



Robot soccer

Claude Sammut*

Robot soccer is a test bed for a variety of robotic and Artificial Intelligence (AI) methods. Its relevance to Cognitive Science is that it confronts the designer with a task that requires the integration of almost all aspects of AI to create an agent that is capable of working in a complex, dynamic environment inhabited by other agents, some of which are cooperative and others competitive. We describe the main elements that make up a robot soccer player and how these players associate to create effective teams. We pay special attention to the architecture of the players. © 2010 John Wiley & Sons, Ltd. *WIREs Cogn Sci*

The various disciplines that contribute to Cognitive Science have their own approaches to characterizing and understanding intelligent behavior. For example, we could consider the neurosciences as being reductionist in that they generally begin with models of neurons and try to build explanations of brain function by composing complex behaviors based on neuronal activity. In contrast, Artificial Intelligence (AI) takes an engineering approach by posing the question, ‘if we wanted to build an intelligent system from scratch, what would we have to do?’ However, AI researchers rarely try to build a complete intelligent system. They usually specialize, working in subdisciplines such as computer vision, natural language understanding, knowledge representation and reasoning, or machine learning. An intelligent system, in the sense that interests us in Cognitive Science, generally consists of many components, and the interaction between them is often as important as the individual components themselves. A robot is, necessarily, an integration of many hardware and AI systems. Thus, robotics contributes to Cognitive Science because building a robot forces us to consider the system as a whole. We must not only have functioning vision systems, problem solvers, etc., but they should also function *together*.

In recent years, competitions have become a popular means for motivating research in robotics. Examples include the DARPA Grand Challenge,¹ the International Planning Competition,² and the Reinforcement Learning Competition.³ As well as generating interest in research, these competitions

serve an important scientific function in providing a way of making objective comparisons between methods. In experimental sciences, there is the well-established principle of confirming results by repeating experiments under as nearly identical conditions as possible. Robotics competitions serve a similar purpose since robot builders must have their systems perform outside their own laboratories under conditions created by the competition organizers, who are independent of the teams.

One of the largest robotics competitions is the RoboCup international robot soccer competition. Soccer was chosen as one of the tasks because a successful robot soccer team must incorporate many different aspects of AI and robotics research. A robot soccer player should have the following capabilities:

- The agent must be able to perceive its environment, which is constantly changing. Therefore, it must be able to perform its sensing efficiently.
- The agent must incorporate the information from its sensors into a model of the world. Although the world is constrained to be the known dimensions of a soccer field, it is still very complex, containing many unknown elements because there are other robots on the field, often behaving unpredictably.
- The agent must be able to reason about its percepts and its world model so that it can make appropriate decisions in the current situation. Because the environment is rapidly changing, due to the actions of other robots, time is critical. Therefore decisions must be made quickly.
- Once a decision is made, it must be executed as an action. This involves the precise control of motors and other kinds of actuators.

*Correspondence to: claude@cse.unsw.edu.au

ARC Centre of Excellence for Autonomous Systems, School of Computer Science and Engineering, University of New South Wales, Sydney, New South Wales 2052, Australia

DOI: 10.1002/wcs.86

- Each agent is part of a team and must act so as to maximize the success of the team. Cooperation is improved when team members have the ability to communicate with each other. The communication may be about each robot's knowledge of the world or of its intentions.
- Many robot soccer players incorporate some form of learning that may be applied to any of the above capabilities. For example, the robot may learn to calibrate its sensors to novel conditions or tune its locomotion to improve its speed. Some teams try to adjust their play to suit their opponents' strategy. Learning may take place during a game or prior to it.

In the following sections, we will describe the robot soccer competition in more detail. We will discuss each of these above capabilities, beginning with the architecture of a robot soccer player. We will end with a discussion of the lessons learned from robot soccer competitions and some of the traps that such competitions can hold.

THE COMPETITION AND THE ROBOTS

There are several international robot competitions, including RoboCup.⁴ The idea of robot soccer as a driver for research in AI and robotics was originated by Mackworth⁵ and developed by Kitano et al.⁶ and Asada et al.⁷ RoboCup is divided into several leagues for different types of robots and tasks. In addition to soccer playing, the competition also includes leagues for urban search and rescue and for robotic helpers at home. Some leagues are simulation only. Some require teams to build their own hardware as well as software, whereas in the 'standard platform league,' all teams use the same hardware and compete in programming the robots differently. In this discussion, we are mainly interested in the way the robots are programmed and we focus primarily on the soccer competition, although we will also make some references to the rescue and 'at home' tasks.

To make our discussion concrete, we will describe the standard platform league in more detail. We choose this league because, unlike the other leagues, the robots are not designed specifically to play soccer. Over the lifetime of the league, two types of robots have been used. From 1998 to 2007, various models of Sony's Aibo quadruped robot were used.⁸ In 2008, the league switched to the Aldebaran Nao humanoid robot⁹ (Figure 1).



