Cooperating Physical Robots: A Lesson in Playing Robotic Soccer*

Bernhard Nebel

Albert-Ludwigs-Universität Freiburg, Institut für Informatik
Georges-Köhler-Allee, Geb. 52
D-79110 Freiburg, Germany

Abstract. Having a robot that carries out a task for you is certainly of some help. Having a group of robots seems to be even better because in this case the task may be finished faster and more reliably. However, dealing with a group of robots can make some problems more difficult. In this paper we sketch some of the advantages and some problems that come up when dealing with groups of robots. In particular, we describe techniques as they have been developed and tested in the area of robotic soccer.

1 Introduction

Having a robot that carries out a task for you, e.g., cleaning the floor or fetching the mail, is certainly of some help. Having a group of robots seems to be even better because in this case the task may be finished faster and more reliably. Sometimes one even needs a group to get the task done. For instance, playing robotic soccer requires a team of robots.

In general, some problems can be more easily, more reliably, or faster solved by a group of robots. For example, distributing mail or messages to many targets can be done faster with a group of robots, as has been demonstrated by the team of SRI robots winning one of the robot competitions at AAAI'96 [13]. Also basic tasks such as *self-localization* can be more reliably solved by a group of robots. On the other hand, dealing with a group of robots can make some problems more difficult. For instance, path planning is easier for one robot than for a group of robots.

While this sounds all very plausible, it also raises the question why a scenario of a cooperating team of robots is interesting from a scientific point of view. A cooperating group of robots appears simply to be a special case of a cooperating group of agents. This is certainly true in the same way as is the statement that robots are "simply" special cases of agents. Mobile robots are special in a number of ways. For this reason, one has to deal with problems that do not arise with other agents, e.g., software agents. Firstly, there is the problem that a robot has to perceive and to act in a physical world. Secondly, sensing and acting is uncertain. Thirdly, connected with the two former points, communication between the robots might not be possible, be restricted to low bandwidth, or

* This work has been partially supported by *Deutsche Forschungsgemeinschaft* as part of DFG project Ne 623/3-1

possible only over restricted distances. Apart from that, however, groups of robots can be viewed as multi-agent systems.

One should note, however, that a group of robots may also be viewed as one centrally controlled multi-bodied robot. While such a viewpoint is indeed possible, there are a number of arguments against such a perspective. Firstly, this multi-bodied robot has a very large number of *degrees of freedom* making it computationally infeasible to control it. Secondly, we might be unable to communicate between the different parts of the robot or the communication bandwidth is very low. Thirdly, we might be unable to estimate a global system state because for some parts of the robot we do not know the state. Fourthly, failures of parts of the multi-bodied robot are much more naturally dealt with when one takes a multi-agent perspective.

In the rest of the paper, we will present some case studies of techniques developed in the context of multi-robot systems. In the next section, we have a look at *cooperative sensing*. In Section 3, we then turn to a particular form of *coordinated behavior*, namely, *cooperative motion planning*. In order to support cooperative behavior in a group, often *roles* are assigned to the group members. How this can be done for a group of robots in a highly dynamic environment will be studied in Section 4. Finally, in Section 5, we will discuss what is needed to build a successful robotic soccer team and in how far the techniques described in this paper can help.

2 Cooperative Sensing

If there is a group of robots that can communicate with each other, it seems natural that the robots share their observations with each other. In this manner they can compensate for sensor limitations that, for instance, restrict the range in which an object can be sensed. Furthermore, by combining estimates, robots may be able to narrow down their hypotheses or to correct their estimates.

As mentioned in the Introduction, all sensor measurements are uncertain. There is always some (normally distributed) noise and in addition there might be some systematic error one cannot anticipate. For example, when using the odometry – measuring how often a wheel has turned – there is a normally distributed measurement error and depending on the floor, there may be an additional systematic error. In particular carpets can can lead to systematic diversions that cannot be anticipated. Finally, there may also be a large displacement from time to time when the robot collides with an obstacle or with another robot. Worse yet, these errors accumulate leading to high uncertainty about the robot's position after a very short time.

For these reasons, other means are used to solve the so-called *self-localization* problem. Measurements from other sensors are used to correct the estimates derived from the odometry. The mathematical tool that is used to deal with this problem is often the *Kalman filter* [17], a method of fusing all measurements in order to arrive at optimal estimates. Intuitively, it involves computing a weighted average over sensor measurements, where sensors which are more accurate have a higher weight than those which are known to be less accurate.

