



# **An Auto-Tuning Scheme of PID Controller Based on a Partial Model Matching Method for an IP Network-based Small DC Motor Speed Control System**

Kenshi Matsuo

Graduate School of Engineering and Resource Science, Akita University, Akita, Japan

**ABSTRACT:** In Internet Protocol (IP) network-based control systems, the control performance is degraded owing to time delays and its jitter over the networks. Hence schemes to overcome the issue are required. In this paper, the author proposes an auto-tuning scheme based on a partial model matching method that tunes PID controllers for speed control of a small DC motor through IP networks with varying time delays. For confirming the effectiveness, speed control experiments are done after the controller of the system is auto-tuned by the proposed scheme under several emulated IP network conditions. As a result, the responses obtained by the experiments show that the proposed scheme is effective for the systems.

**KEYWORDS:** DC motor, IP network, time delay, PID controller, partial model matching method

## **I. INTRODUCTION**

Recently, many applications using the Internet have been developed in various fields. As one of the applications, the development of Internet Protocol (IP) network-based control systems gains more attention (e.g., [1], [2]).

Because the cost of hardware like hubs, routers, network interface cards, etc. is lower, it has been easy to construct IP network-based control systems connected among actuators, sensors and controllers over the networks. However, the control performance is degraded due to varying time delays such as constant time delay and its jitter over the network. Therefore, schemes to overcome the issue are required. A simple scheme is to tune a proportional-integral-derivative (PID) or PI controller properly and automatically according to network conditions. For example, [1] has proposed a scheme that auto-tunes a PI controller. Specifically, this scheme is as follows. It is assumed that varying time delays behave like exponential distributions, and the varying delays are characterised. Then the PI controller is tuned automatically according to the present delays, where the PI parameters are decided such as minimizing an evaluation function with respect to each delay beforehand. However, task for the tuning is often much time-consuming one.

In this paper, another scheme that auto-tunes PID controllers according to varying time delays is proposed. The proposed scheme is that first the PID controller is tuned using a partial model matching method by Kitamori [3], which can automatically choose the PID parameters according to time delays, and then the integral parameter is re-tuned with respect to its jitter. The advantage using the matching method is that proper PID parameters are automatically obtained if the transfer function of the plant model is known. That is, acquiring the value of parameters tuned according to each time delay in advance is not needed. However, the tuning method is still insufficient against jitter. Hence, to overcome the jitter issue, the relationship between the minimal time of varying time delays and the rise time of the step response are examined, where the responses are obtained by control experiment after tuning the controller based on the matching method according to each time delay. Then, based on the relationship to characterize the jitter, a method that re-tunes the integral parameter is developed. Thus, the author proposes an auto-tuning scheme including these two methods.

Finally, an IP network-based speed control system of an actual small DC motor, which is chosen because of a typical servomotor having a simple structure, is constructed, and after applying the proposed scheme, speed control experiments are carried out. From the step responses obtained by the experiments, it is confirmed whether the proposed scheme is effective against IP networks with varying time delays.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

## II. IP NETWORK-BASED CONTROL SYSTEM

Fig. 1 shows an IP network-based speed control system for a small DC motor. The rated voltage, current, power and speed of an actual small DC motor (Sanyo Denki R301T-011) are 24V, 1.25A, 11W and 3000min<sup>-1</sup> respectively. The motor is with an inertial load with 2.5 × 10<sup>-5</sup> kg · m<sup>2</sup>.  $V(s)$ ,  $\Omega_r(s)$ ,  $\Omega_m(s)$  and  $\Omega_e(s)$  are an applied voltage [V] to the motor, a reference speed [min<sup>-1</sup>], a rotor speed [min<sup>-1</sup>], and the error between the reference and the rotor speed, i.e.,  $\Omega_e(s) = \Omega_r(s) - \Omega_m(s)$ , respectively.  $G_m(s)$  is the transfer function of the DC motor. A PID controller  $G_{PID}(s)$  is denoted as

$$G_{PID}(s) = K_p + \frac{K_I}{s} + K_D s = \frac{K_I + K_p s + K_D s^2}{s}, \quad (1)$$

where  $K_p$ ,  $K_I$  and  $K_D$  are the proportional, integral and derivative parameters respectively. In actual experiments described in later section, the derivative term uses not only  $K_D s$ , but also  $K_D s / (1 + \gamma T_D s)$ , where  $\gamma = 5$  and  $T_D = K_D / K_p$ . Both sampling time of each computer to operate the controller and the plant in the figure are 0.001s = 1 ms. Each time delay of signals of  $V(s)$  and  $\Omega_m(s)$  through the IP network is expressed as  $L_1$  [s] and  $L_2$  [s] respectively. However, because  $L_1$  and  $L_2$  are fluctuated due to network traffic conditions, the delays cannot be dealt with as a (constant) dead time in conventional control theory. Hence, in the figure, note that their varying time delays are formally expressed as  $e^{-L_1 s}$  and  $e^{-L_2 s}$ .

