Real-Time Network based Bilateral Robot Control System

Yutaka Uchimura
Kajima Corporation
Technical Research Institute
2-19-1 Tobitakyu, Chofu, JAPAN
yuchi@kajima.com

Kouhei Ohnishi
Keio University
Faculty of Science and Technology
Department of System Design Engineering
ohnishi@sd.keio.ac.jp

Abstract: This paper describes a bilateral robot control system that uses a real-time network and a prioritization scheme for data transmission via a network. The bilateral robot system is controlled by a new controller that improves impedance matching performance. The network system is implemented by a device named responsive processor which is was newly developed to enable real-time networking. Because the network capacity is limited, it is necessary to determine the priority order for the robot control data. The authors analyzed a control system with a time delay and assigned the priority order on the basis of robustness against time delay. To evaluate the priority assignment, experiments on a network-based bilateral robot were conducted. The experiment results confirmed the appropriateness of both the prioritization scheme and the order of the priority assignment.

Key Words: Computer network, Real-time network, Bilateral robot, Tele-operation

1 INTRODUCTION

Computer networks and the services enabled by them are spreading at an explosive rate and are now an indispensable part of our daily lives. Such innovative technologies and their application in the public domain are growing at an incredible speed, and the networks themselves have become part of an important social infrastructure. Advances in motion control technologies, including robot technologies, applications, which supply physical service in the real world, is possibly coming to appear. One such application is teleoperation via a network, and a promising example is the network-based teleoperation of construction machinery.

Teleoperation and similar applications require high-performance real-time data communication. In particular, if the data communication path comprises a feedback loop, this can directly affect the performance of the control system. This well-known problem has motivated a great deal of research. One well-studied approach is a method based on robust control theory, in which the variant time delay factor is modeled as an uncertainty [1] [2]. Another approach is to transform the system into a passive system by applying the scattering transformation to the communication channel with a time delay [3]. Despite being very powerful, these methods nonetheless require a discrete time delay that should ideally be deterministic from the viewpoint of control theory.

As many researchers have revealed, some network systems are not deterministic because of their media access methods. Such systems do not guarantee an upper bound for the time delay in the network transmission. For example, Ethernet, one of the most popular and widely used networks, uses the CSMA/CD arbitration mechanism, which generates a random time delay when a collision arises during transmission [4]. This indefinite delay makes the network nondeterministic.

To overcome this drawback, a novel network processor called a responsive processor was developed [5]. The responsive processor provides a genuine real-time network with two physical network paths, one for data transfer with short delay and the other for high-throughput data transfer. Because these paths are physically separated, the traffic on one path cannot be affected by that on the other, no matter how busy the traffic. This network structure enables the responsive processors to establish a deterministic network.

When using the network system for robot control, control data require a short delay. However, because the capacity (packet size) of the short-delay path is limited, these data must be prioritized. The issue is how to determine the priority order of the data.

This paper describes a network-based bilateral robot system that uses responsive processors. The authors have focused on the design of a controller that improves the impedance matching performance and a prioritization scheme for the data transmitted via the network. The priority order was determined by evaluating the robustness against the time delay. Applying this priority scheme, the authors used responsive processors for a real-time network in a bilateral robot system.

Section 2 describes the design scheme of the bilateral control system. Section 3 describes the responsive processor and the structure of the two physical network paths. Section 4 analyzes robustness against time delay and evaluates the priority assignment. Finally, Section 5 shows the results of the experiment.

2 BILATERAL ROBOT SYSTEM

The configuration of the robot system is shown in Fig. 1. The system consists of two robot manipulators with force feedback, which are connected via a network through responsive processors [6] [7]. Each manipulator has multiple degrees of freedom and employs a force sensor on the tip of its arm. When the operator moves the master manipulator (local manipulator), the slave manipulator (remote manipulator) makes the same motion. Once the slave manipulator touches an object, the operator feels the stiffness of the object through the manipulator.

Ideally, a teleoperation system should enable the operator to feel as though he is touching a remote object directly.

