Direct Internet Control Architecture for Personal Robot

Kuk-Hyun Han and Jong-Hwan Kim
Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology (KAIST),
373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
(Tel: +82-42-869-8048; Fax: +82-42-869-8877; Email:{khhan, johkim}@vivaldi.kaist.ac.kr)

Abstract: This paper introduces a direct internet control architecture for internet-based personal robot, 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 direct 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. Simulation results in the real internet environment demonstrate the effectiveness and applicability of the direct internet control architecture.

Keywords: internet control, mobile robot, remote control, internet interface, telerobotics

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.

This paper introduces a direct internet control architecture for internet-based personal robot, 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 [8], [9]. 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 direct 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 [8]. The graphic user interface(GUI) implemented with Java and the practical applications of the internet-based personal robot were described in [10], and the experimental results on the direct, supervisory, and job sequence control between KAIST, Korea and UC Davis, USA were presented in [11].

This paper is organized as follows. In Section 2, the modeling of a mobile robot and the characteristics of internet time delay are described. In Section 3, the direct internet control architecture is designed step by step. Section 4 presents the simulation results. Concluding remarks follow in Section 5.

2. Modeling

2.1. Mobile Robot

The modeling of a mobile robot is needed for the implementation of the simulator. Two 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 robot and the rotational velocity with respect to the center of robot. 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:

$$\dot{\mathbf{P}}_c = \begin{bmatrix} \dot{x_c} \\ \dot{y_c} \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}(\theta)\ \mathbf{u} \qquad (1)$$

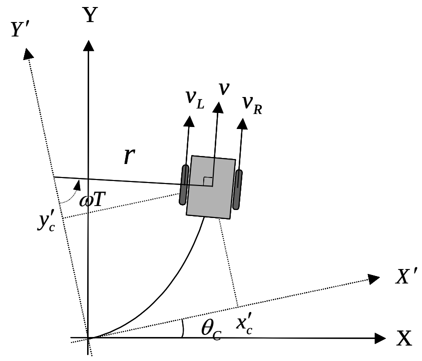


Fig. 1. Trajectory of a mobile robot

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

where v_R is the right wheel velocity, v_L is the left wheel velocity, and L is the distance between the two wheels. To derive the robot's position and angle, (1) should satisfy the following nonholonomic constraint:

$$\mathbf{G}\,\dot{\mathbf{P}}_c = [\sin\theta_c \quad -\cos\theta_c \quad 0] \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix}$$
$$= \dot{x}_c \sin\theta_c - \dot{y}_c \cos\theta_c = 0 \quad (3)$$

where $\mathbf{G}$ is a normal vector with respect to the side of wheel. This means that the instant heading direction is the same as the angle of the front side of the robot.

Control command $\mathbf{u}$ inputs to the robot every sampling time T where $t = kT,\ k = 0, 1, 2, \cdots$. The trajectory of the robot in a sampling time T is shown in Figure 1, where XY is a global coordinate and $X'Y'$ is a local coordinate rotated by θ_c at time $t = (k-1)T$, and the origin is the posture of robot at time $t = (k-1)T$.

2.2. Internet Time Delay

The internet time delay is characterized by the processing speed of nodes, the load of nodes, the connection bandwidth, the amount of data, the transmission speed, etc. Especially the dominating factors are the processing speed and the load of nodes. The internet time delay $T_d(k)$ can be described as follows:

$$\begin{aligned} T_d(k) &= \sum_{i=0}^{n} \left[\frac{l_i}{C} + t_i^R + t_i^L(k) + \frac{M}{b_i} \right] \\ &= \sum_{i=0}^{n} \left(\frac{l_i}{C} + t_i^R + \frac{M}{b_i} \right) + \sum_{i=0}^{n} t_i^L(k) \\ &= d_N + d_L(k) \end{aligned} \quad (4)$$

where l_i is the ith length of link, C the speed of light, t_i^R the routing speed of the ith node, $t_i^L(k)$ the delay caused by the ith node's load, M the amount of data, and b_i the bandwidth of the ith link. d_N is a term which is independent of time, and $d_L(k)$ is a time-dependent term. Because of the term $d_L(k)$ it is impossible to predict the internet time delay at every instant.

