



B-Human 2017 – Team Tactics and Robot Skills in the Standard Platform League

Thomas Röfer^{1,2}(✉), Tim Laue², Arne Hasselbring³, Jesse Richter-Klug²,
and Enno Röhrig²

¹ Deutsches Forschungszentrum für Künstliche Intelligenz,
Cyber-Physical Systems, Enrique-Schmidt-Str. 5, 28359 Bremen, Germany
thomas.roefer@dfki.de

² Universität Bremen Fachbereich 3 – Mathematik und Informatik,
Postfach 330 440, 28334 Bremen, Germany
{tlaue,jesse,roehrig}@uni-bremen.de

³ Hamburg University of Technology,
Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany
arne.hasselbring@tuhh.de

Abstract. The team *B-Human* won the main competition and, together with the team *HULKs*, the Mixed Team Competition in the RoboCup Standard Platform League 2017. In this paper, we argue that in the current state of the league, the development of sophisticated robot behaviors makes a difference between the top teams, while other abilities such as perception, modeling, and motion have been solved to a similar degree by the different top teams. We describe our general tactical approaches for the main competition as well as for the Mixed Team competition. In addition and as an example for a behavior-related robot skill, we present our approach to realtime path planning for humanoid robots with limited processing power.

1 Introduction

B-Human is a joint RoboCup team of the University of Bremen and the German Research Center for Artificial Intelligence (DFKI). The team was founded in 2006 as a team in the Humanoid League, but switched to participating in the Standard Platform League in 2009. Since then, we participated in eight RoboCup German Open competitions, the RoboCup European Open, and nine RoboCups and only lost four official games. As a result, we won all German Open and European Open competitions, the RoboCups 2009, 2010, 2011, 2013, and 2016. This year, we won both the main competition and, together with the team *HULKs* as the team *B-HULKs*, the newly introduced mixed team competition. We also won the technical challenge, i.e., the penalty shootout competition.

This paper is organized as follows: Sect. 2 motivates the focus on behavior development in the Standard Platform League, followed by an explanation of the tactics currently employed by B-Human in Sect. 3. The adjustments made for the Mixed Team Competition are described in Sect. 4. Section 5 gives a detailed

explanation of B-Human's path planner. Finally, Sect. 6 sums up this paper and names potential for future improvements.

2 The Importance of Robot Behaviors in the RoboCup Standard Platform League

The overall challenge of creating successful software for the RoboCup Standard Platform League can be seen as a set of major sub-challenges that have to be solved:

Vision. All major field elements need to be perceived reliably (i.e., without many false negatives and positives) over reasonable distance and with computational efficiency.

Modeling. To keep track of the own position, the position and velocity of the ball as well as of other robots on the field, modeling algorithms have to compute precise and stable state estimates.

Motion. A fast and robust walk, preferentially combined with a flexible and strong kick, is a necessity to perform competitively in the adversarial RoboCup scenario.

Behavior. To select the right actions, given the currently estimated state of the surrounding world, a flexible behavior is needed. Especially for playing in a team of five or more robots, many details, such as a stable role assignment, have to be addressed.

As in other RoboCup soccer leagues, the rules of the game are changed every year to make the overall problem harder and more similar to professional human football. Each year, these changes often only focus on one or two of the aforementioned areas. In recent years, major changes have been the introduction of white goals, the start of play by blowing a real whistle, the black and white ball (all affecting perception and modeling), arbitrary jersey designs (perception), and artificial grass (motion). All top teams have solved these challenges in a robust manner. There exist different solutions that each have certain advantages and disadvantages, but all can be considered to have an overall similar level of quality. For instance, for ball detection, there are pure model-based approaches such as the one used by B-Human [7] as well as many solutions that involve the training of a classifier such as the one by *UT Austin Villa* [3]. In general, when considering the implementations of the top teams, balls can be perceived over distances of several meters, robots can walk with decent velocity, and self-localization is precise and robust. Thus, further major improvements in these areas would not be any game changer.