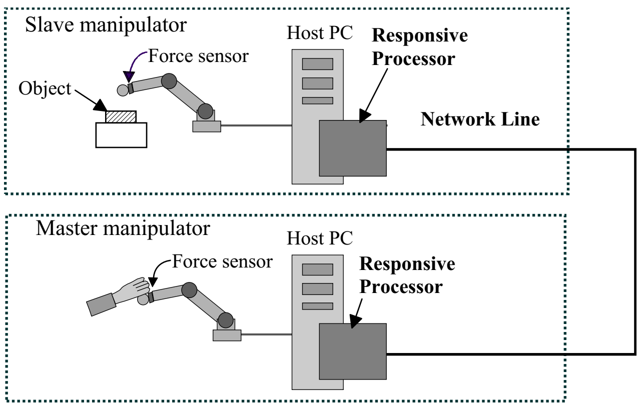

Fig. 1: Network-based bilateral robot system

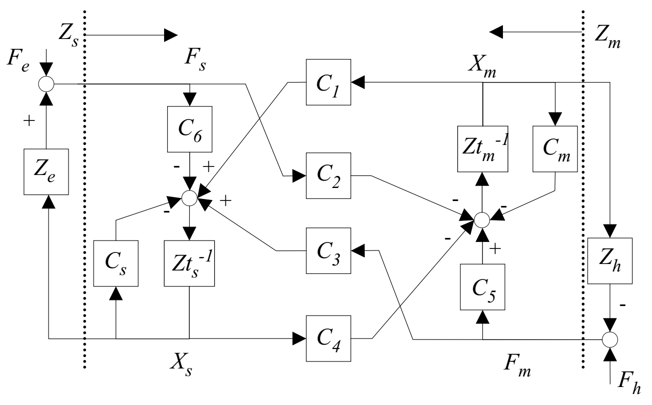

Fig. 2: Configuration of the modified controller

The degree of direct feel is usually measured in terms of transparency. The greater the sensation of direct touching, the higher the transparency. Many researchers have studied various methods of increasing transparency. The 4ch controller proposed by Lawrence [8] is a typical one, although the method does not include force feedback gains in the controller, which restricts the freedom to fine tune the transparency.

The control system shown in Fig. 2 was constructed by adding C_5 and C_6 to a conventional 4ch controller. In the figure, Z_t, Z_h and Z_e are the impedances of the manipulator, the operator, and the object, respectively. X is the position and F is the force. The subscripts m and s indicate the master side and the slave side. Z_m and Z_s are the impedances on the master side and the slave side, respectively, as determined from the relation between their inputs and outputs. Using a 2-port model with a hybrid matrix [9], we can formulate the input-output relations on both the master side and the slave side as in (1).

$$\begin{bmatrix} F_m(s) \\ X_m(s) \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} X_s(s) \\ -F_s(s) \end{bmatrix} \tag{1}$$

The hybrid matrices $H_{11} \sim H_{22}$ are obtained as shown in (2),

$$\begin{aligned} H_{11} &= D\{(Zt_m + C_m)(Zt_s + C_s) + C_1C_4\} \\ H_{12} &= -D\{(Zt_m + C_m)C_6 + C_1C_2\} \\ H_{21} &= D\{(Zt_s + C_s)C_5 - C_3C_4\} \\ H_{22} &= -D(C_5C_6 - C_2C_3) \end{aligned} \tag{2}$$

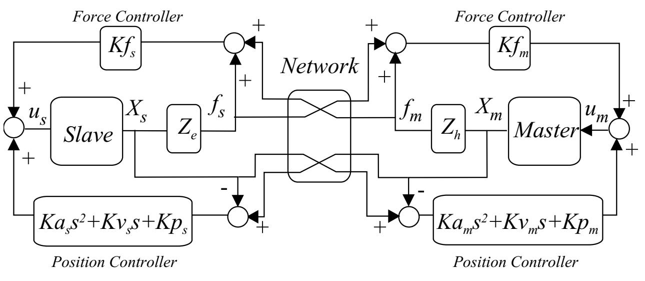

Fig. 3: Block diagram of the proposed controller

where

$$D = (C_1 + C_3 Zt_m + C_3 C_m)^{-1} \tag{3}$$

The conditions for ideal transparency are $H_{11} = 0$ and $H_{22} = 0$. Therefore, if (4) and (5) are true, master-side impedance Z_m matches Z_e as shown in (6), showing that ideal transparency has been achieved.

