Player Positioning in the Four-Legged League

Henry Work
Prof. Eric Chown
Tucker Hermans, Jesse Butterfield, and Mark McGranaghan

Department of Computer Science
Bowdoin College
8650 College Station
Brunswick, ME, 04011, USA
{hwork,echown,thermans,mmcgrana}@bowdoin.edu
jbutterf@cs.brown.edu
http://robocup.bowdoin.edu

Abstract. As RoboCup continues its march towards the day when robots play soccer against people, the focus of researchers' efforts is slowly shifting from low-level systems (vision, motion, self-localization, etc) to high-level systems such as strategy and cooperation. In the Four-Legged League (recently renamed to the Standard Platform League), teams are still struggling with this transition. While the level of play has consistently risen each year, teams continue to remain focused on low-level tasks. Surprisingly few of the 24 Four-Legged teams that competed at RoboCup 2007 were able to self-position at the beginning of the game, despite penalties incurred for not doing so. Things considered to be standard in 'real' soccer – positioning, passing, overall strategies – are still, after 10 years of research, far from a given within the league, and are arguably in their infancy compared to other RoboCup leagues (Small-Sized, Mid-Sized, Simulation). Conversely, for the top teams, many of these low-level systems have been pushed far enough that there is little to be gained in soccer performance from further low-level system work. In this paper we present a robust and successful player positioning system for the Four-Legged League.

1 Introduction

In soccer, possession of the ball is paramount to success. The team controlling the ball has a higher likelihood of scoring goals, and keeping the other team from doing so. Following others [2], we firmly believe that to maximize possession (and thus win games) our team must be the fastest to the ball. For 2007, our positioning system was designed with the sole focus of getting to the ball quickly and ensuring possession.

We begin this paper with some background and then describe our positioning system, including its behavior tree, how our players position around the ball, playing with inactive robots, handling ball out-of-bounds cases, and our potential fields system. We gauge our results with possession statistics from RoboCup

2007. Finally, we conclude with an outlook on future high-level positioning and strategy within the league.

2 Background

The 2007 Four-Legged League [1] robot is the Sony Aibo ERS-7, which all teams must use without any type of hardware modification. The Aibo is a fairly cheap and robust robot, however, the Aibo's sensors are limited and present numerous challenges [6].

As for higher-level systems, few teams at the 2007 competitions were able to accurately position their robots on the field. While lots of research has been conducted on opponent recognition [3] [4], and localization [5], to our knowledge no team uses such technology in games. Passing challenges in 2006 and 2007 encouraged teams to invest more research into passing the ball, however, few deliberate passes were completed during games at RoboCup 2007. These limitations are more a factor of the Aibo's sensor limitations than anything else.

Overall, while the quality of play has consistently risen each year, the league is just beginning to move into high-level strategy and positioning.

3 Positioning

The intention of our behavioral system is to facilitate a coordinated system to allow for high-level team play. The framework we devised is a behavior tree that is strongly influenced by traditional soccer: we have strategies, formations, roles, and sub-roles. It is similar to, but was developed independently from, [10] [2].

First, the system defines overall team characteristics; a team may be more offensive than defensive. Second, we define formations dependent on the position of the ball on the field. For example, when the ball is near the opponent's goal, our robots are positioned to a) score, b) get second chances on missed shots, and c) have a defender bracing for a counter-attack. Third, the system assigns each player robot roles(e.g. defender or chaser) which defines its behavior and position, typically dependent on ball position. Finally, we define sub-roles for each role; if one player is a defender, there are different positioning rules depending on whether the ball is on the offensive or defensive side of the field.

3.1 Strategies

A strategy specifies a general manner of play; a team should be able to play more defensively sometimes, offensively others. While some teams have experimented with strategies [5] [11] [2], a more effective approach is to change strategies on-the-fly, triggered by events during the game. If our team is down a goal with 2:00 minutes left in the game, our team should scramble after the ball in a last ditch effort to score. Further, our team could gather statistics as the game progresses and try to become more or less offensive in order to match the strengths and weaknesses of the opposing team.