FIGURE 1 | (a) Sony's Aibo ERS-7 and (b) the Aldebaran Nao.

Both of these types of robots are legged, which presents interesting problems in controlling locomotion. They both have a color camera as the primary sensor, as well as accelerometers to give some information about the motion of the robot. They have sufficient onboard computing power that they can operate autonomously. That is, no high-powered



FIGURE 2 | The soccer field (from the Standard Platform League Nao Rule Book¹⁰).

base station is allowed to command the robots although they do possess wireless networking for communication with each other. All leagues, except the small size robot league, require the robots to be autonomous. In the small size league, off-board computing resources may be used and cameras may be mounted above the field to provide a global view.

The size of the playing field varies across the different leagues. The standard platform league currently plays on a field approximately 4 m × 6 m (Figure 2; Ref 10). When the Aibo robots were used, four robots made up a team, with one being the goalie. The 2010 Nao competition uses three robots. The goals are colored blue and yellow to allow them to be easily distinguished. Originally, colored markers were placed around the field to help the robots localize themselves. Over time, these markers have been eliminated, requiring the robots to orient themselves by the lines and goals alone. All the RoboCup leagues have made similar adjustments over time. As the robots and their programming have become more sophisticated, the rules of the game, including field size and number of players, have been made tougher to encourage progress.

As an environment, soccer is well structured in the sense that the layout of the field is known in advance and all the objects in the environment are also known. The complexity of the game arises from the fact that there are other robots constantly moving and executing their own behaviors. Thus, sensing and decision making must be very fast to keep up with the dynamic environment.

RoboCup also includes competitions for search and rescue robots and for robots intended for home use. Unlike soccer, the rescue environment is highly unstructured (Figure 3). The competition arena includes elements that simulate those that might be found in a collapsed building. Although localization in soccer is relatively easy since a map of the environment is known, localization in rescue is extremely difficult because an autonomous robot must simultaneously



FIGURE 3 | A RoboCup rescue arena.

perform mapping while also localizing itself.¹¹ To make matters worse, localization depends on being able to identify landmarks. In an unstructured environment, identifying objects that can serve as landmarks is a difficult problem in perception. Although some progress has been made toward autonomous rescue robots, at present, most are tele-operated. This is undesirable when wireless communication is disrupted, as happens often in emergencies.

The RoboCup@Home environment is semi-structured in the sense that the layout of the arena is not known in advance but the elements in it are objects that one would expect to find in most homes. This localization and mapping are somewhat less challenging than rescue. However, this competition introduces another element: physical interaction between robots and humans and between the robots and other objects. Thus, the robots should be able to recognize people and their motions and recognize and manipulate typical objects at home.

In the next section, we introduce a typical architecture for a soccer-playing robot but most mobile robots will have at least the modules we describe, though adapted or generalized to suit their environment and task.

ARCHITECTURE OF A ROBOT SOCCER PLAYER

Almost all soccer-playing robots will have some variant of the software architecture as shown in Figure 4.¹²

Perception usually includes vision to detect the ball, the goals, field markings, and other robots. Other sensors may also be used, including laser range finders and range imagers, infrared detectors, etc.

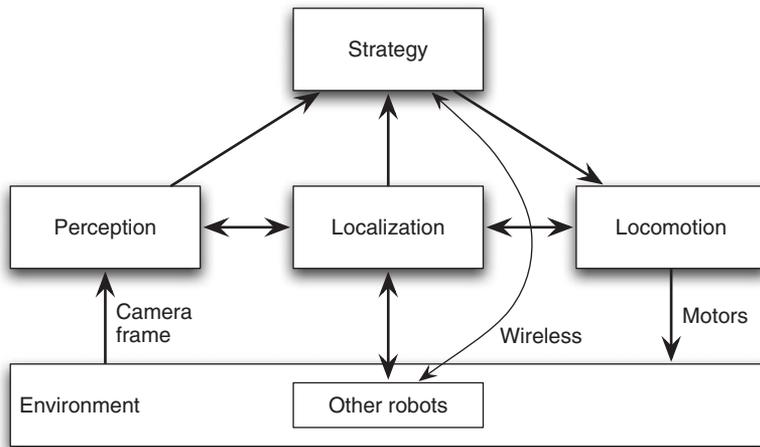


FIGURE 4 | Architecture of a soccer robot.

Perceptual information can either be used to trigger an immediate response or update a model of the world. This is required to localize the robot within the playing field and to localize the other mobile objects, namely, the ball and the other robots. Since team members are able to communicate with each other via a wireless network, they can exchange world models and therefore share information about the location of objects. A strategy module decides the next action of the robot, usually based on the current world model. This may include knowledge about other team members and opponents. Finally, the decision must be translated into motor commands that may move the robot, have it point its cameras in a particular direction, or execute a specific behavior, like a kick.

