Mobility Improvement of an Internet-based Robot System Using the Position Prediction Simulator

Kang Hee Lee[1], Soo Hyun Kim[1] and Yoon Keun Kwak[1,#]

1 Department of Mechanical Engineering, KAIST, Daejeon, South Korea
Corresponding Author / Email: ykkwak@kaist.ac.kr,TEL: +82-42-869-3212, FAX: +82-42-869-5201

KEYWORDS : Internet-based robot, PPS (Position Prediction Simulator), VRML, EAI, Time delay

With the rapid growth of the Internet, the Internet-based robot has been realized by connecting off-line robot to the Internet. However, because the Internet is often irregular and unreliable, the varying time delay in data transmission is a significant problem for the construction of the Internet-based robot system. Thus, this paper is concerned with the development of an Internet-based robot system, which is insensitive to the Internet time delay. For this purpose, the PPS (Position Prediction Simulator) is suggested and implemented on the system. The PPS consists of two parts : the robot position prediction part and the projective virtual scene part. In the robot position prediction part, the robot position is predicted for more accurate operation of the mobile robot, based on the time at which the user's command reaches the robot system. The projective virtual scene part shows the 3D visual information of a remote site, which is obtained through image processing and position prediction. For the verification of this proposed PPS, the robot was moved to follow the planned path under the various network traffic conditions. The simulation and experimental results showed that the path error of the robot motion could be reduced using the developed PPS.

Manuscript received: February 4, 2004 / Accepted: November 29, 2004

1. Introduction

The Internet has been expanding rapidly over the last several years, and today anyone can easily connect to the Internet from anywhere. This vast new group of users has come to rely on the internet for a variety of purposes, including surfing the web, exchanging e-mails, and downloading files. Meanwhile, many companies have attempted to design new programs and applications in order to harness the power of the Internet in innovative new ways. For example, hopes of building home networking and crime prevention systems have been realized by connecting previous off-line equipment to the Internet. With these new applications, the Internet-based robot is also drawing attention.

The Internet-based robot has the same concept with the current tele-operation robot because both use the remote control. However, the Internet-based robot is operated using a public network and a web interface, while the tele-operation controls the robots using a private communication network and interface programs developed for experts. Therefore, unlike the tele-operation technology, the Internet-based robot is widely accessible to the public regardless of the user's terminal type and operating system. Then, the Internet-based system acquires the advantages in the economical network and popularity. However, it also has disadvantages in the uncertainty of data transmission and security. That is, the transmission time via the Internet varies according to the number of nodes and the network load and a large transmission time delay makes some control inputs distorted. Because of these disadvantages, the Internet-based robot has been used by the public in more restricted but focused fields like toy, entertainment, crime prevention, and guidance, while the tele-operation has usually been adopted in hazardous environments, such as space, deep sea conditions, and disaster areas. However, the disadvantages of the Internet-based robot system have been attempted to be solved by many researchers and its applicable areas are expanding.

The first Internet-based robot, the tele-excavation system, was proposed in the Mercury project[1] in 1994, and the UWA telerobot[2] was developed for building block structures by remote controlled robot arm in the same year. In the PumaPaint project[3], the operator ordered the remote robot to draw a picture on a real easel by clicking, holding, and dragging a mouse on the virtual canvas. In addition to these robot arm applications, a significant amount of research has also been conducted and is currently being carried out for the development of the Internet-based mobile robot system. Xavier[4], for example, accepts the commands to go around the different offices in a building and broadcasts the camera images as it travels. Grange et al.[5] have created the web-based robot system, which supports the tele-operation in unknown and unstructured environment. These types of mobile robots were applied to perform the function of a robotic tour guide in museums and enable people at another location to acquire and explore information[6, 7]. In these early Internet-based robot systems, the user inputted a task-based indirect command for the robot's motions. In the indirect control type, the user's task is divided into detailed commands and sequential execution; hence, in this control type, the path planning and the avoidance of obstacles are only determined by the autonomy of the robot.

In order to compensate the Internet time delay problem of the data transmission due to the uncertainty of the Internet, the researchers began to predict the position and state of the robot and provided it to the users. Kim et al.[8] proposed a predictive/preview display technique before the actual robot arm motion, and Kikuchi et al.[9] proposed a virtual time delay method and used a predictive display to capture a moving object with a bilateral arm robot. When the user obtains visual information from the real image of the remote site, there might be an arbitrarily large time delay as the size of the real image is fairly large. To reduce the size of the transmission data, the robot system using a 3D virtual model was developed. Elliot et al.[10] have studied the control of the two-link manipulator camera system and Belousov et al.[11] controlled the robot located in Russia from U.K. Chong et al.[12] proposed a predictive simulator to assist multiple operators for multi-telerobot cooperation. In these systems, path errors were decreased with the prediction of robot position, but the user could not know the information of the robot state when the user's command reached to the robot. Hence, direct control like an emergency stop was impossible.

This research is intended to develop an Internet-based robot system which is insensitive to the unpredictable Internet time delay problem in data transmission. For a novel compensation of the Internet time delay, the PPS (Position Prediction Simulator) is suggested and implemented to the client side of the Internet-based robot system. The user obtains the predicted position of the robot from the projective virtual scene of the PPS and operates the mobile robot using the direct control. This predicted position is based on the time at which the user's command reaches the robot system. Hence, the accurate control, such as stopping the moving robot near the target position became possible. In the mobile robot using the direct control, the mobility improvement means that the robot follows more accurately the planned path by the user command. For the verification of the mobility improvement using the proposed PPS, simulation and experiment were carried out under the various conditions

2. The Internet-based robot system

In the previous research, the OX quiz robot system[13] was developed. The developed Internet-based robot system is equipped with the OX quiz, as "The OX Quiz Show" broadcasted on Korean TV. For this research, the hardware of the OX quiz system was changed and partially reused. The developed Internet-based robot system was constructed as shown in the configuration of Fig. 1.

