To appear in P. Stone, G. Kraetzschmar, T. Balch, *RoboCup 2000*, Springer-Verlag, 2001.

CS Freiburg: Doing the Right Thing in a Group

Thilo Weigel[1], Willi Auerbach[3], Markus Dietl[1], Burkhard Dümler[1], Jens-Steffen Gutmann[2], Kornel Marko[1], Klaus Müller[1], Bernhard Nebel[1], Boris Szerbakowski[3] and Maximilian Thiel[1]

[1]Institut für Informatik
Universität Freiburg
79110 Freiburg, Germany
last-name@informatik.uni-freiburg.de

[2]Corporate Technology
Siemens AG
81730 Munich, Germany
gutmann@ieee.org

[3]Optik Elektronik
SICK AG
79183 Waldkirch, Germany
first-name.last-name@sick.de

Abstract. The success of CS Freiburg at RoboCup 2000 can be attributed to an effective cooperation between players based on sophisticated soccer skills and a robust and accurate self-localization method. In this paper, we present our multi-agent coordination approach for both, action and perception, and our rich set of basic skills which allow to respond to a large range of situations in an appropriate way. Furthermore our action selection method based on an extension to behavior networks is described. Results including statistics from CS Freiburg final games at RoboCup 2000 are presented.

1 Introduction

After winning RoboCup in 1998 and coming in third in 1999, CS Freiburg won the competition in the F2000 league at RoboCup 2000 again. One of the reasons for this success is most probably the accurate and reliable self-localization method based on laser range finders [10]. However, while this was basically enough to win the competition in 1998, it was necessary to work on a number of different problem areas in order to stay competitive.

Our main research aim this year was to develop and test new techniques in the areas of *action selection* and *multi-robot cooperation*. In order to do so we also had to redesign the *action repertoire* and to rethink what kind of problems could be solved by a group of robots in which way. In addition to these issues we further enhanced our *perception technology*. However, most of the work in this area was already done last year [13] and will be described only briefly.

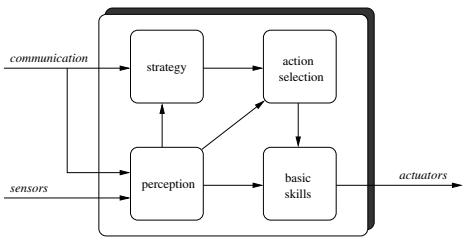


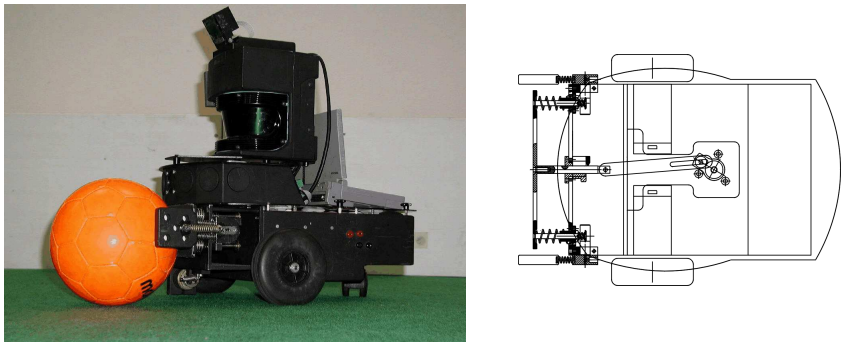
Fig. 1. Player architecture

Figure 1 depicts the software architecture of our players. The *perception* technology as described in Section 3 is the basis of our team. In the area of *cooperative sensor interpretation* we were able to come up with interesting and significant results. As described in Section 3.3, the group estimation of the ball position is done using a combination of *Markov localization* and *Kalman filtering*. For RoboCup 2000 we completely redesigned our *strategy* component. We employ a variation of the dynamic role assignment as used in the *ART* team [5] and a variation of the SPAR [14] technique for determining optimal positions on the field for each player. One of the objectives in the development this year was to enhance the *basic skills* of the soccer agents. In particular, we added a *dribbling* skill and a *ball-shooting* skill. For both of these new skills, it was necessary to modify the hardware and to account for these modifications by incorporating new software. In order to choose the right action in each situation, some *action selection mechanism* is necessary. We designed a new action selection mechanism based on Dorer's [6] extended behavior networks, which are used in Dorer's simulation team [7], the runner-up in the 1999 competition.