When building a robot, the designer is faced with many choices as to how to implement different modules and how to make the modules interact to create an effective complete system. A competition forces the designer to make hard choices about what may be theoretically interesting and what will pay off in a game. This might seem to be a negative aspect of competitions but often researchers become carried away with trying to find elegant but impractical solutions to a problem rather than choosing a design that is practical and works. Finding the balance between the short-term goals of the competition and the long-term goals of research is not always easy but the competition does help keep researchers grounded and at least makes them think about design trade-offs. Sometimes these trade-offs are reflected in real-life. For example, how much deliberation should the robot engage in before making a decision? This is Simon's problem of bounded rationality.¹³ Even though the processing power available to robots is constantly increasing, there is always some point at which more computation becomes either impractical or detrimental to its decision making. Sometimes, building a robot for a

competition requires making an arbitrary decision for the sake of practicality but sometimes thinking about the design forces us to face real problems in the architecture of intelligent systems. These design decisions will become apparent in the following sections.

Perception

In robot soccer, perception has well-defined aims: to find the ball; to identify goals and line markings that can be used for localization; and to identify opponents so they can be avoided. Because the ball and robots are in constant motion, these must be done as quickly as possible, otherwise the robot's decision making will be compromised by obsolete information. A video camera typically grabs 25 or 30 frames per second. So as not to drop frames, the vision system should be fast enough to process a frame in considerably less time than one frame interval, to allow for localization, decision making and motor control to all be executed within that interval. For this reason, robot soccer perception tends to be highly specialized, looking only for those features that are needed to find the ball, goals, lines, and opponents.

As we mentioned earlier, the soccer field originally had color-coded markers around the field to serve as landmarks to help the robots localize themselves. Furthermore, the ball and goals are also painted in distinct colors, so color recognition was and remains a key component of the vision system of most soccer-playing robots. Each pixel in the camera image is represented by three numbers that give the intensity and color of that pixel. There are different conventions for how the information is stored. For example, the three numbers may give the red, green, and blue components or the hue, saturation, and intensity. Video cameras typically use the YUV representation, where Y is the intensity and U and V are the blue and

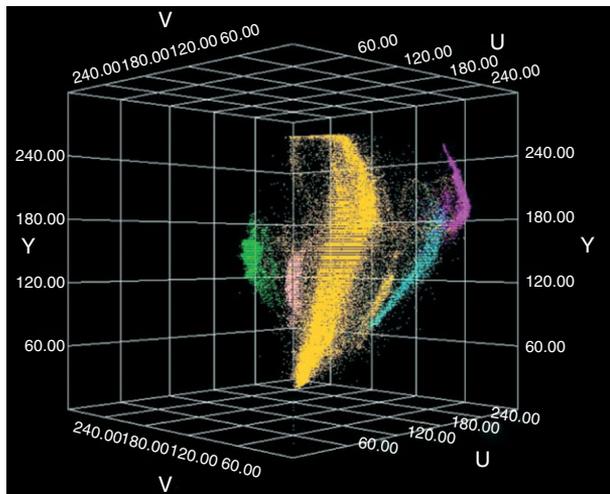


FIGURE 5 | 3D YUV Color space. The data represent pixels that have been manually labeled with the correct color. These can be used as training data for learning color regions.

red chrominance. Whichever representation is used, we have a three-dimensional space in which colors, e.g., the orange of the ball, fall within certain regions. The job of the color detector is to determine if the three numbers associated with a pixel fall within the region of a color (Figure 5).

The main problem in color classification is *color constancy*, that is, different lighting conditions can affect the way a camera detects colors. We are hardly aware of this problem since the human visual system automatically corrects for lighting, so an orange ball appears to be the same color whether seen under fluorescent lights or in sunlight. However, a camera's pixel values for the same object can vary widely under different lights. The most common way of getting around this problem is to use some form of machine learning algorithm to train the color classification system for a particular venue. Thus, when teams arrive at the competition, the first thing they usually do is collect images that are manually labeled by color and these are used as training data for learning.¹⁴ The task for the machine learning algorithm is to identify the regions of the color space that are associated with each color on the field, markers, ball, and robots. One of the difficulties is that these regions may overlap near the boundaries, so colors are not always distinct. In this case, knowledge of the surrounding pixel colors can help disambiguate the classification.

Although supervised learning works in this case, it is cumbersome in requiring manual labeling of training data and unrealistic since vision systems in animals can adapt without explicit training. To this end, there has been research on methods that allow robots to

adjust their color classification automatically.^{15–17} These typically used some form of clustering to find related groups of pixels but they also usually rely on some domain knowledge. For example, we know that when the robot looks down, it is most likely to see the green on the field. A region that is circular and lies above the green field is probably the orange ball, and so on. This knowledge helps the learner to attach the appropriate labels to the clusters that it finds.

