

Implementation of Internet-Based Personal Robot with Internet Control Architecture

Kuk-Hyun Han, Shin Kim, Yong-Jae Kim, Seung-Eun Lee and Jong-Hwan Kim

Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology (KAIST),
Kusong-dong, Yusong-gu, Taejon, 305-701, Republic of Korea
{khhan, skim, yjkim, selee and johkim}@vivaldi.kaist.ac.kr
<http://vivaldi.kaist.ac.kr>

Abstract

This paper describes the implementation of an internet-based personal robot with novel direct internet control architecture which is insensitive to the inherent internet time delay. The personal robot can be controlled by using a simulator provided at a local site. However, a large internet time delay may make some control inputs distorted. Moreover, since it is affected by the number of the internet nodes and loads, this delay is variable and unpredictable. The proposed control architecture guarantees that the personal robot can reduce the path error and the time difference between a virtual robot at the local site and a real robot at the remote site. Simulations and experimental results in the real internet environment demonstrate the effectiveness and applicability of the internet-based personal robot with the proposed internet control architecture.

1 Introduction

Recently it has been observed that many researchers take interest in internet robotics because of the merits of internet which enables users to access any systems on the network cheaply. The robot arm control system [1] through a Web browser was designed, and TeleGarden system [2] and Mars Pathfinder [3] were developed. The sensor-based mobile robot system [4] which can be controlled by using a Web browser and the internet-based supervisory architecture [5] were reported. The concept of a personal tele-embodiment [6] and an intelligent telerobot [7] were introduced recently. Most of them have the supervisory control scheme which enables users to issue high level commands. The internet time delay is variable and unpredictable so that the design of a direct con-

trol scheme which enables users to control the motion of the robot continuously may not be easy. The direct control scheme [8] on the internet was proposed, but the modeling of the internet time delay was not adequate.

This paper describes the implementation of an internet-based personal robot with novel internet control architecture which guarantees that the personal robot can reduce the path error and the time difference between the actions of a virtual robot at the local site and a real robot at the remote site. An internet user can control the real robot using a simulator provided at the local site, and can have information on the real environment at the remote site since the simulator has a virtual environment. The path error and the time difference in the internet-based personal robot system are caused by the unpredictable internet time delay and the difference between the real environment of the remote site and the virtual environment of the local site. It is not easy to model internet time delay, hence a control architecture that is insensitive to time delay, is needed. Main components of the proposed internet control architecture are a command filter to recover the information loss of control commands, a path generator and a path-following controller to reduce the time difference between the real robot and the virtual robot. The difference between the real robot and the virtual robot model of the simulator can be overcome by a posture estimator. The problem caused by the difference between the two environments can be solved by applying a virtual environment supervisor to the control architecture. We have already proposed the control architecture with the virtual environment supervisor [9], and it will not be mentioned in this paper. The graphic user interface(GUI) implemented with Java and the practical applications of the internet-based personal robot were described in [10].

This paper is organized as follows. In Section 2, the developed internet-based personal robot system, the modeling of a mobile robot, and the characteristics of internet time delay are described. In Section 3, the novel internet control architecture is designed step by step, which is insensitive to internet time delay. Section 4 presents the simulation results of the proposed internet control architecture. Real experiments with the developed internet-based personal robot are provided to show the effectiveness and applicability of the proposed internet control architecture. Concluding remarks follow in Section 5.

2 Internet-based Personal Robot

2.1 System description

The internet-based personal robot (IPR), a kind of service robot, can be used for a person's convenient life in a house/office or any indoor environment. It has a personal computer (PC) as a main part, and it can obtain information about environmental changes by using vision cameras, sonar sensors, etc. Actuators enable the robot to move and to carry out some physical work. It has a wireless LAN system for the internet remote control. A user can control the IPR using a simulator provided at a local site. It has the intelligence to gather the data from the sensors and to process them to decide its action.

The overall system consists of computers at the local sites, internet, wireless LAN system and the IPR. Users can access the IPR located at the remote site via internet using a computer at the local site. The wireless LAN system connects the IPR to the internet.