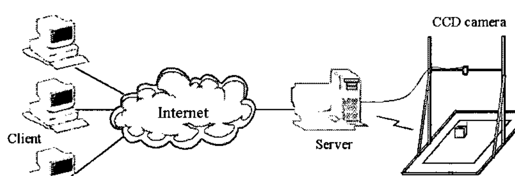
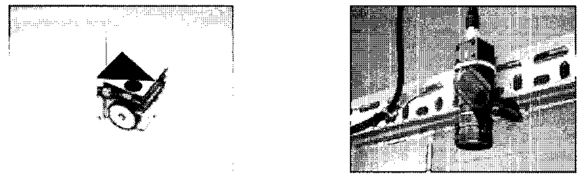


Fig. 1 Configuration of the Internet-based robot system

The robot and server are installed in the remote site, and the clients on the local site are connected to the server by the Internet. As shown in Fig. 2(a), two-wheeled soccer robot was used. This mobile robot has a cube body and its size is 75×75×75 mm. The robot's main controller was constructed with an 80C196KC microprocessor and a 29C256 flash ROM. The robot communicates with the server via the 433 MHz RF module at the speed of 34,800 bps. As shown in Fig. 2(b), a CCD camera, Sony XC-55 with a monochrome progressive scan, was used. The focal length of the camera lens is 8 mm. For the perspective view of the whole operational area in one frame, the camera is installed 1,085 mm above where the robot moves around, even though it can only watch the restricted area without pan-tilt motion. To construct the server, a PC with AMD XP 1800+ CPU, 512 Mbytes of memory, and Windows 2000 operating system was used. For the Internet connection, a 100 Mbps network interface card was installed. The frame grabber board on the server, a Matrox Meteor-II/Multi Channel, was attached to process the captured images. For communication with the robot, an RF control box was attached to the serial port in the server. The operational area that the robot moves on was made of a 640×480 mm size acrylic sheet covered with a grid paper. The size was determined considering the resolution of the CCD camera. The operational area was walled up by a 25mm high screen to prevent the robot from going out of it. Because the robot operates in a known and closed environment, exploration was not considered here.



(a) Mobile robot (b) CCD camera

Fig. 2 Hardware environment

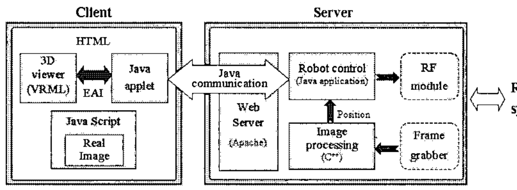


Fig. 3 The software architecture

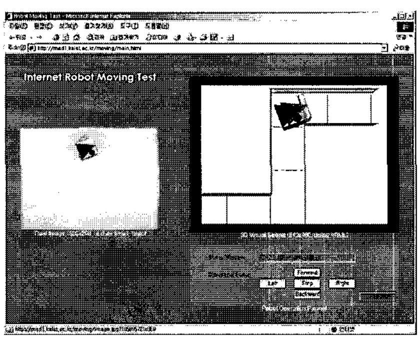


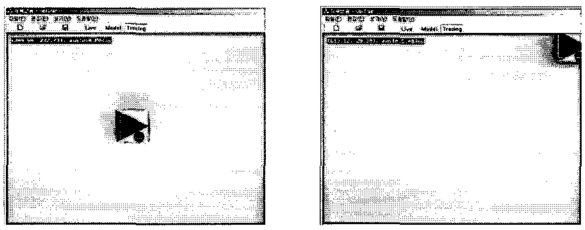
Fig. 4 User interface for the Internet-based robot system

The software architecture in the server and client is shown in Fig. 3. It is supposed that the system proposed here will be easily applicable to other Internet-based robot systems because the system was developed using common software and well-known programming languages. The operating programs in the server can be divided into three parts: web server, robot control, and image processing. In consideration of stability, the web server for the Internet was constructed using Apache HTTP (Hypertext Transfer Protocol) server software. The overall programming of the robot control was carried out using a Java application for smooth client-server communication with the Java applet program of the client. When the user commands the robot to move or stop in the client, that command is transmitted to the robot control part of the server and translated into the appointed packet format that is composed of a preamble, header, motor-speed, and checksum. Then, the command packet is transmitted to the RF control module that is attached to the serial port. For this serial communication, the "javax.comm" API (Application Program Interface) was used. In addition, another main role of the robot control part is to transmit the robot's position and orientation data obtained through image processing part to the client.

A widely available web interface was constructed that a broad range of users can access to operate the robot. As shown in Fig. 4, the user interface window consists of three sections, such as the real image on the left side, the robot operation section on the lower right side, and the PPS scene on the upper right side. The robot operation section was programmed using a Java applet. This allows the interface to be easily deployed and used in any Java enabled web browser. When the user clicks the 4 directional buttons (forward, backward, right, and left) and the stop button to operate the robot, this input is transmitted to the robot control part of the server. Simultaneously, the robot information obtained by the image processing is transmitted to the PPS by this robot operation section.

