, the *score*, the *shots at goal* and the *ball position* histogram can be calculated during the game by the robots (the three *passing* statistics will be harder to calculate). Further analysis of the videos and the gathering of more statistics (mainly *intercepts* and *tackles*) should provide for a more complex set of rules that govern the strategy switching code. The future development of the metrics and heuristics is vital to having the robots successfully implementing real-time adaptive strategies.

When considering the next phase, that is, creating the reasoning system required to take in the all the data and produce a decision, it is best to look at an example. Consider the question:

How can our own team play more aggressively?

The exact answer is unknown as a team can play more aggressively by changing any number of variables. For example aggressiveness can be increased by: changing the strategy to $\mathcal{S}_{aggresive}$, or changing the positions in a formation, or changing the (x, y) in a position, or increasing α and σ (Section 3.6) for the ball in the potential field, or even by reducing $\mathcal{S}$ to 0. The problem is further complicated by not having perfect feedback about the result of any change. For example, a change may be working perfectly (from the perspective of a human overlooking the game), but a bad roll of the ball or a random mistake could easily cause a metric to mislead the robot into thinking the change was bad.

What to change and how to change it may never be clear as it is impossible to test all variables combinations (considering it took 5 days to run the 20 games used in the manual study). This leads to the use of simulation as a mechanism for learning the initial set of rules and transitions. For this reason we have started to investigate (Section 2) whether techniques that have been honed in the simulated leagues can be adapted for use in the Four-Legged League. Of course many problems are evident, for example, the (almost) perfect environment of the simulated leagues differs substantially from the noisy environment of the four-legged league. In the past this has meant that essentially no lessons have been passed between the two leagues. Only recently have the low-level skills of the AIBOs increased to a point that passing and team play is a real option and therefore it may be possible to take advantage of the work being done in the simulated leagues.



(a) Control Match (1-1) (b) *Red* Offensive (0-2) (c) *Red* Defensive (1-2)

Figure 2: Histogram of ball position during a typical match of each experiment type. The blue goal (defended by the *Blue* team) is at $x = 300$ while the yellow goal (defended by the *Red* team) is at $x = -300$. The field size is 400cm $\times$ 600cm

It is hoped that through the use of simulation and then limited game testing we can develop a system that will allow our robots to react to the state of the game and modify its strategy to suit. Since RoboCup code development is done in isolation (teams only see opponent teams once or twice a year), it is impossible to pre-generate rules to counter all opposing strategies. For this reason the system must not only be capable of selecting an appropriate strategy but also be capable of producing new strategies on the fly. This will be the key step in producing a robotic team system that truly learns before adapting.

6 Conclusion

This paper includes results from the first study in the Four-Legged League league to experimentally compare different global team strategies, the question being asked in that study was "has the league progressed to a point in which team strategies really affect the game?". The result has shown that a change in global strategies can indeed affect the result of a game. However, the difference between the alternative strategies employed in the experiments is minor: 40% of the games with different strategies ended up in a draw (30% of the control games were drawn). This indicates that the "style of play" and skills of the robot are still the most critical aspect of the robot system. The answer to the question above would be "strategies can affect the game but only when the two teams are already evenly matched".

However, the study only modified a fraction of the variable available in the system, it all but ignored low-level changes (such as modifying the potential field or changing positions in a formation). Taking the step from "strategy selection" to "strategy creation and selection" is the next evolution and will not only require the use of simulation and real-robot experiments, but it will require extensive work in the area of rule generation and reasoning.

References

[Allison, 1967] M. Allison. *Soccer for Thinkers*. Pelham Books, 1967.

[Bauer, 2002] G. Bauer. *New Soccer Techniques, Tactics & Teamwork*. Sterling Publishing, 2002.

[Dylla *et al.*, 2005] Frank Dylla, Alexander Ferrein, Gerhard Lakemeyer, Jan Murray, Oliver Obst, Thomas Röfer, Frieder Stolzenburg, Ubbo Visser, and Thomas Wagner. Towards a League-Independent Qualitative Soccer Theory for RoboCup. In Daniele Nardi, Martin Riedmiller, Claude Sammut, and José Santos-Victor, editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 611–618. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[Kalyanakrishnan *et al.*, 2007] Shivaram Kalyanakrishnan, Yaxin Liu, and Peter Stone. Half Field Offense in RoboCup Soccer: A Multiagent Reinforcement Learning Case Study. In Gerhard Kraetzschmar, Elizabeth Sklar, Domenico Sorenti, and Tomoichi Takahashi, editors, *RoboCup-2006: Robot World Cup X*. Springer-Verlag, 2007.