One can notice that there have not been any major rule changes directly affecting the behavior. Furthermore, major properties of the overall setup – the size and design of the field as well as the number of robots – remained constant. There will probably some changes in 2018, e.g., the introduction of free kicks, but this has not finally be decided at the point of time when writing



Fig. 1. RoboCup 2017 final between B-Human (black jerseys, playing from left to right) and Nao-Team HTWK (blue jerseys, playing from right to left). The ball is in midfield and both teams distribute their robots over the field according to their respective tactical concepts. (Color figure online)

this paper. Thus, the current behaviors of all teams have evolved over multiple years. Furthermore, the behavior seems to be the one part that is missing in most code releases of the top teams, i.e., one can consider the actual implementations of low level behaviors such as ball handling and dribbling as well as the formulas and parameters for tactics such as roles and positioning as *secret*. In contrast, the formalism in which the behavior is specified is often known. Many teams, such as B-Human, use hierarchical finite state machines, e.g., using *CABSL* [5].

In summary, given similarly competitive solutions for most other tasks and a certain level of secrecy in behavior development, one could say that in the current Standard Platform League, the development of robot and team behaviors is a crucial aspect that makes a difference.

3 Current Tactics

When playing with five robots per team, the number of possible tactics and team formations is quite limited. When assigning the goalkeeper task to one specific robot and letting this robot stay within its own penalty area, as it is done by B-Human and almost all other teams, only four field players remain for specific tasks. It is common among the top teams to not assign fixed roles to field players but to perform a permanent task negotiation via wireless communication. When assigning roles, various information is taken into account by the B-Human robots, such as the information about a robot's maximum velocity, which is not very high compared to the size of the current field and which makes it necessary to maintain a reasonable coverage of the field throughout all game situations.

The current roles in normal games (that slightly differ from those used in the Mixed Team Competition, as described in Sect. 4) are: two defenders that dynamically adapt their positions depending on the current ball position, one striker that always approaches the ball, and one supporter that is mainly waiting in an offensive position to perform a rebound in case of a successful save of the opponent goalkeeper. A typical formation is depicted in Fig. 1.

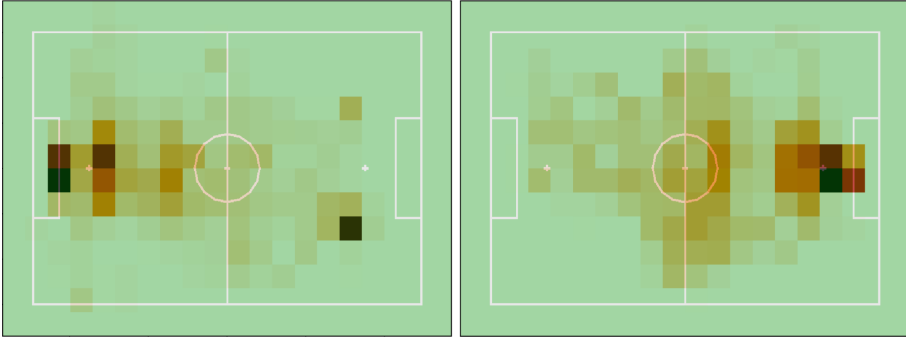


Fig. 2. Heatmaps of the whole teams of the two 2017 finalists: B-Human (left image, playing from left to right) and Nao-Team HTWK (right image, playing from right to left). The darker a square, the more time it has been occupied by a robot of the respective team. The figure has been created by using the *Team Communication Monitor* log files that are publicly available at the Standard Platform League’s website [6].

This approach puts a focus on a strong defense and realizes offensive play by long distance shots – made possible by the kick implementation described in [4] – towards the opponent goal. Overall, this tactic appears to work well. At RoboCup 2017, B-Human was, on the one hand, the Champions Cup team that received the least goals (one goal in 116 min of play) and was, on the other hand, among the teams that scored the most goals (34 goals, only the *Nao Devils* scored more – 36 goals). Detailed results can be found on the league’s website [6]. The heatmap in Fig. 2 shows a summary of the placement of the robots over the whole final game.

The 2017 final opponent, *Nao-Team HTWK*, which is one of the most successful teams in this league, has a different tactical approach: There is no offensive supporter waiting for long balls as the robots do not shoot in general. Instead, ball possession is gained in midfield, where three robots are placed, and the ball is dribbled towards the opponent goal. This is nicely reflected by the heatmap in Fig. 2. Although being quite different, this approach also led to a high number of scored goals (28) and only few received goals (5).