Often, known positions of recognized landmarks are used in the self-localization process for correcting the position estimates. However, when one wants to explore an

unknown territory, there are no known landmarks. With a group of robots that have initially known positions, it is possible to do something similar to landmark-based navigation, though. Some of the robots can be used as landmarks.

2.1 Cooperative Self-Localization

Rekleitis *et al.* [20] proposed a scheme for multi-robot exploration with a group of robots. In this approach it is assumed that the robots can track each other with reasonable reliability and accuracy as long the line of sight between them is free of obstacles. Under this assumption, one or more robots can move using one or more (temporary) immobile robots as landmarks. After a while the roles of the moving and immobile robots can be exchanged. Using such a method, the odometry error can be reduced dramatically [20].

In most applications, however, we already know the environment and "only" have to solve the self-localization problem. In this case it often happens that one robot can come up with multiple position hypotheses. If we now have a group of robots that are able to recognize other group members when they are close enough, it is possible that the robots narrow down the set of position hypotheses when they meet [8].

2.2 Cooperative Object Localization

Yet another scenario for multi-robot cooperative sensing is when we can assume that position and orientation of all robots are almost always accurate and reliably, but there is significant uncertainty and unreliability in sensing other objects. This situation occurs, for instance, in the robotic soccer context.

The players of the *CS Freiburg* team [11, 18, 23] use laser range finders in order to solve the self-localization problem [12], and for this reason can be assumed to know their own position very reliably. However, they are not very good in recognizing the ball and estimating its position on the field – which is done using a monocular vision camera.

There is a significant measurement error for estimating the distance to the ball, an error which increases with the distance between camera and ball. The angular error, on the other hand, is smaller and does not depend on the distance. Additionally, there is a restriction to the maximum distance over which the ball can be recognized, which is approximately 4–5 meters. Finally, the robots often enough recognize *false positives*, i.e., *phantom balls*. Ignoring the latter problem, one can again use a Kalman filter to fuse observations (with time stamps) from different robots in order to get estimates that a more accurate than any single measurement. In fact, this gives us a sort of stereo vision with a group of robots. Assuming that the angular error is much smaller than the distance error gives a triangulation effect as shown in Figure 1.

As already pointed out, sometimes the robots observe phantom balls. An example of such a situation is displayed in Figure 2. Two players see a ball close to the goal and another player sees a ball on the center line.

If we would now take the weighted average of the sensed ball positions, we would get a completely wrong estimate. For this reason, it seems preferable to exclude obviously wrong measurements. One way to do so would be to ignore measurements that are

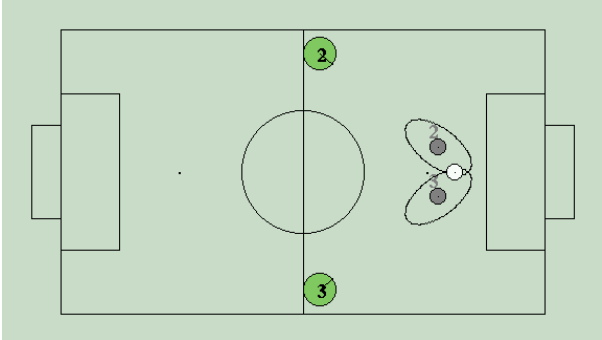

Fig. 1. The Kalman filter for integrating ball observations leads to triangulation. Grey discs denote position estimates for single robots, the ellipses around the grey discs denote measurement errors, and the white disc denotes the fused estimate.

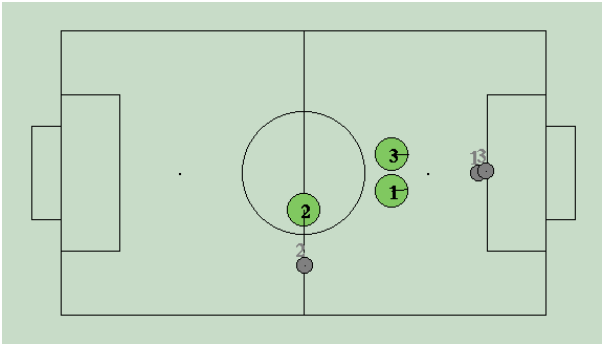

Fig. 2. Player 2 observes a phantom ball, which may lead to an incorrect estimate of the ball position.

implausible given the current estimate. This, however, could lead to a situation where a robot tracks a phantom ball and the other robots are all diverted from sensing the right ball because they believe the hallucinating robot.