Since the internet time delay is affected by the number of nodes and the internet loads, it is variable and unpredictable so that a large internet time delay makes some control inputs distorted. 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, where the test function used is $y(t) = 5\sin(0.2\pi t) + 5$.

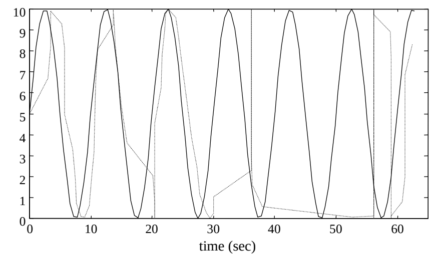


Fig. 2. Influence of the internet time delay

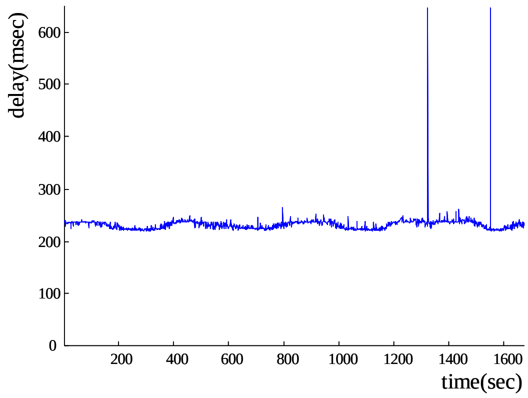


Fig. 3. Internet time delay for 1,675 seconds between KAIST, Korea and UC Daivs, USA

Figure 3 shows the internet time delay for 1,675 seconds between KAIST, Korea and UC Davis, USA. The data were gathered by using PING command. The minimum, average, and maximum time delay were 219 msec, 231 msec, and 646 msec, respectively, and the packet loss was 8%. 22 internet nodes were checked in the internet route between KAIST and UC Davis. The data were used in the simulations.

3. Direct Internet Control Architecture

A user can control the robot at the remote site through internet using a simulator provided at the local site. The user regards the status of the virtual robot at the local site as that of the real robot at the remote site. Since the user cannot recognize the environment of the remote site, it is expected that the real robot moves as the virtual robot does. However, 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 section, a direct internet control architecture is designed step by step to minimize the effect of internet time delay. The architecture is completed in three design steps, and its effectiveness is verified through simulations and experiments.

The direct internet control architecture-I consists of a user interface, simulator, virtual environment and posture estima-