[Kitano *et al.*, 1997a] Hiroaki Kitano, Minoru Asada, Yasuo Kuiyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. RoboCup: A Challenge Problem for AI and Robotics. In Kitano [1998].

[Kitano *et al.*, 1997b] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. RoboCup: A Challenge Problem for AI. *AI Magazine*, 18(1):73–85, Spring 1997.

[Kitano, 1998] H. Kitano, editor. *RoboCup-1997: Robot Soccer World Cup I*, volume 1395 of *LNCS*. Springer, 1998.

[Knorn, 2006] Florian Knorn. Algorithm Development & Testing for Four Legged League Robot Soccer Passing. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, Deutschland (undertaken at the University of Newcastle, Australia), 2006.

[Kok *et al.*, 2005] Jelle R. Kok, Matthijs T. J. Spaan, and Nikos Vlassis. Non-communicative multi-robot coordination in dynamic environments. *Robotics and Autonomous Systems*, 50(2-3):99–114, February 2005.

[Laue and Röfer, 2005] Tim Laue and Thomas Röfer. A Behavior Architecture for Autonomous Mobile Robots Based on Potential Fields. In *RoboCup 2004: Robot world Cup VIII*, volume 3276 of *LNAI*. Springer-Verlag, 2005.

[McMillen and Veloso, 2006] Colin McMillen and Manuela Veloso. Distributed, Play-Based Role Assignment for Robot Teams in Dynamic Environments. In *DARS 2006*, July 2006.

[Murray *et al.*, 2001] Jan Murray, Oliver Obst, and Frieder Stolzenburg. Towards a Logical Approach for Soccer Agents Engineering. In Peter Stone, Tucker Balch, and Gerhard Kraetzschmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, number 2019, pages 199–208. Springer, Berlin, Heidelberg, New York, 2001.

[Quinlan *et al.*, 2005] M.J. Quinlan, S.P. Nicklin, K. Hong, N. Henderson, S.R. Young, T.G. Moore, R. Fisher, P. Douangboupha, S.K. Chalup, R.H. Middleton, and King. R. The 2005 NUbots Team Report. EECS technical report, University of Newcastle, 2005.

[Reis *et al.*, 2001] Luis P. Reis, Nuno Lau, and Eugnio C. Oliveira. Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents. In Markus Hannebauer, Jan Wendler, and Enrico Pagello, editors, *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, LNCS 2103, pages 175–197. Springer, 2001.

[Riedmiller and Merke, 2002] Martin Riedmiller and Artur Merke. Using machine learning techniques in complex multi-agent domains. In I. Stamatescu, W. Menzel, and U. Ratsch, editors, *Perspectives on Adaptivity and Learning*. Springer, 2002.

[Rob, 2004] RoboCup Technical Committee. *Sony Four Legged Robot Football League Rule Book*, may 20, 2004 edition, 2004.

[Rob, 2006] RoboCup Technical Committee. *RoboCup Four-Legged League Rule Book*, april 20, 2006 edition, 2006.

[Röfer *et al.*, 2004] T. Röfer, T. Laue, H-D. Burkhard, J. Hoffmann, M. Jungel, D. Gohring, M. Lotzsch, U. Duffert, M. Spranger, . B. Altmeyer, V. Goetzke, O. v. Stryk, R. Brunn, M. Dassler, M. Kunz, M. Risler, M. Stelzer, D. Thomas, S. Uhrig, U. Schwiegelshohn, I. Dahm, M. Hebbel, Nistico W, C. Schumann, and M.Wacher. GermanTeam 2004. Technical report, 2004.

[Stone *et al.*, 2006] Peter Stone, Gregory Kuhlmann, Matthew E. Taylor, and Yaxin Liu. Keepaway Soccer: From Machine Learning Testbed to Benchmark. In Itsuki Noda, Adam Jacoff, Ansgar Bredenfeld, and Yasutake Takahashi, editors, *RoboCup-2005: Robot Soccer World Cup IX*, volume 4020, pages 93–105. Springer, Berlin, 2006.

[Veloso *et al.*, 2004] M. Veloso, P. Rybski, S. Chernova, D. Vail, S. Lenser, C. McMillen, J. Bruce, F. Tamburrino, J. Fasola, M. Carson, and A. Trevor. CMPack04: Team Report. Technical report, Carnegie Mellon University, 2004.

[Wade, 1988] A. Wade. *Soccer Strategies*. Heinemann Kingswood, 1988.