The best (and most democratic way) to deal with such a situation is to believe in what the majority of robots sense. In the example depicted in Figure 2 we would rather believe the players 1 and 3 than player 2. One way to put such a *voting scheme* into effect is to use the so-called *Markov localization* approach [9] for the ball. In this approach one basically maintains (a discrete) probability distribution for the position probability of the object of interest. Usually, this is the robot itself. In our case, however, it is the ball. Each observation updates the position probability by increasing the probability at the location where the ball has been observed and lowers the probability where no observation has been made (using conditional observation probabilities and Bayes' rule). In addition, the position probability is flattened out for each time step to model the loss

of certainty over time. Using such an approach, it appears more probable that the ball is around the locations where it has been observed by two robots than at the location where it has been observed only by one robot. Combining Markov localization as a plausibility filter with a Kalman filter, one gets a quite reliable and accurate global ball estimation mechanism [6].

While all these methods might not appear to be overly sophisticated, the real value of these approaches is that they are based on a solid theoretical basis and work in practice. Almost all of of these approaches, however, are still passive in the sense that they do not involve the interplay between sensing and acting, i.e., active sensing [3].

3 Cooperative Path and Motion Planning

Latombe starts his book [16] with the following remark:

> "This capability [motion planning] is eminently necessary since, by definition, a robot accomplishes tasks by moving in the real world."

And what is true for single robots is, of course, also true for teams of robots.

The *basic motion planning problem* is usually stated [16] as the problem of moving a single rigid object – the *robot* – in an Euclidian (2- or 3-dimensional) space, the so-called *work space* from an initial position (and orientation) to a target position (and orientation). Of course, there can be *obstacles* in the workspace, which have to be avoided. The problem is usually solved by mapping the problem to the so-called *configuration space*. This space is generated by the *degrees of freedom* the robot has. In the 2-dimensional case, these degrees of freedom are (x, y, θ), i.e., the x and y coordinates of the robot position as well as its heading θ. In this 3-dimensional configuration space, the robot is just a point and we have to find a path from the start to the target configuration avoiding obstacles. In the special case that we have disk-shaped robots, the configuration space can be described by the x and y coordinates alone and so the configuration space is only 2-dimensional.

In the previous section, we have seen how sensing can lead to more accurate and reliable estimates if we have a group of robots. Furthermore, the additional computational costs are reasonable. In contrast to that, path and motion planning is computationally much more difficult if a group of robots is involved. This becomes obvious when one generalizes the configuration space planning method described above to a multi-robot system. In this case, for each robot 3 dimensions have to be added to the configuration space. Of course, this might be an indication that the configuration space approach is not appropriate. However, the multi-robot path planning problem is indeed inherently difficult. It is PSPACE-hard in the number of robots, as follows from results by Hopcroft *et al.* [14].

3.1 Cooperative Path Planning with Global Communication

If we assume that all robots can communicate with each other, the multi-robot path planning problem can be solved centrally, e.g., by using the configuration space approach sketched above. While this guarantees *optimality* and *completeness*, it is usually not efficient enough for even only a moderate number of robots.

Instead of a *centralized* approach, one can use *decoupled planning* [16]. In this approach, one plans first independent trajectories for all robots and then combines them, resolving conflicts when they arise. This reduces the complexity, but it also sacrifices optimality and completeness.

There are two decoupled planning methods that have been considered in the literature. First, there is the *prioritized planning* approach [7], which considers the multi-robot path planning problem as a sequence of path planning problems. One starts with the first robot and all the immobile obstacles. Then one adds another robot and plans a path avoiding all immobile robot and the moving robot from the first phase, and so on. The critical decision is what order one should use. Depending on this order, it is quite possible that no solution is found although there exists one.