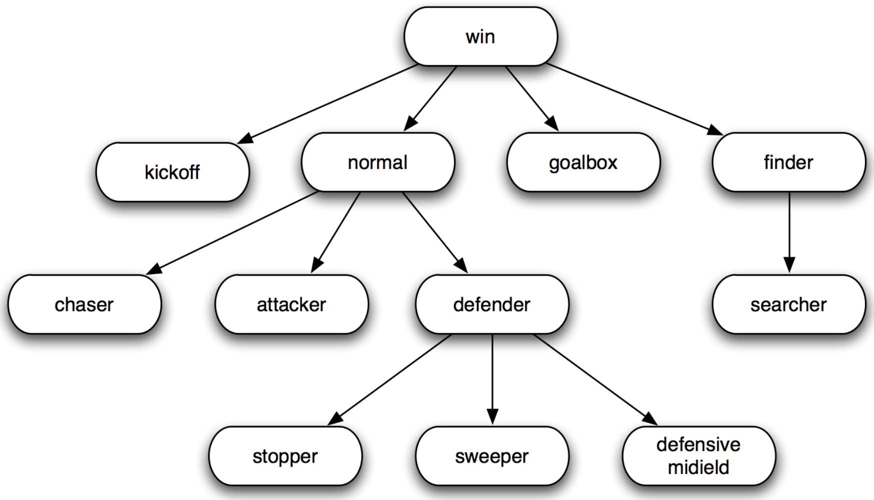

Fig. 1. An overview of our behavior tree. The top level is our single strategy for 2007. The second level are formations, third are roles. The final level is sub-roles, and this chart only shows a small selection of the Defender Role

We define a strategy as a set of formations. For example, we could build a defensive strategy which would require its component formations to include two defenders. The decision-making process for determining which strategies should be used is a complex problem and is a target of current research within our team. While the strategy framework was in place for the 2007 competitions, by competition time, we had only one strategy: *win*. We consider dynamic strategies to be one of our biggest research opportunities in 2008.

3.2 Formations

Unlike human soccer, in the Standard Platform League, each robot is able to switch roles whenever a situation calls for it. A robot may be the primary attacker at one moment, and then a defender the next. The details of our role switching system are described more thuroughly in [9] and are summarized later in this article.

Formations act as a layer above the role switching system to allow for different roles to be specified for selection by the agents as well as sometimes dictating specific roles when well known situations arise. An example of one of our formations is *Kickoff*.

The kickoff is the one situation in a soccer game which is relatively constant (the ball is in the center of the field, the players are not moving, etc). Thus we can ensure that robot A will always setup in position to start playing defense,

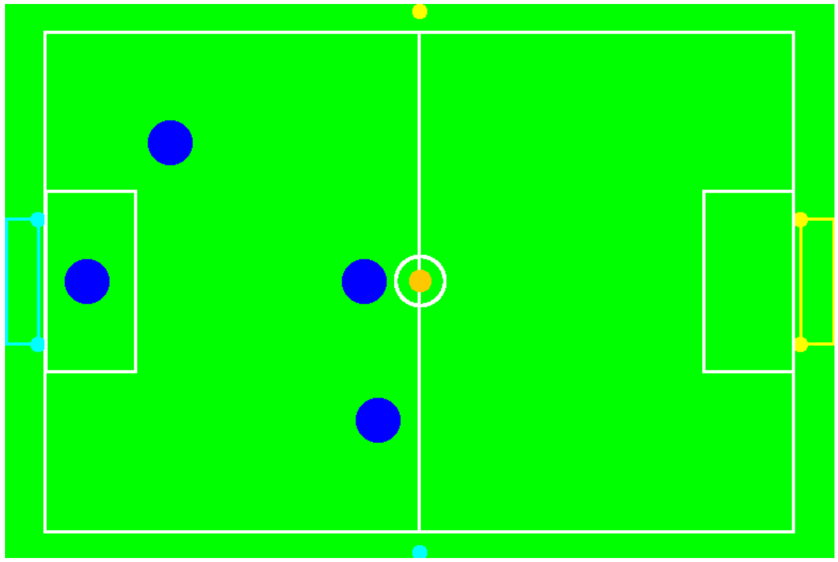

Fig. 2. Offensive Kickoff Position. Immediately after kickoff, the robot closest to the ball will become *Chaser*, the one on top will migrate into the center of the field to become a *Defender*, and the one on the bottom will assume the *Attacker* role