At the local site to control the IPR, a remote control architecture of the IPR system should be designed considering the inherent internet time delay. A basic remote control architecture of the IPR system considered in this paper is described in the following. A user controls a virtual robot in a simulator provided at the local site, and the virtual robot uses a virtual environment for obstacle avoidance. The command signals given by the user are sent to the IPR at the remote site via the internet. The IPR moves like the virtual robot, and avoids obstacles using sensor information. The posture of the virtual robot in the simulator can be updated by feedback of the IPR posture information through the internet.

The developed IPR has a square body of size $45cm \times 52cm \times 75cm$ as shown in Figure 1. The weight is about $70Kg$. It is a 4-wheeled drive with

two fixed wheels and two auxiliary off-centered orientable wheels. It consists of a personal computer (Pentium II 333Mhz), a wireless LAN (Samsung MagicWave, 2Mbps), a head with two vision color cameras, a 12.1 inch TFT monitor, a speaker, a microphone, sonar sensors (10 pairs), a 12V 100Ah battery (5hr 80Ah), and two AC servo motors (LG Industrial Systems, 200W). Two cameras of the head part can rotate around a vertical and a horizontal axis under the command of the three DC motors. The IPR is connected to the internet through the wireless LAN, and it works as a server. The user can connect to the IPR using a Web browser or a TCP/IP application anywhere and also the user can give motion commands to the IPR.

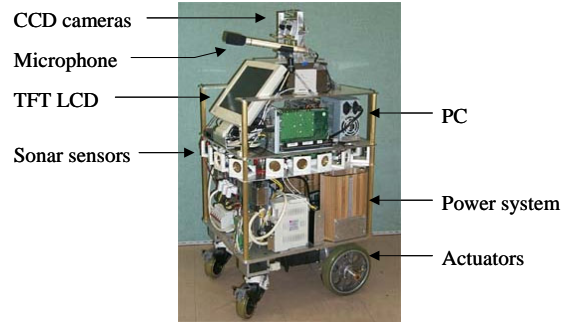


Figure 1: The IPR hardware

2.2 Modeling of the IPR

The modeling of IPR is needed for the implementation of the simulator. Two fixed and two auxiliary off-centered orientable wheeled mobile robots with non-slipping and pure rolling are considered. The velocity vector $\mathbf{u} = [v \ \omega]^T$ consists of the translational velocity of the center of driving wheel axis and the rotational velocity with respect to the center of driving wheel axis. The velocity vector \mathbf{u} and a posture vector $\mathbf{P}_c = [x_c \ y_c \ \theta_c]^T$ are associated with the robot kinematics as follows:

$$\dot{\mathbf{P}}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos \theta_c & -h \sin \theta_c \\ \sin \theta_c & h \cos \theta_c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}(\theta_c) \mathbf{u} \quad (1)$$

$$\mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix} \quad (2)$$

where v_R is the right wheel velocity, v_L is the left wheel velocity, h is the displacement from the center of robot to the wheel axis, and L is the distance between the two wheels.

2.3 Internet time delay

A large internet time delay may make some control inputs distorted. Moreover, since it is affected by the number of the internet nodes and loads, this delay is variable and unpredictable. Figure 2 shows the influence of the internet time delay on the control information. The received data at the remote site was distorted severely, and the information of the sine function was almost lost, when the test function used was $y(t) = 5 \sin(0.2\pi t) + 5$.

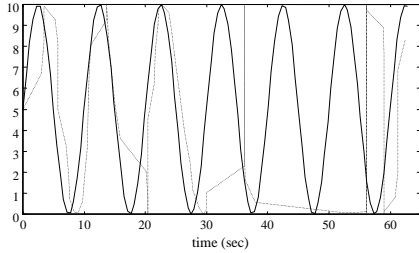


Figure 2: Influence of the internet time delay

3 Internet Control Architecture

A user can control the IPR at the remote site through internet using a simulator provided at the local site. The user regards the status of the virtual IPR at the local site as that of the real IPR at the remote site. Since the user cannot recognize the environment of the remote site, it is expected that the real IPR moves as the virtual IPR does. However, because of time delay we have to compensate for the path error and the time difference between the real IPR and the virtual IPR, which increase as time goes on.

