Electrical Engineering in Japan, Vol. 184, No. 1, 2013
Translated from Denki Gakkai Ronbunshi, Vol. 131-D, No. 12, December 2011, pp. 1442–1450

Self-Tuning PI Control Using Adaptive PSO of a Web Transport System with Overlapping Decentralized Control

TAKESHI NISHIDA,[1] TETSUZO SAKAMOTO,[1] and NICOLA IVAN GIANNOCCARO[2]
[1]Kyushu Institute of Technology, Japan
[2]University of Salento, Japan

SUMMARY

Web transport systems for transporting films, textile material, paper, etc., are usually large-scale systems. The velocity and the tension of the web are controlled by dividing the systems into several subsystems in which strong coupling exists between the velocity and tension control. A self-tuning PI (STPI) controller with an estimator based on a novel adaptive particle swarm optimization method is constructed, and it is applied for controlling an actual web transport system. The controllers are designed on the basis of the methodology of the overlapping decentralized control by taking into consideration online executions performed by a general computer. The effectiveness of the constructed control system is verified on the basis of several experimental results obtained by using an actual experimental web transport system. © 2013 Wiley Periodicals, Inc. Electr Eng Jpn, 184(1): 56–65, 2013; Published online in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/eej.22366

Key words: web transport system; overlapping decentralized control; self-tuning PI control; adaptive particle swarm optimization.

1. Introduction

Web is a collective term for paper, films, and other materials. The equipment used in processing of such materials is usually called web transport. Raw web materials are discharged from unwinders, processed (e.g., printed) while passing over multiple rollers, and finally wound up by motor-driven winders. A change of web transport speed in the course of processing may result in nonuniform processing; furthermore, abnormal tension leads to buckling and breaks. Thus, advanced control is required in order to synchronize all parts of a web for appropriate speed and tension.

Usually, web transport is a large system including numerous driving rollers, and hence decentralized control is often applied to improve maintenance and simplify the configuration of the control system [1]. When designing decentralized control, a system is divided into multiple subsystems. The problem is how to deal with mutual interference among the subsystems. Overlapping decentralized control was proposed as a solution to this problem [2]. Specifically, strongly coupled components were arranged in one subsystem, and shared by neighbor subsystems. In real life, however, the parameters of a web transport system vary with time and the environment, and high-order dynamics is difficult to model; thus, adaptivity remains an issue to be solved. Recently, an STPID (Self-Tuning PID) control design was proposed to solve this problem by providing adaptation to parameter fluctuations [3]. In particular, the proposed control system for web transport has the following features: (1) a system is divided into subsystems using overlapping decentralized control design, (2) using noise-robust GMVC (Generalized Minimum Variance Control) [4], explicit self-tuning control [5] is configured, (3) PSO (Particle Swarm Optimization) is employed for system parameter identification, (4) GMVC control rules are replaced with PID control rules [5, 7, 8]. The effectiveness of the previously proposed control method [3] was verified by simulations, and its robustness to the system's nonlinear characteristics and observation noise was demonstrated. However, actual web transport systems involve dynamics and observation noise that cannot be fully reproduced in simulations, and therefore real-machine verification remained an issue. In addition, the computational complexity in real systems is restricted by the necessity of real-time execution, and hence the control algorithm must be implemented for online execution. In the present study, we consider the improvements required in order to apply the previously proposed method [3] to actual web transport systems. Specifically, we present a method for online estimation of the system parameters based on adaptive PSO (OPSO: Online PSO) that offers good calculation efficiency, and a particular configuration of a self-adjusting PI controller based on the estimated parameters. We also demonstrate the effectiveness of the proposed methods using a web transport system.

The paper is organized as follows. First, related studies are reviewed in Section 2. Then the considered web transport system and control system design are explained in Section 3. Experimental results are presented in Section 4, and conclusions are drawn in Section 5.

2. Related Research

2.1 Research on web transport systems

As regards decentralized control proposed so far for web transport systems, there are methods based on H_∞ control [2, 10, 11], two-DOF H_∞ control [12, 13], and a two-DOF gain-scheduled controller [14]. The effectiveness of these methods was verified in experiments with a three-motor experimental system [12, 14] and in simulations with a relatively large nine-motor system model [13]. In addition, the design parameters were adjusted in advance using detailed models, and were implemented in real machines (simulation-based control design).

Research on model-reference web tension control systems includes feedforward control [15] and adaptive PI control [16]. In such methods, one needs relatively detailed models of the controlled plants, while providing robustness to modeling errors is an important issue of controller design. The effectiveness of these methods has been proven in simulations.

The previously proposed methods were verified mainly by simulations. Even experimental studies have dealt with minimum configurations of the web transport systems; to our knowledge, the real-machine verification in the present study is the first of this scale.

2.2 Research on controller self-tuning based on PSO

There are many methods for PSO-based search of PID gain sets. The most typical are those that select the PID parameters so as to minimize a response waveform index by offline simulations using an identification model [17–22]. Direct search is performed in a space configured of three PID parameters, using a combination of criteria such as IAE (Integral Absolute Error), ISE (Integral Square Error), ITSE (Integral-of-Time multiplied Square Error), or ITAE (Integral-of-Time multiplied Absolute Error). However, when using such criteria to evaluate the performance of a PID controller, one must comprehensively examine the control waveforms, from transient states to steady state, which means that online search for the PID parameters is impracticable.