3. Acquisition of robot position data through image processing

The image processing part of the server was programmed using C^{++} language, and its main operation is to analyze the image and verify information about the robot's position and orientation with the pattern matching method. The size of pre-defined robot pattern, 60×60 pixels, is smaller than the real robot image as shown in Fig. 5(a). Therefore, the robot can be recognized in the edge position as shown in Fig. 5(b). Here, the robot's information includes the two-dimensional coordinates x and y and the steering angle θ, which are measured with respect to the coordinate system of Fig. 6. In this Internet-based robot system, the size of the operational area is 640×480 mm, and the size of the real image captured by CCD camera is 640×480 pixels, so the resolution is determined to be 1 mm/pixel. The captured image is shown in the client, being compressed into about 5 kbytes file in JPEG (Joint Photographic Experts Group) format with a size of 320×240 pixels.



(a) Pre-defined robot pattern (b) Recognition in boundary position

Fig. 5 Obtainment of robot position data through image processing

There are some errors in the robot position obtained through image processing because the fixed camera is susceptible to an optical illusion caused by the height of the robot as shown in Fig. 7(a). That is, the robot position obtained through image processing in Fig. 7(b) is different from the real position of the ground in Fig. 7(c). This error increases when the robot is located far from the center of the operational area.

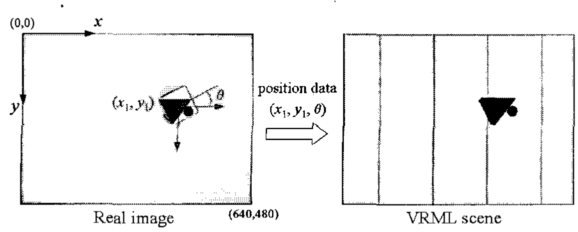
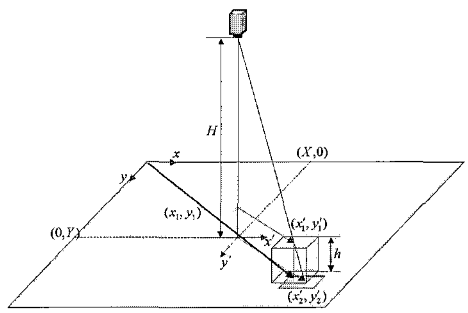
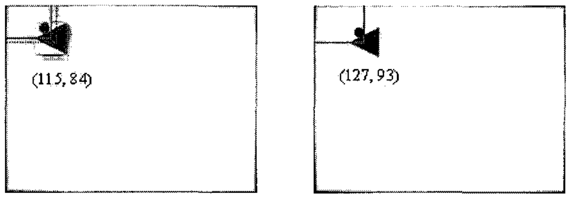


Fig.6 Relation of the robot position between the real image and 3D virtual scene



(a) The coordinate in the calibration



(b) Before calibration (c) After calibration

Fig. 7 Calibration of the robot position

Therefore, the robot position can be calibrated using the geometrical relation in Fig. 7(a) and expressed with the following equations:

$$x_1 = x_1' + X = \frac{(H-h)}{H} x_2' + X \quad (1)$$

$$y_1 = y_1' + Y = \frac{(H-h)}{H} y_2' + Y \quad (2)$$

where x_1, y_1 are the real robot positions and X, Y are the positions of CCD camera, which is fixed at the center of operational area in (x, y) coordinate. x'_2, y'_2 are derived from the image processing, and x'_1, y'_1 are derived after the calibration in (x', y') coordinate whose origin is located at the center of operational area. H is the distance from the ground to the camera, and h is the height of the robot. In this system, H is 1,085 mm, and h is 75 mm. For the verification of calibration, the measurement of the robot position was carried out in 16 different points of the operational area. The true robot position was measured using the grid paper on the operational area. As shown in Table 1, mean error and maximum error were decreased after the calibration.

Table 1 Robot position error in the image processing (before and after calibration)

Axis	Mean value of errors (mm)		Maximum error (mm)	
	Before calibration	After calibration	Before calibration	After calibration
x	10.4	1.0	17.5	2.8
y	8.9	2.1	15.5	4.4

4. PPS (Position Prediction Simulator)

Though the Internet is a public network and is not owned by a private company, the routing in OSI (Open System Interconnection) layer 3 using the IP address, which is the standard for the Internet, does not guarantee the communication QoS (Quality of Service). Hence, the Internet is often irregular and unreliable in data transmission, and it causes the varying time delay that is a significant problem for the construction of the Internet-based robot system. Therefore, the PPS is suggested in this research to compensate for this time delay problem of the Internet. The proposed simulator runs on the client side and shows the predicted position of the robot considering the time delay. The PPS is divided into two parts, including the robot position prediction and the projective virtual scene part.