In this section, a novel internet control architecture is designed step by step to minimize the effect of internet time delay. The proposed architecture is completed in three design steps, and its effectiveness is verified through simulations and experiments.

The internet control architecture-I consists of a user interface, simulator, virtual environment and posture estimator, which can be devised from the basic concept as shown in Figure 3. In the figure, $\mathbf{u}_r(i)$ is the i th control command $[v_r(i) \ \omega_r(i)]^T$ from a user, $\mathbf{u}_r^d(i)$ the i th control command passed through the internet, $\mathbf{P}_c(i)$ the i th robot posture, $\mathbf{P}_c^d(i)$ the i th robot posture passed through the internet, $\hat{\mathbf{P}}_c(i)$ the i th estimated posture, and $\mathbf{P}_c^s(i)$ the i th posture of the virtual robot. In order to correct the posture error between the virtual robot and the real robot, the real robot

generates feedback signals such as posture information of the real robot, to the simulator. The architecture-I can be considered as a basic structure.

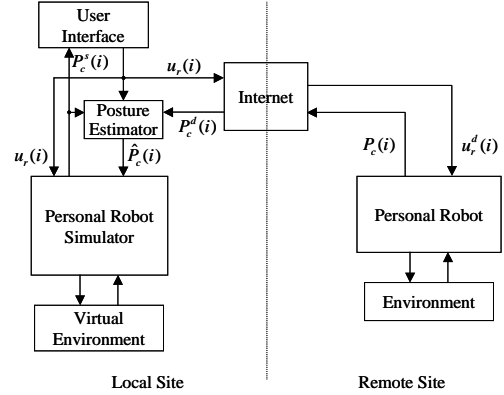


Figure 3: Internet control architecture-I (Step 1)

User Interface which can be implemented by Java, C++, etc., enables a user to control a remote IPR. *Posture estimator* estimates the current posture of the virtual IPR based on the feedback information of the real IPR. *Personal robot simulator* is the same as the virtual mobile robot at the local site. *Virtual environment* has the information of the real environment so that it enables the virtual robot to avoid obstacles. *Personal robot* is the same as the real mobile robot at the remote site. *Environment* is a circumstance where the real IPR is working.

The internet control architecture-I has a weak point that the information loss of control commands increases when internet time delay occurs. The *posture estimator* can recover the information loss eventually, but the time required for the recovery becomes too long. The architecture which can get rid of the cause of the information loss is needed. Figure 4 shows the internet control architecture-II, where a *command filter* is introduced. The *command filter* can recover the information loss of control commands caused by the internet time delay. It means that the filter reduces the path error between the real robot and the virtual robot. The function of the *command filter* is shown in Figure 5. Command signals received at the same time after the internet time delay T_d are regenerated with the sampling time T in the *command filter*. The *command filter* consists of two modules such as a *command queue* and a *command generator*. The *command filter* and the two modules can be defined by DEVS (Discrete Event Systems Specifications) formalism [9]. The *command filter* receives a control command, and stores it in the *command queue*. The *command gener-*

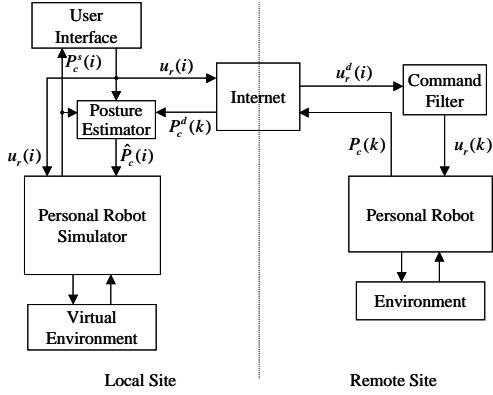


Figure 4: Internet control architecture-II (Step 2)

ator pulls out the command from the *command queue* and outputs it at each sampling time T .