By comparing the different heatmaps of both teams, there is one noticeable issue: the corners of the opponent half are almost not occupied. This is in stark contrast to human soccer, where it is common to dribble towards the ground line and to pass cross the opponent’s goal. It appears to be worth investigating to implement such behaviors in the future.

4 Mixed Team Competition

This year, the *Mixed Team Competition* was held for the first time, replacing the *Drop-In Competition*, which had been the Standard Platform League’s testbed for multi-agent cooperation from 2014 until 2016 and which B-Human won twice.

For the first year of this new competition, each mixed team consisted of a pair of normal teams. This pairing remained fixed over the whole competition and had to be defined in advance, i.e., together with the teams’ application for the main competition.

As we are convinced of the necessity of performing many full system tests under realistic conditions to achieve a high performance in a competition, we were looking for a partner team that we could meet several times for testing and coordination. Given these prerequisites, the HULKs have been the perfect partner. Their university is only about one hour away from Bremen and they were also strongly committed to the Mixed Team Competition.

As in this competition both members of a mixed team use their own soccer competition code base, only two major issues have to be resolved for playing together as a team: agreeing on a strategy for playing with six robots and specifying a standard communication protocol that extends the rather basic elements of the SPL standard message. To provide a real multiple-teams robot cooperation, the standard communication protocol was the only code shared between the two teams. The protocol is documented in the B-Human team report [7].

4.1 Strategy Agreements

Since the normal B-Human five-robot strategy has shown very good results in the past years (in comparison to the state-of-the-art team play in the Standard Platform League), the combined team agreed to use basically the same, but split the former support role (cf. “6.2.1 Roles and Tactic” in [7]).

However, as role names, we now used chess pieces:

King. The goalkeeper robot.

Queen. The ball-playing robot.

Roops. Two robots that are positioned defensively, moving mainly horizontally in front of the goal.

Knight. A robot that is positioned offensively to help the Queen, jumping in when she loses the ball.

Bishop. A robot that is positioned far inside the opponent half to be able to receive a pass.

The general well-tried procedure is that every robot is calculating a role selection suggestion for itself and each connected teammate, but uses the role selection of the captain robot. The captain robot is determined by the lowest (not penalized) connected teammate number. The reason behind this procedure is to always have a suggestion available but inhibit the chance that the robots play with different role selections at the same time. Furthermore, we decided that all B-Human teammates have a smaller number than their HULKs colleagues to additionally support the selection robustness (because we reduce the possibility of switching between slightly different algorithms). True to the wish of maximizing the inter-team soccer play we force our robots to spread themselves over the field equally (by role selection). In fact, the full B-HULKs lineup had a

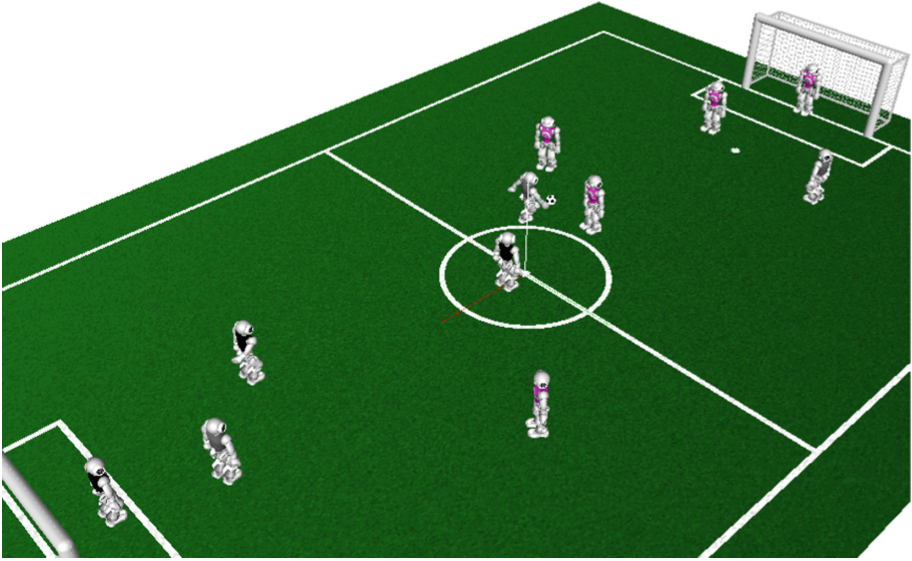


Fig. 3. An example scene of a possible B-HULKS match. B-HULKS robots wearing black (B-Human) or gray (HULKS) jerseys. From left-to-right they are currently performing the roles King, Rook, Rook, Knight, Queen, and Bishop.