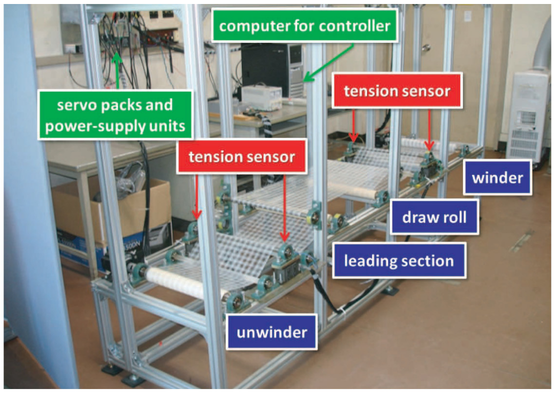
In addition, there have been attempts to combine PSO and PID with other algorithms. For example, a method was proposed in which the PID parameters are predetermined by PSO search under various credible conditions, and then synthesized dynamically by fuzzy rules in system control [23]. In addition, a method for online design of an H_∞ controller was proposed using PSO, intended for optimization of constrained problems (ALPSO: Augmented Lagrangian PSO) [24]. However, these methods employ conventional PSO algorithms that do not provide adaptive search for time-varying systems, and hence the search for control parameters must be applied to static models.

As regards online PSO implementation, there is a method of adjusting the PID parameters by minimization of the errors between a reference model and the output when identification is difficult [25]. Another method has been proposed for tuning of PID control rules using offline and online PSO [26]. However, these methods do not consider adaptation to the system's time variation; PSO search and PID parameter tuning converge over time, which makes it difficult to use these methods for time-varying systems.

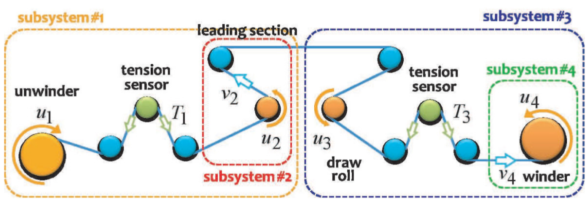
On the other hand, there is a method called MPC-PSO in which PSO is used in online control to determine the input sequence of model predictive control (MPC) [27]. In this method, the predictive control input is selected using conventional PSO. The effectiveness of such an approach was verified for both linear and nonlinear systems. However, this usage of PSO is different from the online parameter identification in the present study; in MPC-PSO, a separate means is needed to acquire a plant model. In addition, here too, adaptation to time-varying systems is not considered.

3. Experimental Web Transport System and Control System Design

The experimental web transport system is shown in Fig. 1. The system includes 12 rollers, of which 4 are provided with servo motors and 2 are provided with tension sensors. The rollers driven by motors are called driving rollers. In addition, the first roller that discharges the web is called the unwinder, and the last roller that winds the web up is called the winder. The present study aims at control to maintain the desired values of the tension $T_1(t)$ and $T_2(t)$ at two points as well as the speeds $v_2(t)$ and $v_4(t)$ of the web on the second and fourth rollers (driving rollers) as shown in Fig. 1(b). Here t denotes time. Other symbols used in this study are explained in Table 1.



(a) Overview of the experimental web transport system



(b) Scheme of the experimental web transport system

Fig. 1. Experimental web transport system. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

3.1 Overlapping decentralized control

In the system shown in Fig. 1, the tension and speed at every roller are related in terms of the viscoelasticity of the web as follows:

$$T_i(s) = \frac{P(s)}{L_i}\{V_{i+1}(s) - V_i(s)\} \tag{1}$$

where the characteristic transfer function $P(s)$ is expressed by the Voigt model as

$$P(s) = A\left(\eta_v + \frac{G_v}{s}\right) \tag{2}$$

Table 1. Conventional symbols

A [m^2]	Cross-sectional area of web
G_v [N/m^2]	Elastic modulus of web material
η_v [Ns/m^2]	Viscosity modulus of web material
r_i [m]	Radius of i-th drive roll
J_i [kgm^2]	Moment of inertia of i-th drive roll
L_i [m]	Web length between i-th and $(i+1)$-th drive roll
u_i [Nm]	Input torque of i-th drive roll
v_i [m/s]	Web velocity on i-th drive roll
T_i [N]	Tension between i-th and $(i+1)$-th drive roll

The input/output relations of the experimental system are shown in Fig. 2. Thus, the transfer functions for the web speed are

$$V_1(s) = \frac{r_1}{J_1 s}(r_1 T_1(s) - U_1(s)) \tag{3}$$

$$V_2(s) = \frac{r_2}{J_2 s}\{U_2(s) + r_2(T_2(s) - T_1(s))\} \tag{4}$$

$$V_3(s) = \frac{r_3}{J_3 s}\{r_3(T_3(s) - T_2(s)) - U_3(s)\} \tag{5}$$

$$V_4(s) = \frac{r_4}{J_4 s}\{U_4(s) - r_4 T_3(s)\} \tag{6}$$

Considering Eq. (1), the transfer functions from $U_1(s)$ and $U_2(s)$ to $T_1(s)$, or from $U_3(s)$ and $U_4(s)$ to $T_3(s)$, can be expressed as follows:

$$T_i(s) = G_i(s)\tilde{U}_i(s) \tag{7}$$

where

$$\tilde{U}_i(s) \triangleq \frac{r_i}{J_i}U_i(s) + \frac{r_{i+1}}{J_{i+1}}U_{i+1}(s) \tag{8}$$

$$G_i(s) \triangleq \frac{\frac{P(s)}{L_i}}{1 + \frac{P(s)}{L_i}\left(\frac{r_i^2}{J_i s} + \frac{r_{i+1}^2}{J_{i+1}s}\right)} \tag{9}$$

