

Keeping the Ball from CMUnited-99

David McAllester and Peter Stone

AT&T Labs — Research

180 Park Ave.

Florham Park, NJ 07932

dmac,pstone@research.att.com

<http://www.research.att.com/~dmac,~pstone>

Abstract. This paper presents preliminary results achieved during our current development of a team for simulated robotic soccer in the RoboCup soccer server [2]. We have constructed a team that plays a simplified “keepaway” game. Playing keepaway against the 1999 RoboCup champion CMUnited-99 team, our new program holds the ball for an average of 25 second with an average distance of 24 meters from the opponents end of the field. CMUnited-99 playing against itself holds the ball for an average of only 6 seconds. Here we describe the design of the keepaway team. The principal technique used is the vector sum of force-fields for governing player motion when they are not in possession of the ball.

1 Introduction

As a first step in developing a team for simulated robotic soccer we have constructed a team for playing a simplified “keepaway” game. We begin by justifying the use of the keepaway game as a simplified model of full soccer. There are two reasons for focusing on the keepaway game. First, it is much easier to measure performance in a keepaway game than in the full game. This means that a hill-climbing approach to program development is more feasible and high performance programs can be developed more rapidly. Second, a team that can hold the ball for extensive periods of time close to the opponent’s goal is likely to have more scoring opportunities than the opponents. So good performance in keepaway should translate into good performance on the full game.

As mentioned above, the first motivation for focusing on keepaway is that keepaway performance is more easily measured than goal scoring performance. One of the great frustrations in the development of a team for simulated robotic soccer is the difficulty of determining whether a given change improves performance. When the effect of a change is dramatic then one can often tell that the change is significant (usually bad) by simply watching the modified team play for a while. However if the effect on performance is only moderate, say a 20% change in scoring rate, then it can be very difficult to see the effect of the change either by watching the play or by counting goals in simulated play. To get statistical confidence intervals for performance measures, such as the rate at which goals are scored, we view the game as a Markov process and assume that the

mixing time of this process is comparable to the possession time — the average period of time between reversals in possession of the ball. We also assume that the expected time between goals is long compared to the possession time (and hence the mixing time). Under these assumptions, detecting a 20% change in scoring rate (up to 95% confidence) requires a run long enough to contain about 100 goals. In high level play, goals are infrequent and running long enough to get 100 goals can take days. For smaller improvements the situation is worse. Detecting a 5% improvement requires about 800 goals.

Using holding time and nearness to the goal as performance metrics results in a simplification of the game of soccer that we call “keepaway.” The keepaway game does not involve scoring but is similar enough, we hope, to the full game that good performance in keepaway can be converted into a high scoring team in the full game.

Our approach to keepaway is based on action generation using vector sums of “force fields”. For example there is a force field “repelling” players from the edge of the field and from each other when players get too close together. Previous research has explored action generation via vector sums. For example, the Samba control architecture [3] uses two behavior layers: the reactive layer which defines action maps from sensory input to actuator output; and the task layer which selects from among the action maps. In the robotic soccer application, a vector sum of action maps is used to determine the player’s actual motion. In this case, the vector sum is not of forces, but of low-level actions.

A previous force-field approach considering sums of attractive and repulsive forces among players and the ball is called strategic positioning using attraction and repulsion, or SPAR [6]. In contrast to our work reported here, these forces were only active over limited regions of the field, and boundaries, such as out-of-bounds and offsides, were treated as hard constraints. SPAR was implemented both in simulation and on real robots.

2 The Keepaway Game

For the experiments described in this paper, we use the RoboCup soccer server [2]. In the keepaway game used here, there is a distinguished offensive team and a distinguished defensive team. The game is played in a series of “trials.” At the beginning of a trial, the ball is placed next to the most open offensive player, i.e., the player farthest from the nearest defensive player. The trial lasts until a defensive player gains control of the ball (is within kicking range of the ball for half a second); the ball is passed in a way that violates the offsides rule; or the ball goes out of bounds. When one trial ends a new trial is started by moving the ball to the most open offensive player.

A first objective for the offensive team is to hold the ball as long as possible, i.e., to make each trial last as long as possible. A second objective is to move the ball as far downfield as possible. In the experiments described here, the players are assigned random positions at the start of the first trial. However, the runs are sufficiently long that performance is dominated by an “equilibrium” player

positioning achieved after the first few trials. The keepaway game has no rules other than those ending a trial as described above. When the defensive team (CMUnited-99) gains possession of the ball, it simply holds the ball in order to end the trial, rather than trying to pass and score. Otherwise, the CMUnited-99 team plays as it would in tournament play, which includes trying to take the ball away from the offensive team.