B-Human robot with the role King, Rook, and Queen-or-else-Knight, completed with HULKS robots as Rook, Bishop, and Queen-or-else-Knight. Figure 3 shows an example of a standard positioning.

4.2 Collaborative Field Coverage

The black-and-white ball cannot be seen across the whole field and the field of view of the robot is not very wide. Therefore, if the ball is lost, it is searched for in a coordinated team effort. B-Human robots use a common model for the field coverage (cf. Fig. 4), which is gradually synchronized through team communication. Based on this, each robot will look at positions nearby, which have not been observed for a long time. To allow each individual robot to still create the model when playing together with HULKS robots, all teammates broadcast their estimated pose on the field, head posture, and all obstacles detected. Using this information together with an occlusion model (cf. Fig. 5), our robots add the field of view of the non-B-Human robots to their own coverage model, allowing them to search for the ball as if being in a B-Human-only team.

4.3 Additional Adjustments

For the positioning in the Ready state, i.e., preparing for a kickoff, the general method used by B-Human (cf. “6.2.6 Kickoff” in [7]) has been implemented by

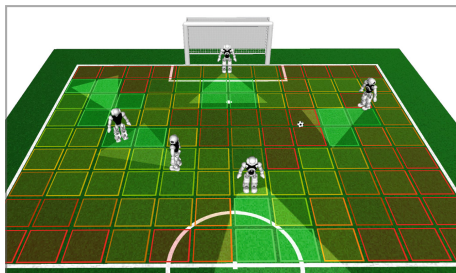


Fig. 4. An illustration of the common coverage model. The field is discretized into 12×18 areas. Green areas were observed recently, while red areas have not been observed for a longer time. (Color figure online)

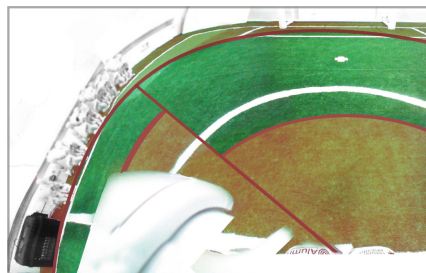


Fig. 5. Occlusion of the field of view with an average head posture. Depending on the head posture, the occlusion caused by the own body differs by several square meters.

the HULKs. However, due to the special role assignment in Mixed Team games, the procedure needed to be slightly adjusted. Otherwise, it could have happened that two robots of the same team take defensive positions, which would cause instant position switching after kickoff because of their different roles (cf. Sect. 4.1). Therefore – in contrast to regular games – the role assignment is also active during the Ready state, and the Ready position selection checks each possible assignment for constraints that disallow certain role-position combinations.

Another piece of team play that has been coordinated between B-Human and the HULKs is the handling of balls near the King (i.e., the goalkeeper). In this case, the King communicates its intention to play the ball. The Queen would then get out of the way of the King.

In the area of modeling, the whistle and the obstacle model have been communicated between HULKs and B-Human robots to build a more complete world model.

4.4 Competition Results

During the competition, the B-HULKs played four games and won all of them, although the final win required an additional penalty shootout. As both teams have robust implementations of all basic abilities required, such as stable walking, ball recognition, and self-localization, all robots on the field were able to play together reliably according to the strategy described above. The robots from B-Human and the HULKs equally contributed to our success in this competition.