robot B will be in the center ready to become the chaser, and robot C will be on the wing ready to become the attacker.

This formation lasts for a specified time amount or until something unexpected occurs (i.e. the ball goes out of bounds, a player is penalized, etc). The style of play in the *Kickoff* formation does not look any different than our typical style of play; the formation simply reinforces the method of soccer we wish to be played against any poor data which could occur during kickoff.

3.3 Roles

Roles define the fundamental activity a robot should be performing from frame to frame and directly influences which behavioral states in the player's FSM are to be used. The five roles used at RoboCup 2007 were *Chaser*, *Attacker*, *Defender*, *Searcher*, and *Goalie*. These roles are fairly broad in scope and thus we define a number of sub-roles to further control the team dynamic.

3.4 Sub-Roles

Sub-roles are divisions of *Roles* which are dynamically assigned by the position of the ball on the field. Each sub-role carries an assignment of a specific point or line on the field for positioning.

Smooth transitions between sub-roles is very important. If the ball were to land on the midfield line, we may have two sub-roles for a *Defender*: one for when the ball is on the offensive side of the field and one for when it is on the defensive side of the field. Because of the particularly noisy ball estimates in our league, there may be oscillation between these two sub-roles. Such oscillations are intolerable in soccer where slight hesitations give opponents an advantage.

Our major strategy for coping with the noise and uncertainty inherent in role switching involves buffering the decision-making process. A robot must be 'sure' of a sub-role decision for about a third of a second before it decides to switch to another sub-role. This means that the robot must have estimates that place the ball's location inside a different sub-role's zone for a constant amount of time before it decides to switch. Further, we overlap the sub-role ball zones so that once a ball is in one zone it must travel leave it convincingly before being considered outside the zone. These features significantly reduce hesitation, and thus improve our odds of maintaining possession of the ball.

3.5 An Example: The Defender

In soccer, the defender's job is prevent the other team from scoring by stopping the ball when on its side of the field. Because we have only one strategy, *win*, the defender's high-level behavior does not change as the game matures (it remains equally defensive during the match). Most of our formations, *normal*, *kickoff*, *goalbox* require a defensive presence and so the *Defender* role is used heavily in games. The *Defender* role consists of three sub-roles: *Stopper*, *Sweeper*, and *Defensive Midfield*. The basic rules we use to define our defender are: a) it should never cross half-field nor enter its own goalbox, and b) it should always position itself between the ball and its own goal to prevent shots.

The main *Defender* sub-role is *Stopper*, which attempts to stay between the ball and the goal. We activate this sub-role when the ball is in the middle of the field. This position relies on ball localization estimates; it is calculated 100 centimeters from the ball on a line between the ball and the back of our own goal. The *Stopper* was integral to our success at competition.

When the ball is at the opponent's end of the field, the *Defender* switches to a *Defensive Midfield* sub-role, which clips its position at half-field. The robot still positions itself between the ball and its own goal, however it will not chase the ball into opponent territory.

Conversely, the *Sweeper* sub-role is active when the ball is very deep within our own territory. The *Sweeper* position is a static (x,y) coordinate just above its goalbox which makes sure the defender does not get in the way of a teammate chasing the ball. If the ball rolled into the goalbox, the *Goalbox* formation would take over and this sub-role would cease.

4 Positioning on the Ball

To go along with our 'possession is king' approach to soccer, one important feature of positioning is that all robots need to be facing the ball at all times.

