

Robot Soccer: Science or Just Fun and Games?

Claude Sammut **

School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
`claude@cse.unsw.edu.au`

Abstract. RoboCup is an international robot soccer competition that has been running since 1997. A frequent criticism of any such competition is that all of a team’s development is narrowly focussed on a very specific task and consequently, little contribution is made to science. In this paper, we describe our involvement in the RoboCup four-legged robot league and show how the competition has, indeed, resulted in contributions beyond robot soccer. We describe innovations in vision, localisation and locomotion. In particular, we discuss the stimulus that RoboCup has given to system integration and cooperative behaviour between multiple agents.

1 Introduction

The RoboCup competition was created to provide an incentive for researchers to work, not only on specific problems in robotics, but to build integrated systems to solve a very complex but well-defined problem. However, a frequent criticism of any such competition is that all of a team’s efforts are narrowly focussed on shortcuts for winning games and consequently, little contribution is made to science. In this paper, we suggest that the answer to the question: “is robot soccer science or fun and games”, is “it’s both”.

The benefits of the competition are evident in the progress from one year to the next. The competitive nature of the research has forced participants to evaluate very critically their methods in a realistic task as opposed to experiments in a controlled laboratory. It has also forced researchers to face practical hardware and real time constraints. In particular, there are very few domains in which such strong emphasis is placed on multi-agent cooperation. Despite the serious nature of the scientific endeavour, the “fun” aspect of RoboCup is important as a way of encouraging bright students into robotics and AI.

In this paper, we will describe the involvement of the rUNSWift team in the RoboCup four-legged robot league and the innovations that have resulted in vision, localisation, locomotion and cooperative multi-agent behaviour. Hopefully, this will answer the question of whether there is science in RoboCup. The work described has been the accumulated effort of UNSW teams from 1999 to 2003

** The work described here is the collective effort of successive UNSW RoboCup teams since 1999. All the team members are listed in the acknowledgements.

and, in 2003, jointly with NICTA. The participation of all team members is gratefully acknowledged. They have been finalists in all five years of the official RoboCup competition¹ and three times world champions, including 2003.

AI has often suffered from either attempting to tackle problems that were overly ambitious or else using toy problems that have been simplified to the extent that the real problems have been removed. Although robot soccer is a highly constrained domain, it is, nevertheless, sufficiently complex that it preserves most of the central problems of robotics. It seems to have struck a good balance between complexity and attainability. Furthermore, the rules of the competition adapt, each year, introducing more challenges as teams come to terms with previously unsolved problems.

For a mobile robot to operate effectively it must be capable of at least the following:

- perceive its environment through its sensors;
- use its sensory inputs to identify objects in the environment;
- use its sensors to locate the robot relative to other objects in the environment;
- plan a set of actions in order to achieve some goal;
- execute the actions, monitoring the progress of the robot with respect to its goal;
- cooperate with other agents that have a the same goal and thwart the efforts of opponent agents.

Robot soccer requires a solution to all of these problems. Most importantly, they must be integrated into a complete system that must operate in real-time in a highly dynamic environment. So the solutions must be practical and robust. In the following sections, we first describe the robot soccer competition and then discuss how we have approached each of the above problems.

2 The Four-legged Robot League

The four-legged robot league is one of several leagues that currently form the RoboCup competition. In some leagues, the teams are required to design and build the robot hardware themselves. In the case of the four-legged robot league, all teams use the same robotic platform, manufactured by Sony. Since all teams use the same hardware, the competition is based on a combination of a team's creativity in programming and also its discipline in making it work reliably.

The robot used in the four-legged league Sony's AIBO. Although designed for the entertainment market, this robot is quite sophisticated. The model ERS-210A robot, used in 2003, has an on board MIPS R5200 384MHz processor, 176x144 colour CMOS camera, accelerometers, contact sensors, speaker and stereo microphones. Each of the four legs has three degrees of freedom, as does

¹ A demonstration competition run in 1998, won by CMU. The principal team member, Will Uther, is now with NICTA

the head. Programs are written off-board in C++ and loaded onto a memory stick that is inserted into the robot. All of Sonys ERS robots run a proprietary operating system called Aperios, which provides an interface for the user-level programs to the sensors and actuators.