All in all, it turned out that the combination of techniques we employed allowed us to stay competitive, to demonstrate that our robots can play an attractive (robotic) soccer game, and to win the competition a second time after 1998.

2 Hardware

The basic hardware setup of CS Freiburg has remained mainly unchanged since the team first participated at RoboCup in 1998 [9]. Figure 2(a) shows a picture of one of our Pioneer 1 robots enhanced by a custom-made kicking device, the Cognachrome vision system for ball recognition, a SICK laser range finder for self-localization and object recognition, a Toshiba Libretto Notebook for local information processing and the WaveLan radio ethernet for communication.



(a) (b)

Fig. 2. A CS Freiburg player (a) and a sketch of the new kicking device (b).

In 1999 we already put effort in improving our perception technology [13]. We replaced the old PLS200 laser range finders with the new LMS200 models which provide depth information for a $180°$ field of view with an angular resolution of $0.5°$ and an accuracy of 1 cm. With a serial connection of 500 KBaud we now get up to 25 scans

per second. We also modified and extended the Cognachrome software to reduce the amount of vision data by discarding all informations not needed for our ball recognition module. Doing so we were able to raise the frame rate to 60 frames per second. The overall performance of our perception module was further improved by exploiting the feature of RTLinux to assign millisecond accurate time stamps to all sensor readings.

The major hardware improvement of this year is a new powerful kicker and ball steering mechanism. Figure 2(b) shows a sketch of the completely rebuild device. The kicker is based on a wind-screen wiper motor which compresses four springs attached to the ball steering plate when turning. The springs can be unlocked by a solenoid with the result of a very strong kick. The ball steering flippers can be turned to an upright position and back by two separately working DC motors. The movable flippers helped to avoid situations where robots would get stuck or accidentally hit the ball while turning towards it. In the future we intend to exploit the new feature for an elaborate ball interception behavior.

3 Perception

Back in 1998, the CS Freiburg team has been designed under the hypotheses that it is of great advantage if every robot would know its exact position and orientation on the soccer field. The general perception approach can roughly be divided into laser-based perception, vision-based perception and multi agent sensor integration.

3.1 Laser-Based Perception

As our research group already got a lot of experience with different self-localization methods using laser range finder data [8] it has been an obvious step to adapt a variant of them for our soccer robots. The method developed for the CS Freiburg team extracts line segments from a laser scan and matches them to an *a priori* line model of the soccer field. Only lines of a certain minimum extend are taken into account for discarding other players that are present in the field of the range finder. The scan-matched position is fused in a Kalman filter with the estimate from odometry.

Several experiments have been performed using this localization method and a comparison with other scan-matching methods shows that the line-based method is faster and more robust than point-based scan-matching methods while still retaining the precision of the other methods [10].

After matching a scan to the *a priori* model of the soccer field, players are extracted from the scan by removing all points belonging to the field walls and clustering the remaining ones. For each cluster the center of gravity is computed and considered as the center of a player. Inherent to this approach is the systematical error due to the different shapes of the robots [17]. At least for some players this error can be reduced, e.g. in our system we assume that the opponent goalkeeper is usually oriented parallel to the goal line, thus adding a constant offset to the center of gravity can reduce the position error for the opponent goalie[1].

[1] Of course this offset depends on the shape of the opponent's goalie and has to be adjusted before the game.

3.2 Vision-Based Perception

For recognizing and tracking the ball we make use of the Cognachrome vision system which delivers pixel-area information (so called *blobs*) of previously trained colors. We extended the supplied software to overcome problems with noisy readings, to enhance color training, and to customize the computed blob information to our needs.

A filter does plausibility tests to discard blobs whose shape, size or position make it very unlikely to correspond to the ball. From the remaining blobs the one closest to the previously selected blob is chosen and – by fitting a circle to it – various properties such as center, radius, or size in pixels are determined. Fitting a circle to the blob seemed to improve the overall position estimation of the ball as localization was still accurate when the ball was partially occluded by other robots.

From the computed blob center the global ball position is determined by using an off-line learned mapping table that contains distance and heading values for each pixel. This lookup table is autonomously learned by the robot before the game by positioning itself at various position on the field and taking measurements of the ball which stays at a fixed position. Interpolation is then used to compute the distance and heading pairs for pixels where no explicit information is available [15].

