RoboCup: The Robot World Cup Initiative

Hiroaki Kitano[1], Minoru Asada[2], Yasuo Kuniyoshi[3], Itsuki Noda[3], Eiichi Osawa[1]

Sony Computer Science Lab.[1] 3-14-13 Higashi-Gotanda Shinagawa, Tokyo 141 Japan

Dept. of Mechanical Engineering[2] Osaka University Suita, Osaka 565 Japan

Electrotechnical laboratory[3] 1-1-4 Umezono Tsukuba, 305 Japan

Mailing-list: `RoboCup@csl.sony.co.jp`

Web: `http://www.csl.sony.co.jp/person/kitano/RoboCup/RoboCup.html`

Abstract

The Robot World Cup Initiative (RoboCup) is an attempt to foster AI and intelligent robotics research by providing a standard problem where wide range of technologies can be integrated and examined. The first **RoboCup** competition will be held at IJCAI-97, Nagoya. In order for a robot team to actually perform a soccer game, various technologies must be incorporated including: design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, robotics, and sensor-fusion. Unlike AAAI robot competition, which is tuned for a single heavy-duty slow-moving robot, RoboCup is a task for a team of multiple fast-moving robots under a dynamic environment. Although RoboCup's final target is a world cup with real robots, RoboCup offers a software platform for research on the software aspects of RoboCup. This paper describes technical challenges involved in RoboCup, rules, and simulation environment.

1 Introduction: RoboCup as a Standard AI Problem

We propose a Robot World Cup (RoboCup), as a new standard problem for AI and robotics research. This is a proposal to use a soccer game as a platform for a wide range of AI and robotics research, such as design principles of autonomous agents, multi-agent collaboration, strategy acquisition, real-time reasoning, and sensor-fusion. Every year, AAAI hosts the robot competition for a single autonomous robot. Although the task of the AAAI competition changes every year, it is designed for a slow-moving and heavy-duty single robot[Nourbaksh et al 93]. The goal of the RoboCup is the opposite. RoboCup aims at providing a standard task for research on fast-moving multiple robots, which collaborate to solve dynamic problems. Although RoboCup's final target is a world cup with real robots, RoboCup offers a software platform for research on the software aspects of RoboCup. In addition, we intend to create an award for an expert robot, which demonstrates a high-level of competence for a specific task, such as shooting, intercepting, etc. Thus, RoboCup consists of three competition: the real robot competition, the software robot competition, and the special skills competition.

Standard AI problems have been the basic driving force for AI research. Research on computer chess, which is the most typical example of a standard problem, lead to the discovery of various powerful search algorithms. Other problems including, the Yale Shooting Problem and the Monkey-Banana, contributed to AI research by illustrating the essential difficulties involved in everyday reasoning. Criticisms against using such problems often focus on the fact that these are abstract tasks, which ignore essential difficulties of real world problem solving. Proponents of such criticism argue that the real world problem must be the target of serious research. While there is truth in such a claim, solving real world problems inherently involves domain-specific constraints and often social and economic constraints, which are not necessary common in other domains. In addition, research on usable real world systems are beyond the manpower and funding of many research groups. This hampers comparative studies of techniques for real world tasks. Thus, we need to setup a standard problem which is realistic, but affordable for many research groups.

The RoboCup is designed to meet the need of handling real world complexities, though in a limited world, while maintaining an affordable problems size and research cost. RoboCup offers an integrated research task covering the broad areas of AI and robotics. Such areas include: real-time sensor fusion, reactive behavior, strategy acquisition, learning, real-time planning, multi-agent systems, context recognition, vision, strategic decision-making, motor control, intelligent robot control, and many more.

2 Viewing a Soccer Game as A Multi-agent Environment

A soccer game is a specific but very attractive real-time multi-agent environment from the viewpoint of distributed artificial intelligence and multi-agent research. If we regard a soccer team as a multi-agent system, a lot of interesting research issues will arise.

In a game, we have two competing teams. Each team has a team-wide common goal, namely to win the game. The goals of the two teams are incompatible. The opponent team can be seen as a dynamic and obstructive environment, which might disturb the achievement of the common team goal. To fulfill the common goal, each team needs to score, which can be seen as a subgoal. To achieve this subgoal, each team member is required to behave quickly, flexibly, and cooperatively; by taking local and global situations into account.