4.1 Robot position prediction

In the robot position prediction part, the robot position is predicted based on the time at which the user's command reaches the robot system for more accurate operation. The kinematic relation of two-wheeled mobile robot can be expressed with equations as follows:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_r \\ \omega_r \end{pmatrix} \quad (3)$$

$$v_r = \frac{V_R + V_L}{2} = r\frac{\omega_R + \omega_L}{2} \quad (4)$$

$$\omega_r = \frac{V_R - V_L}{D} = r\frac{\omega_R - \omega_L}{D} \quad (5)$$

where v_r is the velocity, and ω_r is the angular velocity of the robot as shown in Fig. 8. V_R, V_L are the velocities of the right and left wheel of the robot, and ω_R, ω_L are the angular velocities of the right and left wheel of the robot. r is the radius of the robot wheel, and D is the distance between the two wheels. In the prediction of the robot position, the predicted time delay ΔT and the velocity of the robot were considered. Therefore, the predicted robot position after ΔT can be obtained from the following equations:

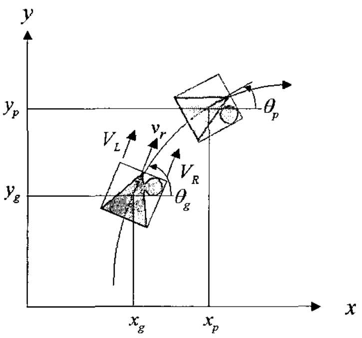


Fig. 8 The kinematic relation of two-wheeled mobile robot

- If, $V_R \neq V_L$

$$\theta_p = \theta_g + \omega_r \Delta T \quad (6)$$

$$x_p = x_g + \frac{v_r}{\omega_r}\left(\sin\theta_p - \sin\theta_g\right) \quad (7)$$

$$y_p = y_g + \frac{v_r}{\omega_r}\left(\cos\theta_g - \cos\theta_p\right) \quad (8)$$

- Else if, $V_R = V_L$

$$\theta_p = \theta_g \quad (9)$$

$$x_p = x_g + \cos\theta_g v_r \Delta T \quad (10)$$

$$y_p = y_g + \sin\theta_g v_r \Delta T \quad (11)$$

where x_p, y_p, θ_p are predicted robot positions, and x_g, y_g, θ_g are the given positions by the server as shown in Fig. 8. It was assumed that the velocity and the angular velocity of the robot would be time invariant values. The predicted time delay ΔT is a time variable and obtained from the following equation:

$$\Delta T_{(i)} = 2T_{t(i)} + T_p \quad (12)$$

$$T_{t(i)} = T_{t(i-1)} + \Delta T_{c(i-1)} - \Delta T_{s(i-1)} \quad (13)$$

The $\Delta T_{(i)}$ in the i-th step is derived from the round-trip transmission time delay between the client and server $2T_{t(i)}$, and the process time delay T_p. The one-way Internet transmission time delay $T_{t(i)}$ is updated from the previous delay $T_{t(i-1)}$, the interval of the system clock in the server $\Delta T_{s(i-1)}$, and the interval of the system clock in the client $\Delta T_{c(i-1)}$. The prediction between the client and server is carried out in each step as shown in Fig. 9.

The initial transmission time delay $T_{t(0)}$ is measured every 10 seconds. For this measurement, the round-trip communication time of the short characters is measured and half of this value is used. However, if the interval of the initial transmission time delay measurement is reduced, it will cause a heavy load on communication because of the large measurement packets. Since it is difficult to synchronize the system clock of the client with that of the server, only the difference between the interval values affects the new transmission time delay. On the other hand, if the transmission time delay is the same as the previous one, the interval value in the server and client will not be changed.

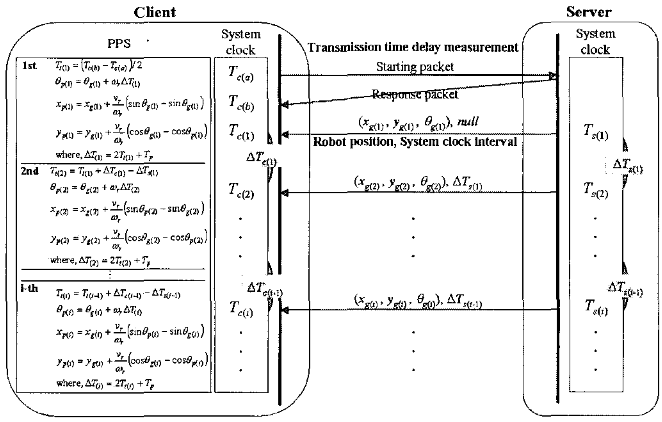
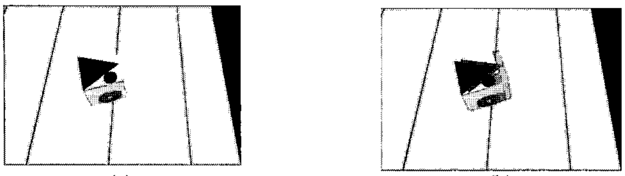


Fig. 9 Schematic diagram of position prediction

4.2 Projective virtual scene

In this study, the VRML (Virtual Reality Modeling Language) was chosen to implement the projective virtual scene of the PPS because this language is easy to embed in a web page and does not require the client to download another program. The VRML viewer can run as an applet within a web browser for easy access by an Internet user and also as a stand-alone application. The virtual environment is expressed with basic information of the robot shape and operational area, both of which are already transmitted to the client in the early connection. Moreover, this 3D virtual scene allows the user to get various viewpoints using features such as zoom (both in and out), rotation, and translation. For the communication between Java and the VRML, the EAI (External Authoring Interface) method was used with the "vrml.external" Java class because this method enables the Java program to modify the position and orientation data in the robot model that is pre-defined in the VRML program.

In the original virtual scene, the position of the mobile robot is obtained through image processing and then displayed as in Fig. 10(a). In order to reduce the network load, only the position and orientation data of the robot is transmitted from the server to the client when it is different from the previous data. Thus, the transmission time decreases, and a much faster update of the robot information in the remote site can be achieved. Nevertheless, even this reduced load of transmission data is still influenced by the delay problem when there is heavy data traffic in the Internet. Hence, the projective virtual scene implemented with the position prediction part shows the overlapped semi-transparent image for the predicted robot position as in Fig. 10(b).