5 Path Planning

Implementing the different roles often requires robots to be able to walk from one position on the field to another one without bumping into other robots, e.g.,

when walking to their kickoff positions or when walking to a distant ball. In these cases, a purely reactive control can be disadvantageous, because it usually would not consider obstacles that are further away, which might result in getting stuck. Therefore, our robots use a path planner in these situations since 2011. Until 2014, it was based on the Rapidly-Exploring Random Tree approach [2] with re-planning in each behavior cycle, i.e., for each new image taken. Although the planner worked quite well, it had two major problems: On the one hand, the randomness sometimes resulted in suboptimal paths and in oscillations¹. On the other hand, it seemed that the RRT approach is not really necessary for solving a 2-D planning problem, as the planner actually did. Thus, it was slower than it needed to be.

5.1 Approach

Therefore, it was replaced by a visibility-graph-based 2-D A* planner (cf. Fig. 6). The planner represents obstacles as circles on which they can be surrounded and the path between them as tangential straight lines. As a result, a path is always an alternating sequence of straight lines and circle segments. There are four connecting tangents between each pair of non-overlapping obstacle circles, only two between circles that overlap, and none if one circle contains the other (cf. Fig. 7). In addition, the current position of the robot and the target position are only points, not circles, which also influences the number of tangents. A robot can either walk in clockwise or in counter clockwise direction around an obstacle. It also always walks forward. This means that it has to leave a circle in the same direction it has entered it. In the path planning problem, this actually results in two nodes per obstacle circle, one for clockwise movement and one for counter clockwise movement, which are not directly connected.

With up to nine other robots on the field, four goal posts, the ball, and the optional requirement to avoid the own penalty area, the number of edges in the visibility graph can be quite high. Thus, the creation of the entire graph could be a very time-consuming task. Therefore, the planner creates the graph while planning, i.e., it only creates the outgoing edges from nodes that were already reached by the A* planning algorithm (cf. Fig. 6). Thereby, the A* heuristic (which is the Euclidean distance) not only speeds up the search, but it also reduces the number of nodes that are expanded. When a node is expanded, the tangents to all other *visible* nodes that have not been visited before are computed. *Visible* means that no closer obstacle circles intersect with the tangent, which would prevent traveling directly from one circle to another. To compute the visibility efficiently, a sweep line approach is used (cf. Fig. 8a). However, correctly ordering the circles by their distance would require quite a lot of bookkeeping, because of their different sizes and their possible intersections (cf. circle 5 in Fig. 8a). Instead, the sweepline is just ordered by the closest distances between circles and to check whether the endpoint of a tangent is reachable, the tangent is intersected with all circles in the sweepline the furthest point of which is closer

¹ Although the planner tried to keep each new plan close to the previous one.

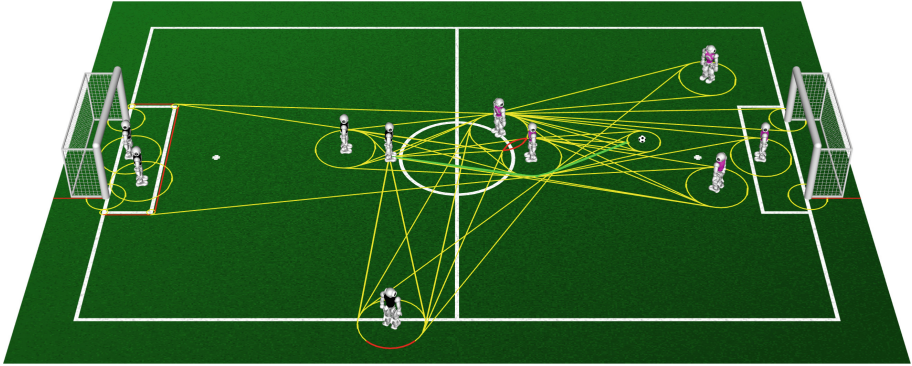


Fig. 6. Visualization of the planning process. The robot on the left of the center circle plans a path to a position suitable to kick the ball on the right towards the opponent goal. The obstacle circles and the edges expanded are shown in yellow. Sectors of the obstacle circles that are not traversable, either because they overlap with another circle or they are too close to the field border, are depicted in red. Barriers the robot is not allowed to cross, either to avoid walking through the goal net or because it should not enter its penalty area, are also shown in red. The shortest path determined is marked in green. (Color figure online)