As the competition has been shifting away from colored markers to relying mostly on field lines for localization, new methods have been developed to detect lines reliably. These methods usually begin with the observation that in robot soccer most of the objects of interest are close to the robot. Near the robot, the camera's resolution is high and as the objects become more distant, fewer pixels can be detected for an object. For example, if the ball is close then it almost fills the camera's field of view but if it is far away, perhaps only a few pixels of orange can be seen. A heuristic to save processing time is to scan the camera image from bottom to top, where the scan lines are initially well separated (e.g., scan every 10th column) but as the scan goes up the image, additional scan lines are introduced since objects further away will not cover as big an area as they do when they are closed up.¹⁸ This is illustrated in Figure 6, where the scan lines are slightly rotated to account for the orientation of the robot's head. As a pixel of a particular color is encountered during a scan, it is processed as either part of the ball, a line, a goal, or an obstacle, that is, another robot.

A criticism of the above approach to vision may be that it is very highly tuned to the narrow task of the soccer competition. Of course, this is true. But is it unrealistic? In their classic paper, *What the Frog's*



FIGURE 6 | Line detection is done by scanning bottom to top with increasing resolution.

Eye Tells the Frog's Brain, Lettvin et al.,¹⁹ found that the frog's retina does not simply detect spots of light but performs quite a lot of processing that amounts to the 'bug detection.' That is, the vision system is tuned to the survival needs of the specific organism in a particular environment. The human vision system is much more complex than a frog's, but does the complexity arise from greater generality or from an accumulation of many specialized functions? In robot soccer, the difficulty of getting a general-purpose vision system to perform in real-time, given limited computing resources, encourages us to follow the model of the vision system as a collection of specialized processes, each adapted for a particular task.

Localization

If the robot is to make a sensible decision, it must know where it is in relation to the ball, the goals, and other robots but this isn't easy. One of the most important lessons that robotics has for AI and Cognitive Science is that nothing is certain and knowledge of that uncertainty must be included in the robot's decision making. Many classical AI systems make the assumption that the uncertainty inherent in the world can be isolated to low-level processing of sensors and actuators and that high-level reasoning can assume a deterministic world. However, the problem of uncertainty is so pervasive that even high-level reasoning, whether they are based on logic or probabilities, must assume that all information is unreliable and that all models of the world are not only approximations but also contain errors. For a soccer robot, the problem is how to maintain the best possible map of the field, accounting for errors in sensing and control.

The information from sensors only represents a small part of the world and it contains errors, often large ones. For example, the distance to the ball can be estimated from the camera image by the area, in pixels, of the orange blob. If the ball is far away, the blob may only consist of few pixels and the pixels on the boundary of the ball will cover objects that are not part of the ball. So the number of pixels does not correspond exactly to the true area of the ball. Thus, distance estimation becomes less accurate when the object is farther away. Similar problems arise with all types of sensors and estimation methods. When the robot performs an action, it will not do exactly what was expected. For example, if the robot is supposed to turn 45° , it may turn 44.5° because the wheels or legs may have slipped a little on the floor. Over time, errors accumulate so that the robot becomes completely lost and relies on dead reckoning. For this reason, the robots must use their sensors to detect landmarks

like goals and field markings to localize themselves. Most robot systems use probabilistic models that not only maintain estimates of the locations of objects but also estimate errors. For example, Figure 7 shows the world models maintained by two robots on the same team. Note that the locations of the robots and other objects are represented by ellipses, not points. The larger the ellipse, the larger is the uncertainty of the position. Many different schemes have been used to update the probabilistic model. These include Kalman filters and Monte-Carlo localization. The reader is referred to Thrun et al.²⁰ for a comprehensive text on these and other methods in probabilistic robotics. In almost all cases, some form of Bayesian update is performed. When new information becomes available, the probability of an event is updated proportional to the prior probability of the event occurring and the probability of the event, given the new information.

We noted earlier that the robots can communicate with each other via a wireless network. Thus, it is possible for them to share their world models to improve their location estimates. Multiple world models can also be used to reduce the problem of the trade-off discussed in the previous paragraph. If it is important for an attacking robot to keep its focus on the ball, e.g., if it is close to a shooting position, it can use information from its teammates to keep itself localized. However, sometimes sharing world models can be tricky, as illustrated in Figure 7. Here two robots see the ball in two completely different locations and their estimates are reasonably confident. This situation can arise when there is a vision error, for example, one robot may pick up an orange blob off the field and recognize it as the ball.