Four robots make a team, with one usually being designated the goalie. The field for in the four-legged robot league is shown in Figure 1. It has an angled white border designed to keep the ball within the field. The game ball is coloured orange and the two goals are coloured yellow and blue. The field also contains six coloured landmark poles to aid the robots in localising themselves in the field. In front of each goal there is the penalty region that only the goalie is allowed to defend.

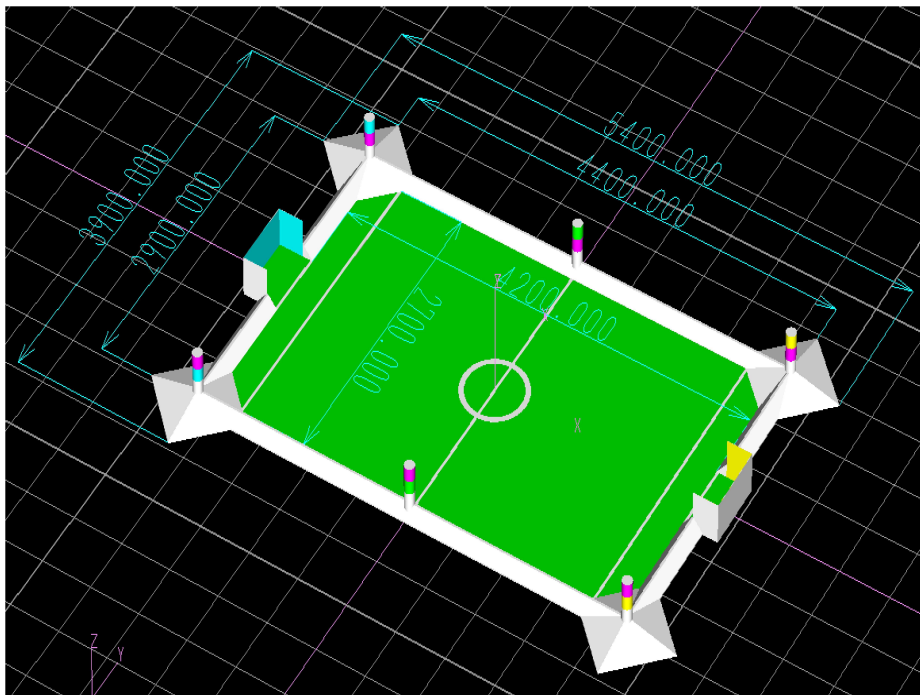


Fig. 1. The RoboCup playing field

Games are 10 minutes per half. The robots from each team wear “uniforms” that are red or blue stickers. The blue team defends the blue goal and the red team defends the yellow goal. Teams sides at half-time and also swap colours. Because colour recognition can be quite difficult, as we will see in the next section, a team may play better as one colour or the other, so to even out this effect, team colours are swapped.

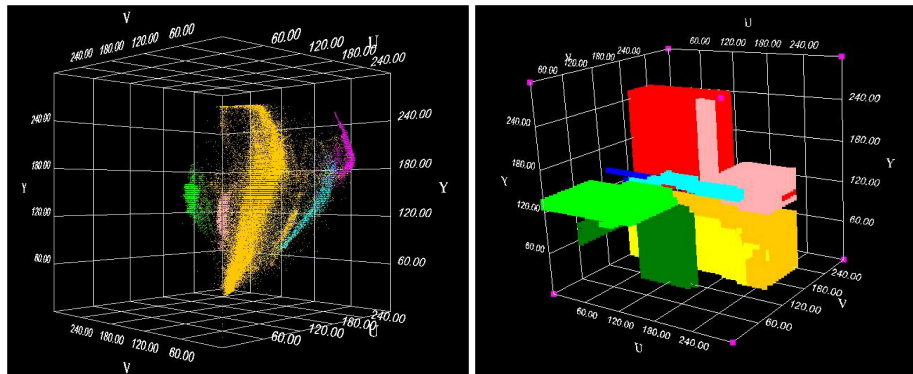
3 Vision

3.1 Colour Classification

A colour camera is the robots primary sensor. The ERS-210A, has a 176x144 resolution camera mounted in the “snout” of the robot. This delivers 25 frames per second. The information in each pixel is in YUV format, where Y is the brightness and U and V are the colour components. Since all the objects on the field are colour coded, the aim of the first stage of the vision system is to classify each pixel into the eight colours on the field: orange for the ball, blue and yellow for the goals and beacons, pink and green for the beacons, light green for the field carpet, dark red and blue for the robot uniforms.

