Internet Control of Personal Robot between KAIST and UC Davis

Kuk-Hyun Han[1], Yong-Jae Kim[1], Jong-Hwan Kim[1] and Steve Hsia[2]

[1] 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, yjkim and johkim}@vivaldi.kaist.ac.kr

[2] Department of Electrical and Computer Engineering,
University of California, Davis, California
hsia@ece.ucdavis.edu

Abstract

This paper describes the implementation of an internet-based personal robot with three control modes, and experimental results on the remote control between KAIST, Korea and UC Davis, USA. The idea is to control a personal robot at the remote site (KAIST) by using a simulator provided at the local site (UC Davis). However, if the information of the current absolute position of the robot at its remote site cannot be estimated, the simulator may be useless. The absolute position of the robot can be determined by comparing a reference map with sensor information from sonar sensors and an electronic compass. A user can use three control modes - direct control mode, supervisory control mode and job scheduling mode, and monitor the current status of the robot using a graphic user interface (GUI) of the simulator implemented with Java.

1 Introduction

Telerobotics has made it possible to control a robot at a remote site. This has made it possible to control the motion of the robot at Mars directly [1]. There are many media for teleoperation - internet, telephone line, artificial satellite, etc. Especially, internet is universal and enables users to access any systems on the worldwide network cheaply. In these days, the tendency of electric home appliances is toward home networking. This means that one will be able to control all household electric appliances and monitor the status of one's house at the remote site.

Internet robotics that combines robot with internet made its appearance in the mid 90's, when internet had become universal. It has been observed that many researchers take interest in the area because of the merits of internet. The robot arm control system [2] through a Web browser was designed, and TeleGarden system [3] and Mars Pathfinder [1] 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. An intelligent telerobot [6] was 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 control scheme which enables users to control the motion of the robot continuously may not be easy. The direct control scheme [7] on the internet was proposed, but the modeling of the internet time delay was not adequate.

In [8] Kopacek stated that personal robot was the final application of the household robot which is a kind of service robot. The concept of a personal tele-embodiment was introduced, and the personal robot which enables a user to talk with other persons via internet was implemented in [9]. The robot which can charge its battery by itself was developed in [10]. The research on the interface between human and robots is also important. Hirukawa et al. proposed the standard human interface for internet control in [11], and Backes and Tharp implemented the Web interface for the internet control of the Rocky7 rover in [12]. The remote control interface which enables users to control the Humanoid robot via ISDN was designed in [13], and a collaborative teleoperation, that is, a many-one

control architecture via internet was described in [14].

This paper describes the implementation of an internet-based personal robot with three control modes, and experimental results on the remote control between KAIST, Korea and UC Davis, USA. The absolute pose of the robot can be calculated by using the map-based localization technique. Users at the local site can control the robot at the remote site using the developed graphic user interface (GUI) implemented with Java. Only the first user connected to the robot can use the control modes of the GUI, and other users can only monitor the status of the robot. There are three control modes in the developed GUI - direct control mode, supervisory control mode and job scheduling mode. The direct control mode is designed by using the internet control architecture already proposed in [15, 16]. The internet control architecture is insensitive to the internet time delay, and can reduce the path error and the time difference between the real robot at the remote site and the virtual robot of the simulator provided at the local site. In the supervisory control mode, if a user decides a goal position, the robot generates a moving path and moves toward the goal position autonomously. In the job scheduling mode, the user can create a job sequence of the personal robot. The internet-based personal robot is a fully autonomous mobile robot.

This paper is organized as follows. In Section 2, the developed internet-based personal robot system, the modeling of a mobile robot and the localization using sonar sensors, an electronic compass and odometric data are described. In Section 3, the three control modes and the developed GUI are presented. In Section 4, real experiments on the three control modes between KAIST and UC Davis are provided to show the usefulness and the applicability of the proposed internet-based personal robot system. 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 robots, 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 a vision camera, sonar sensors, a laser scanner, 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

Figure 1: The developed IPR, Mybot.

IPR using the simulator provided at the 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.