The team might have some sorts of global (team-wide) strategies to fulfill the common goal, and both local and global tactics to achieve subgoals. However, consider the following challenges:

1. the game environment, i.e. the movement of the team members and the opponent team, is highly dynamic.
2. the perception of each player could be locally limited.
3. the role of each player can be different.
4. communication among players is limited, therefore, each agent is required to behave very flexibly and autonomously in real-time under the resource bounded situation.

Summarizing these issues, a soccer team can be viewed as a cooperative distributed real-time planning scheme, embedded in a highly dynamic environment. In cooperative distributed planning for common global goals, important tasks include the generation of promising local plans at each agent and coordination of these local plans. The dynamics of the problem space, e.g. the changing rate of goals compared with the performance of each planner, are relatively large, reactive planning that interleaves the plan generation and execution phases is known to be an effective methodology at least for a single agent [McDermott 78, Agre and Chapman 87, Maes 91, Ishida and Korf 91] to deal with these dynamic problems.

For cooperative plan schemes, there are frequent changes in the problem space or the observation of each agent is restricted locally. There is a trade-off between communication cost, which is necessary to coordinate the local plans of agents with a global plan, and the accuracy of the global plan (this is known as the predictability/responsiveness tradeoff). The study of the relationship between the communication cost and processing cost concerning the reliability of the hypotheses in FA/C [Lesser and Erman 80], and the relationship between the modification cost of local plans and the accuracy of a global plan in PGP [Durfee and Lesser 87] illustrate this fact. Also, Korf addressed it theoretically in [Korf 87].

Schemes for reactive cooperative planning in dynamic problem spaces have been proposed and evaluated sometimes based on the pursuit game (predator-prey) [Benda *et al.* 85, Stephens and Merx 89, Gasser *et al.* 89, Levy and Rosenschein 92, Korf 92, Osawa 95]. However, the pursuit game is a relatively simple game. Tileworld[Pollack and Ringuette 90] was also proposed and studied[Kinny and Georgeff 91, Ishida and Korf 91]. However, the environment is basically for the study of a single agent arichtecture.

We see that a robot soccer game will provide a much tougher, fertile, integrated, exciting, and pioneering evaluation environment for distributed artificial intelligence and multi-agent research.

3 Research Issues for RoboCup with Real Robots

In this section, we discuss several research issues involved in realizing real robots for RoboCup.

- **Design of RoboCup player and their control:** Existing robot players have been designed to perform mostly single behavior actions, such as pushing/dribbling/rolling [Connel and Mahadevan 93a, Asada et al. 95, Sahota 94], juggling [Rizzi and Koditschek 93, Schaal and Atkeson 94], or hitting [Watanabe et al. 94]. A RoboCup player should be designed so that it can perform multiple subtasks such as shooting (including kicking), dribbling (pushing), passing, heading, and throwing a ball; which often involves the common behavior of avoiding the opponents. Roughly speaking, there are two ways to build RoboCup players:

  1. Design each component separately, which is specialized for a single behavior and then assemble them into one.
  2. Design one or two components that can perform multiple subtasks.

  Approach 1 seems easier to design but more difficult to build and *vice versa*. Since the RoboCup player should move around quickly it should be compact; therefore, approach 2 should be a new target for the mechanical design of the RoboCup player. We need compact and powerful actuators with wide dynamic ranges. Also, we have to develop sophisticated control techniques for as few as possible multiple behavior components with low energy consumption. The ultimate goal of a RoboCup player would be a humanoid type, that can run, kick and pass a ball with its legs and feet; can throw a ball with its arms and hands, and can do heading with its head. To build a team of humanoid type robots currently seems impossible, this is just future goal.