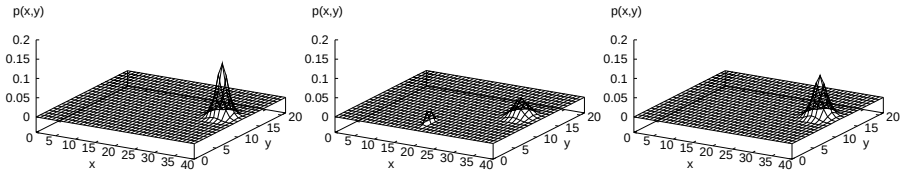
Despite of the applied filters we still observed infrequent wrong ball measurements due to reflections on shiny surfaces or badly trained colors, e.g. we had problems with the white field markings because when training the ball color, shiny reflections on the ball appeared to have a similar color. In order to detect such wrong ball measurements a global sensor integration module compares the observations of all players. This is described in the next section.

3.3 Multi Robot Sensor Integration

The perceptions of all players are sent to the *multi robot sensor integration module* for building and maintaining a global view of the world. Because the global model integrates more data than a single player is able to perceive, it should be more accurate and comprehensive than any local world model of a single player. After merging the player's perceptions into the global world model it is sent back to all players. Using the sensor informations of the whole team enables a player to take into account hidden objects when selecting an action. This proved to be especially advantageous for the global ball position, since the ball is seen almost all the time by at least one player. Furthermore knowing whether an observed object is an opponent or a teammate is, of course, very helpful for an effective cooperative team play.

In the first version of our *multi robot sensor integration module* we were using a simple averaging method for computing global positions of all objects on the field [9]. However, we encountered problems with very inaccurate or even completely wrong measurements, e.g. wrong ball observations as mentioned above, which corrupted a coherent global position. Therefore a more sophisticated method, namely Kalman filtering, for fusing observations and rejecting outliers is now employed. When tracking the ball, however, rejecting outliers by using a validation gate [3] has a big disadvantage: Once the system is tracking the ball based on a wrong ball observation, the filter needs several cycles until it accepts other observations again.

Therefore, we followed the proposal of Gutmann *et al.*[8] and combined Markov localization [4] which employs a multi-modal probability distribution for fusing ball observations with a uni-modal Kalman filter for precise ball localization. This way, the Kalman filter always tracks the ball using observations validated by one or more robots and only few cycles are needed in case the system was tracking a ball based on wrong observations.



(a) (b) (c)

Fig. 3. Probability grids after (a) correct measurements, (b) one additional wrong measurement and (c) another correct one.

Figure 3 illustrates our Markov localization variant on the basis of real measurements. At first the ball estimate is very reliable. An erroneous measurement makes this estimate less likely, but still more probable than the new estimate. Another correct measurement strengthens the assumptions that the previous one was wrong.

4 Strategy

Coordination among our players and their positioning on the field followed a rather inflexible scheme at the last tournaments. In order to achieve a more flexible and more effective coordinated team play we redesigned our strategy component completely this year. As described in the following sections we introduced dynamic *roles* for the players and dynamic target positions for each role.

4.1 Role Assignment

The CS Freiburg players organize themselves in roles, namely *active*, *support* and *strategic*. While the active player always tries to get and play the ball, the supporting player attempts to assist by positioning itself appropriately. The strategic player always occupies a good defensive position.

Each player constantly calculates its utility to pursue a certain role and communicates the result to its teammates. Based on its own and the received utilities a player decides which role it wants to take. This approach is similar to the one taken by the *ART* team [5],[2] however, the CS Freiburg players additionally communicate to their teammates which role they are currently pursuing and which role they desire to take. A role can only be taken from another player if the own utility for this role is the best of all players and the robot currently pursuing the role also wants to change its role. Following this strategy makes it less likely that two or more players are pursuing the same role at the same time than assigning roles based on utility values only.

Figure 4 shows a screenshot of the global view during a game. While the active player dribbles the ball around an opponent the supporting player moves to its target position and the strategic player observes the ball. Roles for the field players are assigned in the order *active, strategic, support*. Since the goalkeeper has a special hardware setup, it's role is fixed by assigning appropriate utility values to all roles.