$$C_4 = -(Zt_m + C_m),\ C_1 = (Zt_s + C_s) \tag{4}$$

$$C_2 = C_3 = C_5 = C_6 \tag{5}$$

$$Z_m = -H_{12} Z_e H_{21}^{-1} = Z_e \tag{6}$$

The authors propose the following controller as one that satisfies the conditions in (4) and (5).

$$\begin{aligned} &C_2 = K_f,\ C_3 = K_f,\ C_5 = K_f,\ C_6 = K_f \\ &C_4 = -Zt_m - Ka_m s^2 - Kv_m s - Kp_m \\ &C_1 = -Zt_s - Ka_s s^2 - Kv_s s - Kp_s \end{aligned} \tag{7}$$

Zt_m and Zt_s in (7) include the dynamics of the manipulator, so model identification is required. The model of a manipulator generally includes a nonlinear component, so solving (7) requires calculation of very complicated dynamic models. To overcome this drawback, the authors incorporated a disturbance observer [10] in the manipulator to make it possible to assume the manipulator's dynamics to be a simple linear model. Since Zt_m and Zt_s are the inverse dynamics of the manipulators, therefore they supposed to be nonproper functions. The authors then designed an appropriate filter in the disturbance observer to make them proper functions.

The block diagram in Fig. 3 depicts the overall control system described above. The control input to the master manipulator, u_m, is obtained from (8).

$$u_m = Kf_m(F_m + F_s) + (Ka_m + Kv_m s + Kp_m)(X_s - X_m) \tag{8}$$

The network function part in Fig. 3 is made up of responsive processors. The following section describes the network system through the responsive processor.

3 RESPONSIVE PROCESSOR

The responsive processor is designed for parallel and distributed real-time control applications. Fig. 4 shows a responsive processor on the PCI form factor. The processor is equipped with a SPARC core MPU, I/O functions, and a network interface in an ASIC chip. Of these functions, the

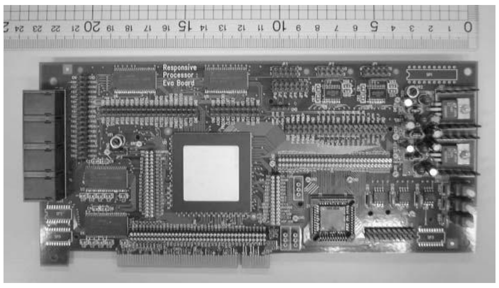

Fig. 4: Responsive processor board on the PCI form factor

network interface - hereafter called *responsive link* - plays a key role in real-time data communication.

The most significant feature of the responsive link is that it has two separate communication links. One of them, *'event link'*, is designed for short-latency packet transmission. The other, *'data link'*, is designed for wide-bandwidth data communication. Packets carried on the event link are called event packets, while those carried on the data link are called data packets. The size of event packets is fixed at 16 bytes, while that of data packets is fixed at 64 bytes. Event packets consist of a 4-byte header, an 8-byte payload, and a 4-byte trailer. Data packets consist of a 4-byte header, a 56-byte payload, and a 4-byte trailer.

Since these two links are physically separate, event packets should never be delayed, no matter how busy the data link. This is the principal design feature that maintains the real-time performance of the network system.

To implement the motion control application, authors have developed a message handler that manages packet transmission between the responsive processors and a host PC. The host PC acquires the sensor data and drives the actuators by generating control signals [7].

When designing the message handler, the principal scheme to enable priority-driven task scheduling was also maintained. Mailbox and DMA (Direct Memory Access) transfer are the main methods of data transfer between the responsive processors and the host PC. The mailbox generates interrupt signals and is therefore used to transfer urgent event packets. The DMA, on the other hand, is used mainly for transferring high-throughput data packets.

