

# パラメータ推定器を切り替えて用いるウェブ搬送装置のセルフチューニングPI制御

九州工業大学 西田 健 崎村 悠登 坂本 哲三

## Self-Tuning PI Control of Web Transport System with Switching Two Parameter Estimators

Takeshi NISHIDA, Yuto SAKIMURA, and Tetsuzo SAKAMOTO

Kyushu Institute of Technology

**Abstract :** A self-tuning PI controller which employs both the recursive least square algorithm and an on-line particle swarm optimization (OPSO) algorithm to estimate the web transport system parameters is proposed. The controller is implemented to an experimental web transport system which has 12 rolls, 4 motors and 4 tension sensors, and the performance is verified by experiments. Several experimental results are shown to evaluate the performance of the proposed controller.

### 1. はじめに

フィルム状素材の製造や加工に用いられるウェブ搬送装置では、多数の駆動ローラを制御することによって、搬送物体の速度や張力を高い精度で一定に保つ。一般にウェブ搬送系は大規模であるため、分散制御系が構成される [1] が、各サブシステムは相互干渉に敏感であり、外乱やパラメータ変動に脆弱であるという問題が発生する。現在までに、相互干渉に関する問題を解決する手法として、重複分割分散制御 [2] が提案され、また、システムパラメータの推定を OPSO (on-line particle swarm optimization) [3] により行い、その結果を利用するセルフチューニング PID 制御法 [4] が提案されている。さらに、この手法はウェブ搬送実験系に実装するための修正が加えられ、12 のローラと 4 つのモータ、4 つの張力センサを有するウェブ搬送実験機により提案手法の有効性が検証されている [5]。本稿では、先行研究における問題の 1 つである、OPSO による推定に起因する初期応答のむだ時間を解消するために、RLS (recursive least square) 法を併用する推定器の構成法を提案する。

### 2. ウェブ搬送系

ウェブ搬送実験システムの概要を Fig. 1 に示す。フィルムはアンワインダから引き出され、リードセクションおよびドローロールを経てワインダで巻きとられる。この系の入力は、時刻  $t$  での各モータへのトルク  $u(t) \triangleq (u_1(t) \ u_2(t) \ u_3(t) \ u_4(t))^T$  である。各サブシステムの計測値は、ドローロールとワインダに組み込まれた張力センサの計測値 ( $T_1(t)$ ,  $T_3(t)$ ) とモータに取り付けられたロータリエンコーダにより計測される速度 (リードセクション  $v_2(t)$  とワインダ  $v_4(t)$ ) であり、これらをまとめて出力を  $y(t) \triangleq (T_1(t) \ v_2(t) \ T_3(t) \ v_4(t))^T$  と表

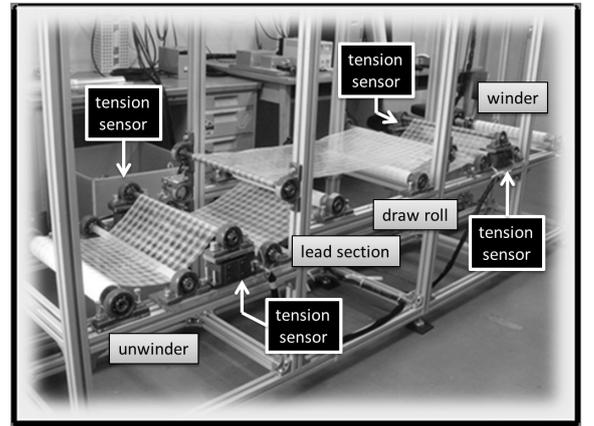
す。重複分割分散制御 [2] に基づき、各サブシステムへの仮想入力を  $\tilde{u}(t) \triangleq (\tilde{u}_1(t) \ \tilde{u}_2(t) \ \tilde{u}_3(t) \ \tilde{u}_4(t))^T$  とする。ここで、

$$u(t) = N\tilde{u}(t), \quad N \triangleq \begin{pmatrix} \frac{r_1}{J_1} & \frac{r_2}{J_2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{r_3}{J_3} & \frac{r_4}{J_4} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \quad (1)$$