[2] Note, however, that our approach was developed independently.

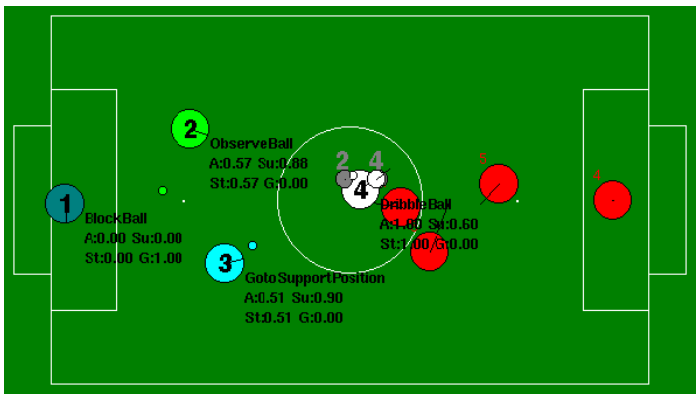
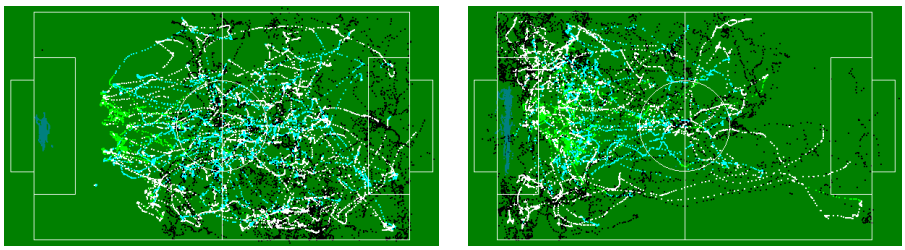


Fig. 4. Visualization of the results of the global sensor integration together with a players utilities for taking a certain role, its current role and its current action. The small white circle denotes the position of the ball as determined by the global sensor integration, whereas the small grey ones correspond to the ball as perceived by the individual players.

4.2 Positioning

The target positions of the players are determined similar to the SPAR method of the *CMU* team in the small size league [14]. From the current situation observed by the robots a potential field is constructed which includes repulsive forces arising from opponent players and attracting ones from desirable positions, e.g. positions from where the ball is visible. Positions are then selected based on the robot's current role, e.g. the position of the active player is set close to the ball, the supporting player is placed to the side and behind the active one, and the strategic player takes a defensive position which is about half way between the own goal and the ball but behind all opponent players.



(a) (b)

Fig. 5. Trace of the positions of the ball (black) and the CS Freiburgs' players during (a) the quarter final and (b) the final.

Figure 5 shows a trace of the positions of our players and the ball during the quarter final against *CMU* and the final against *Golem*. While in the quarter final our team spent most of the time in the opponents half our players were forced into our own half most of the time during the final game. The high correlation between the position of the ball and our players however demonstrates the effective positioning of our team.

5 Tactics: Basic Skills and Action Selection

In order to play an effective and successful game of robotic soccer a powerful set of basic skills is needed. This section describes the basic skills implemented for the goalkeeper and the field players in CS Freiburg, as well as the method of action selection developed for the field players.

5.1 Goalkeeper

As in most other robot soccer teams in the middle size league the hardware configuration of our goalkeeper differs from the one of the field players which is mainly because of the different tasks the goalkeeper and field players are designed for. Our goalie has a special hardware setup where the *head* of the robot, containing the sonar array, laser range finder and vision camera, is mounted $90°$ to one side allowing the robot to move quickly parallel to the goal line (see Figure 6). This kind of setup is quite popular in the middle size league and used by other teams, too, e.g. the *Agilo* team [2].