In this system, the behavior of IP networks between the controller and the plant is emulated by a PC router on which a Linux network emulator netem [5] is installed. Netem can emulate time delays (as mean [ms]), its jitter (as standard deviation [ms]) and distribution (e.g., normal distribution) of varying time delays. A communication protocol used on the network is User Datagram Protocol (UDP). Packets including a control signal may be lost since UDP has no reception confirmation. Moreover a packet sent from the controller earlier may receive at the plant later or vice versa. This issue is packet disorder [6]. In this paper, if a packet has not been received within a sampling period or disorder has occurred, then the receiver employs the latest one obtained at the preceding period. Moreover, it is assumed that the influence of the packets loss can be negligible.

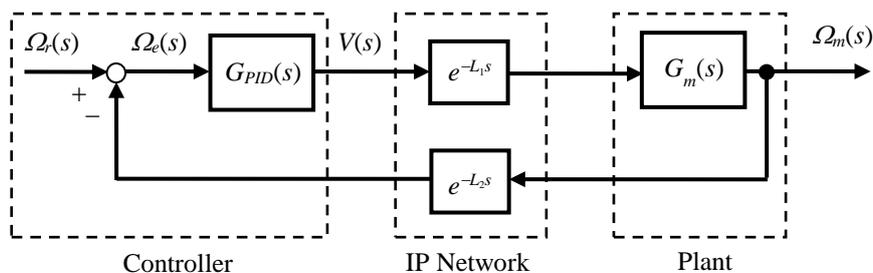


Fig. 1 A block diagram of an IP network-based speed control system for a small DC motor

## III. AN AUTO-TUNING OF THE SYSTEM WITH A DEAD TIME BY PARTIAL MODEL MATCHING METHOD

The partial model matching method by Kitamori [3] can decide PID parameters automatically such that the closed loop transfer function of a provided system including the controller coincides with a function of a reference model, where the response of the reference model is the one that a designer wants to design. In this section, it is explained how the method is applied to a system with a “constant” time delay, i.e., a dead time  $L$  as shown in the upper figure of Fig. 2, not “varying” time delays as in Fig. 1. This method with a dead time is provided as follows [4].

APID controller  $G_{PID}(s)$  is denoted as (1). First, the transfer function of the DC motor model in the plant is described as  $G_m(s) = 1/(g_0 + g_1 s + g_2 s^2)$ , and  $G_{md}(s)$ , which is the function of the motor with a dead time  $L$ , is

$$G_{md}(s) = G_m(s) e^{-Ls} = \frac{1}{(g_0 + g_1 s + g_2 s^2) e^{Ls}}. \quad (2)$$

Let  $g_0 = 4.807 \times 10^{-3}$ ,  $g_1 = 6.346 \times 10^{-4}$  and  $g_2 = 7.232 \times 10^{-8}$ , whose parameters of the motor are empirically derived beforehand. Moreover, let

