

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

○西田 健, 崎村 悠登, 坂本 哲三

九州工業大学大学院 工学研究院

1. はじめに

ウェブとは

薄く長い素材の一般名称

紙・板紙, 繊維, プラスティックフィルム, セラミックシート, 炭素繊維複合材, 不織布, 合成紙, 金属箔, 鋼板, 液晶パネル用各種光学フィルム, 固形高分子膜, 医療用人工生体膜, etc..

ウェブの製造技術

コンバーティング技術

コーティング, ラミネート, プリンティングなど, 学術的に目覚ましく発展している.

ウェブハンドリング技術

- ウェブの巻出し, 搬送, 巻き取りを行う技術
- 生産現場の経験の積み重ねにより練り上げられてきた
- 学術的なバックグラウンドに乏しい

1. はじめに

ウェブ搬送装置

ウェブ搬送装置は、多数の駆動ローラを制御することによってウェブの速度や張力を高い精度で一定に保つ。

- 破損や破断，巻き取りムラを抑制
- 安定した品質の維持

制御を困難にする要素

- 1 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。
 - サブシステム間に相互干渉が発生
 - 外乱やパラメータ変動に脆弱
- 2 時変かつ非線形なダイナミクスを有する。
- 3 精密なモデリングやエミュレートが困難。

1. はじめに

これまでの研究

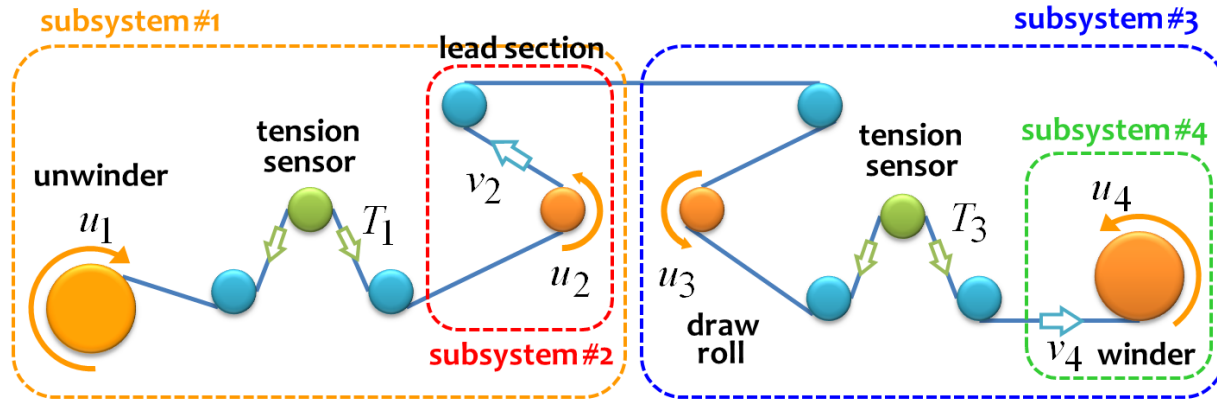
- 1 重複分割分散制御**
サブシステム間の相互干渉力の解消
- 2 新しい適応PSO (OPSO) の提案**
メタヒューリスティクスによるオンラインシステムパラメータ推定
- 3 セルフチューニングPID (STPID) 制御システム**
一般化最小分散制御 (GMVC) とOPSOを用いる制御手法
- 4 ウェブ搬送実験装置への制御系実装**
12のロールと4つのモータ, 4つの張力センサを有する実験機

本研究の目的

- 5 望ましい過渡応答特性と適応性能を両立する制御系の提案**
これまでに構築したセルフチューニングPI制御系の改善

2. 制御システム—重複分割分散制御系—

サブシステムの伝達関数



1st と 3rd サブシステム

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

$$G_i(s) \triangleq \frac{P_v(s)/L_i}{1 + P_v(s)/L_i (r_i^2/J_i s + r_{i+1}^2/J_{i+1} s)}$$

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

2nd と 4th サブシステム

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

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

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

2. 制御システム

近似モデル

各サブシステムを一次遅れと零次ホールド要素で表される伝達関数で近似

$$G(s) = \frac{K}{T_a s + 1}$$

$$H(s) = \frac{1 - \exp(-T_s)}{s}$$



$$Z[H(s)G(s)] = \frac{K\{\exp(T_s/T_a) - 1\}z^{-1}}{1 - \exp(-T_s/T_a)z^{-1}} \triangleq \frac{z^{-1}B(z^{-1})}{A(z^{-1})}$$