3 The Basic Keepaway Program

The players in our experiments are built using CMUnited-99 agent skills [4] as a basis. In particular, their skills include the following:

HoldBall(): Remain stationary while keeping possession of the ball in a position that is as far away from the opponents as possible.

PassBall(t): Kick the ball directly towards teammate t .

GoToBall(): Intercept a moving ball or move directly towards a stationary ball.

In each of the keepaway programs described here, each offensive player is always in one of three modes: “with-ball”, “going-to-ball”, or “supporting-ball”. The player is in with-ball mode if it is within kicking distance of the ball. If no offensive player is within kicking distance then the offensive player that can reach the ball the soonest (as determined by a CMUnited-99 primitive) is put in going-to-ball mode. Since each player is actually in a separate process, each player must decide separately what mode it is in. Because of sensing errors, occasionally two players will both think they can each reach the ball soonest and both go into going-to-ball mode. But this is rare and one can generally think of mode assignment as being centrally determined.

In all of the keepaway teams described here, the with-ball player either executes HoldBall() or PassBall(). When a pass is kicked, the receiver generally becomes the player which can reach the ball the soonest and automatically goes into going-to-ball mode. The player in going-to-ball mode executes GoToBall(): its behavior is identical to that of the CMUnited-99 players in this mode.

In the experiments presented in section 4 the with-ball player is controlled with a somewhat elaborate heuristic. However, based on our experience with controlling the with-ball player, we believe that this elaborate heuristic achieves roughly the same performance as always passing the ball immediately and selecting the receiver that maximizes the minimum angle between the pass and a defensive player no further from the ball than the intended receiver. In the experiments described here we hold the with-ball behavior fixed so that all of the performance differences we observe are a result of differing behaviors of the players in supporting-ball mode.

In all versions of the program described here, the movements of the supporting-ball players are controlled by force fields — each supporting-ball player moves in the direction of a sum of vector fields. Players are kept in bounds with a field that repels the players from the out of bounds lines. This bounds-repellent fields

becomes infinitely strong as a player approaches an out-of-bounds line. More specifically, the bounds-repellent field is defined as follows where B_x and B_y are the x and y coordinates of the field, x and y are the player's current x and y coordinates, and x_{\min} , x_{\max} , y_{\min} and y_{\max} define the in-bounds region.

$$B_x = 5/(x - x_{\min}) - 5/(x_{\max} - x)$$

$$B_y = 5/(y - y_{\min}) - 5/(y_{\max} - y)$$

In general we arrange that a given field will tend to dominate other fields if it has a magnitude large compared to 1. The constant 5 in the above equation causes the out-of-bounds field to become strong if a player is within five meters of the edge of the playing field. At ten meters or further from any edge the bounds-repellent field is weak.

There is also an offsides-repellent field that operates much like the bounds-repellent field to keep players onsides. This offsides-repellent field acts only on the x coordinate of the player and is defined as follows where O_x is the x coordinate of the force field and x_{off} is the x coordinate of the offsides line.

$$O_x \equiv \text{if}^*((x_{\text{off}} - x) \leq^1 5, -5, 0)$$

$$\text{if}^*(p, x, y) \equiv p * x + (1 - p) * y$$

$$x \leq^\delta y \equiv s((y - x)/\delta)$$

$$s(x) \equiv 1/(1 + e^{-x})$$

This fairly complex formula expresses a rather simple idea. If the player is significantly less than five meters from the offsides line then the force field pushes the player away with a force of five. If the player is significantly more than five meters from the offsides line then the force field is negligible. The field varies continuously from a negligible value to a value near 5 as the player crosses a line five meters from the offsides line.

In addition to the bounds-repellent and offsides-repellent force fields, there are force fields between players. For a given offensive player there is a strategic inter-player force due to teammate i , denoted S_i , and defined as follows where d_i is the distance (in meters) to teammate i and U_i is the unit vector pointing in the direction to teammate i (all from the perspective of a player calculating forces on itself due to its teammates).

$$S_i \equiv [(d_i =^{10} 20) - 2(d_i \leq^{10} 20)]U_i$$

$$(x =^\delta y) \equiv e^{-(x-y)^2/\delta^2}$$

This is a limited range force field — the strategic force is negligible when significantly further away than 20 meters. The basic idea is that players should be

within passing distance of each other but far enough apart so that a pass between them would move the ball a significant distance. Note that S_i is a continuous function of d_i and U_i .