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

$$G_{md}(s) = \frac{1}{a_0 + a_1s + a_2s^2 + a_3s^3 + \dots} \quad (3)$$

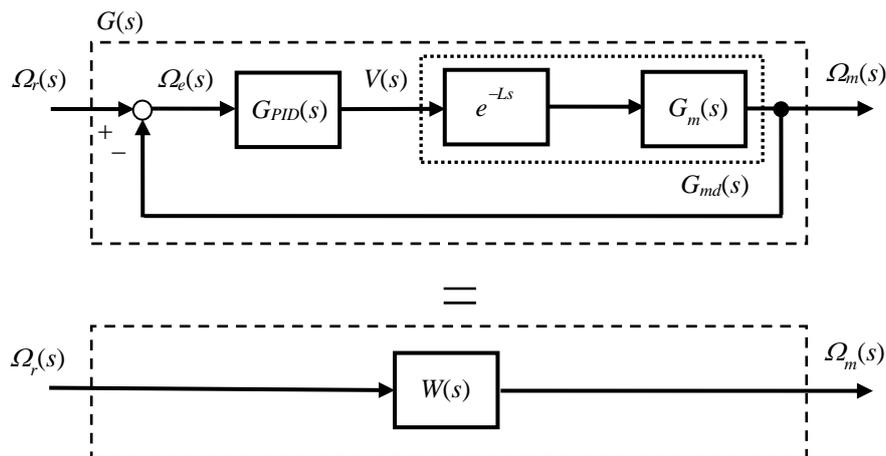


Fig. 2 A block diagram of a speed control system for a small DC motor with a constant dead time (upper figure), and the transfer function of the reference model that should be equivalent to the system of the upper figure (lower figure)

Now, in (2), the Maclaurin series expansion of  $e^{Ls}$  is done, i.e.,  $e^{Ls} = 1 + Ls + L^2s^2/2! + L^3s^3/3! + \dots$ . Then, since (2) is equivalent to (3), the following relations (4) are obtained,

$$\begin{aligned} a_0 &= g_0, & a_1 &= g_1 + g_0L, \\ a_2 &= g_2 + g_1L + g_0 \frac{L^2}{2!}, & a_3 &= g_2L + \frac{g_1L^2}{2!} + \frac{g_0L^3}{3!}. \end{aligned} \quad (4)$$

Second, a reference model transfer function  $W(s)$  is introduced as

$$W(s) = \frac{1}{\alpha(s)} = \frac{1}{\alpha_0 + \alpha_1\gamma s + \alpha_2\gamma^2 s^2 + \alpha_3\gamma^3 s^3 + \alpha_4\gamma^4 s^4 + \dots}, \quad (5)$$

where let  $\{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{1, 1, 17/40, 39/400, 109/7599\}$  [4] and  $\gamma$  is a time scaling parameter. The response of  $W(s)$  in the system shown in the lower figure of Fig. 2 means the desirable one that a designer wants to design in the actual system illustrated in the upper figure. Then PID parameters  $K_P$ ,  $K_I$  and  $K_D$ , are determined such that the coefficient of each term from low to high order in  $s$  of the closed-loop function  $G(s) = G_{PID}(s)G_p(s)/\{1+G_{PID}(s)G_p(s)\}$  matches that of  $W(s)$ , as illustrated in Fig.2. That is, these parameters are done as following formulas:

$$\begin{aligned} K_P &= \frac{a_1}{\gamma} - \alpha_2 a_0, & K_I &= \frac{a_0}{\gamma}, \\ K_D &= \frac{a_2}{\gamma} - \alpha_2 a_1 + \gamma(\alpha_2^2 - \alpha_3) a_0, \end{aligned} \quad (6)$$

where  $\gamma$  is a minimum positive solution, and satisfies the following equation:

$$(2\alpha_2\alpha_3 - \alpha_3^3 - \alpha_4)\alpha_0\gamma^3 + (\alpha_2^2 - \alpha_3)\alpha_1\gamma^2 - \alpha_2 a_2\gamma + a_3 = 0. \quad (7)$$

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

## IV. THE PROPOSED AUTO-TUNING SCHEME ACCORDING TO VARYING TIME DELAYS

The auto-tuning method using the partial model matching method described in the previous section is discussed in the case that a time delay in the system is constant as shown in Fig. 2. However, because time delays vary over IP networks as shown in Fig. 1, a tuning scheme according to fluctuating delays, i.e., the jitter, needs to be developed.

### A. The Relationship Between Minimum Delay Time and Rise Time

It is important to discuss how varying time delays are characterised for developing the auto-tuning scheme to overcome the varying ones. Now let varying time delays be divided into a constant time delay, i.e., a dead time and its jitter. A dead time  $L$  is determined to apply the matching method in the previous section. Let  $L = L_{mean}$ , where  $L_{mean}$  is the mean of round trip times (RTTs), each of which is a time from the controller back to the one via the plant, because varying time delays occur in actual networks, and then the matching method are done.