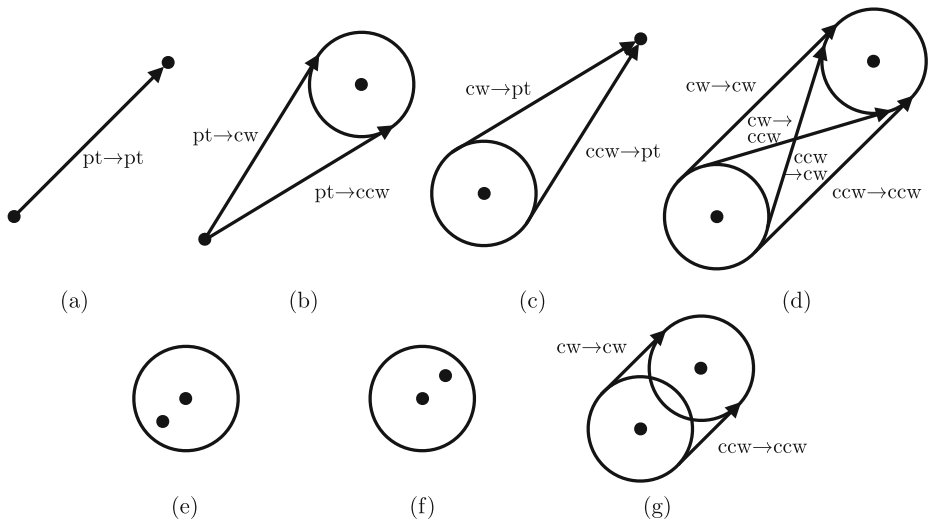


Fig. 7. Seven different combinations of nodes with the corresponding walking directions on the circles (point, clockwise, counter clockwise). (a) Point to point. (b) Point to circle. (c) Circle to point. (d) Circle to circle. (e, f) No tangents if point or smaller circle is inside other circle. (g) Circles overlap.

than the closest point of the tangent’s target circle. This means that not only the first entry in the sweepline is checked, but all entries until the upper bound is reached. However, in most cases, this still means that only a single entry is checked. As a result, the planning process never took longer than 1 ms per behavior cycle in typical games.

5.2 Avoiding Oscillations

Re-planning in each behavior cycle bears the risk of oscillations, i.e., repeatedly changing the decision, for instance, to avoid the closest obstacle on either the left or the right side. The planner introduces some stability into the planning process by adding an extra distance to all outgoing edges of the start node based on how far the robot had to turn to walk in the direction of that edge and whether the first obstacle is passed on the same side again (no extra penalty) or not (extra penalty). Note that this does not violate the requirement of the A* algorithm [1] that the heuristic is not allowed to overestimate the remaining distance, because the heuristic is never used for the outgoing edges of the start node.

5.3 Overlapping Obstacle Circles

The planning process is a little bit more complex than it appears at first glance: As obstacles can overlap, ingoing and outgoing edges of the same circle are not necessarily connected, because the robot cannot walk on their connecting circle segment if this is also inside another obstacle region. Therefore, the planner manages a set of walkable (non-overlapping) segments for each circle, which reduces the number of outgoing edges that are expanded when a circle is reached from a certain ingoing edge (cf. Fig. 6). However, this also breaks the association between the obstacle circles and the nodes of the search graph, because since

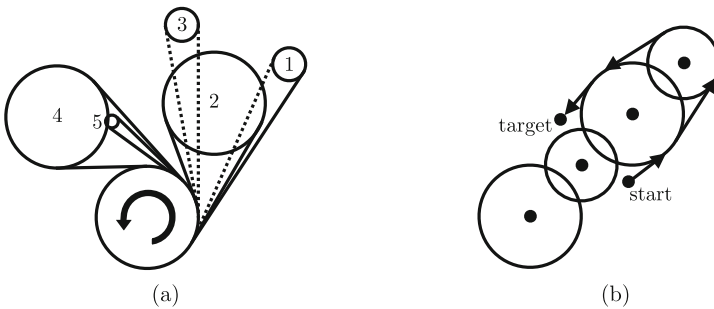


Fig. 8. (a) Computing edges from tangents using a sweepline ordered by distance (here for tangents starting in counter clockwise direction). All tangents are processed ordered by their direction. Right tangents enter a circle into the sweepline, left tangents remove a circle from the sweepline. The tangents with an endpoint that is the closest in that direction are kept (depicted as solid lines), all other tangents are removed (depicted as dotted lines). (b) An example for reaching the same circle twice.

some outgoing edges are unreachable from a certain ingoing one, the same circle can be reached again later through another ingoing edge that now opens up the connection to other outgoing edges (cf. Fig. 8b). To solve this problem, circles are cloned for each yet unreached segment, which makes the circle segments the actual nodes in the search graph. However, as the graph is created during the search process, this cloning also only happens on demand.