(a) (b)

Fig. 10 Concept of the PPS (a) Original scene in given position (b) Projective virtual scene in predicted position

4.3 Process time delay of the system

As shown in equation 12, the time delay in the Internet-based robot system can be divided into two categories: the transmission time delay via the Internet and the process time delay of the system. To predict a robot position precisely, the process time delay and the Internet transmission time delay must be known. In this research, the process time delay of the overall system is defined considering the process time in the software, the RF communication time between the server and robot, the response time of the robot, and the sliding time of the robot. However, the RF communication time and the response time of the robot could not be measured individually. Hence, the process time delay of the overall system was calculated from the distance error of a robot stop test. The test is to stop the moving robot near the target line when there was no data traffic. For this measurement of the distance error, the client was connected to the server on the same PC. This stop test was carried out 15 times in each condition of the robot velocity, 60 and 120 mm/sec, and the process time delay was simply calculated from the mean value of the distance errors divided by the robot velocity. The results were 120 msec and 141 msec for the robot velocities of 60 mm/sec and 120 mm/sec respectively as shown in Table 2. For verification of this result, the robot stop test using the PPS that utilizes the obtained process time delay was conducted under the same conditions. In this case, the mean value of the distance errors was decreased to less than 2.1 mm.

Table 2 Measurement of the overall system process time delay using the robot stop test

Robot velocity (mm/sec)	Distance error without the PPS (mm)	Calculated process time (msec)	Distance error with the PPS (mm)
60	7.2	120	1.1
120	16.9	141	2.1

5. Simulation and experiment

5.1 Conditions of the robot motion test

For the verification of the proposed PPS, the robot was moved following the planned paths, which were straight-line and S-curve. The motion experimental conditions are shown in Fig. 11. In the straight-line test, after moving 150 mm, the mobile robot stops three times and pauses for one second per stop as shown in Fig. 11(a). For the S-curve motion, the user must command 6 times in Fig. 11(b). This motion experiment was carried out in the constant velocity, 60 and 120 mm/sec. The movement trajectory of the robot was stored in a data file by the server program. For the convenient operation, the virtual guideline can be displayed in the PPS. For example, the start and stop line were expressed in the straight-line test as shown in Fig. 12(b), and the virtual wall and command guideline were shown in the S-curve test as shown in Fig. 12(c).

A simulation was carried out under the same conditions as the experiment in order to compare with the experimental results. The simulation was performed using the MATLAB. The sequence of data transmission between the client and server was considered in the time-delayed condition. Assumptions in the simulation are as follows. The process time delay of the system was divided into two categories, the server process time and the robot response time. The server process time was assumed to be a mean value of the measured results, 69 msec, and it was varied 10 % using random generation because the image processing time was varied by the robot position. The robot response time was assumed to be a fixed value of 51 and 72 msec in each case of the robot velocity, 60 and 120 mm/sec. The robot velocity was considered as a fixed value, and the acceleration and deceleration of the robot motion were assumed to be negligible. The user could obtain visual information only from the 3D virtual scene of the robot.

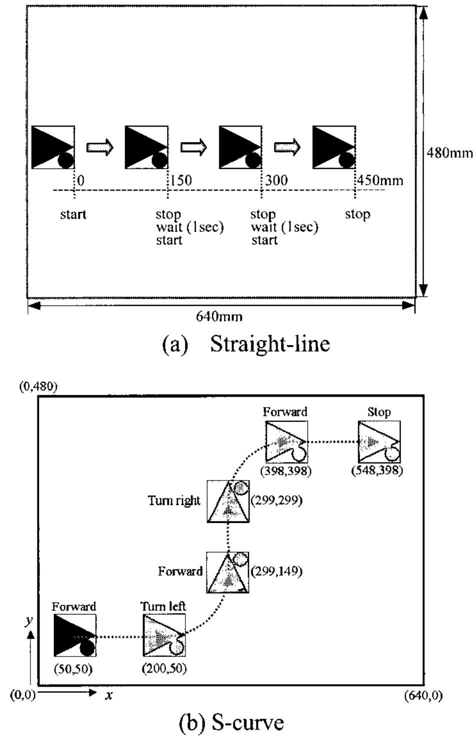
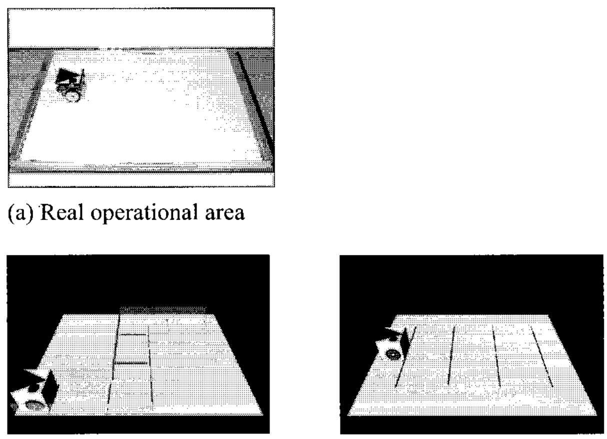