Because of problems like this, there is no way of making a soccer robot completely reliable. In general, we must assume that mistakes will happen and have to build behaviors that will be tolerant to errors. Another significant lesson from building integrated systems is that when things go wrong, the errors can sometimes be caught by other modules in the same robot or by other robots on the team. So while it is important to make each module as robust as possible, it is also important to think about how other components of the team can be used as backups when a module fails. This will be illustrated further when we discuss programming behaviors.

Locomotion

The method of locomotion varies across the different leagues of RoboCup. Some leagues used wheeled robots while others were legged. The legged robots may be bipedal humanoid robots or, as in the previous

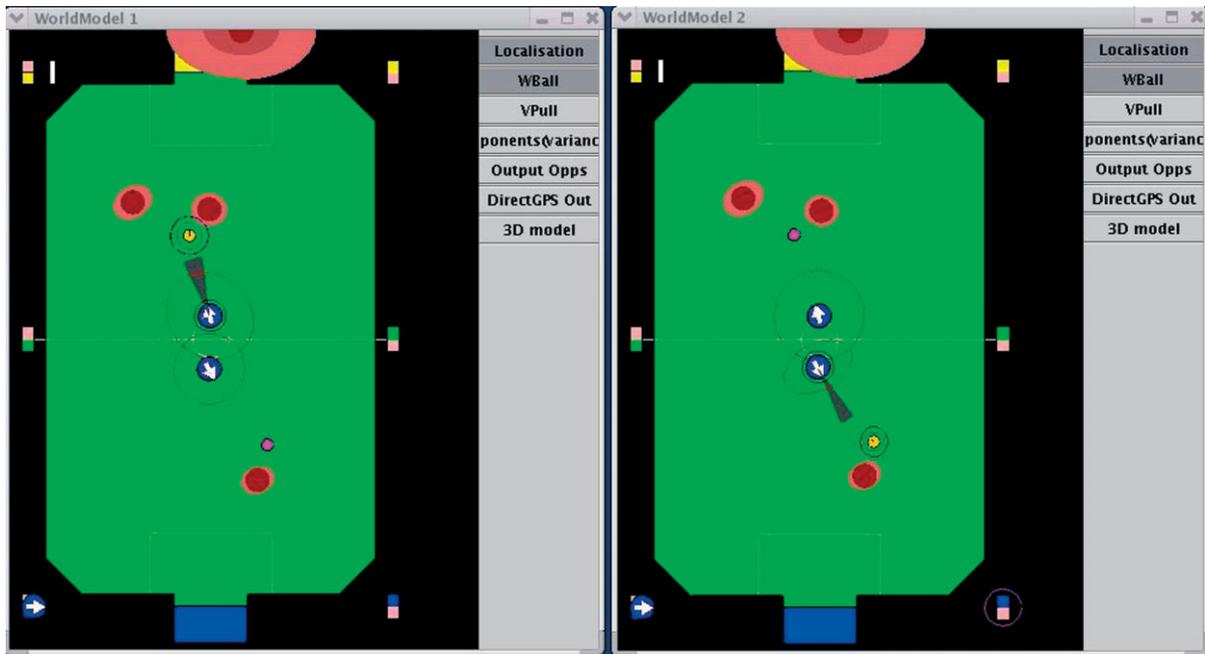


FIGURE 7 | The world model of two robots on the 'blue' team. Blue ellipses represent blue robots; red ellipses represent red robots; orange represents the robot's estimate of where the ball is; and pink is the other teammate's estimate of where the ball is. The gray arc is the robot's heading.

standard platform league competitions, quadrupedal. All forms of locomotion require a control strategy. In this section, we will discuss legged locomotion because the range of possible motor actions is very large and finding an effective set of actions is challenging.

It is well-known in computer science that choosing a good representation can make a problem tractable, whereas a poor representation can make it almost impossible to solve. Representation was key to making it possible to program the quadrupedal Aibo to be agile enough to play soccer. Hengst et al.²¹ developed a representation for the motion of the robot, which allowed them to easily program the robot to move in any direction and in any orientation. The motion of each 'paw' was constrained to follow a simple pattern. Initially, the trajectory described a rectangle. Many different gaits could be created by altering the dimensions of the rectangle, its position under the body of the robot, its orientation, and how fast each side of the rectangle was covered (Figure 8). The ease with which experiments could be performed allowed them to create motions that were quite unintuitive but were stable and could be executed quickly. In successive competitions, many teams adopted this approach, trying new shapes for the trajectories of the paws.