(i = 1, 3). Now the transfer function from U_2 to $V_2(s)$, or from U_4 and $V_4(s)$, can be expressed as follows:

$$V_i(s) = G_i(s)\tilde{U}_i(s) \tag{10}$$

where

$$\tilde{U}_i(s) \triangleq U_i(s) \tag{11}$$

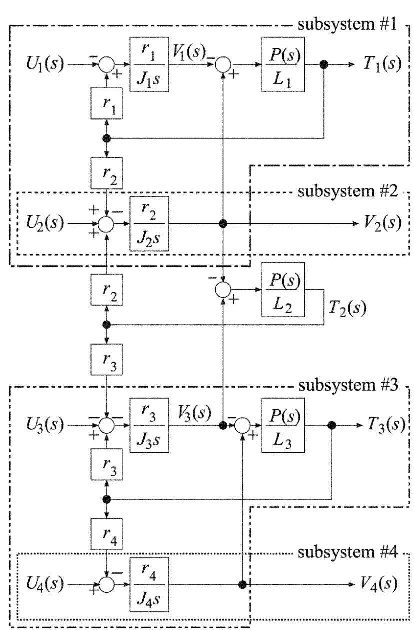


Fig. 2. Block diagram of system.

$$G_i(s) \triangleq \frac{r_i}{J_i s} \tag{12}$$

($i = 2, 4$). Thus, the experimental system can be divided into four overlapping subsystems in Eqs. (7) and (10). Let the manipulated variable of every subsystem be expressed as

$$\tilde{\boldsymbol{u}}(t) \triangleq (\tilde{u}_1(t)\ \tilde{u}_2(t)\ \tilde{u}_3(t)\ \tilde{u}_4(t))^T \tag{13}$$

Then the input of every motor is

$$\boldsymbol{u}(t) = \boldsymbol{N}\tilde{\boldsymbol{u}}(t) \tag{14}$$

Here $\boldsymbol{u}(t) \equiv (u_1(t)\ u_2(t)\ u_3(t)\ u_4(t))^{\mathrm{T}}$ and the following is the transformation matrix of the manipulated variables:

$$\boldsymbol{N} \triangleq \begin{pmatrix} r_1/J_1 & r_2/J_2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & r_3/J_3 & r_4/J_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \tag{15}$$

3.2 Discrete-time model

In order to implement a control algorithm in the experimental system, every subsystem is represented by a discrete-time model as follows:

$$A_j(z^{-1})y_j(k) = z^{-k_m}B_j(z^{-1})u_j(k) \tag{16}$$

Here $y_j(k)$ denotes the output at discrete time point k, and k_m denotes the minimum dead time. In this study, we assume $k_m = 1$ and consider only the dead time of zero hold. In addition, j is the subsystem number; below this index is omitted because the control system is configured in the same way for each subsystem.

In this study, we configure a mechanism to sequentially identify $A(z^{-1})$ and $B(z^{-1})$ for a response within about 1 s from the current instant* using adaptive PSO (to be explained below). Thus, in this study the response (9) is approximated by a sequentially updated first-order delay system, and the transfer functions of all subsystems are synthesized from first-order delay elements $G(s) = K/(T_a s + 1)$ and zero-hold elements $(1 - \exp(-T_s s))/s$, and approximated by the z-transform:

$$\mathcal{Z}[H(s) \cdot G(s)] = \frac{K\{1 - \exp(-T_s/T_a)\}z^{-1}}{1 - \exp(-T_s/T_a)z^{-1}} \tag{17}$$

Here T_s is the sampling time; T_a and K are the time constant and gain, respectively. From the above, the following polynomials are obtained for the system parameters:

$$A(z^{-1}) = 1 - \exp(-T_s/T_a)z^{-1} \triangleq 1 + a_1 z^{-1} \tag{18}$$

$$B(z^{-1}) = K\{1 - \exp(-T_s/T_a)\} \triangleq b_0 \tag{19}$$

*We set this time period assuming it to be sufficient for estimation of the dominant mode in every subsystem.

3.3 Online identification of system parameters by OPSO

PSO is a method of search for statistically optimal solutions. The method imitates swarm intelligence and proves effective for nonlinear programming problems [6]. The robustness and high speed of PSO in optimal solution search have been proved by numerous studies dealing with nonlinear, nondifferentiable, multimodal, and other problems, and this method is now applied in a wide range of fields [28, 29]. PSO was first proposed for solving static optimization problems. Recently, the method was modified to deal with real dynamical systems, and new PSO algorithms can adapt to environmental changes caused by measurement noise or the system's time variation, that is, to a varying search space [30–32]. In addition, the authors have proposed an adaptive OPSO featuring high calculation efficiency in online identification of time-varying systems [9]. In the present study, this algorithm is used to configure a mechanism for online identification of system parameters.

3.3.1 Algorithm

The position and velocity of particles in the search space are expressed by $\boldsymbol{x}_m(k) \equiv (a_{1m}(k)b_{0m}(k))^{\mathrm{T}}$ and $\boldsymbol{v}_m(k) \in \mathbb{R}^2$, respectively. Here $m = [1, M] \in \mathbb{N}_+$ is the particle's number, and $a_{1m}(k)$ and $b_{0m}(k)$ are system parameters estimated by the m-th particle. OPSO is used to seek the optimal solution to the following problem with objective function $f_k : \mathbb{R}^2 \to \mathbb{R}$:

$$\min_{\boldsymbol{x}} f_k(\boldsymbol{x}) \geq 0 \tag{20}$$

First, the best solution $\hat{\boldsymbol{x}}_m(k-1)$ of every particle obtained at the previous time point is reevaluated using the evaluation function at the present time point, and the minimum value is found:

$$\tilde{\boldsymbol{x}}^g(k) = \arg\min\{f_k(\hat{\boldsymbol{x}}_m(k-1))\} \tag{21}$$