Fig. 11 Conditions of the robot motion test



(a) Real operational area

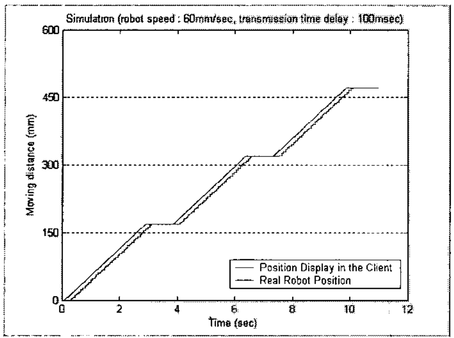
(b) Straight-line test (c) S-curve test

Fig. 12 Virtual environment for the robot motion test

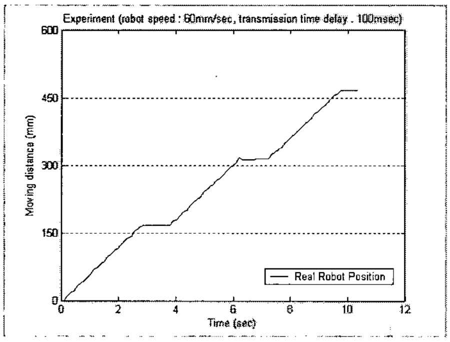
5.2 Motion test in fixed transmission time delay

For the motion test of the robot in fixed transmission time delay, a waiting function of 100 msec is added to the Java application program in the server. The test was carried out when the server and client were connected in the same LAN (Local Area Network) switch. The results of straight-line test without the PPS are shown in Fig. 13. The graph shows the movement trajectory of the robot in x-direction.

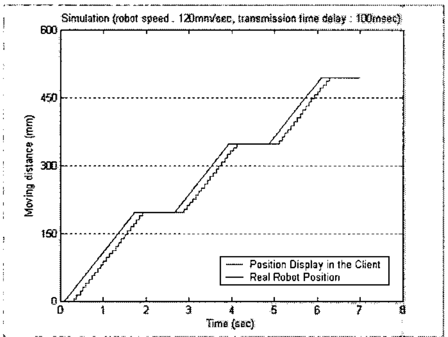
In the simulation results, the solid line shows the moving position of the robot, and the dotted line shows the position information on the client side. Without the PPS, the user obtained lagged information about the robot's position. Simulation and experimental results show the same trend. The results of straight-line test with the PPS are shown in Fig. 14. Using the PPS, the user obtained lead information of the robot position considering the time delay. The mean values of three distance errors between actual stop point of the moving robot and the target line in planned path are shown in Table 3.



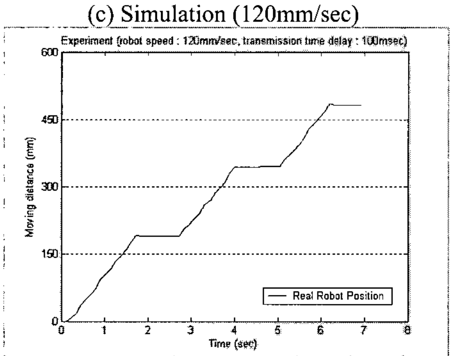
(a) Simulation (60mm/sec)



(b) Experiment (60mm/sec)

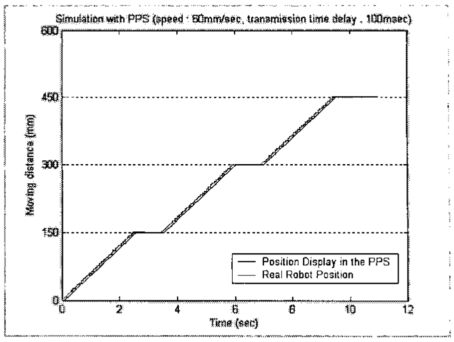


(c) Simulation (120mm/sec)

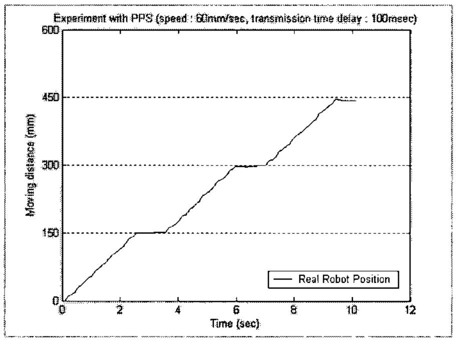


(d) Experiment (120mm/sec)

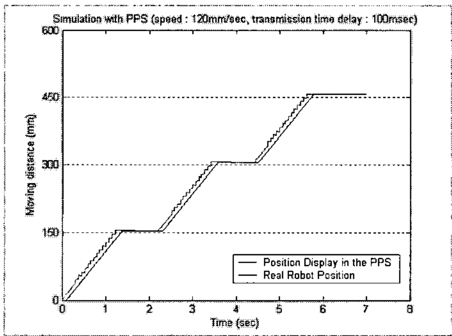
Fig. 13 Results in straight-line motion test without the PPS



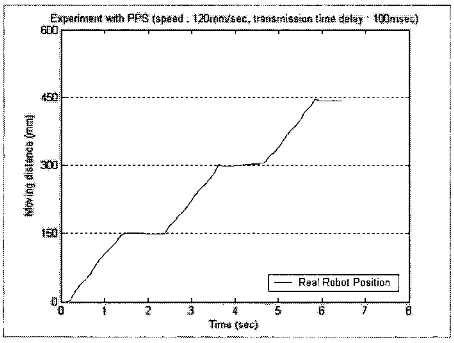
(a) Simulation (60mm/sec)