The other decoupled approach is the so-called *path coordination* method [19], where the robots plan their paths independently (see Figure 3), and afterwards coor-

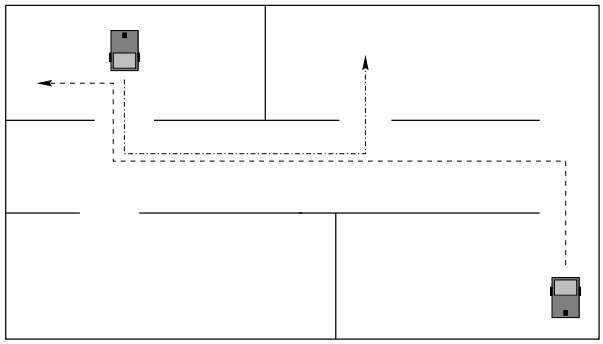

Fig. 3. Paths planned in a decoupled approach

dinate their movements without leaving their planned paths. As is obvious from this example, there is a chance that the two robots collide if they follow their planned paths without coordination. What we need here is a *collision-free schedule* the robots can follow. For two robots, this problem can be solved using so-called *coordination diagrams*, as the one shown in Figure 4. While this coordination diagram shows that with two robots the problem can be easily solved, it also gives a hint that the problem might become computationally difficult when the number of robots increase. The generalized coordination diagram would contain as many dimensions as there are robots. For this reason, often prioritization schemes are used [1].

Path coordination is, of course, even more restrictive than the prioritized planning approach and for this reason may find fewer solutions, which may be less cost-effective. This is indeed the case even in natural environments, as has been shown by Bennewitz and Burgard [2].

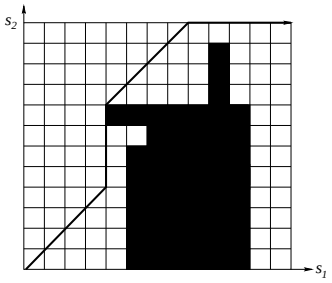

Fig. 4. Coordination diagram. The s_1 and s_2 axes represent the length robot 1 or robot 2, respectively, has already traveled on the respective planned path. Black cells represent collisions. The bold line shows a collision-free schedule.

3.2 Cooperative Path Planning with only Local Communication

Although the decoupled path planning methods do not attempt to control the group of robots as one entity – and reduce the algorithmic complexity by that – these approaches still presuppose that there is a *central coordinator* and a *global communication network*. If one assumes that only *local communication* between pairs of physically close robots is possible, then the decoupled approaches do not work.

Similar to the decoupled approaches, we will assume that the paths are planned independently by each robot. However, instead of relying on a central component that deals with conflicts, we will now assume that only local coordination is possible [15].

If two robots are close to each other, they establish a coordination link, which means that they create a coordination diagram, which determines their schedule. It now can happen that one robot (A) has to wait for the other one (B). Unfortunately, it might happen that also robot B may have to wait for another robot C, which in turn waits for A, i.e., we have a *deadlock*. These deadlocks have, of course, to be detected and resolved. Resolution of a deadlock can here mean that one tries to find an alternative path in the coordination diagram or that a new path (segment) is planned [15]. The right tool to use here are *distributed algorithms* for deadlock detection and resolution [5].

4 Role Assignment in Dynamic Environments

The previous two sections have focused on problems such as sensing and path planning. In these cases, the solutions appeared to be very robotics specific and the overlap with multi-agent system seems to be minimal. However, there are, of course, other multi-robot problems that have a true multi-agent flavor.

One such problem is the *assignment of roles* to members of a group [21]. Such an assignment serves the purpose of associating a set of behavioral patterns with the agents in order to support the coordination between the agents. For instance, in soccer, we distinguish at least between the roles *goalie*, *defender*, and *forward player*. There may be additional roles such as *midfielder* and *supporter*. In general, we want to determine a one-to-one function from the set of agents A to the set of roles R. Sometimes, we

may want also to consider different sets of roles, i.e., different *formations*. In soccer, for instance, we may want to deal with a 4-3-3 and a 3-3-4 formation and to switch between these formations.

A very simple way of dealing with this problem is to use a fixed assignment, as for example the *CS Freiburg* team did in 1998 [11]. Each robotic soccer player has a fixed role, which has an associated *home position* and an *area of competence*, as shown in Figure 5.

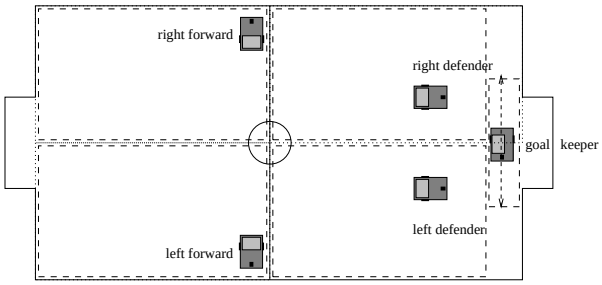

Fig. 5. Role assignment and areas of competence

When the ball moves into such an area of competence, the respective robot becomes active and tries to move the ball into the direction of the opponent goal – without leaving its area of competence. While this strategy does not appear to be optimal, it avoids the problem that a swarm of robot approaches the ball. In fact, only one robot of the team can be at the ball.