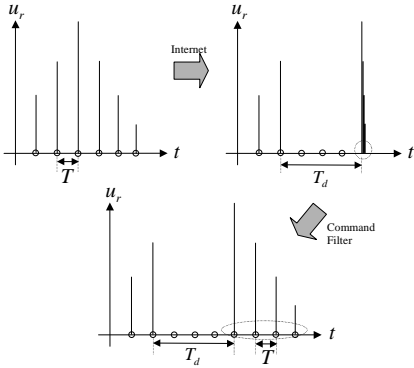


Figure 5: Function of *command filter*

The internet control architecture-II can recover the information loss of control commands, though internet time delay exists, but it still has the serious problem that the time difference between the real robot and the virtual robot increases, as internet time delay T_d is accumulated in the *command filter*. In order to solve this problem, the internet control architecture-III is finally designed.

The internet control architecture-III guarantees that the path error and the time difference between the real IPR at the remote site and the virtual IPR at the local site can be reduced. The proposed architecture includes a *path generator* and a *path-following controller*. The *path generator* restores the moving path of the virtual robot. The *path-following controller* guarantees that the real robot follows the generated path. The time difference between the real robot and the virtual robot can be reduced by the *path generator* and the *path-following controller*. As the control input of

the real robot is separated from the control command passed through the internet by the two modules, the *command generator* in the *command filter* can be modified by replacing sampling time T with the processing time T_p which is shorter than T . The processing time is the computing interval for generating a path segment for one control command.

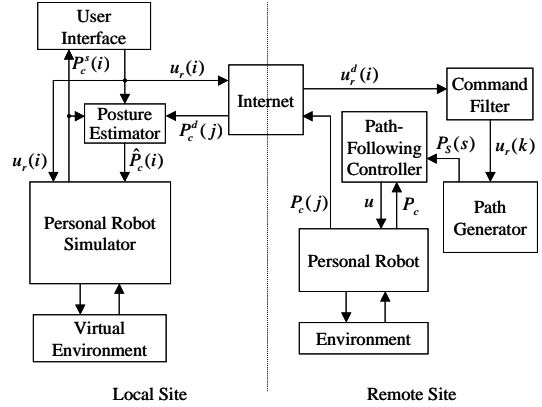


Figure 6: Internet control architecture-III (Step 3)

Figure 6 shows the internet control architecture-III, where $\mathbf{P}_S(s)$ is the moving path of the virtual robot, \mathbf{u} is the control input of the *path-following controller*, \mathbf{P}_c is the current posture of the real robot, and $\mathbf{P}_c(j)$ is the j th robot posture to be fed back to the simulator.

In this paper, the *path-following controller* is implemented with the uni-vector field navigation method [11]. The uni-vector field makes the mobile robot converge to a desired path.

4 Simulations and Experiments

4.1 Simulation results

Simulations were performed in the real internet environment. The physical distance between the local site and the remote site was about 300Km in Korea. Figure 7 shows the path error and the time difference between the two robots.

It should be noted that the architecture-I had cumulative errors. Because of this, it might be inconvenient for the user to control the robot in the simulation environment by the architecture-I. By the architecture-II the path error could be reduced, but the time difference increased continuously. However, by the architecture-III, the time difference as well as the path error were very small, although the internet time delay was quite variable.

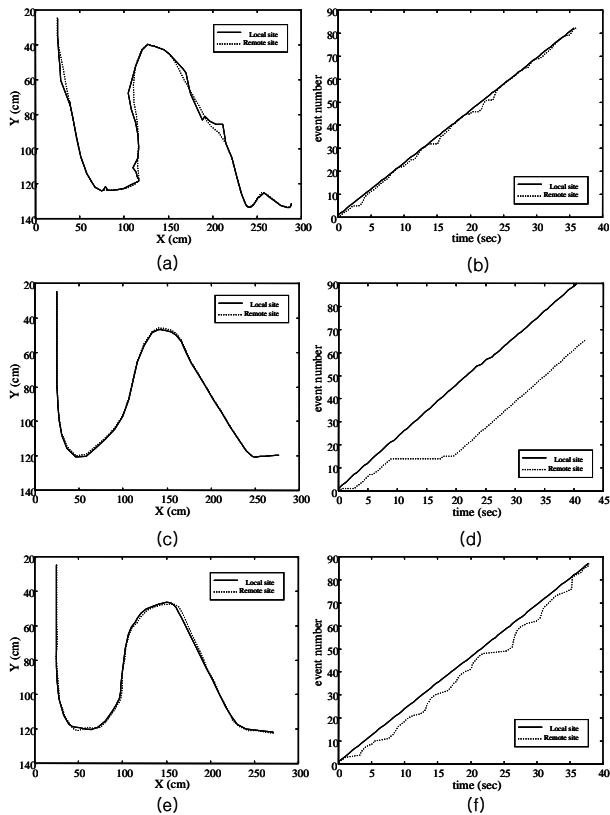


Figure 7: Simulation results. The path error and the time difference. (a),(b) architecture-I. (c),(d) architecture-II. (e),(f) architecture-III.