(b) Experiment (60mm/sec)



(c) Simulation (120mm/sec)



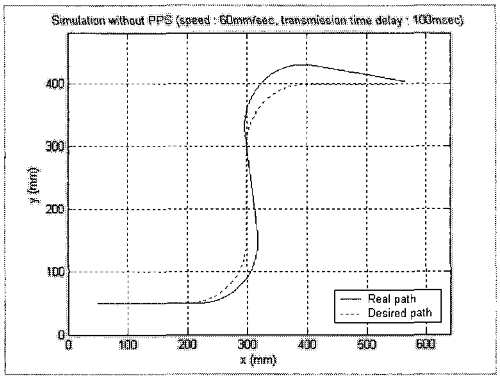
(d) Experiment (120mm/sec)

Fig. 14 Results in straight-line motion test with the PPS

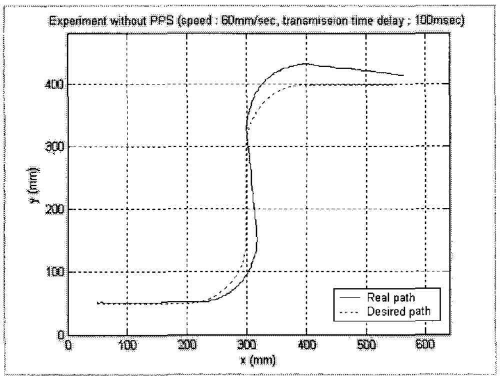
Table 3 Distance error results in straight-line motion test under fixed transmission time delay

Robot velocity (mm/sec)	Mean value of the distance errors in stop positions (mm)			
	without the PPS		with the PPS	
	Simul.	Exp.	Simul.	Exp.
60	22.3	15.7	0.3	2.7
120	44.3	38.7	5.3	2.7

The results of S-curve test without the PPS are shown in Fig. 15. The test was conducted with a robot velocity of 60 mm/sec. The solid line shows the moving position, and the dotted line shows the planned S-curve path. Simulation and experimental results show the same trend as in the straight-line test. The results of S-curve test with the PPS are shown in Fig. 16. As can be seen in the results, the robot implemented with the PPS follows more accurately the planned path than that without the PPS in both cases of straight-line and S-curve tests.



(a) Simulation (60mm/sec)



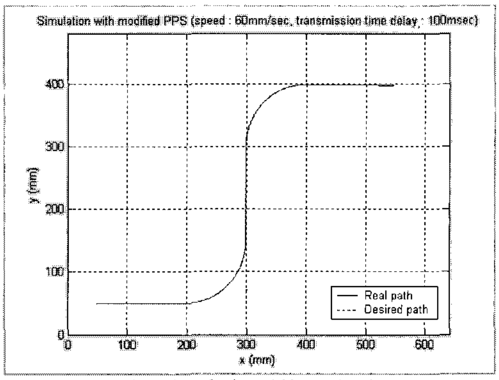
(b) Experiment (60mm/sec)

Fig. 15 Robot motion path in S-curve test without the PPS

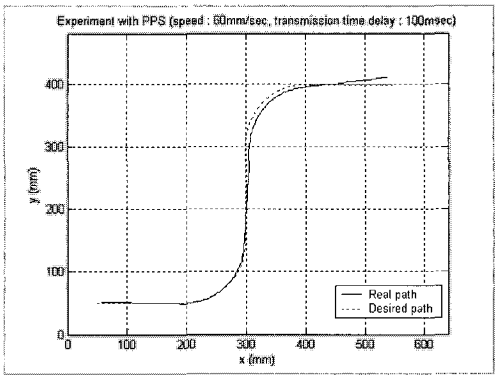
5.3 Stop test in variable transmission time delay

For the stop test of the moving robot in variable transmission time delay, the three test PCs including the server, client, and temporary PC were connected to the campus LAN and located in nearby buildings. For the test in the real network traffic conditions, which are set to be varied like the Internet, the Sniffer Pro 4.6 software was used in the server side. First, packets of 1,472 bytes were sent from the server to the temporary PC using the "ping" command, and 1,514 bytes live packets were captured by Sniffer in the server. To generate network traffic artificially, the captured packets are sent from the server to the temporary PC continuously using the Sniffer that provides the traffic generation function. The line utilization between the server and the temporary PC is varied using the interval regulation of the generated packets. This line utilization is related to the change of the percentage of the communication line usage between the server and the temporary PC as shown in Table 4.

A robot stop test was conducted which attempted to stop the moving robot with a velocity of 60 mm/sec, near the target line under various network traffic conditions. After the robot was stopped by the user command, the distance error between the actual stop point of the robot and the target line was measured. In this test, the error source would be the traffic generation amount and the usage of the PPS. When the line utilization varies between 0 and 75%, the mean values of the distance errors are shown in Fig. 17 and Table 5. Each value is an absolute mean value of 5 results of the test carried out under the same conditions. As presented clearly, in the same traffic generation condition, the distance error of the experiment that is implemented with the PPS is smaller than without the PPS, and the slope of the former is lower than that of the latter. The experimental results show that the distance error can be reduced more than 63% using the developed PPS, and this system is also insensitive to the Internet time delay in various network traffic environments.