Then the position and velocity of each particle are updated as follows:

$$\begin{aligned} \boldsymbol{v}_m(k) = \omega\boldsymbol{v}_m(k-1) &+ c_1 r_1\{\tilde{\boldsymbol{x}}^g(k) - \boldsymbol{x}_m(k-1)\} \\ &+ c_2 r_2\{\hat{\boldsymbol{x}}_m(k-1) - \boldsymbol{x}_m(k-1)\} \end{aligned} \tag{22}$$

$$\boldsymbol{x}_m(k) = \boldsymbol{x}_m(k-1) + \boldsymbol{v}_m(k) \tag{23}$$

Here ω, c_1, and c_2 are configuration parameters; the initial values $\boldsymbol{x}_m(0) = \hat{\boldsymbol{x}}_m(0)$ and $\boldsymbol{v}_m(0)$ are set using random numbers. Then the position of every particle offering the minimum evaluation value and its value $f_k(\hat{\boldsymbol{x}}^{(m)}(k))$ are found:

$$\hat{\boldsymbol{x}}_m(k) = \begin{cases} \boldsymbol{x}_m(k) & \text{if } f_k(\boldsymbol{x}_m(k)) < f_k(\hat{\boldsymbol{x}}_m(k-1)) \\ \hat{\boldsymbol{x}}_m(k-1) & \text{otherwise} \end{cases} \tag{24}$$

Finally, the best solution for the entire swarm, that is, the OPSO estimate at time point k, is found:

$$\hat{\boldsymbol{x}}^g(k) = \arg\min\{f_k(\hat{\boldsymbol{x}}_m(k))\} \triangleq (\hat{a}_1(k)\ \hat{b}_0(k))^T \tag{25}$$

OPSO is different from conventional PSO algorithms as follows: (1) due to the use of a time-varying evaluation function, the values $\hat{\boldsymbol{x}}_m(k)$ (usually referred to as *pbest*) and $\hat{\boldsymbol{x}}^g(k)$ (usually referred to as *gbest*) do not decrease monotonically, (2) no new design parameters are added, (3) the calculation procedure is simple, (4) a strong adaptive ability is obtained while the only additional calculation is Eq. (21).

3.3.2 Particle evaluation

The following evaluation function is used to estimate the system parameters by OPSO:

$$f_k(\boldsymbol{x}_m(k)) = \sum_{i=0}^{I} \left|\boldsymbol{y}(k-i) - \boldsymbol{\psi}^T(k-i)\boldsymbol{x}_m(k)\right| \tag{26}$$

where

$$\boldsymbol{\psi}^T(k) \triangleq (y(k-1)\ \ u(k-1)) \tag{27}$$

I is the number of estimation steps. Because of the constraint on the stability of the system parameters to be estimated, search is not performed in the regions where $\hat{a}_{1m}(k) > 0$ or $\hat{b}_{0m}(k) < 0$. The greater the value of I that is set, the more effectively the influence of measurement noise can be suppressed; however, the computational complexity increases directly with I, which results in a longer delay time in order to reflect the changes of the system parameters in the control rules. In this study, the sampling time of the experimental system is $T_s = 10$ ms, and I is set to 100. This means that the parameters representing the system's dynamics are calculated using response waveforms within 1 s at most.

3.4 Self-tuning of PID controller based on GMVC

The control systems of every subsystem of the web transport system considered in this study are configured using a method that couples minimization of the performance index and PID control rules [7]. Transformation into PID control rules has the following advantages: (1) easy implementation on PID-controlled equipment, (2) easy examination of system behavior in the frequency domain, (3) the behavior and implications of the self-tuned control inputs are easy to understand for specialists experienced in PID control. In the framework of the present study, points (2) and (3) are especially beneficial. Namely, the experimental system is susceptible to mutual interference among the subsystems and the influence of measurement noise, and hence it is important to select PI parameters for efficient control in the low-frequency region. In this context, the representation of control rules by means of PID parameters is very convenient to evaluate the validity of self-tuning controllers. Below we explain briefly how this method is applied to the control system in this study.

First consider the minimization of the following performance index based on GMVC:

$$J = E\left[\phi^2(k + k_m + 1)\right] \tag{28}$$

The generalized output $\phi(k + k_m + 1)$ is specified as follows:

$$\begin{aligned}\phi(k + k_m + 1) := {} & P(z^{-1})y(k + k_m + 1) + \lambda\Delta u(k) \\ & - R(z^{-1})w(k + k_m)\end{aligned} \tag{29}$$

Here $\omega(k)$ is the target value and λ is a design parameter that weights the control input. In addition, $\Delta = 1 - z^{-1}$. The control rule $u(k)$ is expressed as follows:

$$\begin{aligned}F(z^{-1})y(k) + & \left\{E(z^{-1})B(z^{-1}) + \lambda\right\}\Delta u(k) \\ & -R(z^{-1})w(k) = 0\end{aligned} \tag{30}$$

Here $E(z^{-1})$ and $F(z^{-1})$ are found from the following Diophantine equation:

$$P(z^{-1}) = \Delta A(z^{-1})E(z^{-1}) + z^{-(k_m+1)}F(z^{-1}) \tag{31}$$

where

$$E(z^{-1}) = 1 + e_1 z^{-1} \tag{32}$$

$$F(z^{-1}) = f_0 + f_1 z^{-1} \tag{33}$$

In addition, $P(z - 1)$ is defined as the following design polynomial to adjust the rise time and damped oscillation characteristics:

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2} \tag{34}$$

where

$$p_1 = -2e^{-\frac{\rho}{2\mu}} \cos\frac{\sqrt{4\mu - 1}}{2\mu}\rho \tag{35}$$

$$p_2 = e^{-\frac{\rho}{\mu}} \tag{36}$$

$$\rho = \frac{T_s}{\sigma} \tag{37}$$

$$\mu = 0.25(1 - \delta) + 0.51\delta \tag{38}$$

The quantities δ and σ can be used to adjust the system response characteristics. In the present study, $\delta = 0$ is fixed to prevent overshoot, and therefore the above expressions can be simplified as follows, with σ being the only parameter that defines $P(z^{-1})$:

$$p_1 = -2e^{-\frac{2T_s}{\sigma}} \tag{39}$$

$$p_2 = e^{-\frac{4T_s}{\sigma}} \tag{40}$$

On the other hand, the control rules of a digital PID controller can be expressed as follows:

$$\Delta u(k) = k_p \left(\Delta + \frac{T_s}{T_I} + \frac{T_D}{T_s} \Delta^2 \right) e(k) \tag{41}$$

Here $e(k) \equiv \omega(k) - y(k)$. This can be rewritten as follows:

$$C(z^{-1})y(k) + \Delta u(k) - C(z^{-1})w(k) = 0 \tag{42}$$

where

$$\begin{aligned} C(z^{-1}) &\triangleq c_0 + c_1 z^{-1} + c_2 z^{-2} \\ &= k_p \left(1 + \frac{T_s}{T_I} + \frac{T_D}{T_s} \right) - k_p \left(1 + \frac{2T_D}{T_s} \right) z^{-1} \\ &\quad + \frac{k_p T_D}{T_s} z^{-2} \end{aligned} \tag{43}$$

Now, placing the emphasis on the steady-state response and approximating $E(z^{-1})B(z^{-1}) \approx E(1)B(1)$, Eq. (30) can be described as follows:

$$\frac{F(z^{-1})}{\nu} y(k) + \Delta u(k) - \frac{R(z^{-1})}{\nu} w(k) = 0 \tag{44}$$

where

$$\nu \triangleq B(1)E(1) + \lambda \tag{45}$$

Therefore, as follows from a comparison with Eq. (42), GMVC and PID control can be considered approximately equivalent with the following design:

$$R(z^{-1}) = F(z^{-1}) \tag{46}$$

$$C(z^{-1}) = F(z^{-1})/\nu \tag{47}$$

Therefore, the PID parameters can be derived as follows using the system parameters $\hat{a}_1(k)$ and $\hat{b}_0(k)$ estimated by OPSO:

$$k_p = -f_1/\nu \tag{48}$$

$$T_I = -f_1 T_s/(f_0 + f_1) \tag{49}$$

However, the system dynamics in this study is approximated by low-order polynomials, and T_D cannot be calculated. Using Eq. (31) and the system parameters estimated by Eq. (40), the following can be obtained:

$$f_0 = p_2 + \hat{a}_1(k) + (1 - \hat{a}_1(k))e_1 \tag{50}$$

$$f_1 = e_1 \hat{a}_1(k) \tag{51}$$

$$\nu = \hat{b}_0(k)(1 + e_1) + \lambda \tag{52}$$

$$e_1 = p_1 - \hat{a}_1(k) + 1 \tag{53}$$

Figure 3 shows the control system configured by applying this method to the web transport system.

The design parameters of such STPI control based on GMVC are λ and σ. All subsystems of the web transport system should be controlled synchronously, and hence these parameters should be designed the same for all subsystems. Thus, we adjust the four subsystems using the common parameters λ and σ. The physical meaning of these parameters is obvious, which makes their setting relatively easy.

In the previous study [3], we repeated PSO search 40 or more times at each discrete optimization step, and in addition we sought five system parameters assuming a second-order estimation model. On the other hand, in the present study, the processing time is reduced by a factor of about 100 due to the introduction of OPSO and the use of an estimation model of lower order, which makes possible real-time implementation of the control algorithm.

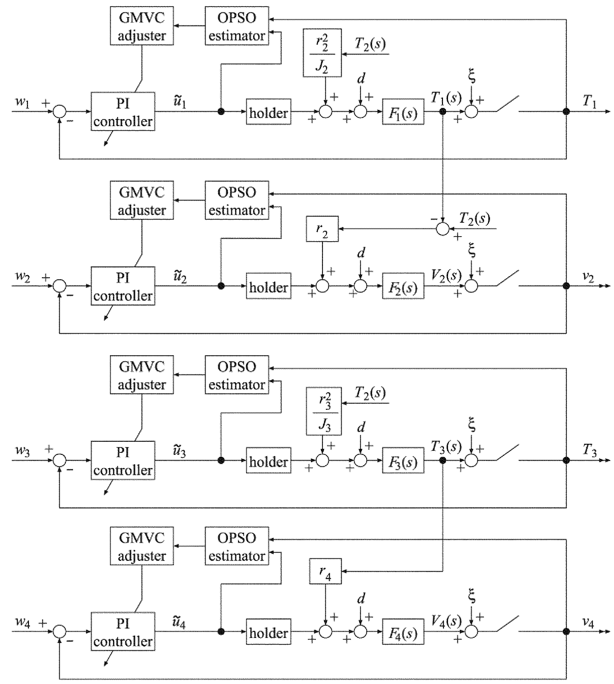


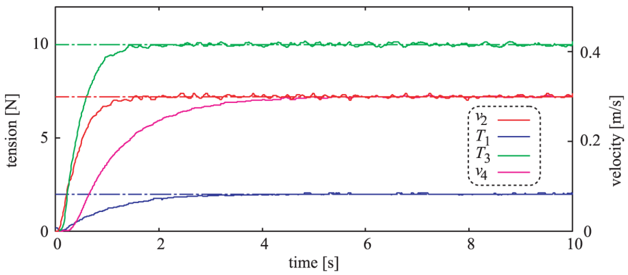
Fig. 3. Block diagram of control system.