- **Vision and sensor fusion:** Visual information is a rich source of information to perceive, not only the external world, but the effects of the robot's actions as well. Computer Vision researchers have been seeking an accurate 3-D geometry reconstructing from 2-D visual information, believing in that the 3-D geometry is the most powerful and general representation. This could be used in many applications, such as view generation for an video database, robot manipulation and navigation. However, the time-consuming 3-D reconstruction may not be necessary nor optimally suited for the task given to the RoboCup player. In order to react to the situation in real time, the RoboCup player quickly needs information to select behavior for the situation. we are not suggesting a special-purpose vision system, just that the vision is part of a complex system that interacts in specific ways with the world [Aloimonos 94]. RoboCup is one of these worlds, which would make clear the role of vision and evaluate the performance of image processing which has been left ambiguous in the computer vision field.

In addition to vision, the RoboCup player might need other sensing devices such as: sonar, touch, and force/torque, to discriminate the situations that cannot be discriminated from only the visual information nor covered by visual information. Again, the RoboCup player needs the real time processing for multi-sensor fusion and integration. Therefore, the deliberative approaches with rough estimation using multi-sensor system does not seem suitable. We should develop a method of sensor fusion/integration for the RoboCup.

- **Learning RoboCup behaviors:** The individual player has to perform several behaviors, one of which is selected depending on the current situation. Since programming the robot behaviors for all situations, considering the uncertainties in sensory data processing and action execution is unfeasible, robot learning methods seem promising. As a method for robot learning, reinforcement learning has recently been receiving increased attention with little or no *a priori* knowledge giving higher capability of reactive and adaptive behaviors [Connel and Mahadevan 93b]. However, almost all of the existing applications have been done only with computer simulations in a virtual world, real robot applications are very few [Asada et al 94a, Connel and Mahadevan 93a]. Since the prominence of the reinforcement learning role is largely determined by the extent to which it can be scaled to larger and complex robot learning tasks, the RoboCup seems a very good platform.

  At the primary stage of the RoboCup tournament, one to one competition seems feasible. Since the player has to take the opponent's motions into consideration, the complexity of the problem is much higher than that of simple shooting without an opponent. To reduce the complexity, task decomposition is often used. Asada et al. [Asada et al 94b] proposed a method for learning a shooting behavior avoiding a goal keeper. The shooting and avoiding behaviors are independently acquired and they are coordinated through the learning. Their method still suffers from the huge state space and the perceptual aliasing problem [Whitehead and Ballard 90], due to the limited visual field. Sahota [Sahota 94] proposed a reactive deliberation approach to the architecture for real time intelligent control in a dynamic environment. He applied it to a one to one soccer-like game. Since his method needs global sensing for robot positions inside the field, it does not seem applicable to the RoboCup that allows the sensing capability only through the agents (see the rule section).

  At the final stage, a many-to-many competition is considered. In this case, collective behaviors should be acquired. Defining all the collective behaviors as a team seems infeasible, especially, the situations where one of multiple behaviors should be performed. It is difficult to find a simple method for learning these behaviors, definition of social behaviors [Mataric 94]. A situation would not be defined as the exact positions of all players and a ball, but might be perceived as a pattern. Alternatives, such as "coordination by imitation," should be considered.

In addition to the above, the problems related to the RoboCup such as task representation and environment modeling are also challenging ones. Of course, integration of the solutions for the problems mentioned above into a physical entity is the most difficult one.

4 Rules for RoboCup

This section describes rules for RoboCup. RoboCup consists of three sections, the real robot section, the simulation section and the special skill section. In the real robot section, real robots are controlled by themselves or computers play matches. Competitors need to construct real robots and their control programs. The simulation section is the computer-simulation version of the real robot section, so that competitors need to construct only control programs for each player. In the special skill section, competitors compete for one of the special skills, for example, a penalty kick (PK), goal saving, human-like actions, and so on.