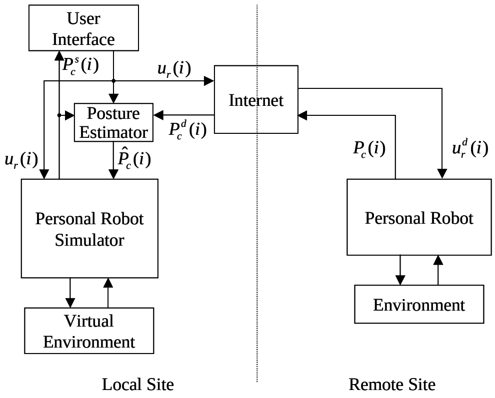


Fig. 4. Direct internet control architecture-I

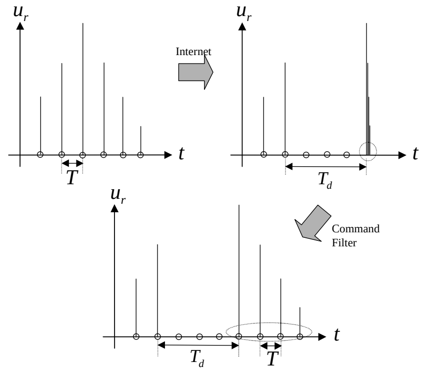


Fig. 6. Function of *command filter*

tor, which can be devised from the basic concept as shown in Figure 4. In the figure, $\mathbf{u}_r(i)$ is the ith control command $[v_r(i)\ \ \omega_r(i)]^T$ from a user, $\mathbf{u}_r^d(i)$ the ith control command passed through the internet, $\mathbf{P}_c(i)$ the ith robot posture, $\mathbf{P}_c^d(i)$ the ith robot posture passed through the internet, $\hat{\mathbf{P}}_c(i)$ the ith estimated posture, and $\mathbf{P}_c^s(i)$ the ith 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.

User Interface which can be implemented by Java, C++, etc., enables a user to control a remote robot. *Posture estimator* estimates the current posture of the virtual robot based on the feedback information of the real robot. *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 robot is working.

The direct 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 5 shows the direct 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 6. 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 [8]. The *command filter* receives a control command, and stores it in the *command queue*. The *command generator* pulls out the command from the *command queue* and outputs it at each sampling time T.

The direct 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 direct internet control architecture-III is finally designed.

The direct internet control architecture-III guarantees that the path error and the time difference between the real robot at the remote site and the virtual robot at the local site can be reduced. The architecture includes a *path generator* and a *path-following controller*. The *path generator* restores the moving path of the virtual robot. The *path-following con-*