5.4 Forbidden Areas

There are two other extensions in the planning process. Another source for unreachable segments on obstacle circles is a virtual border around the field. In theory, the shortest path to a location could be to surround another robot outside of the carpet. The virtual border makes sure that no paths are planned that are closer to the edge of the carpet than it is safe (cf. Fig. 6). On demand, the planner can also activate lines surrounding the own penalty area to avoid entering it. The lines prevent passing obstacles on the inner side of the penalty area. In addition, edges of the visibility graph are not allowed to intersect with these lines. To give the planner still a chance to find a shortest path around the penalty area, four obstacle circles are placed on its corners in this mode. A similar approach is also used to prevent the robot from walking through the goal nets.

5.5 Avoiding Impossible Plans

In practice, it is possible that the robot should reach a position that the planner assumes to be unreachable. On the one hand, the start position or the target position could be inside obstacle circles. In these cases, the obstacle circles are “pushed away” from these locations in the direction they have to be moved the least to not overlap with the start/target position anymore before the planning is started². For instance in Fig. 6, the obstacle circle surrounding the ball was slightly moved away to make the target position reachable.

On the other hand, due to localization errors, the start and target location could be on different sides of lines that should not be passed. In these cases, the closest line is “pushed away”. For instance, if the robot is inside its penalty area although it should not be, this would move the closest border of the penalty area far enough inward so that the robot’s start position appears to be outside for the planning process so that a solution can be found.

6 Conclusion and Future Work

For successfully playing robot soccer and winning the RoboCup, several sub-problems need to be solved. In this paper, we represented the hypothesis that in

² Note that when executing the plan, these situations are handled differently to avoid bumping into other robots.

the current state of the RoboCup Standard Platform League, the development of sophisticated robot behaviors makes a difference between the top teams, in contrast to other tasks such as perception or motion, which have converged to solutions of similar quality.

As examples, we described our general tactical approaches for the main competition as well as for the Mixed Team competition, which we also won together with the HULKS. Furthermore, as an example for robot skills, our realtime path planning approach has been presented.

As there are still many open issues, regarding team tactics as well as robot skills, that have not been solved yet by any team, one major focus for RoboCup 2018 will be on the further improvement of the robot behaviors. In addition, the Standard Platform League currently plans to introduce some major changes such as free kicks that will require even more behavior development.

References

1. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* 4(2), 100–107 (1968)
2. LaValle, S.M.: Rapidly-exploring random trees: a new tool for path planning. Technical report TR 98–11, Computer Science Department, Iowa State University (1998)
3. Menashe, J., et al.: Fast and precise black and white ball detection for RoboCup soccer. In: *RoboCup 2017: Robot World Cup XXI*, Nagoya, Japan, July 2017
4. Müller, J., Laue, T., Röfer, T.: Kicking a ball – modeling complex dynamic motions for humanoid robots. In: Ruiz-del-Solar, J., Chown, E., Plöger, P.G. (eds.) *RoboCup 2010*. LNCS (LNAI), vol. 6556, pp. 109–120. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20217-9_10
5. Röfer, T.: CABSL – C-based agent behavior specification language. In: Akiyama et al. (eds.) *RoboCup 2017*. LNAI, vol. 11175, pp. 135–142. Springer, Cham (2018). <http://www.informatik.uni-bremen.de/kogrob/papers/RC-Roefer-18.pdf>
6. RoboCup Federation: Website RoboCup Standard Platform League (2017). <http://spl.robocup.org>
7. Röfer, T.: B-human team report and code release 2017 (2017). <http://www.b-human.de/downloads/publications/2017/coderelease2017.pdf>