Despite the large amount of activity in digital camera technology, the recognition of colours remains a difficult problem and is an area in which RoboCup has made a significant contribution. The main difficulty in colour recognition is that the characteristics of the light source have a major influence on how a camera detects colour. Therefore, we need a vision system that can be easily adapted to new surroundings. One way of doing this is by training the vision system to recognise colours, given samples of images from a particular environment.

The first step is to take sample images of different objects at different locations on the field. Then for each image, every pixel is classified by “colouring in” the image by hand, with the help of a custom painting tool. All these pixels form the set of training data for a learning algorithm. Each pixel can be visualised as a point in a 3 dimensional space, as shown in Figure 2(a).



(a) Colour data

(b) C4.5 regions

Fig. 2. Using C4.5 to learn to recognise colours

Figure 2(b) shows the regions grown by Quinlans C4.5 decision tree learning program [3]. C4.5’s input is a file of examples where each example consists of

a set of attribute values followed by a class value. In this case, the attributes are the Y, U and V values of a pixel and the class value is the colour manually assigned to that pixel. C4.5 turns the training data into a set of classification rules that should be able to determine the colour of a new, unclassified pixel.

Although colour classification with C4.5 generated decision trees has proved to be quite reliable, the disadvantage of this approach is that it requires quite a lot of manual labour to collect and label the sample images. This means that it takes time to recalibrate if conditions change, for example, of crowds around the field cast shadows, etc. Therefore, it is desirable to further automate colour calibration. Recently, Cameron and Barnes [1] have developed a method for using the constraints of the field to suggest the colour of region, without requiring human assistance. Unsupervised learning, using clustering techniques may also help to automate colour recognition.

3.2 Object Recognition

Once colour classification is completed, the object recognition module takes over to identify the objects in the image. The possible objects are: the goals, the beacons, the ball and the blue and red robots. A blob formation algorithm links neighbouring pixels of the same colour in the image to form blobs. Based on the blobs, we identify the objects, along with their distance, heading and elevation relative to the camera and the base of the neck of the robot.

Objects are identified in the order: beacons first, then goals, the ball and finally the robots. Since the colour uniquely determines the identity of an object, once we have found the bounding box around each colour blob, we have enough information to identify the object and compute its parameters. Because we know the actual size of the object and the bounding box determines the apparent size, we can calculate the distance from the snout of the robot (where the camera is mounted) to the object. We also calculate heading and elevation relative to the nose of the robot and the bounding box's centroid. However to create a world model, needed for strategy and planning, measurements must be relative to a fixed point. The neck of the robot is chosen for this purpose. Distance, elevations and headings relative to the camera are converted into neck relative information by a 3D transformation using the tilt, pan, and roll of the head.

While giving objects unique colours makes object recognition much easier than in a completely unconstrained vision problem, there are still many subtle difficulties to be overcome. For example, every beacon is a combination of a pink blob directly above or below a green, blue or yellow blob. One side of the field has the pink on the top of the colour in the beacons, while the other has it below. The beacons are detected by examining each pink blob and combining it with the closest blob of blue, yellow or green. This simple strategy works most of the time but fails occasionally. When the robot can just see the lower pink part of a beacon and the blue goal, it may combine these two blobs and call it a beacon. A simple check to overcome this problem is to ensure that the bounding boxes of the two blobs are of similar size and the two centroids are not too far apart.

Over the years of that we have been competing, the vision system has accumulated a large library of such situation-specific heuristics. Knowledge-based image understanding seems to be a requirement for a reliable vision system. However, like all knowledge acquisition tasks, we have the problem of building a consistent and reliable set of heuristics. It is very easy to add new rules to the knowledge base that interfere with previous rules. Discovering such interactions is difficult because they might occur in only a few frames, remembering that the camera images are processed at 25 frames per second. If a heuristic triggers incorrectly at a crucial moment, for example, while attempting to grab the ball for a kick, its effects could be serious. For this reason, most teams create debugging tools for monitoring image processing. Since each robot has an onboard wireless LAN card, it is possible to transmit debugging information to another computer. We currently have a semi-automated system for capturing the firing of heuristics and checking them against the desired result. An obvious improvement will be to apply machine learning.