However, the jitter still affects the step responses obtained when tuning based only on a dead time  $L_{mean}$  and controlling the system. Therefore, the jitter must be characterized for tuning properly under time-varying network conditions. Now it is explained that the relationship between the minimum time  $L_{min}$  of RTTs and the rise one  $T_r$  of the step response. First,  $L_{std}$  and  $T_{std}$  standardizing  $L_{min}$  and  $T_r$  are defined as

$$L_{std} = \frac{L_{min}}{L_{mean}}, \quad T_{std} = \frac{T_r}{T_{rconst}}, \quad (8)$$

respectively. In (8),  $T_{rconst}$  is the rise time of a step response obtained when controlling the system with a dead time  $L = L_{mean}$ , which is shown in the upper figure of Fig. 2, not time-varying. The relationship of  $L_{std}$  and  $T_{std}$  can be utilized for characterizing varying delays as described in next subsection.

For examining the relationship, in the system shown in Fig. 1, under each network condition set by netem, speed control experiments are carried out after auto-tuning PID parameters by the matching method based on  $L_{mean}$  of RTTs. Netem can set a network condition of one-way trip times (OTTs). Let  $\Omega_r(s) = 1500\text{min}^{-1}$ . Experiments are done ten times under each condition of all combinations of these netem parameters as shown in Table 1, where the parameters settings are same in the both direction from the controller to the plant and vice versa in Fig.1. Under the netem settings in the cases that the delay is 50, 100, and 150 ms,  $T_{rconst}$  are 290.2, 450.4 and 589.0 ms respectively, where  $T_{rconst}$  is derived empirically in advance. The distributions set to exponential one, approximated in [1], and normal one.

The results are shown in Fig. 3. As shown in the figure, it seems that there is a linear relationship between  $L_{std}$  and  $T_{std}$ , not depending on the distributions. Hence the relationship  $f(L_{std}) = 1.5 - 0.572L_{std}$  is obtained by a least squares method.

TABLE I  
Netem parameter configuration for the experiments.

Delay [ms]	50, 100, 150
Jitter [ms]	5, 15, 25, 35, 45
Distribution	Normal, Exponential

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

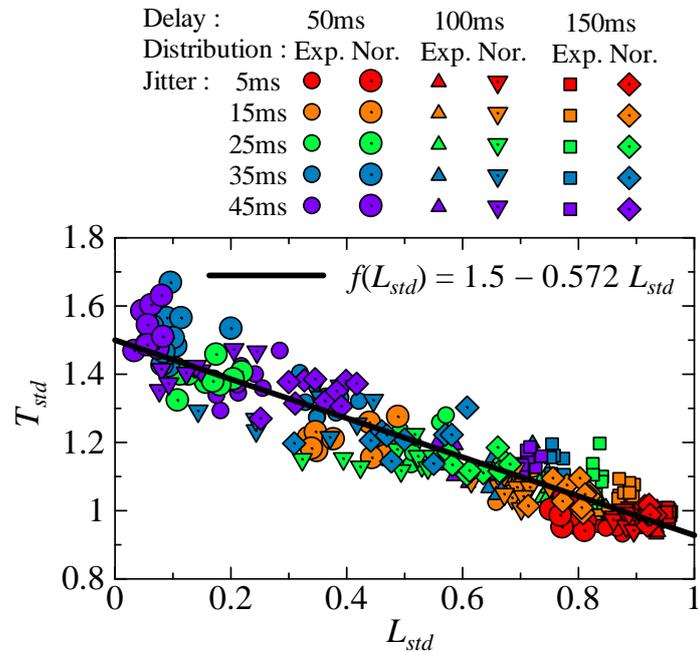


Fig. 3 The relationship between minimum time of RTTs and rise one of step responses