On the host PC side, priority-driven scheduling was implemented by using RT-Linux. A higher priority was assigned to the thread that handles event packets than to that for data packets. There is often a trade-off when choosing the event link or the data link. A general solution is to evaluate the priority of the transmitted data, then assign the event link for the higher-priority data. This is discussed in greater detail below.

4 PRIORITIZATION OF STATE VARIABLES

Because the event link can transmit packets with a shorter delay than the data link, all control data should be sent via the event link. However, all of the control data cannot be transmitted at the same time, since the payload of the event packets is too small. For example, to control a manipulator with three degrees of freedom requires data for three sets of position and force.

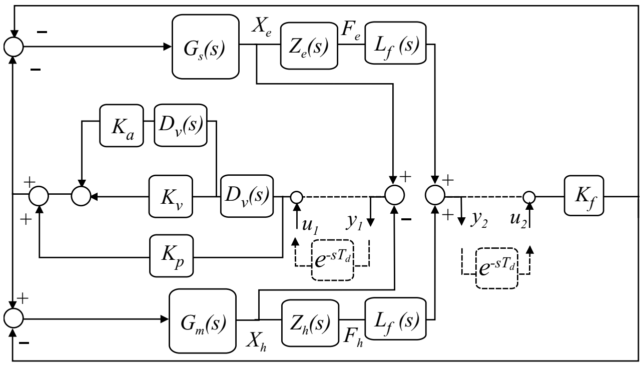

Fig. 5: Analysis model for evaluating robustness against time delay

If these data are sent in a 32-bit float format, the payload must be 24 bytes ($3 \times 2 \times 32 \div 8 = 24$). Therefore, one event packet with an 8-byte payload cannot accommodate all of the data in one transmission. Consequently, simultaneous transmission of position data and force data is technically difficult.

Because of this constraint, the priority of the state variables must be assigned. Variables that have higher priority should be transmitted via the event link. The remaining data are sent via the data link.

4.1 State variable priority assignment

The order of priority of the state variables was assigned by creating an analysis model based on the proposed controller described in Section 2. Using this model, the robustness of each state variable was evaluated against time delay. Fig. 5 shows the model used in the analysis.

To simplify the analysis, it was assumed that the controller gains on the master side and on the slave side were the same, and the controller part was wrapped up as shown in the middle of the figure. In the figure, $G_m(s)$ and $G_s(s)$ represent the master and the slave manipulators, respectively, $Z_h(s)$ and $Z_e(s)$ are the impedances of the operator and of the environment, respectively, $L_f(s)$ is a low-pass filter for the force sensor. The time delay is modeled as e^{-sT_d}, where T_d is the magnitude of the time delay. $D_v(s)$ is a pseudo-derivative used to obtain the acceleration from the velocity and the velocity from the position, respectably. Because the acceleration and the velocity is calculated using the velocity and the position value, it was assumed that both of the acceleration, the position and the velocity are delayed by the same amount. K_a, K_v, K_p and K_f are feedback gains with fixed values.

The time delay causes a phase lag whose magnitude increases as the frequency increases, whereas the gain is constant. Using this fact, robustness against time delay was evaluated as follows:

Step 1: The feedback gains were designed assuming no time delays.

Step 2: The open-loop transfer function from u_1 to y_1 in Fig. 5 was obtained. (as the position feedback loop).

Step 3: The open-loop transfer function from u_2 to y_2 in Fig. 5 was obtained. (as the force feedback loop).