However, it is also clear that this delegation of duties has a number of severe problems. First of all, a defending robots can never run with the ball over the entire field and score a goal. They always have to pass the ball to a forward player. For this reason, very early on a "shouting" protocol was implemented that permits a robot with the ball to make a run to the opponent goal without being stopped by its own team members (see Figure 6).

A second disadvantage of the scheme described above is the disjoint decomposition of the field. It happened that the ball was in one of the competence areas, but the respective player was unable to go for the ball for some reason. Then no other player would come to help this player. This problem can be (and has been) solved by allowing overlaps of the competence areas and using the "shouting" protocol to avoid that two players block each other at the ball. Finally, there is the problem that once a robot breaks down, its role will not be filled by another robot on the field, even if the role is very important.

Although the *CS Freiburg* team became RoboCup world champion of the F2000 league in 1998, this was certainly not because its role assignment and coordination method were superior to that of the other teams. Indeed, there are a number of issues one has to address in order to build a flexible and robust team:

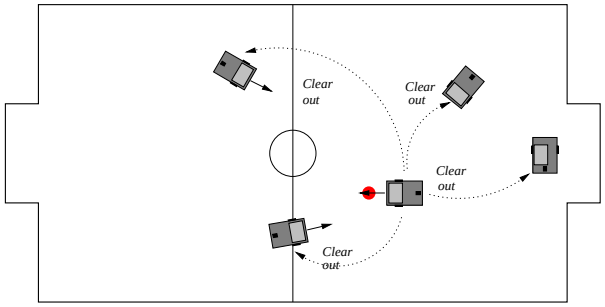

Fig. 6. Shouting in order to get a free run to the opponent goal

– role assignments should be changed dynamically to account for the current positioning and to support team reconfigurations after the break down or removal of individual team members; and
– flexible positioning that takes into account the entire situation on the field.

The latter point has been addressed by *CMUnited's* SPAR method [22], a method that tries to find the optimal position by using a linear programming method given the values for a number of important parameters such as ball position, position of team members, etc. Stone and Veloso [21] also addressed the issue of dynamic role re-assignments. However, this was approach was built on so-called *locker room agreements*, i.e., pre-built plans, and on filling more preferable roles if they are vacant.

A more flexible scheme for the dynamic assignment of roles has been proposed and used by the *ART Italy* team in 1999 [4]. They consider roles such as

– *active player*, the player which possesses the ball or goes to the ball;
– *supporter*, the player that moves parallel with the active player;
– *defender*, the player staying behind defending the goal.

Each agent can contribute some utility when filling a role. For instance, if a robot is already close to the defending position, it can contribute a high utility value when it fills the *defender* role. If it is close to the ball, it can contribute a high utility value if it fills the *active player* role. Each robot determines these utility values for each role and transmits the computed values to all other robots.

The roles for the field players are ordered by importance and assigned dynamically (in the importance order) to the player that can contribute most by filling the role according to the computed values. In fact, this assignment is done in a distributed manner, i.e. each robots decides on the basis of the received utility values which role to take. This can lead to situations, where a role is temporarily filled by two players. However, this does not happen very often and is resolved after a fraction of a second [4].

While this scheme appears to work very well, its efficiency seems to rely on ordering the roles by importance. A more general scheme would view the role assignment problem as an optimization problem, where we want to maximize the social welfare of

the entire group. While this sounds like a combinatorial, i.e., a computationally difficult, problem, it is simply the problem of finding a *maximal weighted match* [10], which can be solved in polynomial time. In the *CS Freiburg* team, this more general scheme together with a communication protocol for changing roles in a consistent manner is used [23]. This is complemented by a variant of the SPAR method [22] to find the right position for each role.

5 Conclusions and Discussion: Playing Robotic Soccer

As should have become obvious from what has been said so far, robotic soccer is a rich source of inspiration for multi-robot and multi-agent research. It has already led to the development of a number of interesting new methods, and it is an attractive testbed for comparing different methods.