While always having visual contact with the ball would be ideal, this is not possible. First, the frequency of occlusion and obstruction of the ball is quite high: other robots get in the way; referees step on the field to remove penalized robots; and the ball occasionally ‘teleports’ when it gets placed back on the field after going out-of-bounds. Further, to stay properly localized, positioning robots must continually scan for landmarks.

Our positioning robots split their time between tracking the ball and self-localizing. If they detect any significant velocity of the ball, the robots track the ball until it stops moving. Staying well localized improves ball localization for other robots who may have an obstructed view (particularly the goalie).

Positioning robots also try to keep the center of their bodies facing the ball, ready to become the *Chaser*. Because local data is more trustworthy, when a robot sees the ball, it uses relative estimates for aligning itself; when it does not see the ball, it uses its global estimate of the ball’s position.

One important note is that the *Chaser* robot always keeps its eye on the ball. Unfortunately, this negatively affects the *Chaser*’s self-localization (it must rely purely on odometry). But keeping track of the ball is paramount to possession; teams that choose to do quick head scans to re-localize when chasing the ball increase their chances of losing the ball. We have seen many instances of a robot looking away, only to lose sight of the ball when it looks back.

5 Inactive Teammates

It is crucial for a team to adjust its strategies according to how many robots are actually on the field. In the Four-Legged League there is no stoppage of play; teams are forced to play shorthanded whenever a penalty is called or a robot shuts off. The detection of the operability of teammates has lots of useful extensions for general multi-agent systems research. For our purposes, if our team does not recognize that we have inactive defender, we may give up a goal.

The news of a 30-second penalty must propagate across the entire team as quickly as possible. In our implementation, whenever a robot is penalized, the offending robot will immediately broadcast a packet to its teammates. A penalized robot continues to broadcast its status until it becomes unpenalized. Similarly, when a robot literally shuts off, the ‘dead’ robot broadcasts an emergency packet to its teammates on shutdown. When the robot is turned on and starts sending packets again, its teammates know that it is alive and well.

We architect our formations and roles to consider all cases of teammate inactivity. When we have one robot inactive, we have no attacker. Two robots inactive, and we don’t have a chaser that goes beyond half-field. There were many times during competition where this work saved us from getting scored upon (especially when our defender received a penalty).

6 Ball Out-of-bounds

A crucial part of our positioning system is taking into consideration the ball going out-of-bounds as per the 2007 Four-Legged League rules [8]. There is no pausing of the game when the ball leaves the field; referees place the ball back onto the field according to rules designed to penalize the team which knocked the ball out. We don't believe in trying to insure that the ball stays inbounds; we assume that the ball will be bumped out-of-bounds constantly. In our three matches during the round of eight, the ball went out of bounds on average more than 42 times per match, which translates to an average of more than twice per minute of play.

When the ball is closer to any edge of the field, our positioning anticipates the ball leaving the field. One teammate covers the ball's placement if our team kicks it out, another is positioned if the other team does. Our third robot, the *Chaser*, goes for the ball. If we have one inactive robot, we always cover the more defensive out-of-bounds placement (with two, we just have a *Chaser*).

Where out-of-bounds strategies particularly comes in handy is with goal-line situations. The worst out-of-bounds penalty in the game comes when a robot shoots or accidentally kicks the ball out-of-bounds along the opposing team's goalline. This penalty of ball being moved half the field length and happens quite often in our games (an average of 13 times per game during our three round of eight games in 2007). Our central strategy is to have our *Defender*, who never crosses midfield, always 'anticipate' the side of the field on which the ball is more likely go out. The *Defender* keeps track of side-to-side position of the ball down field and tries to stay parallel with its motion. As the ball moves from side-to-side on the field, so does *Defender*. Ideally, if the ball goes out-of-bounds, it amounts to a de facto pass to our defender. Top teams employ similar strategies, but none seem to execute them as effectively as we have: of the 39 times it occurred in our final three games of RoboCup 2007, we retrieved the ball 32 times. Picking up the ball in this situation is critical to keeping the ball in the opponent's territory.