The developed IPR which is called Mybot is a fully autonomous mobile robot. It can navigate among the obstacles without any supervisor commands. It has its own battery. Its autonomy is for about 5 hours. It can also use the external power via the conventional wired plug. The IPR has a square body of size $45cm \times 52cm \times 75cm$ as shown in Figure 1. The weight is about $75Kg$. It has two drive wheels and two auxiliary off-centered casters. It consists of a personal computer (Pentium III 850MHz), a wireless LAN (Cisco, 11Mbps), an electronic compass module, a head with one vision color camera and two DC motors, a 12.1 inch TFT monitor, a speaker, a microphone, sonar sensors (12 pairs), a 12V 100Ah battery (5hr 80Ah), and two brushless DC servo motors (LG Industrial Systems, 200W). Its maximum acceleration rate is $0.52m/sec^2$ and maximum velocity is $2.09m/sec$. Considering safety, its maximum translational velocity is limited to $50cm/sec$ and the maximum angular velocity is limited to $45\deg/sec$. The CCD camera of the head part can rotate around a vertical and a horizontal axis under the command of the two 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 program anywhere.

For the localization purpose, it is equipped with a laser scanner. It gives range information in front of the sensor from $0\deg$ to $180\deg$ at the same time. Its angular resolution is programmable between $0.25\deg$ and $1.0\deg$. The measured data is transmitted by the RS232/RS422 serial communication. In the case of $180\deg$ scanning, it gives 733 bytes. The transmission

rate can be raised to 25Hz but the limitation of the serial port of the main computer restricts the rate to 5Hz. In this paper, considering the cost, sonar sensors and an electronic compass are used along with the odometric data for the localization.

2.2 Modeling

The modeling of the IPR is needed for the implementation of the simulator. A two fixed and two auxiliary off-centered orientable wheeled mobile robot with non-slipping and pure rolling is considered. The schematic figure of the developed robot is shown in Figure 2. In the figure, $O_W - X_W - Y_W$ is the global frame and $O_R - X_R - Y_R$ is the robot frame. The
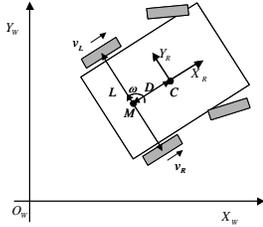

Figure 2: Modeling of Mybot.

translational velocity v_M and the angular velocity ω_M at the center of the wheel base M is obtained from left and right wheel velocities v_L, v_R as follows:

$$\begin{bmatrix} v_M \\ \omega_M \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix} \qquad (1)$$

where L is the wheel base length. v_R and v_L are obtained from the encoders of the two motors.

The robot pose $P_R^W = \begin{bmatrix} x_R^W, y_R^W, \theta_R^W \end{bmatrix}^T$ is defined at the center of the robot with respect to the global frame. The kinematic equation of the robot in the global frame is established as follows:

$$\begin{bmatrix} \dot{x}_R^W \\ \dot{y}_R^W \\ \dot{\theta}_R^W \end{bmatrix} = \begin{bmatrix} \cos\theta_R^W & -D\sin\theta_R^W \\ \sin\theta_R^W & D\cos\theta_R^W \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_M \\ \omega_M \end{bmatrix} \qquad (2)$$

where D is the offset distance between M and C.

2.3 Localization

The absolute position and the heading angle of the robot can be determined by comparing a reference map with a partial map built by using sensor information. A laser scanner is suitable for building an accurate map, but it is so expensive. The experimental

results on the localization using a laser scanner have already published in [17, 18].

Instead of using a laser scanner, in this paper, sonar sensors, an electronic compass and odometric data were used for the localization of the robot. While a laser scanner gives accurate data on the surrounding obstacles, sonar sensors give sparse data on them. It means that it is very difficult to build a partial map using sonar sensors. To solve this problem, the following estimation scheme for the robot pose is used:
1. Calculate the robot pose using odometric data, which can be obtained from the kinematic equation (2).
2. Correct the heading angle of the robot using the electronic compass data as follows:

$$\begin{aligned} e_\theta &= \theta_{compass} - \theta_R^W \\ \theta_R^W &= \theta_R^W + \alpha e_\theta \end{aligned}$$