In each section, play basically conforms to the rules of human soccer. Two teams of up to 11 robots compete with each other, trying to kick a ball into the goal of the opponent. However, because of differences between robots and humans, it is necessary to simplify and modify the rules for the robots. Major changes are as follows:

- Field size is 1/20 of the World Cup soccer field, but the width of the goal is enlarged.
- Total number of robots in a team is limited. This agrees with the real soccer game. We set the upper limit to 11. We allow each team to consist of a smaller numbers of robots, such as 3 robots or 4 robots. It is the design decision of each team.
- Total weight of robots in a team is limited. Weight of a robot is an important factor because there is a lot of physical contact in a soccer match.
- Total size of robots is also limited. This is necessary to prohibit making a wall of robots to cover the goal.
- Robots may move with wheels or caterpillars instead of legs. Considering the current state of technology, it is a hard task to build a robot that can walk on two legs. Therefore it is reasonable to allow wheels.
- Robots must not *hold* [1] a ball for more than 5 seconds. In order to 'kick' the ball, robots may have imitation legs. In this case, the robot can control the

[1] 'A robot holding a ball' means that the robot makes the ball not accessible to other robots. For example, the followings are considered *holding*:

- To surround a ball with its arms.
- To pick up the ball.

ball by using its legs like hands. In order to avoid "holding". We introduce the "hand-ball" rule.

- Fouls concerned with the intentions of player and flows of plays, such as obstruction, dangerous charging etc., are ignored. Considering the current state of technology, it is difficult to determine the intentions of robots. Therefore it is difficult to judge these fouls.
- In the simulation section, we will use a soccer server system, which provides an interface to control a player via networks for each client program. A client program can control a player on a field with "dash", "turn" and "kick" commands. The client can get information about the field and messages from other players and referees.

In the early stages of RoboCup, human referees will control the match. Many soccer rules require the subjective interpretation of the referee. Therefore, it is difficult to judge plays automatically. However building a referee program and robot is an important open problem.

These RoboCup rules described in this section are tentative and details are currently under discussion.

5 RoboCup Simulators

5.1 The Soccer Server

In the simulation section, we will use Soccer Server, a simulator of **RoboCup** developed by Dr. Itsuki Noda, ETL, Japan, which is a network-based graphical simulation environment for multiple autonomous mobile robots in a 2D space. Using the soccer server, each client program can control each player on a soccer field via UDP/IP. This allows us to compare different types of multi-agent systems through the server, and test how well techniques of cooperation of agents work in dynamical varied situations.

The soccer server provides a virtual field where players of two teams play a soccer (association football) game. Each player is controlled by a client program via local area networks. Control protocols are simple in that it is easy to write client programs using any kind of programming system that supports UDP/IP sockets.

Control via Networks: A client can control a player via local area networks. The protocol of the communication between clients and the server is UDP/IP. When a client opens a UDP socket, the server assigns a player to a soccer field for the client. The client can control the player via the socket.

Physical Simulation: The soccer server has a physical simulator, which simulates movement of objects (ball and players) and collisions between them. The simulation is simplified so that it is easy to calculate the changes in real-time, but the essence of soccer is not lost.

The simulator works independently of communications with clients. Therefore, clients should assume that situations on the field change dynamically.

Referee: The server has a referee module, which controls each game according to a number of rules. In the current implementation, the rules are: (1) Check goals; (2) Check whether the ball is out of play; (3) Control positions of players for kick-offs, throw-ins and corner-kicks, so that players on the defending team keep a minimum distance from the ball.

Judgments by the referee are announced to all clients as an auditory message.

5.2 MARS

The other possible simulator is MARS. MARS is intended for use in intelligent robotics research on single/multiple mobile robot(s), such as RoboCup, behavior learning, map building, collective intelligence, multi-robot cooperation, cooperative learning, etc. **MARS** is a network-based graphical simulation environment for multiple autonomous mobile robots in a 2D space. This simulator is under development by Electrotechnical Laboratory, using the EuLisp object-oriented programming environment, developed by Dr. Toshihiro Matsui of ETL.

Modular Architecture: The MARS system adopts a modular architecture; a physical simulation module, robot control modules, a graphical user interface module (GUI), and network interface modules.

Physical Simulation: Our current physical simulation module handles four types of objects: **walls** (static obstacles, used for goal posts), **area** (static area with containment checking used for soccer court representation), **blocks** (passive objects), and **a robot-body** (active objects).

Robot Models: There is a clear interface between the physical simulation module and the robot control modules. Any number of **robot** models can be generated and simulated in (pseudo-)parallel. Each **robot** consists of a pair of modules, a **robot-body** and a **robot-brain**. A **robot-body** defines physical properties of a robot and a **robot-brain** defines how the robot behaves.