Since new gaits could be created or existing ones be tuned by the adjustment of a relatively small set of parameters, the next development was to automatically perform a search through the space of

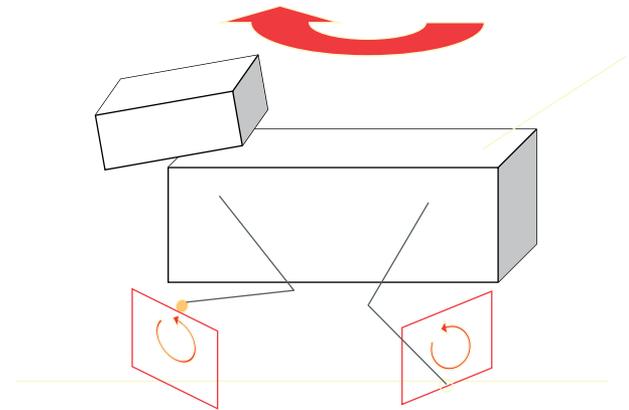


FIGURE 8 | The trajectory of 'paws' can be described by a simple figure whose parameters are easy to tune for different walks.

possible parameter values. Kim and Uther²² developed a method in which a robot could repeatedly walk backward and forward across the field, measuring its speed as it went. On each traversal, it used an optimization algorithm to adjust the parameters of the gait. If the speed improved, the robot received a positive reward; otherwise it got a negative reward. Thus, it was able to improve its performance by learning. This work was inspired by earlier research in evolutionary learning by Hornby et al.²³ and followed up by other researchers.²⁴ Similar techniques can be used to learn gaits for bipedal robots. However, this

task is more difficult because bipeds are inherently less stable than quadrupeds.

Like vision, the quality of locomotion in the standard platform league depended on learning. Often, problems in robotics are too complicated to work out entirely analytically, so learning is used to search for an adequate solution with a space constrained by the programmer's analysis and background knowledge.

Behaviors

In Figure 4, vision, localization, and locomotion formed the bottom layer of our architecture. Behaviors bring these together to create a soccer player. Since soccer play is generally very fast, robots usually do not have much time to deliberate over their actions. For this reason, most strategy modules are implemented as sets of reactive behaviors. This contrasts with the classic three-layer architecture.^{25,26} In Figure 4, the lowest layer directly accesses and controls the robot's hardware. The second layer is what Gat called the 'sequencer.' That is, it is responsible for selecting sequences of behaviors in response to the state of the world. Gat's uppermost layer, the 'deliberator,' is missing in this version of the architecture. The deliberator is where time-consuming, long-term planning is done. In soccer, there is no time for such long-term planning during the game. Generally, teams will run practice matches in which the behavior of the robots is observed. These behaviors are modified by the programmers who are responsible for planning, which is mostly done prior to a competition game. Later, we will describe research in which the robot does some planning and how behaviors can be learned.

There are many ways in which reactive behaviors can be implemented. Here we describe just one method¹² that uses a decision tree. A very simplified version of the tree is shown in Figure 9.

At the bottom of the tree, we have the 'atomic' behaviors. These are the basic skills that are fully preprogrammed, such as walking forward, turning, kicking, turning toward a goal, etc. Higher nodes in the tree represent composite behaviors. For example, if a robot sees the ball, it will first employ the 'go to ball' behavior that consists of walking forward, while tracking the ball. Once the ball has been reached, the behavior may switch to grabbing the ball, turning toward the goal and shooting. This behavior had to be executed within 3 s since that is the maximum time allowed for the robot to grab the ball.

At the higher level, the robot must decide upon its role. For example, the robot nearest the ball may be designated the attacker while the next closest team member becomes the supporter. Recall that in our discussion of localization, we emphasized the

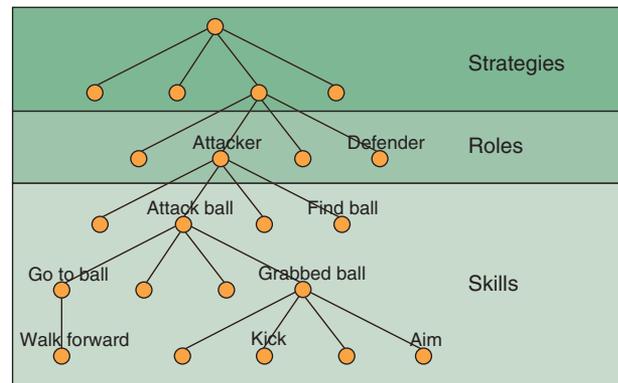


FIGURE 9 | Behaviors implemented as a decision tree.

importance of assuming mistakes would occur and behaviors have to be programmed to recover from these mistakes. If a robot has control of the ball, it is common to lose it, either because the robot's hardware is too imprecise to provide reliable control or the vision system loses the ball or an opponent knocks the ball away. The job of the supporter is to stay near the attacker in anticipation of the ball coming loose. In that case, the supporter will step in and attempt to gain control of the ball and become the new attacker.