(a) Simulation (60mm/sec)



(b) Experiment (60mm/sec)

Fig. 16 Robot motion path in S-curve test with the PPS

Table 4 Relation between the line utilization and network load in the traffic generation

Line utilization (%)	25	50	75
Number of generated packets: 1,514 bytes (number/sec)	2,041	4,078	6,022
Network load (Mbps)	24.7	49.5	73.0

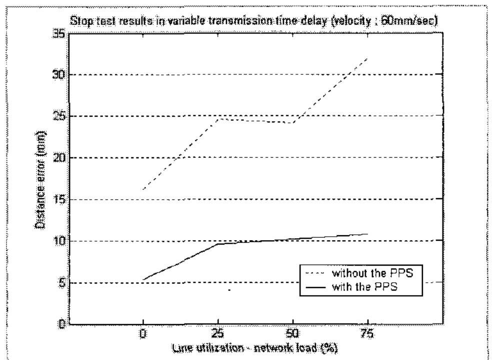


Fig. 17 Distance error results in the robot stop test (robot velocity: 60mm/sec)

Table 5 Distance error results in the robot stop test under various traffic condition

Line utilization (%)	Distance error (mm)		Reduction of the distance error (%)
	without the PPS	with the PPS	
0	16.2	5.4	67
25	24.6	9.6	61
50	24.2	10.2	58
75	32.0	10.8	66
Mean value	24.3	9.0	63

6. Conclusions

An Internet-based mobile robot system moving in obedience to the user's direct command was developed. In addition, a widely available web interface that a broad range of users can access and control the robot system was constructed. For the time delay compensation in data transmission via the Internet, the PPS (Position Prediction Simulator) was proposed. The PPS was added to the client side of the system to show the predicted robot position considering the process time in the system and the Internet transmission time. Through the projective virtual scene using the VRML, much faster update of the robot information in the remote site became achievable and the user can get the various viewpoints. For the verification of the proposed PPS, the tests to move the robot in a planned path and stop the moving robot near the target line were carried out. The simulation and experimental results showed that the robot mobility can be improved using the developed PPS, and this system is also insensitive to the Internet time delay in various network traffic environments.

Moreover, a simulator predicting the motion of the robot is needed because its effect becomes significant when the time delay exceeds the certain specific value. In addition, the modified PPS with display of the appearance of new obstacles by mixture of virtual scene and a real image should be developed. It is supposed that the developed system and the PPS will easily be applicable to other Internet-based robots such as home monitoring and vacuum cleaner robot, because the system proposed here is adopted with the common software and well-known programming languages.

ACKNOWLEDGEMENT

This work was supported by the Brain Korea 21 Project in 2004.

REFERENCES

1. Goldberg, K., Gentner, S., Sutter, C. and Wiegley, J., "The Mercury Project: A Feasibility Study for Internet Robots," IEEE Robotics & Automation Magazine, Vol. 7, No.1, pp. 35-40, 2000.
2. Taylor, K., Dalton, B. and Trevelyan, J., "Web-based telerobotics," Robotica, Vol. 17, pp. 49-58, 1999.
3. Stein, M. R., "Interactive Internet Artistry," IEEE Robotics & Automation Magazine, Vol. 7, No.2, pp. 28-32, 2000.
4. Simmons, R., Fernandez, J. L., Goodwin, R., Koenig, S. and O'Sullivan, J., "Lessons Learned from Xavier," IEEE Robotics & Automation Magazine, Vol. 7, No. 2, pp. 33-39, 2000.
5. Grange, S., Fong, T. and Baur, C., "Effective Vehicle Teleoperation on the World Wide Web," Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA 2000), pp. 2007-2012, 2000.
6. Schulz, D., Burgard, W., Fox, D., Thrun, S. and Cremers, A. B., "Web Interfaces for Mobile Robots in Public Places," IEEE Robotics & Automation Magazine, Vol. 7, No. 1, pp. 48-56, 2000.
7. Maeyama, S., Yuta, S. and Harada, A., "Remote Viewing on the Web using Multiple Mobile Robotic Avatars," Proc. of the 2001 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, Oct. 29-Nov. 03, pp. 637-642, 2001.
8. Kim, W. S. and Bejczy, A. K., "Demonstration of a High-Fidelity Predictive/Preview Display Technique for Telerobotic Servicing in Space," IEEE Transactions on Robotics and Automation, Vol. 9, No. 5, pp. 698-702, 1993.
9. Kikuchi, J., Takeo, K. and Kosuge, K., "Teleoperation system via computer network for dynamic environment," Proc. of IEEE International Conference on robotics and Automation, pp. 3534-3539, 1998.
10. Elliott, E. D. and Eagleson, R., "Web-Based Tele-Operated Systems Using EAI," Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, IEEE Intl. Conf., Vol. 1, pp. 749-754, 1997.
11. Belousov, I. R., Tan, J. and Clapworthy, G. J., "Teleoperation and Java3D Visualization of a Robot Manipulator over the World Wide Web," Information Visualization, IEEE Intl. Conf., pp. 543-548, 1999.
12. Chong, N. Y., Kotoku, T., Ohba, K., Sasaki, H., Komoriya, K. and Tanie, K., "Multioperator Teleoperation of Multirobot Systems with Time Delay: Part II-Testbed Description," Presence: Teleoperators and Virtual Environments, Vol. 11, No. 3, pp. 292-303, 2002.
13. Lee, K. H., Lee, Y. B., Kim, S. H. and Kwak, Y. K., "Distance Error Compensation of Internet-based Robot System Using Position Prediction Simulator," Journal of the Korean Society of Precision Engineering, Vol. 20, No.5, pp. 108-115, 2003.