where $\theta_{compass}$ is the measured heading angle from the compass module, e_θ the heading angle error, and α the coefficient of angle error correction rate. If α is too big, the heading angle is sensitive to a noise. If too small, the convergence speed of the angle is too slow. In this paper, 0.05 was used for α.
3. Correct the position of the robot using the sensing data from sonar sensors as follows:

$$\begin{aligned} \begin{bmatrix} e_x \\ e_y \end{bmatrix} &= k\sum_{i=1}^{N_s} \gamma_i (l_{s_i} - l_{c_i}) \begin{bmatrix} \cos(\theta_{n_i}) \\ \sin(\theta_{n_i}) \end{bmatrix} \\ \begin{bmatrix} x_R^W \\ y_R^W \end{bmatrix} &= \begin{bmatrix} x_R^W \\ y_R^W \end{bmatrix} + \beta \begin{bmatrix} e_x \\ e_y \end{bmatrix} \end{aligned}$$

where N_s is the number of sonar sensors, k the number of valid data, l_{s_i} the distance measured from the ith sonar sensor, l_{c_i} the calculated distance between the ith sonar sensor and the nearest object line in a reference map, θ_{n_i} the angle which is normal to the object line sensed from the ith sonar sensor, and γ_i, β are a variable coefficient and a static coefficient, respectively. If the measured data l_{s_i} is not valid, γ_i is 0. If valid, γ_i is 0.33. The validity condition of data from sonar sensors was used as $|l_{s_i} - l_{c_i}| < D_{max}$. $1.5m$ and 0.02 were used for D_{max} and β, respectively.

3 User Interface

3.1 Control Modes

A user can control the IPR at the remote site via internet using the developed GUI 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.

There are three control modes in the GUI implemented with Java - direct control mode, supervisory control mode and job scheduling mode. The user can control the motion of the robot directly using the direct control mode. In the supervisory control mode, if the user decides a goal position, the robot generates a moving path and moves toward the goal position autonomously. In the job scheduling mode, the user can create a job sequence for the personal robot.

In the direct control mode, because of time delay we have to compensate for the path error and the time difference between the real robot and the virtual robot, which increase as time goes on. In this paper, the internet control architecture proposed in [15, 16] is considered. The architecture is insensitive to the internet time delay, and can reduce the path error and the time difference between the real robot at the remote site and the virtual robot of the simulator provided at the local site. The performance of the architecture was proved in [15], and the experimental results using the developed IPR were described in [16].

In the supervisory control mode, the goal position and the final heading angle on the map are the control commands. After the goal position is determined, the shortest moving path is generated by using the "via" points which were already given in the map. The via points can be considered as all the points in the path through which the robot passes.

In the job scheduling mode, one job sequence can be determined by using the several jobs given in the menu of the GUI. In the GUI of Mybot, four jobs are given, such as MOVE, VOICE, MESSAGE and WAIT. MOVE means 'move toward the goal position,' VOICE 'speak the words provided by the user,' MESSAGE 'show the messages written by the user to the monitor,' and WAIT 'wait for the specified time.'

3.2 GUI

The developed GUI was implemented with Java 2 (JDK 1.3). So as to use the GUI, the user only needs to connect the personal robot using a Web browser. The Web browser connected to the robot will load the Java class file of the GUI automatically. The GUI loaded in the Web browser is shown in Figure 3. The GUI is linked with the robot using TCP/IP. Only the first user connected to the robot can use the control modes of the GUI, and other users can only monitor the status of the robot. In the center of the GUI, the
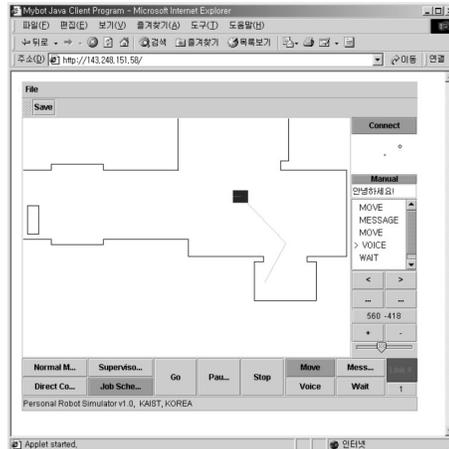

Figure 3: GUI implemented with Java.

map and the status of the robot are displayed. In the bottom of the GUI, there are four buttons for the selection of the control modes, and several buttons for generating the commands. On the right side of the GUI, the controls for the motion of the CCD camera mounted on the top of the robot and for editing the job sequence exists. The order and the status of connection can be verified using the right bottom of the GUI.