While most behaviors in robot soccer are programmed, some work has been devoted to learning and planning behaviors. For example, Stone and Sutton²⁷ developed a reinforcement learning system for multiple robots to learn to keep the ball away from opponents. This was done for simulated robot soccer since it is much easier to perform repeated trials in a simulated game than with real robots. Since that works, the use of learning in simulated soccer has been extended and some effort has been devoted to moving the learning of behaviors to real robots.²⁸

Cognitive Robotics is aimed at giving robots and software agents reasoning capabilities through the use of logic-based languages that incorporate some planning.²⁹ These concepts have been extended to dynamic domains like robot soccer. Languages like READYLOG³⁰ allow a programmer to write reactive behaviors that are programs that respond directly to the current state of the world. However, the programmer may also specify a high-level goal in a program, which, when encountered, is handed over to a planner to automatically devise a sequence of actions to achieve the goal. This system has been used to control robots in the RoboCup midsize league. When this capability is available, the architecture of Figure 4 is extended to include the previously missing deliberative layer.

Although planning is slower than reacting, it gives the robot the ability to achieve goals when there

may be no preprogrammed behavior available. Thus, there is a trade-off between speed and generality. Humans make similar trade-offs. When we are novices at some task, we usually have to think much more about what we are doing compared to an expert, whose skill is so automatic that it is often subconscious. That is, as expertise grows, more skills are reactive. We only have to think about something when it is novel.

CONCLUSION

In this review of robot soccer, we have tried to extract some lessons about intelligent systems that go beyond playing football.

- To achieve real-time performance, sensing, modeling, and control have to be specialized for a given task. Generality will come from the accumulation of many special skills rather than from a single skill that is general.
- Nothing is certain. Robots must be programmed assuming that things will go wrong. Often robustness to errors can be achieved by one module within a robot compensating for errors in another module or, in multi-robot systems, other robots can try to cover the mistakes of teammates.
- Learning is often necessary to deal with the complexity of controlling robots in uncertain and dynamic environments. It is often too difficult for a programmer to anticipate all the circumstances that the robot will face, so the robot may use learning from example and trial-and-error learning to acquire the skills needed to handle the variety of situations it may get itself into.
- Resources are bounded, so there will always be trade-offs between reacting quickly and spending time to become more certain about the correct course of action.

Robot soccer has proved to be a very fruitful problem domain to further research in AI and Robotics. When approaching this task, we try to use it as a test bed for ideas that go beyond soccer. The competitive nature of the domain is a strong inducement to work hard and to make things work well and reliably. However, as it is a competition, everyone likes to win. All the teams have to guard against the temptation of making things so highly optimized for the competition that nothing about AI or Robotics can be learned. To try to avoid this, teams are encouraged to not only publish their methods but to release their code so that the best from all the teams can be incorporated into subsequent competitions. Each year, rules are revised to make problems tougher and more realistic. Leagues can even completely change their nature to push along new developments.

In 1968, John McCarthy and Donald Michie made a bet with then Scottish chess champion, David Levy, that within 10 years a computer program would be able to beat him.³¹ It took a bit longer than 10 years (nearly 30) but eventually such programs came into being. In the same spirit of a grand challenge, RoboCup aims, by the year 2050, to develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team. To achieve this, or come even close, the robots will have to be able to sense and act in completely unstructured environments. This will require major advances in perception, decision making, learning, and cooperative behaviors. Not forgetting that robots are integrated hardware and software systems, significant advances will also be needed in sensors, actuators, energy storage, and materials. These research results will be the real benefits of such competitions. Soccer and the other competition challenges are only the means to focus on the research. In the words of Alan Turing, in the conclusion of his 1950 paper on Computing Machine and Intelligence,³² ‘We can only see a short distance ahead, but we can see plenty there that needs to be done.’

REFERENCES

1. *DARPA Grand Challenge*. Available at: <http://www.darpa.mil/grandchallenge/>. (Accessed July 7, 2010).
2. *International Planning Competition*. Available at: <http://ipc.icaps-conference.org/>. (Accessed July 7, 2010).
3. *Reinforcement Learning Competition*. Available at: <http://www.rl-competition.org/>. (Accessed July 7, 2010).
4. *RoboCup*. Available at: <http://www.robocup.org/>. (Accessed July 7, 2010).
5. Mackworth AK. On seeing robots. In: Basu A, Li X, eds. *Computer Vision: Systems, Theory, and Applications*. Singapore: World Scientific Press; 1993, 1–13.
6. Kitano H, Tambe M, Stone P, Veloso M, et al., eds. *The RoboCup Synthetic Agent Challenge* 97.