Sensor Models: A user can choose and attach any number of sensors to any place of a **robot-body**. Currently available sensor models are as follows: **distance** (odometry), **angle** (rotation), **touch**, **infrared**, **radar** (sonar), and **eye** (object-name sensor). Noise or uncertainty is not considered in the current version.

Robot Brains: A **robot-brain** is a user defined module for processing the simulated sensor data and generating action commands. It must accept sensor data and return action commands. In addition, it must be written as a re-entrant program, time-sliced by a **:step** message, sent by the global control loop. As long as these constraints are met, a user can adopt any cognitive architecture. The system provides an example behavior-based type architecture as a default.

Network Extension: Each **robot-brain** can be configured to be connected to an external process, via an asynchronous socket connection (UDP/IP). In this case, a user can use an arbitrary language (C, Prolog, Scheme, Perl, etc...) and an arbitrary control scheme (sequential, multi-thread, etc.) to write the **remote-brain**. Thanks to the asynchronous connection, a user does not have to worry about handshaking or timing problems concerning communication. Moreover the simulator server is not affected by communication failures.

GUI: The **MARS** main window has a menu-bar with several buttons for controlling the system. Also, the system has a built-in graphical editor for creating/modifying the physical environment, including the robot-body. You can save/load a physical environment definition together with agent definitions to/from a file.

Soccer Specific Extensions: Soccer specific brain-body message protocol is given in Table 1. The messages are defined on top of the primitive messages/actions provided in MARS.

6 Summary

In this paper, we proposed a RoboCup as a new standard AI problem. RoboCup provides rich research issues for a wide range of AI and robotics studies. We are currently inviting participation to this initiative, in order to define rules of play, develop a common research environment, and to host competitions and workshops. Those who are interested in RoboCup, please send e-mail to `RoboCup@csl.sony.co.jp`. Or write to: Robot World Cup Initiative (RoboCup), c/o Hiroaki Kitano, Sony Computer Science Laboratory, 3-14-13 Higashi-Gotanda, Shinagawa, Tokyo 141 Japan, or Prof. Minoru Asada, Department of Mechanical Engineering, Osaka University, Suita, Osaka, 565 Japan.

References

[Agre and Chapman 87] P. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pp. 268–272, 1987.

[Aloimonos 94] Y. Aloimonos. "Rply: What i have learned". *CVGIP: Image Understanding*, 60:1:74–85, 1994.

[Asada et al. 95] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. Vision-based reinforcement learning for purposive behavior acquisition. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1995.

[Asada et al 94a] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda. "purposive behavior acquisition on a real robot by vision-based reinforcement learning". In *Proc. of MLC-COLT (Machine Learning Conference and Computer Learning Theory) Workshop on Robot Learning*, pages 1–9, 1994.

[Asada et al 94b] M. Asada, E. Uchibe, S. Noda, S. Tawaratsumida, and K. Hosoda. "coordination of multiple behaviors acquired by vision-based reinforcement learning". In *Proc. of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994 (IROS '94)*, pages 917–924, 1994.

[Benda *et al.* 85] M. Benda, V. Jagannathan, and R. Dodhiawalla. On Optimal Cooperation of Knowledge Sources. Technical Report BCS-G2010-28, Boeing AI Center, 1985.

[Connel and Mahadevan 93a] J. H. Connel and S. Mahadevan. "Rapid task learning for real robot". In J. H. Connel and S. Mahadevan, editors, *Robot Learning*, chapter 5. Kluwer Academic Publishers, 1993.

[Connel and Mahadevan 93b] J. H. Connel and S. Mahadevan, editors. *Robot Learning*. Kluwer Academic Publishers, 1993.

[Durfee and Lesser 87] E. Durfee and V. Lesser. Using Partial Global Plans to Coordinate Distributed Problem Solvers. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, 1987.

[Gasser *et al.* 89] L. Gasser, N. Rouquette, R. Hill, and J. Lieb. Representing and Using Organizational Knowledge in Distributed AI Systems. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence, Volume II*, pp. 55–78. Morgan Kaufmann Publishers, Inc., 1989.