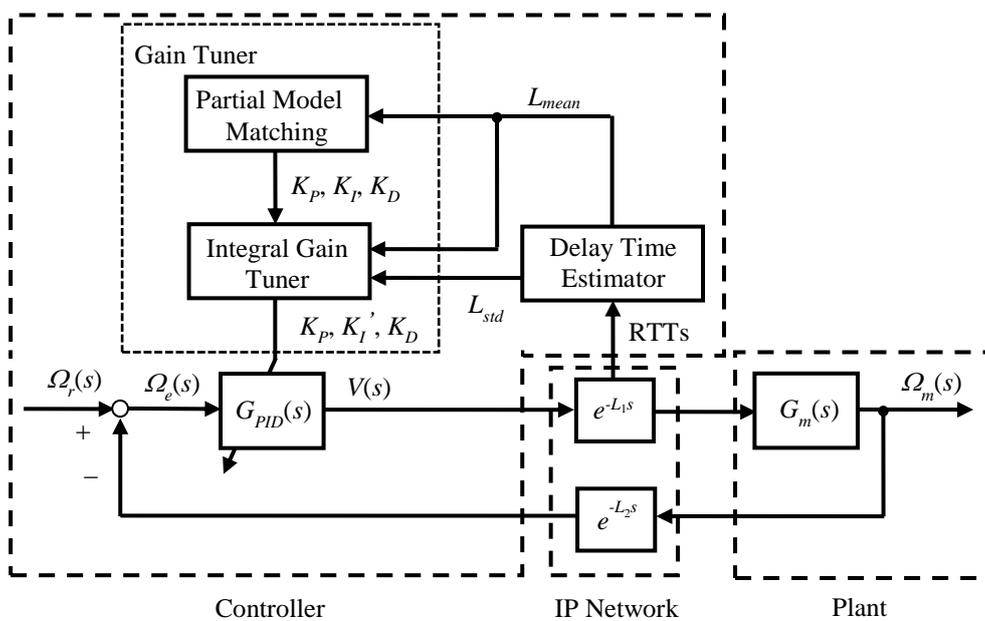


Fig. 4 An IP network-based speed control system with an auto-tuning scheme for a small DC motor

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

## B. The Proposed Scheme

The author proposes an auto-tuning scheme based on the partial model matching considering varying time delays for the system shown in Fig. 1. The scheme is shown in Fig. 4, which is composed of tuning by a partial model matching method based a dead time and re-tuning done on the characterization of the jitter by the relationship described in previous subsection. Specifically, first, delay time estimator derives 1000 RTTs, and  $L_{mean}$  and  $L_{std}$  from  $L_{mean}$  and  $L_{min}$  are computed. Second, the partial model matching of the gain tuner outputs the PID parameters  $K_P$ ,  $K_I$  and  $K_D$  from  $L_{mean}$ . Third, the integral gain tuner modifies  $K_I$  in accordance with varying time delays, i.e., outputs  $K_I' = K_I \cdot f(L_{std})$ , based on the relationship described in the previous subsection, from  $L_{std}$ , where the formula modifying  $K_I$  is empirically adjusted. Finally, the controller  $G_{PID}(s)$  is updated by  $K_P$ ,  $K_I'$ , and  $K_D$ .

## V. EXPERIMENT

For confirming the effectiveness of the proposed scheme, in the system shown in Fig. 4, the speed control experiments are done under network conditions set by netem. Now the netem settings in the both direction between the controller and the plant are as follows:

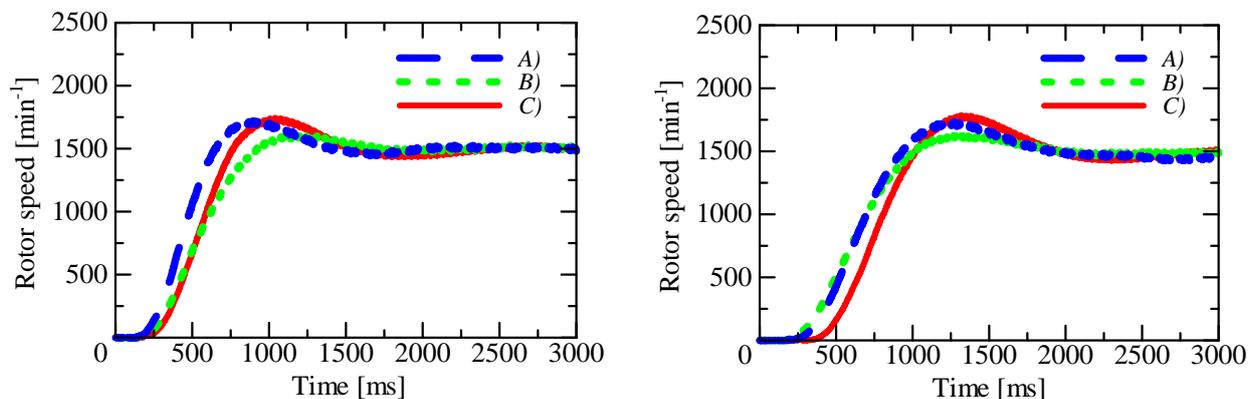
- 1) delay, jitter and distribution are 83ms, 23ms and normal one,
- 2) delay, jitter and distribution are 112ms, 37ms and exponential one.