This half-field recovery situation also presents an interesting problem due to the large displacement involved: ball capture. From the robot's perspective, this situation seems very odd. At one moment, the ball is near the opponent's goal-line and has some velocity. Then suddenly, the ball disappears (as it is being picked up by a referee), and reappears in a very different place with zero velocity. This ball capture problem is particularly acute with our localization system, an Extended Kalman Filter [12]. To the filter, this situation seems a lot like noise at first–steady readings that radically oscillate. And so our defenders often appeared sluggish in their responses to the ball (despite their good positioning). Instead of capitalizing on their positioning and scooping up the ball and sending it back into enemy territory, they behaved tentatively. The solution to this problem is to conditionally change the parameters of the filter. When a robot is in a defensive role and is near midfield, its ball model is run with different parameters such that it will respond to the ball far more quickly. This change, implemented right before our final game, brought an immediate boost in performance.

7 Potential Fields

To get our robots to specific, strategic positions, we need each one to avoid running into other teammates, running off the field, running into their own goalbox, or blocking teammates' shots on goal. The system we implemented for these purposes is flexible and avoids unmanageable increases in the number of cases as our decision-making becomes more sophisticated. Inspired by other work in the league [7] [3], we use a potential field positioning system.

7.1 Charges Overview

At the core of the potential field system are 'charges'. Charges either attract robots to or repel robots from certain points or line segments in the playing field. By placing several of these charges around the playing field, we can easily influence the movement and positioning of the robots.

A charge influences a robot through the charge's force function, which we can also think of as the heights of the potential field that the charge creates. For example, we say that a repulsive point charge generates a 'potential hill' with a peak at its point location and height that decays with increasing distance from the point. A robot using potential fields to navigate in the presence of such a charge will tend to move away from the charge point towards locations with lower potential charges, thus keeping the robot away from the point as desired.

To model situations in which the robot must consider many such objects when deciding how to move, we use multiple paramaterized charges. For example, we might place repulsive line charges around the sidelines, a repulsive charge at the location of each of the other robots on the field, and an attractive charge at the point to which we would like the robot to move. With such a potential field set up, we could aggregate the positioning influences of all of the charges to determine how to move.

7.2 Charges Implementation

Following [7], we use exponential functions to model the height of potential fields and the partial derivative of these height functions with respect to the x and y location of the robot to determine how it should move.

7.3 Equilibrium Detection

In the course of normal play, robots will often reach points of equilibrium in the potential field, for example in a potential 'trough' or 'cup'. It is important that as the robots approach and move past such positions that they recognize them as such and stop moving instead of oscillating repeatedly across the equilibrium. To implement equilibrium detection we find the value of the height function at some small distance away from the robot in each of the 4 cardinal directions as well as at its current location. The heights obtained can then be used to determine if the robot is at an effective equilibrium point. For example, if the height to the left and the height to the right are both greater than the current height, we say that the robot is at equilibrium with respect to its movement in x direction.