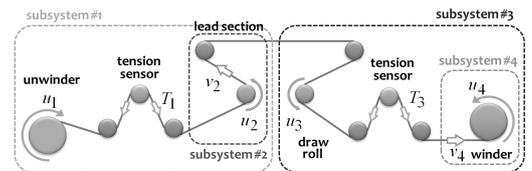
である。重複分割分散制御の導入により、各サブシステム間の相互干渉を外乱として扱うことが可能となる。

次に、実験機に対するコントローラを構築するために、各サブシステムを次のようにモデル化する。

$$A_j(z^{-1})y_j(k) = z^{-k_m}B_j(z^{-1})u_j(k), \quad (2)$$



(a) Overview of the experimental system.



(b) Scheme of the experimental system.

Fig. 1 The experimental web transport system.

ここで  $y_j(k)$  は離散時刻  $k$  における  $j$  番目のサブシステムの出力であり,  $k_m$  はむだ時間の最小見積りである. 本研究の実験では  $k_m = 1$  とした. 同様のコントローラを各サブシステムに対して構成するので, 以下では簡単のため  $j$  を除いて議論する. 本研究では, 各サブシステムは, 一次のシステム  $G(s) = K/(T_a s + 1)$  と ZOH 要素  $H(s) = (1 - \exp(-T_s s))/s$  の合成から構成される以下のような  $z$  変換関数によって近似可能であると仮定する.

$$\mathcal{Z}[H(s) \cdot G(s)] = \frac{K \{\exp(T_s/T_a) - 1\} z^{-1}}{1 - \exp(-T_s/T_a) z^{-1}}, \quad (3)$$

ここで  $T_s$ ,  $T_a$ , および  $K$  は, それぞれサンプリング時間, 時定数, ゲインを表す. これらの関係より, 式 (2) の系の極は

$$A(z^{-1}) = 1 - \exp(-T_s/T_a) z^{-1} \triangleq 1 + a_1 z^{-1} \quad (4)$$

$$B(z^{-1}) = K \{\exp(T_s/T_a) - 1\} \triangleq b_0 \quad (5)$$

と表される.

### 3. GMVC に基づく ST-PI 制御

GMVC によって算出される入力を PID パラメータに変換する手法 [6] を適用する. 制御入力  $u(k)$  は以下で与えられる.

$$F(z^{-1})y(k) + \{E(z^{-1})B(z^{-1}) + \lambda\} \Delta u(k) - R(z^{-1})w(k) = 0 \quad (6)$$

ここで  $w(k)$  は目標値,  $\lambda$  は制御入力の重みパラメータであり,  $\Delta \triangleq 1 - z^{-1}$  である.  $E(z^{-1})$  と  $F(z^{-1})$  は以下の Diophantine 方程式で与える.

$$P(z^{-1}) = \Delta A(z^{-1})E(z^{-1}) + z^{-(k_m+1)}F(z^{-1}) \quad (7)$$

$$E(z^{-1}) = 1 + e_1 z^{-1}, \quad F(z^{-1}) = f_0 + f_1 z^{-1} \quad (8)$$

また, システムの立ち上がり時間と減衰率を指定するために,

$$P(z^{-1}) \triangleq 1 + p_1 z^{-1} + p_2 z^{-2} \quad (9)$$

$$p_1 \triangleq -2e^{-\frac{\rho}{2\mu}} \cos \frac{\sqrt{4\mu-1}}{2\mu} \rho, \quad p_2 \triangleq e^{-\frac{\rho}{\mu}}$$

$$\rho \triangleq T_s/\sigma, \quad \mu \triangleq 0.25(1 - \delta) + 0.51\delta$$

と設定する. ここで, オーバースhootを防ぐために  $\delta = 0$  と設定するので,  $P(z^{-1})$  の設計パラメータは  $\sigma$  のみになり, 上述の式は次のように簡略化できる.

$$p_1 = -2e^{-\frac{2T_s}{\sigma}}, \quad p_2 = e^{-\frac{4T_s}{\sigma}} \quad (10)$$

一方, 離散 PID 制御は次のように記述できる.

$$\Delta u(k) = k_p \left( \Delta + \frac{T_s}{T_I} + \frac{T_D}{T_s} \Delta^2 \right) e(k) \quad (11)$$