Players near the ball are influenced by two tactical inter-player force fields. The first, T_i , is a purely repulsive force between the offensive players. The second tactical force field, the get-clear force, denoted C , pushes a potential receiver away from defenders. The force T_i is defined as follows where again d_i is the distance to teammate i and U_i is the unit vector in the direction of teammate i .

$$T_i \equiv \mathbf{if}^*(d_i \leq^3 8, -5, 0)U_i$$

Note that T_i is again a continuous function of d_i and U_i . The precise magnitude and direction for the get-clear force is somewhat complex and could probably be simplified without influencing the performance of the program. We do not present it here.

Intuitively, the strategic forces apply to players far from the ball and the tactical forces apply to players near the ball. The shift from “near” to “far” is done smoothly. The overall force on a supporting-ball player, denoted F is defined as follows where S is the sum over teammates i of S_i , T is the sum over teammates i of T_i , and d_b is the distance of the player from the ball.

$$F \equiv B + O + \mathbf{if}^*(d_b \leq^{10} 20, T + C, S)$$

A supporting-ball player always tries to run, as fast as possible, in the direction of the combined force F .

4 Variations on the Basic Program

Here we consider two additional strategic force fields for controlling the supporting-ball players. The toward-ball strategic force S^b is a force of unit magnitude directly toward the ball. This force pushes supporting-ball players that are far from the ball toward the ball. The forces repelling players from each other, S and T , keep them from bunching up around the ball. The down-field strategic force, S^d , is a force of unit magnitude directly toward the opponents end of the field. In all of the variations of the program considered here, the total field controlling a supporting-ball player has the following form where the strategic field S^* is one of the fields, S , $S + S^b$, $S + S^d$ or $S + S^b + S^d$.

$$F \equiv B + O + \mathbf{if}^*(d_b \leq^{10} 20, T + C, S^*)$$

The possession time and average x position of the ball for CMUnited and the four variations of the basic program are shown in table 1. The possession time is given as a “95% confidence interval” defined by the mean possession time over a sequence of trials plus or minus $2\sigma/\sqrt{n}$ where σ is the observed standard deviation of the possession time of a trial and n is the number of trials in the run.

Program	Possession Time	Mean Ball x Position
CMUnited	5.7-6.6	-19.5
S	16.9-18.7	-33.6
$S + S^b$	24.8-27.9	-35.9
$S + S^d$	22.2-25.2	25.7
$S + S^b + S^d$	23.7-26.8	26.6

Table 1. Offensive possession time and average x position of the ball when the offensive team is CMUnited and four variations of the basic program. The defensive team is CMUnited in all cases.

Of the four versions of the basic program, all except the basic program have essentially equivalent possession times (the differences in possession times are not statistically significant). It seems that the basic version of the program gets stuck in an unusually cramped position on the left end of the field. This cramped configuration can be “broken” in a variety of ways. e.g., by moving players nearer to the ball or moving players downfield. Once the cramped configuration is broken, a variety of behaviors have equivalent possession times. In particular, a much better ball position without significantly changing the possession time.

In summary, we believe that keepaway is a good development task because holding time, as opposed to goal rate, can be meaningfully measured, and because being able to hold the ball for longer periods should lead to better scoring performance. Our results indicate that good performance on the keepaway task can be achieved with a force field approach to action control.

References

1. H.-D. Burkhard, M. Hannebauer, and J. Wendler. AT Humboldt — development, practice and theory. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 357–372. Springer Verlag, Berlin, 1998.
2. I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.
3. J. Rieki and J. Roening. Playing soccer by modifying and combining primitive reactions. In H. Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, pages 74–87. Springer Verlag, Berlin, 1998.
4. P. Stone, P. Riley, and M. Veloso. The CMUnited-99 champion simulator team. In M. Veloso, E. Pagello, and H. Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, Berlin, 2000. Springer Verlag.
5. P. Stone, M. Veloso, and P. Riley. The CMUnited-98 champion simulator team. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*. Springer Verlag, Berlin, 1999.
6. M. Veloso, P. Stone, and M. Bowling. Anticipation as a key for collaboration in a team of agents: A case study in robotic soccer. In *Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II*, volume 3839, Boston, September 1999.