- IJCAI97*. San Francisco, CA: Morgan Kaufmann; 1997.
7. Asada M, Kuniyoshi Y, Drogoul A, Asama H, Mataric M, et al. The RoboCup physical agent challenge: phase-I. *Appl Artif Intell* 1998, 2:251–263.
 8. Fujita M, Kitano H. Development of an autonomous quadruped robot for robot entertainment. *Auton Robots* 1998, 5:7–18.
 9. Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, et al. The NAO humanoid: a combination of performance and affordability. CoRR2008;abs/0807.3223.
 10. *Standard Platform League Nao Rule Book*. Available at: <http://www.tzi.de/spl/pub/Website/Downloads/Rules2009.pdf>. (Accessed July 7, 2010).
 11. Dissanayake G, Newman P, Clark S, Durrant-Whyte H, Csorba M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans Rob Auton* 2001, 17:229–241.
 12. Sammut C, Hengst B. The evolution of a robot soccer team. In: Jarvis RA, Zelinsky A, eds. *Robotics Research: The Tenth International Conference*. Berlin: Springer; 2003, 517–529.
 13. Simon H. A behavioral model of rational choice. *Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting*. New York: John Wiley & Sons; 1957.
 14. Chen S, Siu L, Vogelgesang T, Yik TF, Hengst B, et al. The UNSW RoboCup 2001 Sony Legged Robot League Team. In: Birk A, Coradeschi S, Tadokoro S, eds. *RoboCup 2001: Robot Soccer World Cup V*. Berlin: Springer; 2002, 39–48.
 15. Cameron D, Barnes N. Knowledge-based autonomous dynamic colour calibration. In: Polani D, Bonarini A, Browning B, Yoshida K, eds. *Proceedings of the RoboCup Symposium*. Padua; 2004, 226–237.
 16. Sridharan M, Stone P. Color learning on a mobile robot: towards full autonomy under changing illumination. *The International Joint Conference on Artificial Intelligence*, India; 2007, 2212–2217.
 17. Henderson N, King R, Chalup S. An automated color calibration system using multivariate Gaussian mixtures to segment HSI color space. In: Kim J, Mahony R, eds. *Australasian Conference on Robotics and Automation*; Canberra: ARAA; 2008.
 18. Rofer T, Jungel M. Fast and robust edge-based localization in the Sony four-legged robot league. In: Polani D, Bonarini A, Browning B, Yoshida K, eds. *RoboCup 2003: Robot Soccer World Cup VII*. Berlin: Springer; 2004, 262–273.
 19. Lettvin JY, Maturana HR, McCulloch WS, Pitts WH. What the frog's eye tells the frog's brain. *Proc IRE* 1959, 47:1940–1951.
 20. Thrun S, Burgard W, Fox D. *Probabilistic Robotics*. Cambridge, MA: MIT Press; 2005.
 21. Hengst B, Pham SB, Ibbotson D, Sammut C. *Omnidirectional Locomotion for Quadruped Robots. RoboCup 2001: Robot Soccer World Cup V*. Berlin: Springer; 2002, 368–373.
 22. Kim MS, Uther W, eds. Automatic gait optimisation for quadruped robots. *Australasian Conference on Robotics and Automation*; 2003.
 23. Hornby GS, Fujita M, Takamura S, Yamamoto T, Hanagata O, eds. Evolution of gaits with the Sony quadruped robot. *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, CA: Morgan Kaufmann; 1999.
 24. Kohl N, Stone P, eds. Policy gradient reinforcement learning for fast quadrupedal locomotion. *Proceedings of the IEEE International Conference on Robotics and Automation*; 2004, 2619–2624.
 25. Gat E, ed. Integrating planning and reaction in a heterogeneous asynchronous architecture for controlling mobile robots. *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI)*; San Jose, CA: AAAI Press; 1992.
 26. Gat E. Three-layer architectures. In: Kortenkamp D, Bonasso RP, Murphy R, eds. *Artificial Intelligence and Mobile Robots*. Menlo Park, CA: AAAI Press; 1998.
 27. Stone P, Sutton RS. Scaling reinforcement learning toward RoboCup soccer. *The Eighteenth International Conference on Machine Learning*, Williamstown, MA, 2001, 537–544.
 28. Gabel T, Lange S, Lauer M, Riedmiller M, eds. Bridging the gap: learning in the robocup simulation and midsize league. *Proceedings of the 7th Portuguese Conference on Automatic Control*; Lisbon, APCA; 2006.
 29. Levesque HJ, Reiter R, Lin F, Scherl RB. GOLOG: a logic programming language for dynamic domains. *J Logic Program* 1997, 31:59–84.
 30. Ferrein A, Lakemeyer G. Logic-based robot control in highly dynamic domains. *Rob Auton Syst* 2008, 56:980–991.
 31. Levy D. *Robots Unlimited: Life in a Virtual Age*. Natick, MA: A K Peters Ltd; 2005.
 32. Turing AM. Computing machinery and intelligence. *Mind* 1950, 59:433–460.

FURTHER READING

All of the RoboCup competitions are accompanied by a symposium in which the scientific contributions of the teams are presented. All of the proceedings from 1997 to 2008 have been published in Springer's Lectures Notes in Artificial Intelligence series.