ここで  $e(k) \triangleq w(k) - y(k)$  である. この式は次のように変形できる.

$$C(z^{-1})y(k) + \Delta u(k) - C(z^{-1})w(k) = 0, \quad (12)$$

ここで,

$$C(z^{-1}) \triangleq c_0 + c_1 z^{-1} + c_2 z^{-2}$$

$$= k_p \left( 1 + \frac{T_s}{T_I} + \frac{T_D}{T_s} \right) - k_p \left( 1 + \frac{2T_D}{T_s} \right) z^{-1} + \frac{k_p T_D}{T_s} z^{-2}$$

定常応答に重点を置くために  $E(z^{-1})B(z^{-1}) \simeq E(1)B(1)$  とすると, 式 (6) は

$$\frac{F(z^{-1})}{\nu} y(k) + \Delta u(k) - \frac{R(z^{-1})}{\nu} w(k) = 0, \quad (13)$$

となる. ここで  $\nu \triangleq B(1)E(1) + \lambda$  である. 式 (12) と (13) の比較より, GMVC と PID の制御則を近似的に等価とする関係が以下のように求まる.

$$R(z^{-1}) = F(z^{-1}), \quad C(z^{-1}) = F(z^{-1})/\nu$$

PI パラメータは推定されたシステムパラメータ  $\hat{\theta}(k)$  を利用して以下のように求めることができる.

$$k_p = -f_1/\nu \geq 0, \quad T_I = -f_1 T_s / (f_0 + f_1) \geq 0$$

ここで

$$f_0 = p_2 + \hat{a}_1(k) + (1 - \hat{a}_1(k))e_1, \quad f_1 = e_1 \hat{a}_1(k)$$

$$\nu = \hat{b}_0(k)(e_1 + 1) + \lambda, \quad e_1 = p_1 - \hat{a}_1(k) + 1$$

である. ただし, 本研究の対象システムの次元が 1 次であるため  $T_D$  は算出されない.

### 3.1 システムパラメータ推定

#### 3.1.1 逐次最小二乗法 (RLS)

システムパラメータは忘却付き RLS 法によって以下のように推定できる.

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \frac{\mathbf{P}(k-1)\boldsymbol{\psi}(k)}{\lambda_f(k) + \boldsymbol{\psi}^T(k)\mathbf{P}(k-1)\boldsymbol{\psi}(k)} \epsilon(k)$$

$$\mathbf{P}(k) = \frac{1}{\lambda_f(k)} \left\{ \mathbf{P}(k-1) - \frac{\mathbf{P}(k-1)\boldsymbol{\psi}(k)\boldsymbol{\psi}^T(k)\mathbf{P}(k-1)}{\lambda_f(k) + \boldsymbol{\psi}^T(k)\mathbf{P}(k-1)\boldsymbol{\psi}(k)} \right\}$$

$$\epsilon(k) = y(k) - \boldsymbol{\psi}^T(k)\hat{\theta}(k-1)$$

ここで  $\hat{\theta}(k) \triangleq (\hat{a}_1(k) \hat{b}_0(k))^T$ ,  $\boldsymbol{\psi}(k) \triangleq (-y(k-1) \tilde{u}(k-1))^T$  である. また, 忘却係数は  $\lambda_f(k) = e^{-1/\tau_f} \lambda_f(k-1) + (1 - e^{-1/\tau_f})$ , である. 後述の実験では, 初期パラメータを  $\hat{\theta}(0) = \hat{\theta}_0$ ,  $\mathbf{P}(0) = \gamma \mathbf{I}$  と設定した. ここで  $\gamma$  は正の定数であり, 本研究では  $\gamma = 1$  とした. また,  $\hat{\theta}_0 = \mathbf{0}$ ,  $\tau_f = 300$ ,  $\lambda_f(0) = 0.97$  と設定した. これらのパラメータは推定の安定性と正定時間などを考慮して定めた.