The NetMeeting program included in MS Windows was used for transmitting the image of the CCD camera and the sound.

4 Experiments

Experiments were performed with Mybot in the real internet environment. The remote site was KAIST, Korea, and the local site was UC Davis, USA as shown in Figure 4. Mybot was connected to inter-
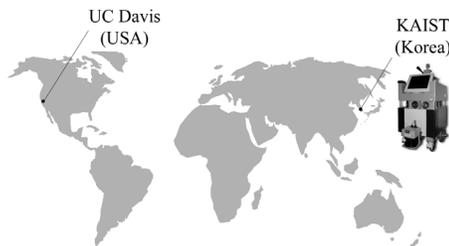

Figure 4: Local site and remote site.

net by using wireless LAN at the remote site, and the notebook computer (Pentium III $600MHz$, $192MB$) for a user was connected to internet by using LAN at

the local site. The number of internet nodes was 14 between KAIST and UC Davis, and the approximate round trip time via internet was about $310msec$.

4.1 Direct control

The user conducted a direct control operation using the direct control mode of the GUI loaded in a Web browser. Figure 5(a) shows the experimental result. The point S was the initial position of Mybot, and two obstacles (black circles) were put on the corridor. The result shows that Mybot followed the virtual robot path well in spite of the two obstacles.

4.2 Supervisory control

In this experiment, the user indicated only a goal position in the map of the GUI, and then Mybot generated a moving path and came to the position autonomously. The result is shown in Figure 5(b). After turning on the corner of the corridor, small path error could be found. This can be considered as the error of the localization of the robot pose.
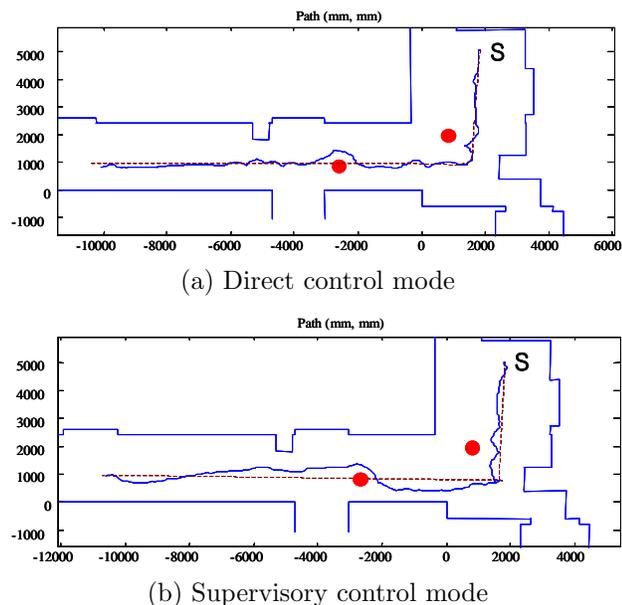

(a) Direct control mode

(b) Supervisory control mode

Figure 5: Experimental results, where the dashed line is the path of the virtual robot in the local site, and the solid line is the path of Mybot.

4.3 Job scheduling

In this experiment, a scenario was provided with a job sequence. The point I on the map shown in Figure 6(a) was the initial position of the robot, and the point V was the final position.

The scenario for the experiment was as follows: Mybot was to move from I toward III in front of a sofa (B). The user stopped the robot on the way to III, since he wanted to see some pictures on the wall (A). He could see the pictures using the camera control panel on the GUI. After that, Mybot moved to the point III and said "Hello!" to the person who sat on the sofa (B). After saying that, Mybot showed a message, "Please give me a paper!" After the person put it on the top of the robot and touched the OK switch, Mybot moved toward the point IV. Mybot said "Here is a message" and showed a message, "Please copy this paper!" After someone at (C) copied the paper and put them on its top, Mybot moved toward the final position V.