To achieve reasonable performance, most “real-world” applications of robotics require a lot of task-specific engineering. Thus, any method that can help in automating the acquisition of domain knowledge is of interest. The RoboCup environment, once again, provides a challenging, but bounded, test bed for such work.

3.3 Line Detection

RoboCup imposes a very strict discipline on the implementor. Being a fast moving game, it is important that the robots are able to process incoming sensor data and make decisions as quickly as possible. It is usually preferable to compromise on the amount of processing done so that the system can keep up with the frame rate of the camera. For this reason, we try to find the simplest image processing operations that yield enough information to be able to play effectively. Of course, how much processing we can afford changes as hardware improves. However, the real-time constraints force us to develop very efficient algorithms, which often requires some radical rethinking of how something should be done.

A recent addition to the robot’s repertoire, thanks to a faster CPU, is the detection of lines such as the field boundaries, centre and goal lines. In the early stages of development, we attempted to use well-known techniques such as the Hough Transform. However, this proved to be both time consuming and not well-suited to the task for which it was intended, namely, helping to localise the robot on the field. After much experimentation, another knowledge-based method was used. Knowing the lines that ought to be visible on the soccer field, the system collects edge-points and attempts to map them onto the lines that should be in the robot’s field of view, assuming it has a rough idea of where it is. There are usually several candidate lines. One is chosen according to which mapping has the smallest error.

One might argue that such domain-specific methods are of little general use and therefore, the scientific merit of robot soccer is diminished. On the contrary, we believe that to achieve adequate performance in practical tasks, domain specific

knowledge is essential. Our challenge is to understand this kind of knowledge and develop ways of learning domain-specific methods.

4 Localisation

A common problem in mobile robots of all kinds is localisation, that is, maintaining an accurate estimate of its position. The most complex form of the problem requires the robot to build a map of an unknown environment while simultaneously locating itself within the map. In robot soccer, this is simpler because we have exact measurements of the field. Even given an accurate map, it is still not straightforward to localise since all sensory data are inherently noisy. Another important variant of the localisation problem occurs when there are several agents that can share information. In principle, sharing information should improve localisation, but since no robot knows its exact location and all sensors are noisy, how can the diverse streams of information be merged? This is the problem of *distributed data fusion*.

Information from the robot's vision system, as well as odometry from the locomotion module, are combined to create a world model. Since the dimensions of the field are known in advance, the world model is constructed by placing the mobile objects on a map of the field. As objects are observed to move, their positions in the map are updated. However, since observations can be inaccurate, the update must be done cautiously. We use a method based on the Kalman filter [2], which maintains an estimate of the error in the current position.

One of the innovations in 2003 was to combine information from objects recognised by their colour with the new line detection algorithms described earlier. There are situations in which localisation based on edges is more accurate than by using landmarks. The ball often strays to the field boundaries or into the corners. When a robot approaches either area, it very often loses sight of the landmarks and soon loses localisation. It gets lost. In such cases, we use line detection as a backup. Being somewhat more expensive to execute, line detection is used sparingly. While our method of line detection was the most accurate to appear in the 2003 competition, the German Team developed a method that was faster and therefore could be used more frequently. Arguably, they made the better trade-off between speed and accuracy.

Figure 3 shows a snapshot of the world model being maintained by one of the blue robots during a game. The ellipses around the objects indicate the size of the error in x and y while the shaded sector indicates the error on heading of the robot. As long as the robot has sight of at least two coloured landmarks, its position estimate will be quite accurate. Since distance is estimated by the size that an object appears in an image, distance to landmark is usually less accurate than the robot's angle, relative to the landmark, hence the elliptical shape of the error estimate.