### 3.1.2 OPSO

OPSO[3] は時変システムの適応同定が可能なアルゴリズムであり、これを用いてオンラインパラメータ推定器を構築する。探索空間内の粒子の位置を  $\mathbf{x}_m(k) \triangleq (a_{1m}(k) \ b_{0m}(k))^T$  とし、その速度を  $\mathbf{v}_m(k) \in \mathbb{R}^2$  とする。ここで  $m = [1, M] \in \mathbb{N}_+$  は粒子の番号を表す。また、 $\hat{\mathbf{x}}_m(k)$  を  $m$  番目の粒子の最良解 ( $pbest$  と呼ぶ) とする。

まず、前時刻の  $pbest$ 、すなわち各粒子の  $\hat{\mathbf{x}}_m(k-1)$  を現時刻の評価関数によって再評価する。

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

各粒子の位置と速度を次のように更新する。

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

ここで  $\omega, c_1, c_2$  は設計パラメータである。この時、 $pbest$  の位置も次のように更新する。

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

また、評価値  $f_k(\hat{\mathbf{x}}_m(k))$  を格納する。最後に、これらの値を用いて、時刻  $k$  における最適な結果もしくは OPSO の推定値は以下のように導出される。

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

ここで  $\hat{\mathbf{x}}^g(k)$  は  $gbest$  と呼ばれる大局解である。

以上の OPSO アルゴリズムを用いてシステムパラメータ同定を行う際には、以下の評価関数を用いる。

$$f_k(\mathbf{x}_m(k)) = \sum_{i=0}^{I_e} \left\| \mathbf{y}(k-i) - \psi^T(k-i) \mathbf{x}_m(k) \right\|^2$$

ここで  $I_e$  は評価ステップ数である。この値を大きく設定すると観測ノイズの影響を減らすことができるが、計算コストが上昇し、システムパラメータの更新がコントローラへ反映されるために必要なむだ時間が長くなる。以降の実験では  $I_e = 100$  と設定した。

### 3.1.3 推定器の切り替え

RLS 法では時刻  $k = 1$  から推定を開始でき、収束も比較的高速であるが、時変システムのパラメータ変化に対する適応は困難である。一方、OPSO による推定は時変システムに対する適応が可能であるが、 $I_e$  ステップの入出力データセットが評価関数の計算に必要なため、

初期応答において推定のためのむだ時間が生ずる。したがって、本研究では二つの推定手法を組み合わせる手法を提案する。すなわち、初期時刻においては RLS により推定を行い、定常応答に移行した後はシステムパラメータの変動に備えるために OPSO による推定に切り替える。

目標値との誤差の収束を判定するために、 $e_{th} \triangleq r_{th} w(k)$  を制御偏差  $e(k) \triangleq w(k) - y(k)$  のしきい値として用いる。ここで  $r_{th}$  は収束の割合を表す変数であり、実験では 0.05 と設定する。まず推定を RLS によって開始し、サブシステムは  $e(k) > e_{th}$  を満たす間はその推定を採用する。次に、 $e(k) \leq e_{th}$  がすべてのサブシステムで満たされた場合に、その時刻を  $k_e$  として保持し、OPSO による推定をバックグラウンドジョブとして開始する。さらに、その時点での推定結果  $(\hat{a}_1(k_e), \hat{b}_0(k_e))$  を初期のパラメータとして RLS 推定器から OPSO 推定器に渡す。もし、OPSO 推定器がバックグラウンドで実行されている最中に条件  $e(k) \leq e_{th}$  が満たされなくなった場合には、OPSO と  $I_e$  のカウントを停止する。応答が順調で時刻が  $k = k_e + I_e$  となった場合には、RLS による推定は停止し OPSO による推定に切り替える。以上の切り替えの流れを図 2 に示す。この切り替えにより OPSO の推定におけるむだ時間を無くすることができる。さらに、RLS によって初期値が与えられるので、OPSO の探索領域を狭くすることができる。

## 4. 実験結果

OPSO の粒子数は  $M = 100$ 、GMVC の設計パラメータは  $\lambda = 10$  および  $\sigma = 1$  とした。目標値は  $w_1 = w_4 = 0.3$  [m/s]、 $w_2 = 2$  [N]、および  $w_3 = 10$  [N] とした。また、15 [s] 後には  $w_1 = w_4 = 0.5$  [m/s]、 $w_2 = 3$  [N]、および  $w_3 = 12$  [N] と変更した。初期値はすべてのサブシステムで  $\hat{a}_1(0) = 0$  および  $\hat{b}_0(0) = 0$  とした。実験の出力と入力推移を Fig. 3(a), (b) に示す。これらより、出力値は目標値にオーバーシュート無く収束しており、定常応答も安定していることがわかる。また、目標値の変化に適応して、遅れ時間が発生していないことがわかる。次に、各サブシステムの推定パラメータの時間推移