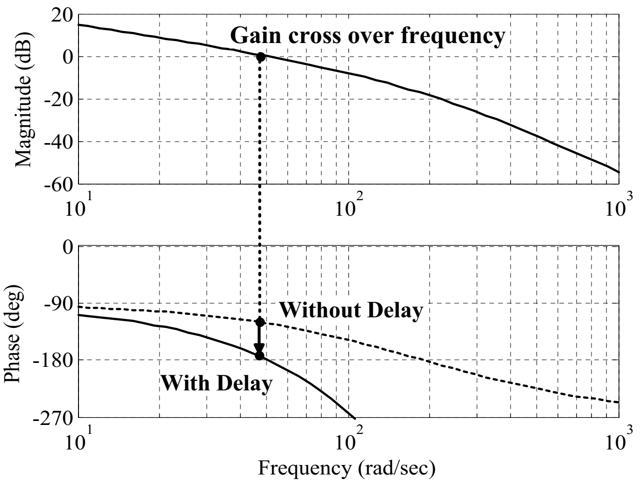

Fig. 6: Bode plots for with and without force data delay

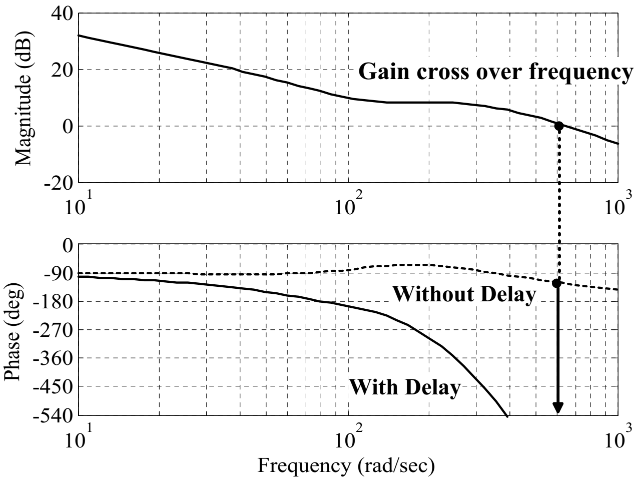

Fig. 7: Bode plots for with and without position data delay

Step 4: The phase margin and the gain crossover frequency were calculated for each feedback loop.

Step 5: The phase lag (due to the time delay) was added and the phase margin at the gain crossover frequency was evaluated.

Step 6: Robustness against time delay was then measured as the total phase margin.

Note that adding the phase lag to the open-loop transfer function is equivalent to adding time delay to the system.

4.2 Evaluation using a numerical model

The phase lag of the position control and the force control were analyzed by using a numerical model in consideration of the dynamics of the robot manipulators and the objects to be touched. The model is the followings:

$$G_m(s) = G_s(s) = \frac{1}{(s/300+1)s},\ L_f = \frac{1}{s/120+1},$$

$$D_v = \frac{s}{s/250+1},\ T_d = 0.02,\ Z_e = 1000,\ Z_h = 100,$$

$$K_p = 200,\ K_v = 0.2,\ K_f = 0.05 \qquad (9)$$

In the model, feedback gains were chosen such that the position feedback loop and the force feedback loop had the same phase margins without time delay. These numerical values are a typical example for the model system. Tests were conducted using various combinations of values, and qualitative results virtually the same as the above were obtained.

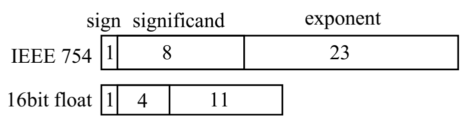

Fig. 9: 32bit float format (IEEE 754) and 16bit float format

Fig. 6 shows Bode plots comparing the cases with time delay and without time delay for force data. In the phase plot, the broken line shows the case without time delay and the solid line shows the case with time delay. Similarly, Fig. 7 shows Bode plots for the position data with and without time delay.

A comparison of the phase lags at the gain crossover frequency shows that the position control delay fell below -180 degrees, as shown in Fig. 7, meaning that the system transgresses the boundary of the stable area. Conversely, as shown in Fig. 6 , the phase lag for the force control remained above -180 degrees (positive phase margin). These results were obtained because the gain crossover frequency for the position control is higher than that for the force control. Since the phase lag due to the time delay is larger at higher frequencies, the position feedback is affected more by the time delay than is the force feedback. Consequently, the state variables associated with position feedback are less robust against time delay.

From these numerical results, it was decided that the position data should be assigned higher priority than the force data, and thus the position data should be sent via the event link.