The experiment with this scenario was successful as shown in Figure 6(b). The experiment demonstrated the applicability of Mybot.
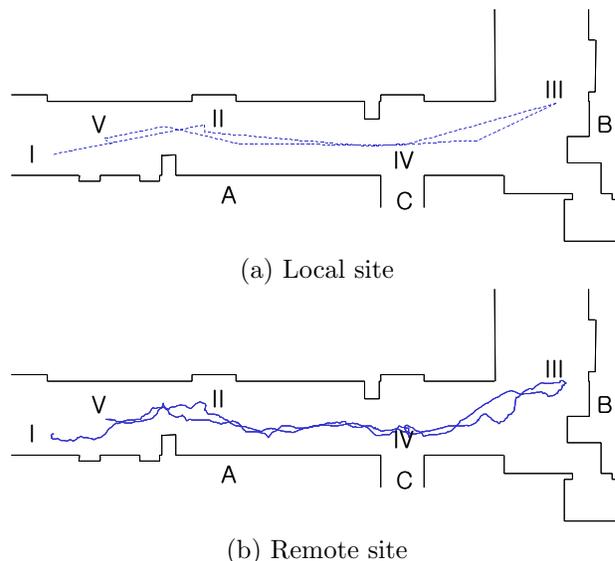

(a) Local site

(b) Remote site

Figure 6: Experimental results on the job scheduling.

5 Conclusions

This paper presented the implementation of the internet-based personal robot system with the three control modes. The developed IPR was a fully autonomous mobile robot. The absolute pose of the robot could be estimated by using sonar sensors and an electronic compass. A user could control the personal robot at the remote site using the developed GUI loaded in a Web browser, where three control modes - direct control mode, supervisory control mode and

job scheduling mode were provided. The GUI implemented with Java enabled the user to control and monitor the robot. The experimental results on the remote control between KAIST (Korea) and UC Davis (USA) demonstrated the usefulness and the applicability of the Mybot system.

References

[1] 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.

[2] K. Taylor and B. Dalton, "Issues in Internet Telerobotics," in *Int. Conf. on Field and Service Robotics*, Dec. 1997.

[3] C. Sutter and J. Wiegley, "Desktop Teleoperation via the World Wide Web," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 654-659, May 1995.

[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.

[6] 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.

[7] 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.

[8] P. Kopacek, "Service Robots Present Situation and Future Trends," in *Proc. of 5th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, pp. 29-33, June 1996.

[9] E. Paulos and J. Canny, "Designing Personal Tele-embodiment," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 3173-3178, May 1998.

[10] J. Ota and T. Arai, "A Hybrid Technique to Supply Indoor Service Robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 89-94, May 1998.

[11] H. Hirukawa, T. Matsui, and H. Onda, "Prototypes of Teleoperation Systems via a Standard Protocol with a Standard Human Interface," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 1028-1033, Apr. 1997.

[12] P. G. Backes and G. K. Tharp, "The Web Interface for Telescience (WITS)," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 411-417, Apr. 1997.

[13] H. Takanobu, E. Guglielmelli, H. Tabayashi, S. Narita, A. Takanishi, and P. Dario, "Waseda-SSSA Joint Research for Human and Humanoid Robot Interaction," in *Proc. 1st IEEE-RAS Int. Conf. on Humanoid Robots*, Sep. 2000.

[14] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, and G. Smith, "Collaborative Teleoperation via the Internet," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 2019-2024, Apr. 2000.

[15] 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.

[16] K.-H. Han, S. Kim, Y.-J. Kim, S.-E. Lee, and J.-H. Kim, "Implementation of Internet-Based Personal Robot with Internet Control Architecture," in *Proc. IEEE Int. conf. Robot. Automat.*, pp. 217-222, May 2001.

[17] J.-H. Kim, K.-H. Han, S. Kim, and Y.-J. Kim, "Internet-Based Personal Robot System using Map-Based Localization," in *Proc. the 32nd Int. Sym. on Robotics*, pp. 1282-1287, Apr. 2001.

[18] S. Kim and J.-H. Kim, "A Simultaneous Localization and Map Building using a Relative Map," in *Proc. the 32nd Int. Sym. on Robotics*, Apr. 2001.