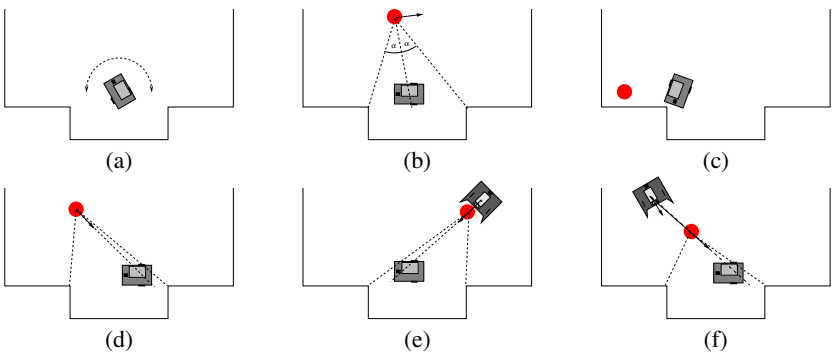


Fig. 6. Goalkeeper's tactics for saving CS Freiburg's goal: (a) ball searching, (b) minimizing area of attack, (c) turning to corner, (d) intercepting ball, (e) *intelligent* interception using opponent heading, and (f) *intelligent* interception using opponent to ball heading.

Our goalkeeper uses six skills sketched in Figure 6 for keeping the ball from rolling over our goal line. If the robot doesn't know where the ball is, it rotates left and right searching for it (*SearchBall*). If the ball does not directly roll towards our goal, the area of attack is minimized by moving to a position from where both sides to the robot allow approximately the same amount of angular space for the ball passing between goalie and goal posts (*BlockBall*). On the left and right side of the goal, the robot turns towards the corners giving less chances for an opponent to score a direct goal. The last three skills concern cases of direct goal danger where the ball or an opponent is moving directly towards our goal (*InterceptBall)*. Here the robot moves to an interception point based on the heading of the ball (Figure 6(d)), the heading of an opponent owning the ball (Figure 6(e)), or the heading of an opponent to the ball (Figure 6(f)). The last two and more sophisticated tactics are based on the assumption that the attacking robot will kick the ball in a straight way which is true for most robot teams participating in the middle size league but is not true for teams like *Golem* or *Alpha*++. Therefore these two tactics can be turned on or off before a game starts.

5.2 Basic Skills for Field Players

To get hold of the ball a player moves to a position behind the ball following a collision-free trajectory generated by a path planning system which constantly (re)plans paths based on the player's perception of the world (*GoToBall*). The system is based on potential fields and uses A^* search for finding its way out of local minima. If close to the ball a player approaches the ball in a reactive manner to get it precisely between the fingers while still avoiding obstacles (*GetBall*). Once in ball possession, a player turns and moves the ball carefully until facing in a direction which allows for an attack (*TurnBall*). If the player is right in front of the opponents goal it kicks the ball in a direction where no obstacles block the direct way to the goal (*ShootGoal*). Otherwise it first heads towards a clear area in the goal and turns sharply just before kicking in case the opponent goalkeeper moved in its way (*MoveShootFeint*). However if obstacles are in the way to the goal, the player tries to dribble around them (*DribbleBall*) unless there is not enough room. In this case the ball is kicked to a position close to the opponents goal by also considering rebound shots using the walls (*ShootToPos*). In the event of being too close to an opponent or to the field border the ball is propelled away by turning quickly in an appropriate direction (*TurnAwayBall*). If a player gets stuck close to an obstacle it tries to free itself by first moving away slowly and (if this doesn't help) then trying random moves (*FreeFromStall*). However a player doesn't give way if the ball is stuck between himself and an opponent to avoid being pushed with the ball towards his own goal (*WaitAndBlock*).

Against fast playing teams our robots were often outperformed in the race for the ball when approaching it carefully. We therefore developed two variants of a skill for situations in which speed is crucial. Both let the robot rush to the ball and hit it forwards while still avoiding obstacles. In offensive play *BumpShootOffense* is employed to hit the ball into the opponents goal when very close to it. In defensive play the use of *BumpShootDefense* can be switched on or off according to the strength of the opponent.