4. Experiments

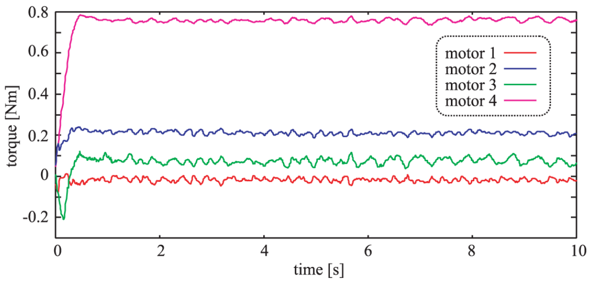
4.1 Response to step input

Here we present the results of experiments with the above control method applied to the web transport system. The number of particles in PSO was 100, and the number of evaluation steps for the function in Eq. (26) was $I = 100$. The design parameters of GMVC were $\lambda = 10$ and $\sigma = 1$. The target values were set to $\omega_1 = \omega_4 = 0.3$ m/s, $\omega_2 = 2$ N, $\omega_3 = 10$ N. The initial values of the sought parameters were $\hat{a}_{1j}(k) = 0$ and $\hat{b}_{0j}(k) = 0$. Here $j = 1, 2, 3, 4$ is the number of the subsystem.

The time evolution of the output values and input torque is illustrated in Fig. 4. As indicated by the diagram, the controlled variables converge to the respective target



(a) Time evolution of output values



(b) Time evolution of input torques of each motor

Fig. 4. Experimental results. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

values without overshoot, while the steady-state response changes smoothly. The time evolution of the estimated parameters $\hat{a}_{1j}$ and $\hat{b}_{0j}$ is shown in Fig. 5. The time evolution of the PI parameters tuned online is shown in Fig. 6. As can be seen from these results, the estimated system parameters vary adaptively over time, and the PI parameters are tuned accordingly. The estimated parameters fluctuate because of measurement noise and other disturbances, which is reflected in oscillatory fluctuations of the PI parameters; however, k_p is adjusted within a relatively narrow range of 0 to 0.2, except for the initial part,* thus not exerting a strong effect on input variation and the control results.

4.2 Variation of response characteristics with system parameters

In the proposed method, there are two configuration parameters, λ, which imposes a constraint on input variation, and σ, which adjusts the response time. Here we present experimental results pertaining to the influence of these parameters on the system's response.

*In Fig. 6, k_p of subsystems 2 and 4 goes beyond this range in some places; in particular, 0.977 at 0.06 s and 0.501 at 0.07 s in the former case, and 0.259 at 0.08 s and 0.220 at 0.30 s in the latter case. These deviations were very brief and were not so large as to degrade the system's stability.

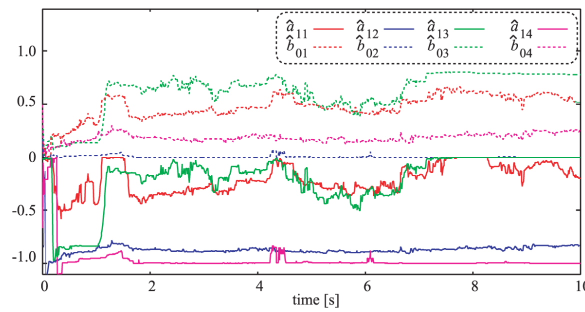


Fig. 5. Time evolution of estimated system parameters of each subsystem. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Experimental results obtained for various λ are shown in Fig. 7. As can be seen from the graphs, greater variation of the input is allowed when λ is set smaller, and convergence to the target becomes faster. However, at $\lambda <$ 3, the proportional gain becomes too high, which results in instability.

Experimental results obtained for various σ are shown in Fig. 8. As can be seen from the graphs, the convergence worsens when σ is set small. However, at $\sigma =$ 0.3 a strong fluctuation of v_2 was observed for about 1 s; that is, an anomalous web speed may occur when the parameter is set too small. When σ is set above 1, the second and third terms on the right-hand side of Eq. (34) become very small, and the effect of σ on the system's response weakens.

According to the above results, first σ should be set sufficiently large unless there is a need to provide a slow response; then λ should be set as small as possible while assuring the system's stability.

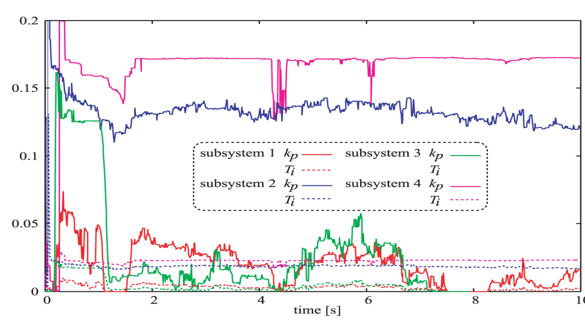
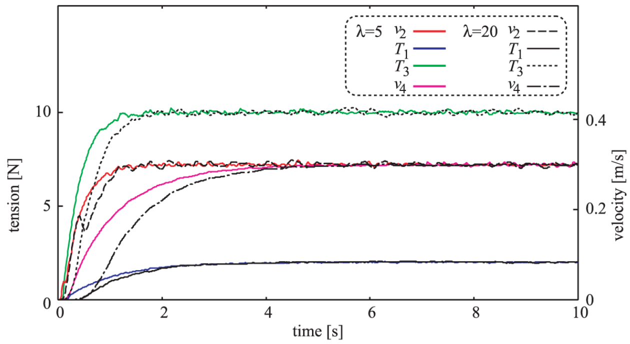
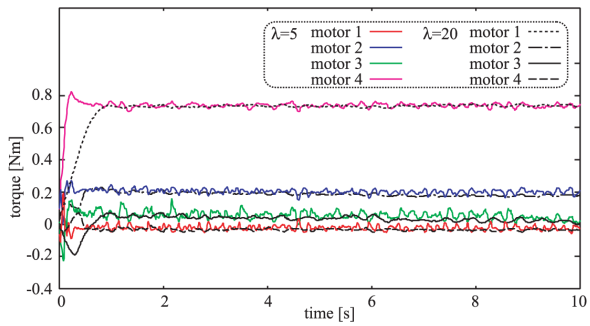