5 EXPERIMENT RESULTS AND EVALUATION

5.1 Overview of experiment setup

Experiments were conducted to evaluate the effect due to time delay and to examine the validity of the priority assignment. Fig. 8 shows an overview of the experiment setup. The manipulator in the foreground is the master manipulator, which was maneuvered by the operator. The slave manipulator is in the background, and both manipulators have three degrees of freedom. The objects to be touched were a sponge block and a hard paper box. The two manipulators were connected via a real-time network through responsive processors. The sampling time of the control system was 1 ms, and data were transmitted every 1 ms.

Command data such as start and stop (16 bits) and position data for three axes (48 bits) were assigned to the event packets using the priority order described in the previous section. To enable three sets of position data to be carried, a 16-bit float format with a 4-bit significand and an 11-bit exponent was used instead of the standard IEEE 754 format, as shown in Fig. 9. The data packets contained force data for three axes. The implementation of the packets is shown in Fig. 10.

Before showing the results with time delay, it is useful as a reference to describe some results without time delay.

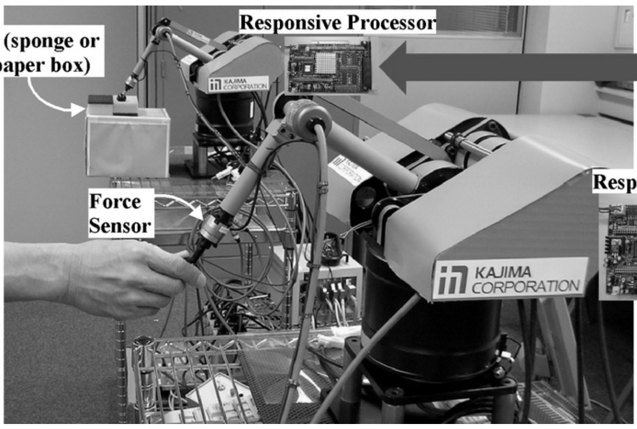

Fig. 8: Experiment setup

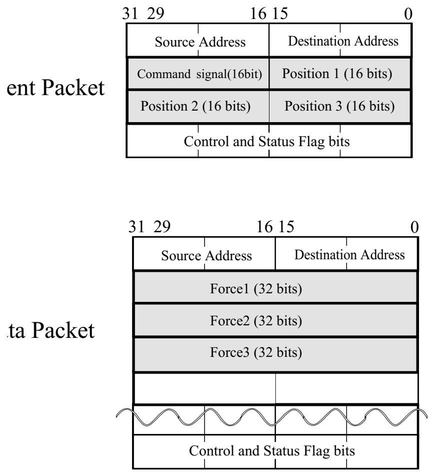

Fig. 10: Assignment and implementation for event and data packets

Fig. 11 and Fig. 12 show the results for two different objects without time delay. Fig. 11 shows those for a hard object (hard paper box), and Fig. 12 shows those for a soft object (sponge block). In each figure, the upper graph shows the force response in the direction of the z-axis (the gravity direction). The lower graph shows the position of the tip of the manipulator in the direction of the z-axis. The solid lines indicate the response of the master side, and the dashed lines indicate that of the slave side. In both cases, the position and force-tracking performances are more than sufficient.

The force response against the hard object and that against the soft object were similar, and their magnitudes reached approximately 15 N. However, the peak-to-peak position changes were much larger in the case of the sponge block than in that of the paper box. This result indicates that the operator feels that the sponge is apparently softer because of the larger deformation. These results show that the control algorithm functioned as desired and that the impedance matching between the master and the slave was satisfactory.

Next, let us look at the results with time delay. A time delay was simulated by software implementation using a FIFO buffer, which stored the current data in a buffer memory then output past (delayed) data.

Assuming that some of the variables are transmitted via the event link and that the rest are sent via the data link with time delay, tests were conducted using various combinations of variables and time delays. The results shown here are those for the cases when the time delay was set to 20 ms.