One of the interesting questions is in how far the multi-robot and multi-agent methods sketched in the previous sections could contribute to creating a competitive robotic soccer team. An answer depends, of course, on what *league* one has in mind. In the *simulation league*, multi-agent considerations are most probably of utmost importance. In the real robot leagues, however, the single agent capabilites are most important. If these capabilities (such as sensor interpretation or ball skills) are flawed, then even the best cooperation technique will not help. However, as mentioned in Section 2.2, cooperative sensing can compensate for the shortcomings of sensors. In fact, a number of goal scoring attempts by other teams could be stopped because of the global ball position estimation technique used by the *CS Freiburg* team.

Furthermore, with the increase of the single-agent capabilities over the years, cooperative behavior becomes more and more important. This development is also mirrored in the evolution of the *CS Freiburg* team. While in 1998 a robust and accurate sensor self-localization method together with simple ball skills and a very basic cooperation mechanism (see Section 4) was enough to win, team play proved to be one of the important skills in winning the RoboCup competition again in 2000 [23].

References

1. K. Azarm and G. Schmidt. A decentralized approach for the conflict-free motion of multiple mobile robots. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'96)*, pages 1667–1675. IEEE/RSJ, 1996.
2. M. Bennewitz and W. Burgard. An experimental comparison of path planning techniques for teams of mobile robots. In *Autonome Mobile Systeme*. Springer-Verlag, 2000.
3. W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, Aug. 1997. Morgan Kaufmann.
4. C. Castelpietra, L. Iocchi, M. Piaggio, A. Scalza, and A. Sgorbissa. Communication and coordination among heterogeneous mid-size players: ART99. In P. Stone, G. Kraetzschmar, and T. Balch, editors, *RoboCup-2000: Robot Soccer World Cup IV*, Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, New York, 2001. Springer-Verlag. To appear.
5. K. M. Chandy, J. Misra, and L. M. Haas. Distributed deadlock detection. *ACM Transactions on Computer Systems*, 1(2):144–156, May 1983.

6. M. Dietl, S. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. 2001. Submitted paper.
7. M. A. Erdmann and T. Lozano-Pérez. On multiple moving objects. *Algorithmica*, 2(4):477–521, 1987.
8. D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Collaborative multi-robot localization. *Autonomous Robots*, 8(3), 2000.
9. D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.
10. Z. Galil. Efficient algorithms for finding maximum matchings in graphs. *ACM Computing Surveys*, 18:23–38, 1986.
11. J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, and T. Weigel. The CS Freiburg team: Playing robotic soccer based on an explicit world model. *The AI Magazine*, 21(1):37–46, 2000.
12. J.-S. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in polygonal environments. *Advanced Robotics Journal*, 2001. To appear.
13. D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige. Many robots make short work: Report of the SRI International mobile robot team. *The AI Magazine*, 18(1):55–64, 1997.
14. J. E. Hopcroft, J. T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: PSPACE-hardness for the `warehousman's problem'. *International Journal of Robotics Research*, 3(4):76–88, 1984.
15. M. Jäger and B. Nebel. Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. 2001. Submitted paper.
16. J.-C. Latombe. *Robot Motion Planning*. Kluwer, Dordrecht, Holland, 1991.
17. P. S. Maybeck. The Kalman filter: An introduction to concepts. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, Berlin, Heidelberg, New York, 1990.
18. B. Nebel, J.-S. Gutmann, and W. Hatzack. The CS Freiburg '99 team. In M. Veloso, E. Pagello, and H. Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*. Springer-Verlag, Berlin, Heidelberg, New York, 2000.
19. P. A. O'Donnell and T. Lozano-Pérez. Deadlock-free and collision-free coordination of two robot manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'89)*, pages 484–489, 1989.
20. I. M. Rekleitis, G. Dudek, and E. E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1340–1345, Nagoya, Japan, Aug. 1997. Morgan Kaufmann.
21. P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.
22. P. Stone, M. Veloso, and P. Riley. The CMUnited-98 champion simulator team. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, pages 61–76. Springer-Verlag, Berlin, Heidelberg, New York, 1999.
23. T. Weigel, W. Auerbach, M. Dietl, B. Dümler, J.-S. Gutmann, K. Marko, K. Müller, B. Nebel, B. Szerbakowski, and M. Thiel. CS Freiburg: Doing the right thing in a group. In P. Stone, G. Kraetzschmar, and T. Balch, editors, *RoboCup-2000: Robot Soccer World Cup IV*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, Heidelberg, New York, 2001. To appear.