During a game, each robot on the field receives information from every other robot. This means that a robot must reconcile the world models of all three other robots and its own. Each robot maintains an additional, *wireless model*

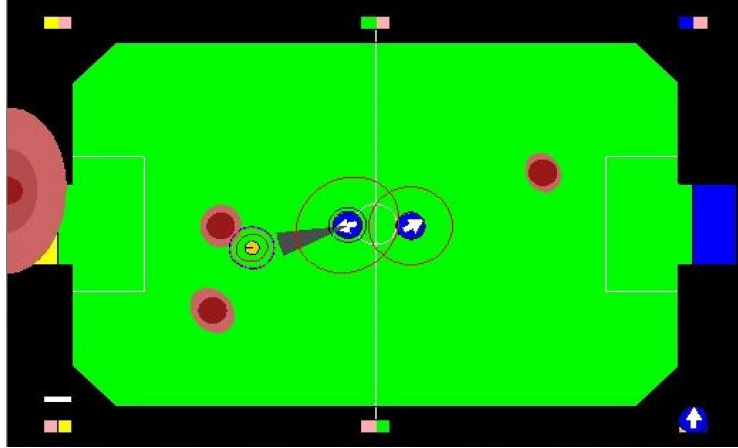


Fig. 3. The world model

that is a representation of the world as given by a robot's team mates. It is kept separate from the robot's own world model. We associate with each team mate a variance which represents our confidence in the accuracy of the information from that robot. The variance of an object in the wireless model is the sum of the team mate's variance, the object's variance, as believed by the team mate, and a small variance for the latency over the wireless network.

The general problem of distributed data fusion remains a major challenge for multi-agent systems. Among the problems to be solved is in combining several world models how do we know that if they refer to the same object. For example, Figure 4, shows two world models on the left. One robot is aware of two red robots and one blue robot. The other is aware of three red robots and one blue robot. Our present solution is a heuristic. If all the supposed robots are placed in one world model, where the variances of two robots overlap, they are combined into one.

Once the information is merged, the best (lowest variance) four robots of each colour are stored in the wireless model. The same process is also applied to combine information for balls. In this case, only the best ball is selected. In the wireless model, three extra robots are also stored. These describe the location of the team mates according to their own belief.

The solutions to the distributed data fusion problem being developed for robot soccer are relevant to other applications. Another competition in RoboCup is concerned with rescue. Teams in the rescue league construct robots that are intended to search through collapsed buildings and similar dangerous environments, searching for survivors and delivering aid. Because of the complexity of the environment, this presents many problems that are avoided when moving about a soccer field. However, any real rescue operation will involve multiple robots, possibly of different construction. They will have to share information to