Fig. 13 shows the response against a paper box when the force data were delayed for 20 ms whereas the position data were not delayed. The figure shows a stable response, although slight fluctuations can be observed at the peak response. However, the feel of touch was almost as good as in the case without delay.

Fig. 14 shows the response against a paper box when the position data were delayed for 20 ms whereas the force data were not delayed. The figure shows an oscillating response near the peak value, which was induced by an unstable contact. This result shows that delaying the position value has a large effect on the control system.

These results indicate that the appropriate priority order is that of position first followed by force. The appropriateness of this order can also be seen from the point of view of bandwidth, in that the bandwidth of the position feedback is wider than that of the force feedback. This is because the bandwidth of the low-pass filter is so narrow that the magnitude of feedback gain is limited.

6 CONCLUSION

This paper has described a bilateral robot system that uses a real-time network system implemented on responsive processors. The system is driven by a newly developed controller, which improves impedance matching performance. The authors also propose a prioritization scheme for the control data transmitted via a real-time network system by the responsive processors. Owing to the limitation of the capacity of the short-latency path, a priority order was assigned for state variables.

The control system was analyzed with and without time delay, and the priority order was determined in consideration of the robustness of the control data against time delay. The prioritization scheme was applied to a bilateral system composed of two robot manipulators connected via responsive processors. Experiments with the bilateral robot system showed that the priority assignment was effective and thus supported the prioritization scheme.