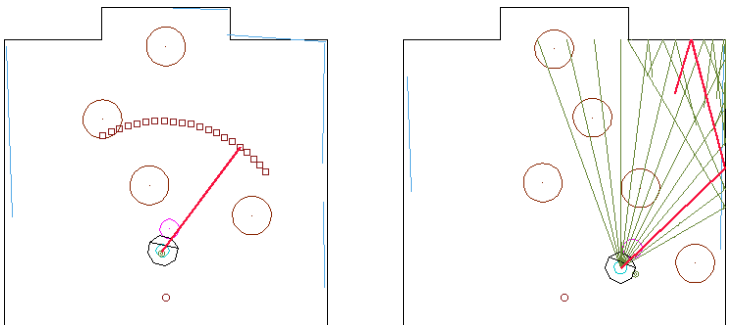
Players fulfilling strategic tasks position themselves following collision-free paths to dynamically determined positions (*GoToPos*). From these positions the players either search the ball if not visible (*SearchBall*) by rotating constantly or observe it by turning until facing it (*ObserveBall*). In offensive play a strategic player may also take a position from where it should be able to score a goal directly (*WaitForPass*). Once in such a position he signals to his teammates that he is waiting to get the ball passed. The decision is then up to the ball owning player whether to pass the ball (*PassBall*) or try to score a goal by itself. Especially against fast playing teams we rarely got the chance to try out ball passing between players. Equipped with the strong kicking device shooting at the opponents goal always appeared more promising than passing the ball to another teammate.

To comply with the "10-seconds rule" a player keeps track of the time he is spending in a penalty area. Whenever he spent more than the allowed time he leaves the area following a collision-free path generated by the same path planning system as employed in the GoToBall skill (*LeavePenaltyArea*).

At RoboCup 2000, our team seemed to be one of the few teams capable of effectively dribbling with the ball and the only one which exploited deliberately the possibility of rebound shots using the walls. Therefore these skills will be described in more detail.

Figure 7(a) shows a screenshot of a player's local view while dribbling. In every cycle, potential continuations of the current play are considered. Such continuations

are lines to points closer to the opponents goal within a certain angle range around the robot's heading. All the possible lines are evaluated and the direction of the best line sample is taken as the new desired heading of the robot. A line is evaluated by assigning it a value which is the higher the further it is away from objects, the less turning is necessary for the player and the closer its heading is to the opponents goal center. Determining the robots heading this way and adjusting the wheel velocities appropriately in every cycle lets the robot smoothly and safely dribble around obstacles without loosing the ball. The CS Freiburg team scored some beautiful goals in this year's tournament after a player had dribbled the ball over the field around opponents along an S-like trajectory. Of course, all this only works because the ball steering mechanism allows for a tight ball control.



(a) (b)

Fig. 7. A CS Freiburg player's view of the world while (a) dribbling and (b) ball-shooting. Circles denote other robots and the small circle in front of the player corresponds to the ball. Lines almost parallel to the field borders are perceived by the laser range finder. The other lines leading away from the player are evaluated by the skills.

Figure 7(b) shows a screenshot of a player during ball-shooting. For this skill the lines are reflected at the walls and are evaluated to find the best direction where to kick the ball. A lines value is the higher the further away from obstacles it is, the closer its endpoint is to the opponents goal and the less turning is required for the player to face in the same direction. Taking into account that the ball doesn't rebounce at the field borders in a perfect billiard-like manner we calibrated manually the correlation between the angles of reflection. Using the passing skill our players were able to play the ball effectively to favorable positions and even to score goals directly.

Figure 8 shows statistics of how long a skill was active for a certain role. Since we constantly improved and modified the action selection mechanism during the preliminary games the statistics are based on the three final games only. We found it quite surprising that the players were searching for the ball up to 10 % of the playing time. However analysis of the log-files showed that the high numbers can be either attributed to communication problems or to one of few situations where no player was seeing the ball. In the final games the ball was seen by the team in 95% of the playing time.

The statistics show that the goalkeeper was most of the time minimizing the area of attack but not under direct threat since the InterceptBall skill was only active in 11 % of the time. The supporting and strategic player spent most of their time observing the ball. This was intended since moving more often or more accurately to their target positions