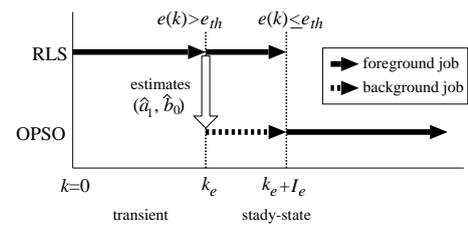


Fig. 2 Switching flow of the estimators.

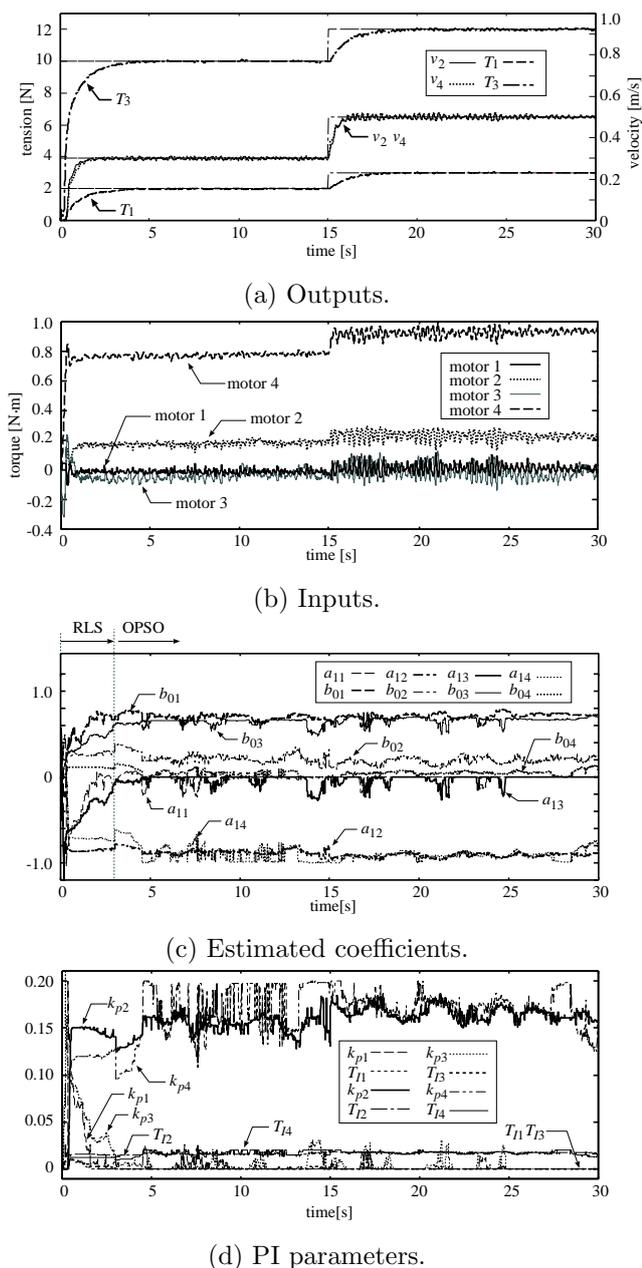


Fig. 3 Time evolution of the experimental results with proposed estimators.