[Ishida and Korf 91] T. Ishida and R. Korf. Moving Target Search. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 204–210, 1991.

[Kinny and Georgeff 91] D. Kinny and M. Georgeff. Commitment and Effectiveness of Situated Agents. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 82–88, 1991.

[Korf 87] R. Korf. Planning as Search: A Quantitative Approach. *Artificial Intelligence*, Vol. 33, No. 1, pp.65–88, 1987.

[Korf 92] R. Korf. A Simple Solution to Pursuit Games. In *Proceedings of the Eleventh International Workshop on Distributed Artificial Intelligence*, 1992.

[Lesser and Erman 80] V. Lesser and L. Erman. Distributed Interpretation: A Model and Experiment. *IEEE Transactions on Computers*, Vol. 29, No. 12, pp.1144–1163, 1980.

[Levy and Rosenschein 92] R. Levy and J. Rosenschein. A Game Theoretic Approach to Distributed Artificial Intelligence. In Eric Werner and Yves Demazeau, editors, *Decentralized A.I. 3*. Elsevier/North Holland, 1992.

[Maes 91] P. Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pp. 49–70. The MIT Press, 1991.

[Mataric 94] M. J. Mataric. Learning to behave socially. In *Proc. of the 3rd Int. Conf. on Simulation and Adaptive Behaviors – From animals to animats 3 –*, pages 453–462, 1994.

[McDermott 78] D. McDermott. Planning and Action. *Cognitive Science*, Vol. 2, pp.71–110, 1978.

[Nourbaksh et al 93] I. Nourbaaksh, et al., The Winning Robots from the 1993 Robot Competition. *The AI Magazine*, Vol. 14, No. 4, 51-62, The AAAI Press, 1993.

[Osawa 95] E. Osawa. A Metalevel Coordination Strategy for Reactive Cooperative Planning. In *Proceedings of the First International Conference on Multi-Agent Systems*, 1995.

Command	Return Message	Description
(turn Dir)		Turn the body to Dir.
(dash Power)		Accelerate the body by Power in the current heading direction.
(lookup)	(ok lookup . Result) Result ::= (ObjInfo ObjInfo ...) ObjInfo ::= (ObjName Direction Distance)	Detect objects around.
(kick Dir Power)		Kick toward Direction with Power.
(say Message)		Put Message on a global blackboard. It expires after a fixed duration.
(hear)	(ok hear . MessageList) MessageList ::= (MessInfo MessInfo ...) MessInfo ::= (Name Time Message) Name ::= Originator of the Message Time ::= Time stamp of the Message	Fetch all the messages from the blackboard.

Table 1: Soccer specific message protocol between a robot-brain and a robot-body.

[Pollack and Ringuette 90] M. Pollack and M. Ringuette. Introducing the Tileworld: Experimentally Evaluating Agent Architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pp. 183–189, 1990.

[Rizzi and Koditschek 93] A. A. Rizzi and D. E. Koditschek. Further progress in robot juggling: The spatial two-juggle. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 919–924, 1993.

[Sahota 94] M. K. Sahota. Reactive deliberation: An architecture for real-time intelligent control in dynamic environments. In *Proc. of AAAI-94*, pages 1303–1308, 1994.

[Schaal and Atkeson 94] S. Schaal and C. G. Atkeson. Robot learning by nonparametric regression. In *Proc. of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994 (IROS '94)*, pages 478–485, 1994.

[Stephens and Merx 89] L. Stephens and M. Merx. Agent Organization as an Effector of DAI System Performance. In *Proceedings of the Ninth Workshop on Distributed Artificial Intelligence*, pp. 263–292, 1989.

[Watanabe et al. 94] H. Watanabe, Y. Nihna, y. Masutani, and F. Miyazaki. Vision-based motion control for a hitting task - hanetsuki -. In *Proc. of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 1994 (IROS '94)*, pages 910–916, 1994.

[Whitehead and Ballard 90] S. D. Whitehead and D. H. Ballard. "Active perception and reinforcement learning". In *Proc. of Workshop on Machine Learning-1990*, pages 179–188, 1990.