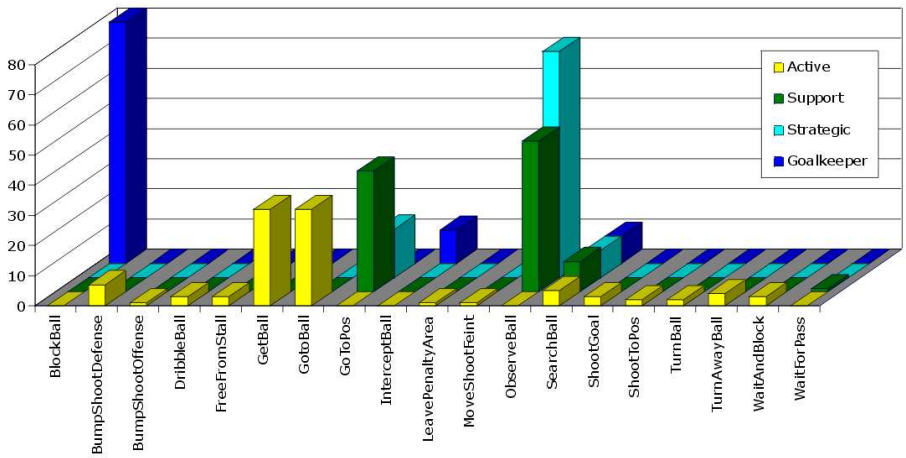


Fig. 8. Time in percent a skill was active in the final games broken down for the different roles.

results in a very nervous behavior as we could observe in one of our preliminary games. It turned out that for our rather clumsy robots staying at one spot was more effective than moving and turning constantly. Because the strategic player usually occupied a good defending position we allowed it to move even less.

At first view it seems surprising that the active player was most of the time (64 %) occupied with getting hold of the ball. However the fact that after kicking the ball the active player usually starts to follow the ball again explains the high numbers for GoTo and GetBall. Nevertheless it made use of all the skills available to it. This demonstrates that the active player in fact distinguished between a large number of different game situations.

5.3 Action Selection for Field Players

Our action selection module is based on extended behavior networks proposed by Dorer [6]. They are a revised and extended version of the behavior networks introduced by Maes [11] and can be viewed as a particular form of decision-theoretic planning. The main structural elements in extended behavior networks are *competence modules* which consist of a certain behavior to be executed, preconditions and positive or negative effects. *Goals* can explicitly be specified and can have a situation-dependent relevance, reflecting the agent's current motivations. The state of the environment, as it is perceived by the agent, is described via a number of continuous propositions $p_i \in [0..1]$. Competence modules are connected with goals if they are able to influence goal conditions and also with each other, if a competence module has an effect that is a precondition of another. Along the resulting network connections an *acitivation spreading mechanism* is defined, with goals being the source of activation. An action is selected by considering each competence module's executability and received activation.

Figure 9 shows a part of the extended behavior network that was used in this year's competition [12]. The ellipses represent the competence modules with their preconditions below and their effects on top of them. Currently our players have two goals: Shoot a *soccergoal* or *cooperate* with teammates. The relevance condition *role_active*

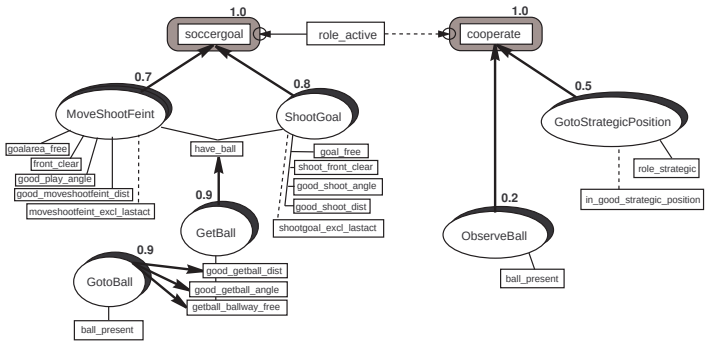


Fig. 9. A part of CS Freiburg's extended behavior network.

(player has role *active*) ensures that only one of these goals is relevant at a time depending on the player's current role. The strength of the effect connections (indicated by the numbers next to arrows) are set manually so far and reflect priorities among the competence modules.

6 Conclusion

The games played by CS Freiburg at RoboCup 2000 looked much more dynamic than the games in 1999 and 1998. So, the cooperation mechanism and the action selection (with the new action repertoire) seem to have improved our play considerably.

However, also the other teams have improved their play. In general, it seems to be the case that the performance of all F2000 teams have increased over the years and that there are more very good teams now – provided we measure the performance by goals/minute. As can be seen in Fig. 10, the average goal rate has almost doubled from 1997–2000,[3] and the goal rate of the two best teams became better over the years.