4.2 Experimental results

Experiment was performed with the developed IPR system equipped with an overhead CCD camera for global positioning.

In this experiment, the proposed internet control architecture was implemented as a TCP/IP application version, and the physical distance between the local site and the remote site was about 300Km, which was the same condition as that of computer simulations. The *Posture Estimator* was removed from the architecture so as to find out the exact characteristics of each control structure. Figure 8 shows the path error and the time difference between the two robots. In the experimental results of the internet control architecture-I, the path error was caused by the information loss of control commands. The information loss of control commands made the real robot path different from the virtual robot path. In the results by the internet control architecture-II, the time difference between the actions of the two robots in-

creased continuously, since internet time delay accumulated as time goes on. In the results by the internet control architecture-III, the path error and the time difference were quite small. The experimental results demonstrated the effectiveness and the applicability of the proposed internet control architecture-III as the simulation results did.

5 Conclusions

An internet-based personal robot with novel internet control architecture was developed. The proposed architecture was insensitive to internet time delay and guaranteed that the path error and the time difference between a real IPR and a virtual IPR could be reduced. Simulations and experimental results in a real internet environment demonstrated the effectiveness and the applicability of the proposed internet control architecture.

Acknowledgements

The authors would like to thank the support given by MIC(Ministry of Information and Communication), Korea, to the development of the internet-based personal robot.

References

- [1] K. Taylor and B. Dalton, "Issues in Internet Telerobotics," in *Int. Conf. on Field and Service Robotics*, Dec. 1997.
- [2] C. Sutter and J. Wiegley, "Desktop Teleoperation via the World Wide Web," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 654-659, May 1995.
- [3] R. Volpe, J. Balaram, T. Ohm and R. Ivlev, "The Rocky 7 Mars Rover Prototype," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1558-1564, Nov. 1996.
- [4] T. M. Chen and R. C. Luo, "Remote Supervisory Control of An Autonomous Mobile Robot Via World Wide Web," in *Proc. IEEE Int. Symposium on Industrial Electronics*, vol. 1, pp. ss60-ss64, July 1997.
- [5] K. Brady and T. J. Tarn, "Internet-Based Remote Teleoperation," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 65-70, May 1998.