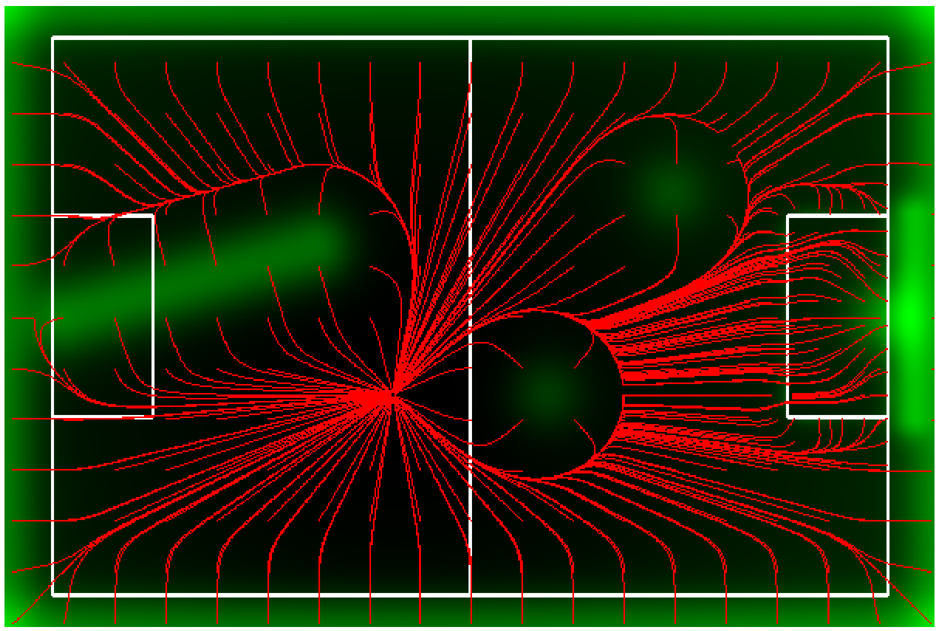

Fig. 3. *An example of the output of the potential fields visualizer. Lighter green indicate regions of higher potential, red lines indicate expected paths of the robots.*

7.4 Implementation

We start with a basic set of charges for the field: charges for the sidelines, and particularly for our own goalbox (see above). Second, every teammate has a charge in the system, and every packet received from a teammate updates the location of their charge to their self-localization estimates. When one player has possession of the ball, we expand its charge to a wider area so that all other teammates do not disrupt that player's advancement of the ball. Further, we create a line-charge between that player's (x, y) estimate and the back of the opponent's goal. This keeps robots from getting in the way of their teammates' shooting chances.

The potential fields system sits at the bottommost rung of the positioning hierarchy. When the positioning system has chosen a formation, role, sub-role, and produces an (x, y) coordinate on the field to move to, the potential fields system takes this desired position, considers the other charges in the system, and then suggests the best direction of movement. Potential fields were an integral part of our overall positioning efforts.

8 Results

RoboCup provides obvious metrics to measure the total quality of the team: wins, losses, goals scored, and goals against. However, to effectively measure possession, we have chosen three statistics: the time the ball spent in either half of the field, out-of-bounds situation handling, and number of attempted grabs. In order to avoid skewing our results we have limited our analysis to games in the final three rounds of the tournament where only the top competitors remained. The results presented here were gleaned by human analysis of video taken during the matches. As a side note, we feel it would be beneficial for the RoboCup community to begin to develop performance metrics other than goals scored.

How much time the ball spent on either half of the field is a good indicator of general possession. Each match is exactly twenty minutes. The results are summarized in Table 1. As you can see, the ball spent more time on the opponent's side of the field in each of our games, particularly against Team B (when our win margin, not shown, was the highest).

Another important metric is our ability to regain possession after a ball gets kicked out-of-bounds. The results are summarized in Table 2. Overall we got to the ball first more than twice as often as our opponents when the ball was replaced after going out-of-bounds. When the out-of-bounds occurred over an endline this shot up to a ratio better than four to one.

In our league robots generally attempt to trap the ball against their chests before getting ready to kick. It stands to reason that if our goal is to get to the ball more quickly then a good measure of success is the number of times such 'traps' are attempted. We have analyzed how often each team attempted to trap (and how often the traps were successful). Since robot speeds are relatively equal, this is another case where positioning should be the deciding factor. Once again we will limit our results to matches played from the quarterfinals on. The results are summarized in Table 3. The results show that our team attempted traps more than twice as often as our opponents. We also trapped successfully at a similar ratio to our opponents. The successful trap ratio reflects the fact that many of the basic soccer skills in RoboCup are close to being optimized for the top teams.

Table 1. Time in minutes in which the ball was in each of the two halves of the field. The first column refers to the opponent's defensive side, the second column to our defensive side.