Fig. 6. Time evolution of calculated PI parameters of each subsystem. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

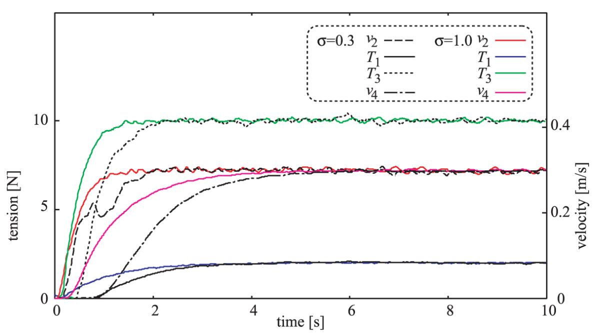


(a) Time evolution of output values

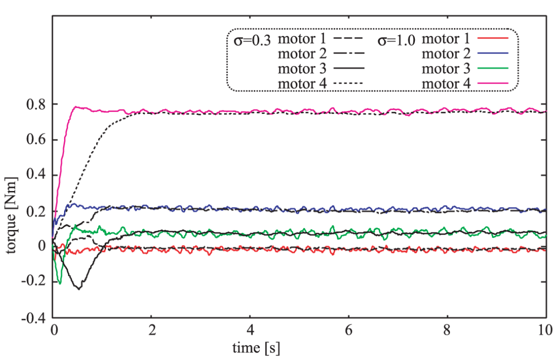


(b) Time evolution of input torques of each motor

Fig. 7. Experimental results using various λ. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]



(a) Time evolution of output values



(b) Time evolution of input torques of each motor

Fig. 8. Experimental results using various σ. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 2. PI parameters for each subsystem

	1st	2nd	3rd	4th
k_p	0.01	0.15	0.30	0.15
T_I	0.013	0.020	0.175	0.020

4.3 Comparison in performance with other methods of PI parameter tuning

Here we demonstrate the effectiveness of the proposed method by experimental comparison with other settings of PI parameters.

First, in preliminary tests we confirmed that the web cannot be transported without breakage by using conventional methods such as PI or CHR. In such adjustment algorithms, the proportional gains in subsystems 2 and 4 (speed adjustment systems) tended to be set large; thus, we adjusted the PI parameters of subsystems 1 and 3 (tension control system) using the CHR algorithm and then tuned the PI parameters of subsystems 2 and 4 by trial and error. The obtained PI parameters are listed in Table 2, and the experimental results obtained using these parameters are shown in Fig. 9. In order to evaluate the response with respect to variation of the target values, the initial target values $\omega_1 = \omega_4 = 0.3$ m/s, $\omega_2 = 2$ N, $\omega_3 = 10$ N were changed to $\omega_1 = \omega_4 = 0.7$ m/s, $\omega_2 = 4$ N, $\omega_3 = 13$ N after 10 s. The other conditions were the same as in the previous experiments. As can be concluded from these results, the proposed method provides fast adaptation to variation of the target values and is more efficient than conventional PI controllers with complicated adjustment procedures.

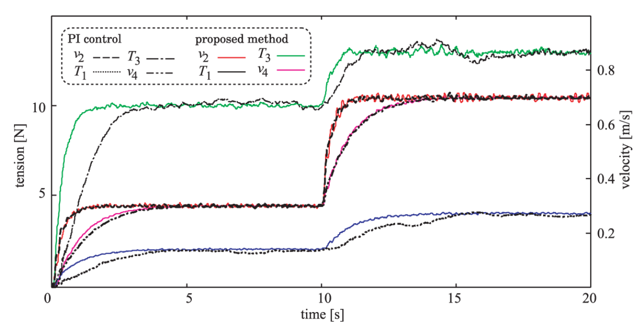


Fig. 9. Experimental results by using fixed PI parameters and using proposed method. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

5. Conclusions

In this study, we configured an experimental web transport system, and designed a control system based on overlapping decentralized control. The controller for each subsystem was designed using STPI control based on GMVC, and a mechanism was provided for online estimation of the system parameters required for control using OPSO. In addition, we performed experiments with the proposed method, which demonstrated that the control system designed in this study was capable of simultaneous control of web speed and tension.

The proposed method has the following features:

- Real-time processing can be implemented on a general-purpose computer.
- Only two parameters, λ and σ, are used to adjust the control characteristics of the overlapping subsystems.
- The adjustment parameters have clear physical meanings, which makes adjustment easy.
- The system parameters of the controlled plant are estimated online, so that there is no need for detailed models, and greater versatility is achieved than in conventional methods.

Topics for further research include comparative experiments with previously proposed H_∞ control algorithms, which requires the development of an accurate emulator. In addition, the control performance should be verified on a large-scale real system.