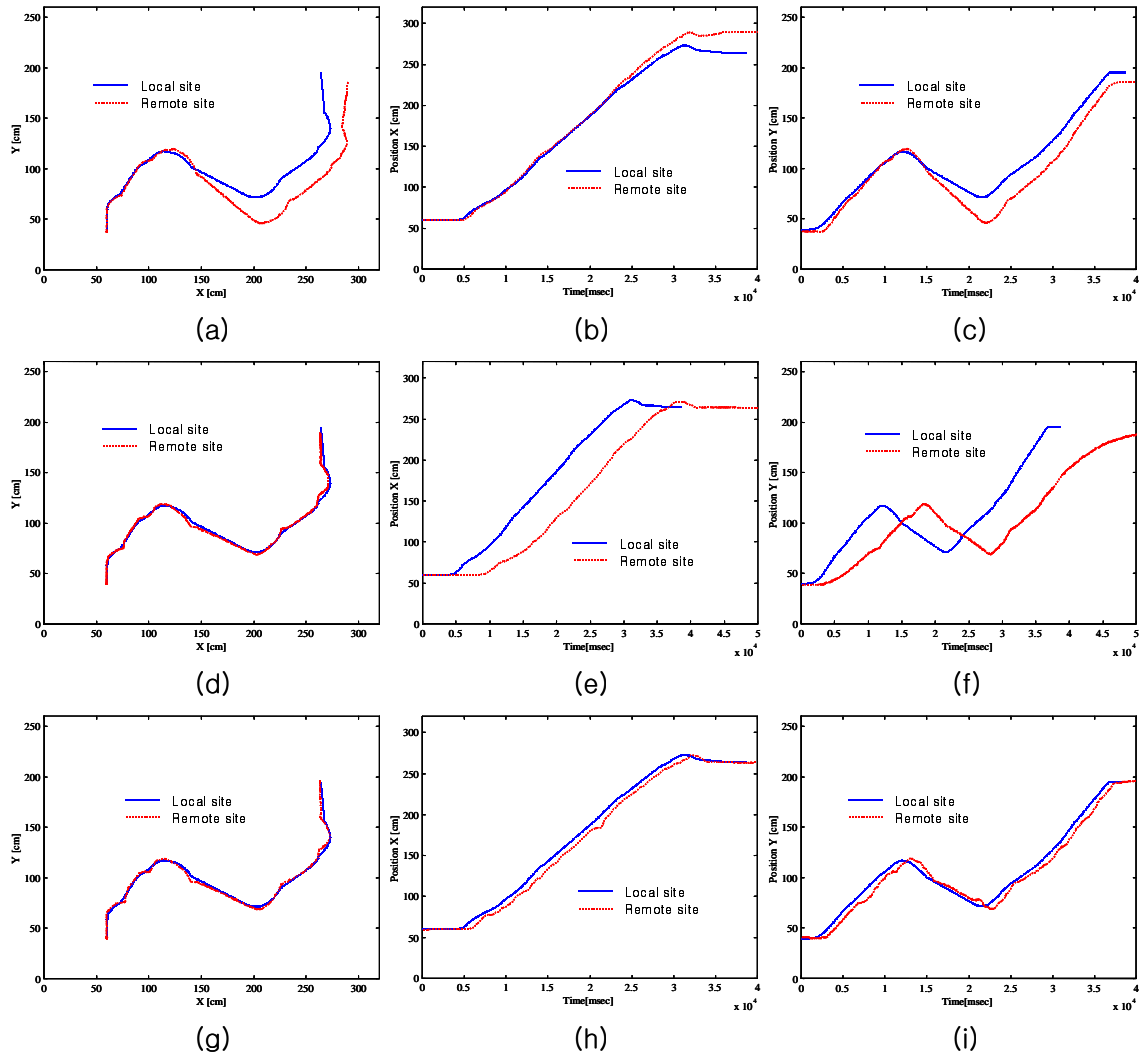


Figure 8: Experimental results. The path error and the time difference. (a),(b),(c) architecture-I. (d),(e),(f) architecture-II. (g),(h),(i) architecture-III.

- [6] E. Paulos and J. Canny, "Designing Personal Tele-embodiment," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 3173-3178, May 1998.
- [7] E.P.L. Aude, G.H.M.B. Caneiro, H. Serdeira, J.T.C. Silveira, M.F. Martins and E.P. Lopes, "CONTROLAB MUFA: A Multi-Level Fusion Architecture for Intelligent Navigation of a Telerobot," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 465-472, May 1999.
- [8] R. Oboe and P. Fiorini, "A Design and Control Environment for Internet-Based Telerobotics," *Int. Journal of Robotics Research*, vol. 17, no. 4, pp. 433-449, Apr. 1998.
- [9] K.-H. Han, S. Kim, Y.-J. Kim and J.-H. Kim, "Internet Control Architecture for Internet-Based Personal Robot," *Autonomous Robots Journal*, Kluwer Academic Publishers, vol. 10, no. 2, pp. 135-147, Mar. 2001.
- [10] K.-H. Han, S. Kim, Y.-J. Kim and J.-H. Kim, "Internet-Based Personal Robot System using Map-Based Localization," in *Proc. the 32nd Int. Sym. on Robotics*, Apr. 2001.
- [11] J.-H. Kim, K.-C. Kim, D.-H. Kim, Y.-J. Kim and P. Vadakkepat, "Path Planning and Role Selection Mechanism for Soccer Robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 3216-3221, May 1998.