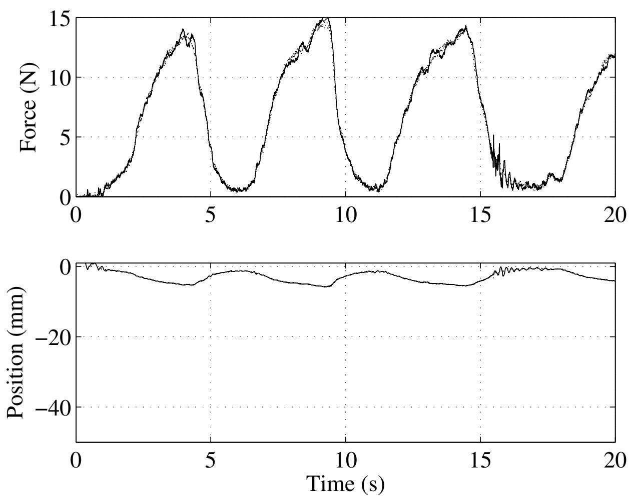

Fig. 11: Force and position tracking against a hard object

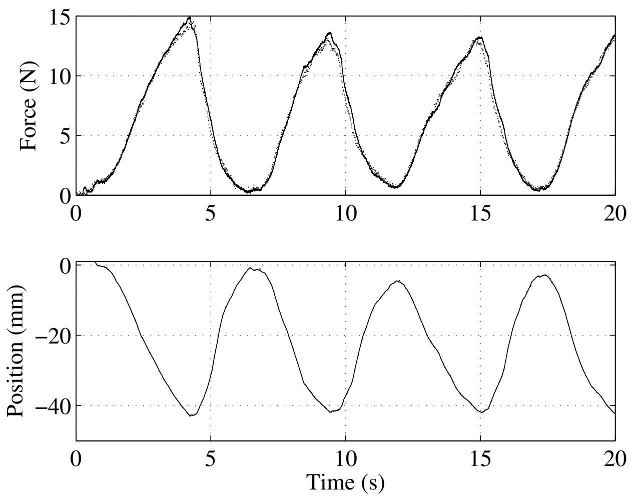

Fig. 12: Force and position tracking against a soft object

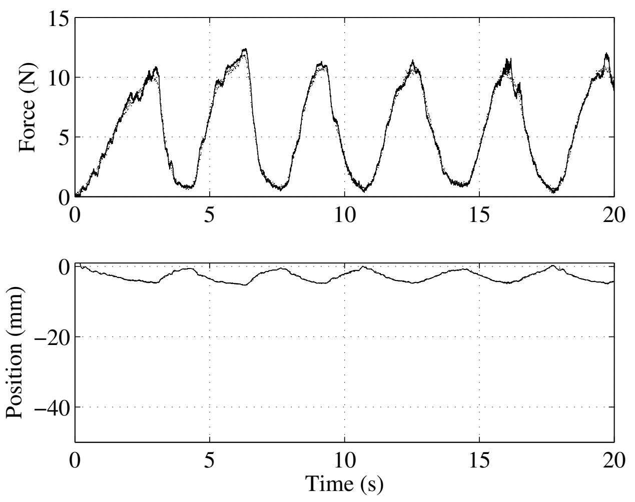

Fig. 13: Response with time delay of force

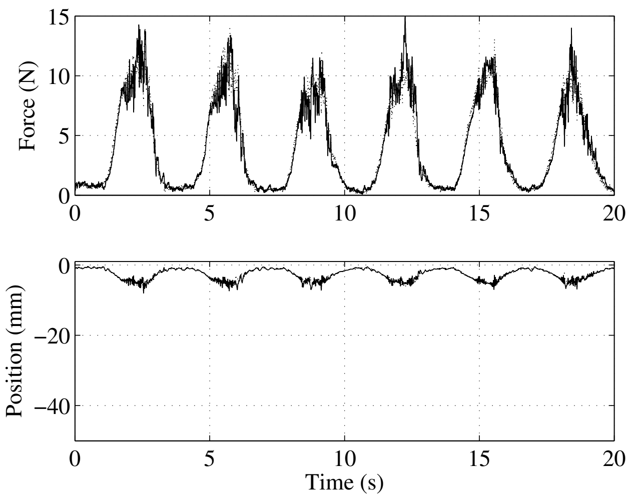

Fig. 14: Response with time delay of position and velocity

ACKNOWLEDGMENT

This study was performed through Special Coordination Funds of the Ministry of Education, Culture, Sports, Science and Technology of the Japanese Government.

References

[1] W.R. Ferrel, "Remote manipulation with transmission delay", IEEE Trans. on Human Factors in Electronics (HFE-6), pp.24-32, September (1965)

[2] G. Leung, B. Francis, J. Apkarian, "Bilateral Controller for Teleoperators with Time Delay via μ-Synthesis", IEEE Trans. on Robotics and Automation, vol. 11, no. 1, pp. 105–116, (1997)

[3] R. J. Anderson, M. W. Spong, "Bilateral Control of Teleoperators with Time Delay," IEEE Trans. on Automatic Control, Vol.34, No.5, pp. 494-501, (1989)

[4] J.D. Wheels, "Process control communications: Token bus, CSMA/CD, or token ring ?", ISA Trans., Vol. 32, pp. 193-198, (1993)

[5] N. Yamasaki, "Responsive Processor for Parallel / Distributed Real-Time Control", Proc. International Conference on Intelligent Robots and Systems, pp. 1238-1244, (2001)

[6] Y. Uchimura, T. Yakoh, K. Ohnishi, "Bilateral Robot System on the Real Time Network Structure", Proc. IEEE Conf. on Advanced Motion Control, pp. 63-68, (2002)

[7] Y. Uchimura, T.Yakoh, N.Yamasaki, K.Ohnishi, "Real-time Network System by Responsive Processor and Its Application to Bilateral Robot Control", Proc. IEEE Conf. of the Industrial Electronics Society (IECON 2003), pp. 1209-1214, (2003)

[8] D.A. Lawrence, "Stability and Transparency in Bilateral Teleoperation", IEEE Trans. on Robotics and Automation, Vol. 9, No. 5, pp. 624-637, (1993)

[9] B. Hannaford, "A Design Framework for Teleoperators with Kinesthetic Feedback", IEEE Trans. on Robotics and Automation, Vol. 5, No. 4, pp. 426-434, (1989)

[10] S.Komada, M.Ishida, K.Ohnishi, T.Hori,"Disturbance Observer-Based Motion Control of Direct Drive Motors",IEEE Transaction on Energy Conversion, Vol.6, No.3, pp. 553-559, (1991)