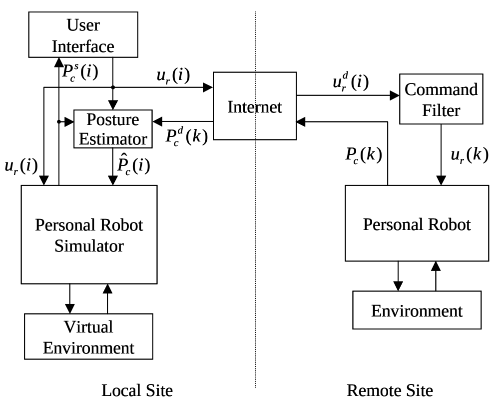


Fig. 5. Direct internet control architecture-II

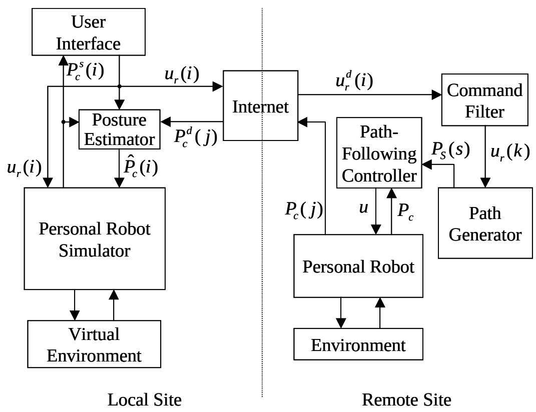
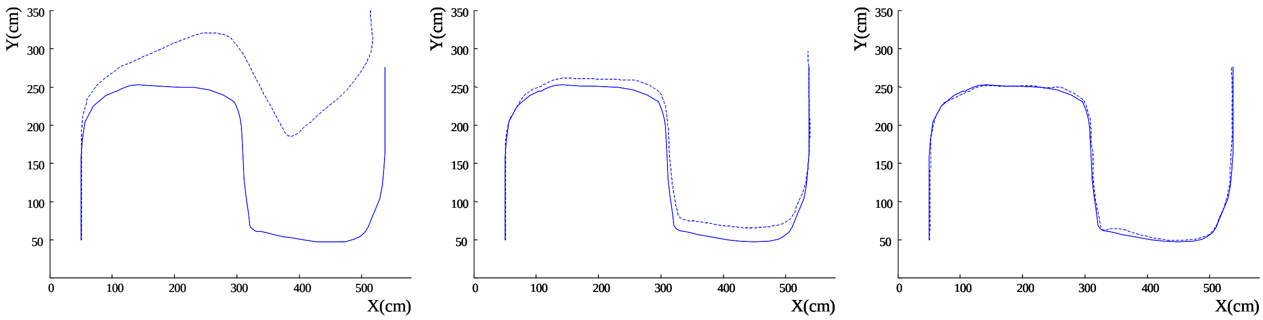
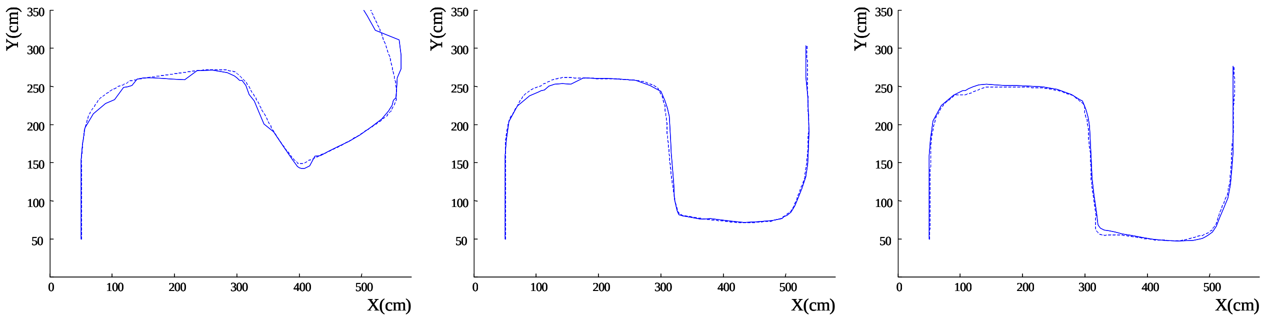


Fig. 7. Direct internet control architecture-III



(a) Architecture-I (b) Architecture-II (c) Architecture-III

Fig. 8. Path error between the robot path of the local site and the robot path of the remote site. *Posture estimator* was not used. The solid line is the robot path of the local site, and the dashed line is the robot path of the remote site.



(a) Architecture-I (b) Architecture-II (c) Architecture-III

Fig. 9. Path error between the robot path of the local site and the robot path of the remote site. *Posture estimator* was used. The solid line is the robot path of the local site, and the dashed line is the robot path of the remote site.

troller 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 7 shows the direct 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 jth robot posture to be fed back to the simulator. The *path-following controller* is implemented with the uni-vector field navigation method [12]. The uni-vector field makes the mobile robot converge to a desired path.

4. Simulations

Simulations were performed with the developed simulator using the real internet time delay data gathered between KAIST, Korea and UC Davis, USA. The local site and the remote site were set in the same machine, and a reflector was provided at another machine. The reflector returns the received data after the internet time delay shown in Figure 3.

Figure 8 shows the path errors of the robot path of the local site and the robot path of the remote site for three types of direct control architecture without *posture estimator*. 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-III, the path error was very small. Figure 9 shows the path errors of the robot path of the local site and the robot path of the remote site for three types of direct control architecture with *posture estimator*. In the results, by the architecture-III the path error was the smallest.

5. Conclusions

This paper has introduced a direct internet control architecture for internet-based personal robot, which is insensitive to the inherent internet time delay. The direct internet control architecture has guaranteed that the path error and the time difference between a real robot and a virtual robot could be reduced. Simulation results in the real internet environment have demonstrated the effectiveness and the applicability of the direct internet control architecture.

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] 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.

[9] 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.*, IEEE Press, pp. 217-222, May 2001.

[10] 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.

[11] K.-H. Han, Y.-J. Kim, J.-H. Kim, and S. Hsia, "Internet Control of Personal Robot between KAIST and UC Davis," in *Proc. IEEE Int. conf. Robot. Automat.*, May 2002.

[12] 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.