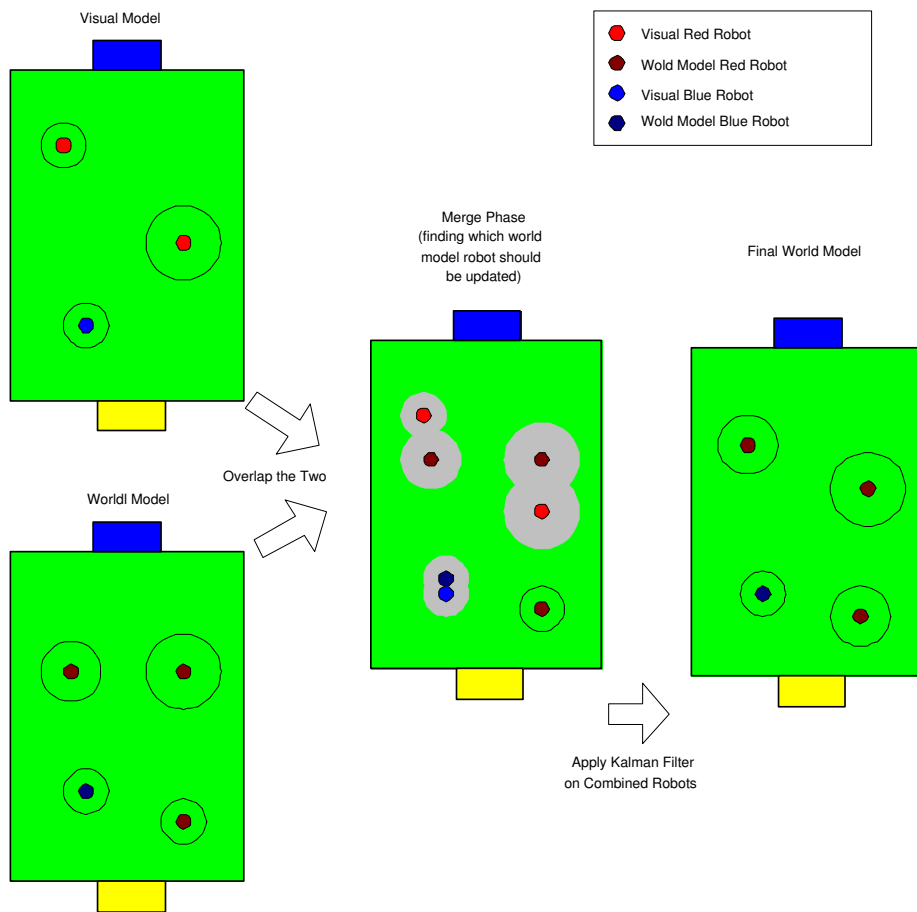


Fig. 4. Updating Multiple Robot Locations

coordinate their efforts and they will have to decide on the identity of objects just as described above. The techniques being tested in the soccer competition for this problem, and others, will carry across to the more socially relevant tasks such as rescue.

5 Locomotion

In RoboCup, speed is crucial. The team that is able to reach the ball first, usually wins. Speed results from a combination of fast and reliable vision, localisation, decision making and, obviously, locomotion.

Although Sony provides a library of walking routines with the Aibo, these are not designed for playing soccer and lack the speed and manoeuvrability needed to play effectively. UNSW's 2000 team [4] wanted to be able to drive the robots as if controlled by a three degree of freedom joystick. That is, the robot should be able to travel in any direction, regardless of which way it was facing. To achieve this, they devised a parameterised walk. The paw of each leg described a rectangle as it moved. A particular kind of walk or turn could be specified by giving the dimensions and orientations of the rectangles. The joint angles are automatically computed by inverse kinematics. This innovation was such a success that the team never had less than double digit scores in its games.

Each year, teams publish reports on their efforts, usually accompanied by source code. In subsequent years, almost all of the teams adopted the "UNSW walk". As a result of this sharing of ideas, the standard of the league has steadily risen, putting pressure on the leading teams to make further leaps forward. To improve the speed of the robot, the team re-examined the parameterised walk. Perhaps there were trajectories other than a rectangular path that would allow the robot to walk faster or more smoothly? Min Sub Kim and Will Uther employed an optimisation algorithm to try to discover a better walk. During the 2003 competition, a robot could be seen pacing back forth across the field, collecting data for the optimisation. At the end of the process the walk was 10% faster than any other teams and also smoother, making the camera shake less. The trajectories found by the optimisation are shown in Figure 5.

The process began with an initial rectangular trajectory. During the process of learning to walk faster, the corner points were shifted to obtained a more efficient walk. Previously, front and back legs had followed the same path, but through learning, it was discovered that having different trajectories improved the walk.

6 Game Play Behaviours

Developing game playing behaviours involves lengthy and painstaking work testing and tuning different skills for approaching the ball, kicking, handling special cases such as when the robot is against the field boundary, etc. These skills are

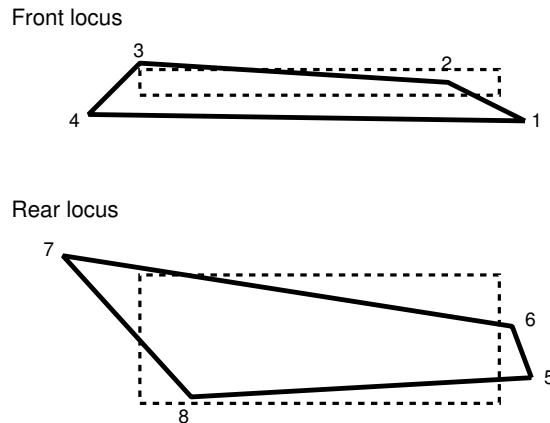


Fig. 5. The robots paw describes a quadrilateral. The robots walk can easily be changed by adjusting the dimensions and orientation of the rectangle

described in detail in reports available online². In this section we discuss the importance of developing strategies that are robust.

No matter how good the robot's skills are, it will inevitably fumble the ball. Opponent robots (and even friendly robots) are constantly attacking the ball so even if the ball handling skills do not fail, something unexpected will happen because of the interaction with other robots. One of the most important lessons that students entering the project learn is that they must program for when things go wrong. An example of this kind of "fault tolerant programming" can be seen in the cooperative behaviour of the rUNSWift forwards.