Let  $\Omega_r(s) = 1500\text{min}^{-1}$ . Under the above network conditions, following experiments are done. First, each PID controller is tuned by

- A) the proposed scheme, whose output parameters are  $K_P$ ,  $K_I'$ , and  $K_D$ ,
- B) the only partial model matching method, whose output parameters are  $K_P$ ,  $K_I$ , and  $K_D$ ,
- C) for comparing C) with A) and B) and showing that A) is more effective than B), the only partial model matching method in the system with a dead time  $L = L_{mean}$ .

The control system in the case C) is a conventional one with a dead time as shown in Fig. 2, not IP network-based one as in Fig. 4. The response obtained in the experiment of this system is regard as one in the case without jitter, whereas A) and B) with jitter. Hence, by comparing C) with A) and B), it can be examined whether the proposed scheme is effective against the jitter.

Next, after the tuning, speed control experiments are carried out. As a result, the obtained step responses are shown in Fig. 5, and the parameters in Table II, where  $\sigma$  is a standard deviation of the measured RTTs. In the case C),  $L$  is set to the time close to  $L_{mean}$  in A) and B) for comparing them as shown in the table. The results show that, in the both cases of 1) and 2), the responses of A) can obtain the similar ones of C) in the overshoot and the rise time. Meanwhile, the ones of B) are obviously affected by the jitter in the overshoot. Therefore, it is found that A) can work well according to varying time delays, especially the jitter.



(a) The step responses under network condition 1)

(b) The step responses under network condition 2)

Fig. 5 Step responses obtained by speed control experiments under two network conditions

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 3, March 2014

TABLE II  
The parameters computed and measured by the scheme in the experiments

Experiment	Parameter	Network Condition	
		1)	2)
A)	$K_P$	0.002316	0.001693
	$K_I'$	0.014025	0.009806
	$K_D$	0.000057	0.000044
	$L_{mean}$ [ms]	178.8	249.0
	$L_{min}$ [ms]	77.0	170.0
	$\sigma$ [ms]	32.2	52.5
B)	$K_P$	0.002262	0.001680
	$K_I$	0.013646	0.009722
	$K_D$	0.000057	0.000043
	$L_{mean}$ [ms]	183.4	250.9
	$\sigma$ [ms]	31.7	51.0
C)	$K_P$	0.002301	0.001686
	$K_I$	0.013923	0.009762
	$K_D$	0.000057	0.000044
	$L$ [ms]	180.0	250.0

## VI. CONCLUSION

An auto-tuning scheme of PID controller has been proposed considering time-varying network conditions based on a partial model matching method. For confirming the effectiveness of the proposed scheme, speed control experiments have been done in an IP network-based control system of an actual small DC motor. The experimental result has been that the step response obtained using the scheme has been similar to one done after tuning the system with the constant delay time and has been improved against the jitter. Therefore it has been found that the scheme has worked well against time delays and its jitter. As a result, it has been confirmed that the scheme has been effective.

## ACKNOWLEDGMENT

The author would like to thank K. Shiroyama for the experimental assistant.

## REFERENCES

1. Tipsuwan, Y. and Chow, M.-Y. (2004), 'Gain Scheduler Middleware: A Methodology to Enable Existing Controllers for Networked Control and Teleoperation-Part I: Networked Control', IEEE Transactions on Industrial Electronics, Vol. 51, No.6, pp. 1218-1227.
2. Matsuo, K., Miura, T. and Taniguchi, T. (2007), 'A Speed Control Method of Small DC Motor through IP Network Considering Packet Loss', IEEE Transactions on Electrical and Electronic Engineering, Vol. 2, No.6, pp. 657-659.
3. Kitamori, T. (1979), 'A Method of Control System Design Based upon Partial Knowledge about Controlled Processes', Transactions of the Society of Instrument and Control Engineers, Vol. 15, No. 4, pp. 549-555. (in Japanese)
4. Abe, N. and Kojima, A. (2007), 'Control in Time-delay and Distributed Parameter System', pp. 28-33, Corona Publishing. (in Japanese)
5. netem, <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem/>
6. Zhao, Y.-B., Liu, G.-P. and Rees D. (2010), 'Actively Compensating for Data Packet Disorder in Networked Control Systems', IEEE Transactions on Circuits and Systems II - Express Briefs, Vol.57, No.11, pp.913-917.