Opponent	Opponent's Side	Our Side
Team A	11:37	8:23
Team B	14:35	5:25
Team C	12:13	7:47
Totals	38:25	21:35

Table 2. Analysis of how often our team got to the ball first on different out-of-bounds situations. The first number in each column is how many times our team got to the ball first, the second number is the total number of times the ball went out of bounds.

Opponent	Sideline	Endline	Total
Team A	18 / 30	11 / 15	29 / 45
Team B	17 / 27	7 / 8	24 / 35
Team C	18 / 31	14 / 16	32 / 47
Totals	53 / 88	32 / 39	85 / 127

Table 3. Number of traps attempted, and successes, by our team and our opponents by game. The first two columns represent the figures for our team, the next two columns for our opponents.

Opponent	Attempts	Success	Attempts	Success
Team A	93	67	49	32
Team B	123	71	39	19
Team C	100	67	57	45
Total	316	205	145	96

9 Conclusion

While hardly the most sophisticated system on paper, our positioning system has been proven to work under the adverse conditions found in competition. This reflects one of our primary research goals - to build systems that work as well in the real world as they do in the lab.

Over the next few years, we firmly believe that improvements to the overall quality of play will come from smarter and more situationally-aware positioning, faster information propagation, and from integrated positioning systems such as potential fields. We hope that our research encourages teams to continue to focus on high-level behaviors and coordination even as the league moves to a new platform.

References

1. The RoboCup Federation. http://www.robocup.org (2005).
2. Quinlan, M., Henderson, N., Nicklin, S., Fisher, R., Knorn, F., Chalup, S., King, R.: The 2006 NUbots Team Report. Technical Report, University of Newcastle (2006).
3. Quinlan, M., Nicklin, S., Hong, K., Henderson, N., Young, S., Moor, T., Fisher, R., Douangboupha, P., Chalup, S., Middleton, R., King, R.: The 2005 NUbots Team Report. Technical Report, University of Newcastle (2005).
4. Röfer T., Brose, J., Carls, E., Carstens, J., Gohring, D., Juengel, M., Laue, T., Oberlies, T., Oesau, S., Risler, M., Spranger, M., Werner, C., Zimmer, J.: The German National RoboCup Team. Technical Report, Darmstadt University, Bremen University, Humboldt University, Berlin (2006).
5. Hebbel, M., Nistico, W., Schwiegelshohn, U.: Microsoft Hellhounds Team Report 2006. Technical Report, Dortmund University (2006).
6. Nistico, W., Rofer, T.: Improving Percept Reliability in the Sony Four-Legged Robot League. In RoboCup 2006: Robot Soccer World Cup X. Springer Berlin (2006).
7. Knorn, F.: Algorithm Development and Testing for Four Legged League Robot Soccer Passing. Dissertation (2006)
8. The 2007 RoboCup Four-Legged League Rules. http://www.tzi.de/4legged/pub/Website/Downloads/Rules2007.pdf (2007).
9. Work, H., Chown, E., Hermans, T., Butterfield, J.: Robust Team-Play in Highly Uncertain Environments. In: Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (2008).
10. Morioka, N.: Road To RoboCup 2005: Behaviour Module Design and Implementation System Integration. Technical Report, The University of New South Wales (2005).
11. Röfer, T., Burkhard, H., von Stryk, O., Schwiegelshohn, U., Laue T., Weber M., Juengel, M., Gohring, D., Hoffmann, J., Altmeyer, B., Krause, T., Spranger, M., Brunn, M., Dassler, M., Kunz, M., Oberlies, T., Risler, M., Hebbel, M., Nistico, W., Czarnetzki, S., Kerkhof, T., Meyer, M., Rohde, C., Schmitz, B., Wachter, M., Wegner, T., Zarges, C.: German Team: RoboCup 2005. Technical Report, Technical Report, Dortmund University, Darmstadt University, Bremen University, Humboldt University, Berlin (2005).
12. Kalman, R.E.: A new approach to linear filtering and prediction problems. In: Transactions of the ASME - Journal of Basic Engineering, v. 82, pp. 35–45 (1960).