The three forwards are dynamically assigned the roles of: primary attacker, supporter and wide supporter.

- The primary attacker is the robot closest to the ball that is heading towards the opponent goal. Its job is to take the ball towards the goal and if it is not being challenged by opponents, it may shoot for goal.
- The supporter is the next closest robot. It follows the ball, but backs away from the primary attacker so as not to interfere with it.
- The wide supporter stays on the side of the field opposite to where the ball is. When the ball is nearer its own half, the wide supporter adopts a defensive role. It is prepared for the ball popping into the clear so that it can move the ball up field. When the ball is close to the opponent goal, it will move in, ready for a cross from its team mates.

As the ball moves around, these roles change dynamically. For example, if the primary attacker loses the ball behind it, the supporter will move in and become

² <http://www.cse.unsw.edu.au/~robocup>

the primary attacker, while the other two robots reassign themselves depending on their proximity to the ball.

The effect of these roles, as observed during the 2003 competition, was to create a wave of attacks. The first robot, as primary attacker would move the ball towards the opponent goal. If it lost the ball, a second robot would take over, moving the ball closer. If it also lost the ball, the third robot was nearby to finish off the goal. Thus, by providing constant backup to team mates, the rUNSWift team mounted an attack that almost always ended in a score.

A potential weakness of the rUNSWift strategy was that in being very aggressive, it could leave the goalie on its own to defend. However, no opponents could take advantage of this until the final against the University of Pennsylvania team. Their robots were programmed with a very powerful kick that could take the ball from an opponent and move the ball far down field. They also had another player positioned to attack when the ball came its way. The final settled into a pattern of rUNSWift attacks and UPenn counterattacks.

Fortunately, we had anticipated this as a possibility. The wide supporter was programmed so that if the ball got behind the rUNSWift attackers, it would execute a fast “run” back and circle behind the ball to block the opposition’s attack. Nevertheless, the game was very close one, ending 4-3 in rUNSWift’s favour.

7 Conclusion

As the average quality of the teams improves each year, we are seeing more close games that may be won or lost because of relatively small differences. We are also, seeing a shift from gaining an advantage from better low-level skills to trying to gain the upper hand through cooperation and strategy. As these trends continue, we learn more about how to create robust behaviour, how to anticipate potential weaknesses and how to prepare for the unexpected. All of these are excellent training for applications of robotics in a wide variety of tasks outside of competitions.

The RoboCup four-legged robot league has been a source of many interesting research problems for robotics and artificial intelligence. In succeeding years, the rules of the game are changed to introduce new challenges and gradually move the game closer to human soccer. The official aim of the RoboCup federation is, by 2050, to have a humanoid robot soccer team that can take on, and beat, the human world champions. The unofficial, and much important aim, is that as a side effect of the grand challenge, the competition will advance robotics and AI to benefit science and society in many practical ways.

Acknowledgements

The results described here are due to the efforts of successive UNSW RoboCup teams: John Dalgliesh and Mike Lawther (1999); Bernhard Hengst, Son Bao

Pham and Darren Ibbotson (2000); Spencer Chen, Martin Siu, Tom Vogelgesang and Tak Fai Yik (2001); James Brooks, Albert Chang, Min Sub Kim, Benjamin Leung, Canh Hao Nguyen, Andres Olave, Timothy Tam, Nick von Huben, David Wang and James Wong (2002); Ricky Chen, Eric Chung, Ross Edwards, Eileen Mack, Raymond Sheh, Nicodemus Sutanto, Terry Tam, Alex Tang and Nathan Wong (2003). Special thanks to team co-supervisors Bernhard Hengst, Will Uther (both with NICTA), Tatjana Zrimec and Brad Hall. The financial support of the UNSW School of Computer Science and Engineering, the Faculty and Engineering and National ICT Australia is gratefully acknowledged.

References

1. Cameron, D. and Barnes, N. (2003) Knowledge-Based Autonomous Dynamic Colour Calibration. In *Proceedings of the Robocup Symposium*, Padua: Springer.
2. Kalman, R.E. (1960) A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME—Journal of Basic Engineering*, **82**(D), pp 35-45.
3. Quinlan., J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
4. UNSW United Team Report.
<http://www.cse.unsw.edu.au/~robocup/2002site/2000PDF.zip>