REFERENCES

1. Sakamoto T. Analysis and control of web tension control system. Trans IEE Jpn 1997;117-D:274–280. (in Japanese)
2. Sakamoto T, Tanaka S. Overlapping decentralized controller design for web tension control system. Trans IEE Jpn 1998;118-D:1272–1278. (in Japanese)
3. Mizoguchi K, Sakamoto T. Self-tuning decentralized controller design of web tension control system. Proc of EUROSIM Congress on Modeling and Simulation, 2010.
4. Wellstead PE, Zarrop MB. Self-tuning systems: control and signal processing. Wiley; 1991.
5. Omatsu S, Yamamoto T. Self-tuning control. Corona; 1996. (in Japanese)
6. Kennedy J, Eberhart RC. Particle swarm optimization. Proc of IEEE Int Conf on Neural Network IV, p 1942–1948, 1995.
7. Yamamoto T, Kaneda M. A design of self-tuning PID controllers based on the generalized minimum variance control law. Trans ISCIE 1998;11:1–9. (in Japanese)
8. Yamamoto T, Mitsukura Y, Kaneda M. A design of PID controllers using a genetic algorithm. Trans SICE 1999;35:531–537. (in Japanese)
9. Nishida T, Giannoccaro NI, Sakamoto T. Online identification of time-varying nonlinear systems by particle swarm optimization. Record of 2010 Joint Conference of Electrical and Electronics Engineers in Kyushu, CD-ROM 06-2A-03. (in Japanese)
10. Knittel D, Gigan D, Laroche E. Robust decentralized overlapping control of large scale winding systems. Proc of American Control Conf Anchorage, p 1805–1810, 2002.
11. Yulin X. Modeling and LPV control of web winding system with sinusoidal tension disturbance. Proc of Chinese Control and Decision Conference, p 3815–3820, 2009.
12. Knittel D, Laroche E, Gigan D, Koc H. Tension control for winding systems with two-degree-of-freedom H∞ controllers. IEEE Trans Ind Appl 2003;39:113–120.
13. Benlatreche A, Knittel D, Ostertag E. Robust decentralised control strategies for large-scale web handling systems. Control Eng Pract 2008;16:736–750.
14. Claveau F, Chevrel P, Knittel K. A 2DOF gain-scheduled controller design methodology for a multi-motor web transport system. Control Eng Pract 2008;6:609–622.
15. Knittel D, Arbogast A, Vedrines M, Pagilla P. Decentralized robust control strategies with model based feedforward for elastic web winding systems. Proc of American Control Conf, p 1968–1974, 2006.
16. Wang B, Zuo J, Wang M, Hao H. Model reference adaptive tension control of web packaging material. Proc of IEEE Int Conf on Intelligent Computation Technology and Automation, p 395–398, 2008.
17. Gaing Z. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Trans Energy Convers 2004;19:384–391.
18. Kaneko T, Matsumoto H, Mie H, Nishida H, Nakayama T. Offline tuning of positioning control system using particle swarm optimization considering speed controller. IEE Japan 2007;127-D:52–59. (in Japanese)
19. El-Telbany ME. Employing particle swarm optimizer and genetic algorithms for optimal tuning of PID controllers: A comparative study. ICGST-ACSE J 2007;7:49–54.
20. Oi A et al. Development of a PID parameter tuning method by PSO. Papers of Technical Meeting on IIC, IEEJ, p 52–59, 2008. (in Japanese)
21. Allaoua B, Gasbaoui B, Mebarki B. Setting up. DC motor speed control alteration parameters using par-

ticle swarm optimization strategy. Leonardo Electron J Pract Technol 2009;14:19–32.
22. GirirajKumar SM, Jayaraj D, Kishan AR. PSO based tuning of a PID controller for a high performance drilling machine. Int J Comput Appl 2010;1(19):12–18.
23. Kao C, Chuang C, Fung R. The self-tuning PID control in a slider-crank mechanism system by applying particle swarm optimization approach. Mechatronics 2006;16:513–522.
24. Kim T, Maruta I, Sugie T. Robust PID controller tuning based on the constrained particle swarm optimization. Automatica 2008;44:1104–1110.
25. Tanaka K, Murata Y, Nishimura Y, Rahman FA, Oka M, Uchibori A. Variable gain type PID control using PSO for ultrasonic motor. J JSAEM 2010;18:294–299. (in Japanese)
26. Pillay N, Govender P. Particle swarm optimization of PID tuning parameters. LAP LAMBERT Academic Publishing; 2010.
27. Yousuf MS. Nonlinear predictive control using particle swarm optimization. VDM Verlag Dr. Müller; 2010.
28. Ishigame A. Particle swarm optimization: global optimization technique. J SICE 2008;47:459–465. (in Japanese)
29. Martínes JLF, Gonzalo EG. The PSO family: deduction, stochastic analysis and comparison. Swarm Intell 2009;3:332–344.
30. Blackwell TM. Swarms in dynamic environments. LNCS 2003;2723:1–12.
31. Cui X, Potok TE. Distributed adaptive particle swarm optimizer in dynamic environment. Proc of IEEE Int Parallel and Distributed Processing Symposium, p 244–250, 2007.
32. Zhang X, Qin YDZ, Qin G, Lu J. A modified particle swarm optimizer for tracking dynamic system. LNCS 2005;3612:592–601.

AUTHORS (from left to right)



Takeshi Nishida (member) received a bachelor's degree from Kyushu Institute of Technology (School of Engineering) in 1998, completed the doctoral program in 2002, and became a research associate. He has been an associate professor since 2007. His research interests are outdoor mobile robots. He holds a D.Eng. degree, and is a member of RSJ, SICE, JNNS, and other societies.

Tetsuzo Sakamoto (senior member) completed the doctoral program at Kyushu University in 1984 and joined the faculty as a research associate. He became a research associate at Kyushu Institute of Technology in 1985, then a lecturer and an associate professor; professor since 2002. His research interests are linear drives, maglev and web tension systems. He holds a D.Eng. degree, and is a member of SICE and other societies.

Nicola Ivan Giannoccaro (nonmember) received a bachelor's degree from Polytechnic University of Bari (Italy) in 1996 and completed the doctoral program in 2001. He is now a researcher at University of Salento. He was a visiting researcher at Kyushu Institute of Technology in 2006 and 2010. His research interest is mechatronics. He holds a Ph.D. degree.