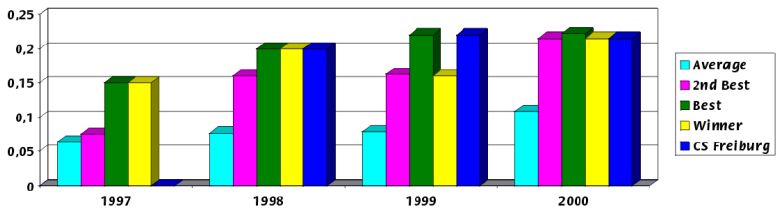


Fig. 10. Goal rates at RoboCup tournaments in goals per minute.

Furthermore, at RoboCup 2000 there was a large number of teams with a performance very close to the performance of CS Freiburg. In fact, while in previous years, CS Freiburg was the only team with an average goal rate of more than 0.2 goals/minute, this year there were 5 teams with a similar goal rate. Furthermore, it was obvious to everybody that the mechanical design of *CE Sharif* and *Golem* was definitely superior to

[3] Note, however, that the conclusion that the goal rate in 2000 was higher than the goal rate in 1997 is statistically significant on the 90% level only.

ours. That we nevertheless were able to stay ahead can at least partially attributed to the careful design of our actions, the action selection mechanism as well as the cooperation mechanisms. It should also be noted, however, that some game results were very close. In particular, the final and the third/fourth place games were decided by penalty kicks.

In the future, we intend to enhance our robot base to make it faster, to use learning techniques for tuning skill parameters, to refine the method of action selection, and to enhance team play.

References

1. M. Asada and H. Kitano, editors. *RoboCup-98: Robot Soccer World Cup II*. Lecture Notes in Artificial Intelligence. Springer-Verlag, 1999.
2. Thorsten Bandlow, Robert Hanek, Michael Klupsch, and Thorsten Schmitt. Agilo RoboCuppers: RoboCup Team Description. In Veloso et al. [16], pages 691–694.
3. Robert G. Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997.
4. Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *AAAI'96*, 1996.
5. Claudio Castelpietra, Luca Iocchi, Maurizio Piaggio, Alessandro Scalzo, and Antonio Sgorbissa. Communication and coordination among heterogeneous mid-size players: ART99. In *Proceedings of the 4th International Workshop on RoboCup*, pages 149–158, Melbourne, Australia, August 2000.
6. Klaus Dorer. Behavior networks for continuous domains using situation-dependent motivations. In *IJCAI'99*, pages 1233–1238, 1999.
7. Klaus Dorer. The magmaFreiburg Soccer Team. In Veloso et al. [16], pages 600–603.
8. Jens-Steffen Gutmann, Wolfram Burgard, Dieter Fox, and Kurt Konolige. An experimental comparison of localization methods. In *IROS'98*, Victoria, October 1998.
9. Jens-Steffen Gutmann, Wolfgang Hatzack, Immanuel Herrmann, Bernhard Nebel, Frank Rittinger, Augustinus Topor, Thilo Weigel, and Bruno Welsch. The CS Freiburg robotic soccer team: Reliable self-localization, multirobot sensor integration, and basic soccer skills. In Asada and Kitano [1], pages 93–108.
10. Jens-Steffen Gutmann, Thilo Weigel, and Bernhard Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *IROS'99*, October 1999.
11. Pattie Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 49–70. MIT Press, Cambridge, MA, 1990.
12. Klaus Müller. Roboterfußball: Multiagentensystem CS Freiburg (in German). Diplomarbeit, Albert-Ludwigs-Universität Freiburg, Institut für Informatik, 2000.
13. Bernhard Nebel, Jens-Steffen Gutmann, and Wolfgang Hatzack. The CS Freiburg '99 team. In Veloso et al. [16], pages 703–706.
14. P. Stone, M. Veloso, and P. Riley. CMUnited-98: Robocup-98 simulator world champion team. In Asada and Kitano [1].
15. Maximilian Thiel. Zuverlässige Ballerkennung und Positionsschätzung (in German). Diplomarbeit, Albert-Ludwigs-Universität Universität Freiburg, Institut für Informatik, 1999.
16. M. Veloso, E. Pagello, and H. Kitano, editors. *RoboCup-99: Robot Soccer World Cup III*. Lecture Notes in Artificial Intelligence. Springer-Verlag, 2000.
17. Thilo Weigel. Roboter-Fußball: Selbstlokalisierung, Weltmodellierung, Pfadplanung und verhaltensbasierte Kontrolle (in German). Diplomarbeit, Albert-Ludwigs-Universität Freiburg, Institut für Informatik, 1999.