が Fig. 3(c) に示されている。これらの結果より、初期応答におけるむだ時間の発生が無いことがわかる。また、図 3(d) には調整された PI パラメータの推移を示した。これより、推定されたパラメータに基づいて PI 制御器のゲインが自動的に調整されていることがわかる。

一方、図 4(a) と (b) は、それぞれ推定器として OPSO 法のみもしくは RLS 法を利用した場合の結果を示している。これらの図より、前者では入力変化への適応はできていないものの評価ステップ  $I_e$  だけのむだ時間が発生しており、後者ではむだ時間の発生はないものの、入力の変

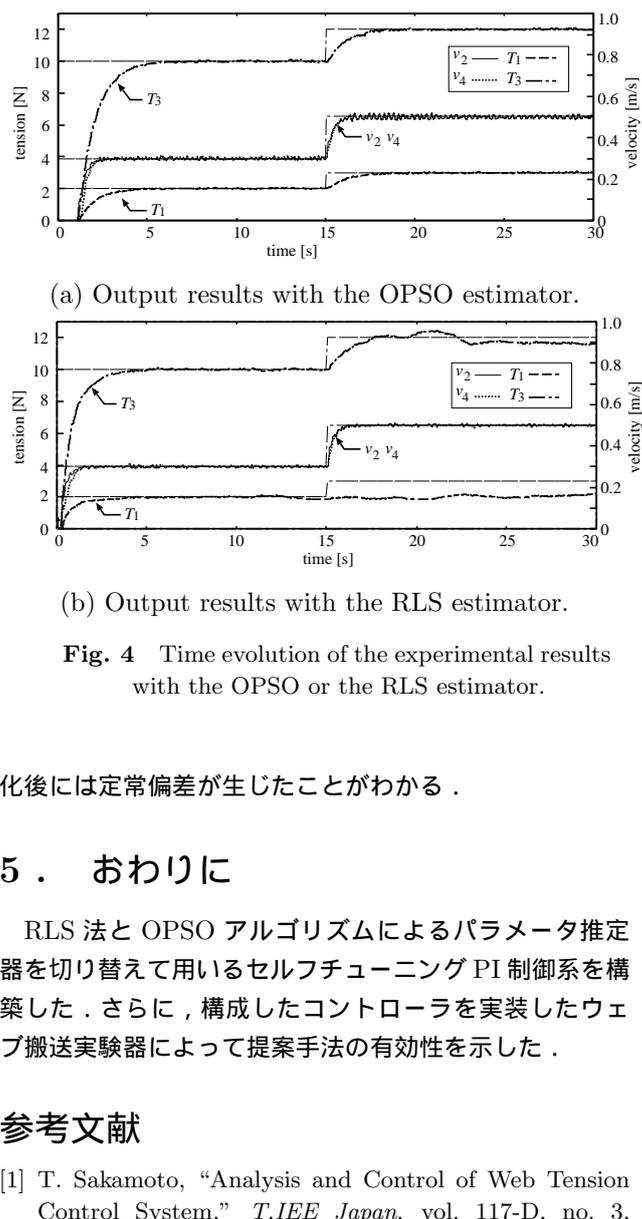


Fig. 4 Time evolution of the experimental results with the OPSO or the RLS estimator.

化後には定常偏差が生じたことがわかる。

## 5. おわりに

RLS 法と OPSO アルゴリズムによるパラメータ推定器を切り替えて用いるセルフチューニング PI 制御系を構築した。さらに、構成したコントローラを実装したウェブ搬送実験器によって提案手法の有効性を示した。

## 参考文献

- [1] T. Sakamoto, "Analysis and Control of Web Tension Control System," *T.IEE Japan*, vol. 117-D, no. 3, pp.274–280, 1997.
- [2] T. Sakamoto and S. Tanaka, "Overlapping Decentralized Controller Design for Web Tension Control System," *T.IEE Japan*, vol. 118-D, no.11, pp.1272–1278, 1998.
- [3] 西田, 坂本, "時変システムのオンライン同定のための適応 PSO," *電学論*, vol. 131-C, no. 9, pp. 1642–1649, 2011.
- [4] K. Mizoguchi and T. Sakamoto, "Self-tuning decentralized controller design of web tension control system," *Proc. of EUROSIM congress on Modeling and Simulation*, CR-ROM, 2010.
- [5] 西田, 坂本, ジアノカッロ, "適応 PSO を用いるセルフチューニング PI コントローラによるウェブ搬送系の重複分散制御," *電学論*, vol. 131-C, no. 12, 2011.
- [6] T. Yamamoto and M. Kaneda, "A Design of Self-Tuning PID Controllers Based on the Generalized Minimum Variance Control Law," *Trans. ISCIE*, vol. 11, no. 1, pp.1–9, 1998.