離散モデル

$$y(k) = \frac{z^{-1}B(z^{-1})}{A(z^{-1})}u(k)$$

$$A(z^{-1}) \triangleq 1 - \exp(-T_s/T_a)z^{-1} \triangleq 1 + a_1z^{-1}$$

$$B(z^{-1}) \triangleq K\{\exp(T_s/T_a) - 1\}z^{-1} \triangleq b_0z^{-1}$$



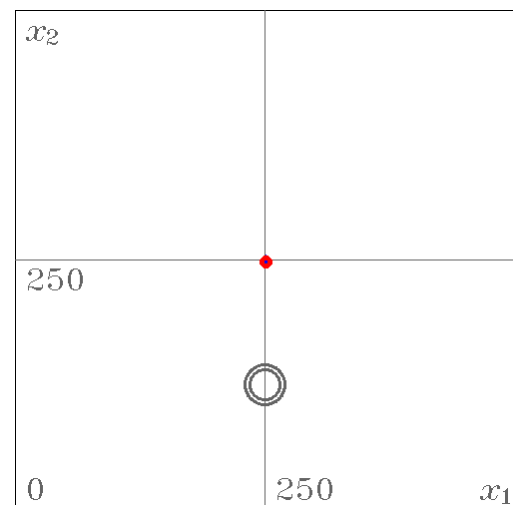
システムパラメータ(a_1 b_0)
が得られればGMVFCによつて適切な制御が可能

3. システムパラメータのオンライン推定 OPSO

PSO (particle swarm optimization)

メタヒューリスティクスの一手法であり, 確率的な最適解の探索を行う

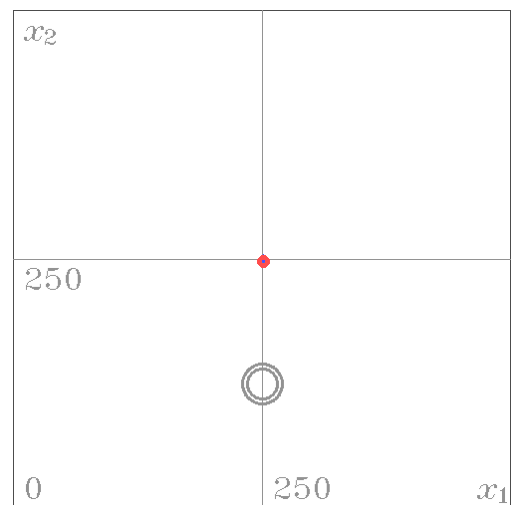
静的な最適解探索問題の解法として提案された



OPSO (online-type particle swarm optimization)

PSOの改良により, 動的関数の最適解探索を行うことが可能な手法.

オンラインで実行可能であり, 収束性能が良い



3. システムパラメータのオンライン推定

粒子の評価

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

評価ステップ数 I_e

出力 $y(k-i)$

推定出力 $\boldsymbol{\psi}^T(k-i)\boldsymbol{\theta}_m(k)$

第 m 粒子の推定 $\boldsymbol{\theta}_m(k)$

出力と入力の組み合わせ $\boldsymbol{\psi}^T(k-i) \triangleq (-y(k-i-1) \ u(k-i-1))$

I_e の値を大きく設定すると
計測ノイズの影響を低減化できる

trade
off

- 計算量が増加する
- 推定の無駄時間が増加する

- 実験では $I_e = 100$ に設定
- 実験機のサンプリング時間は $T_s = 10$ [ms]
現在までの1秒間の応答を利用して適応的な推定を行う
- ! 推定にむだ時間や遅れが生ずる

3. システムパラメータのオンライン推定 ～RLS法～

RLS (逐次最小二乗近似)

推定ベクトル: $\hat{\boldsymbol{\theta}}(k) = (\hat{a}_1, \hat{b}_0)$

忘却付き逐次最小二乗法

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

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

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

忘却率 $\lambda_f(k) = \exp(-1/\tau_f) \lambda_f(k-1) + (1 - \exp(-1/\tau_f))$

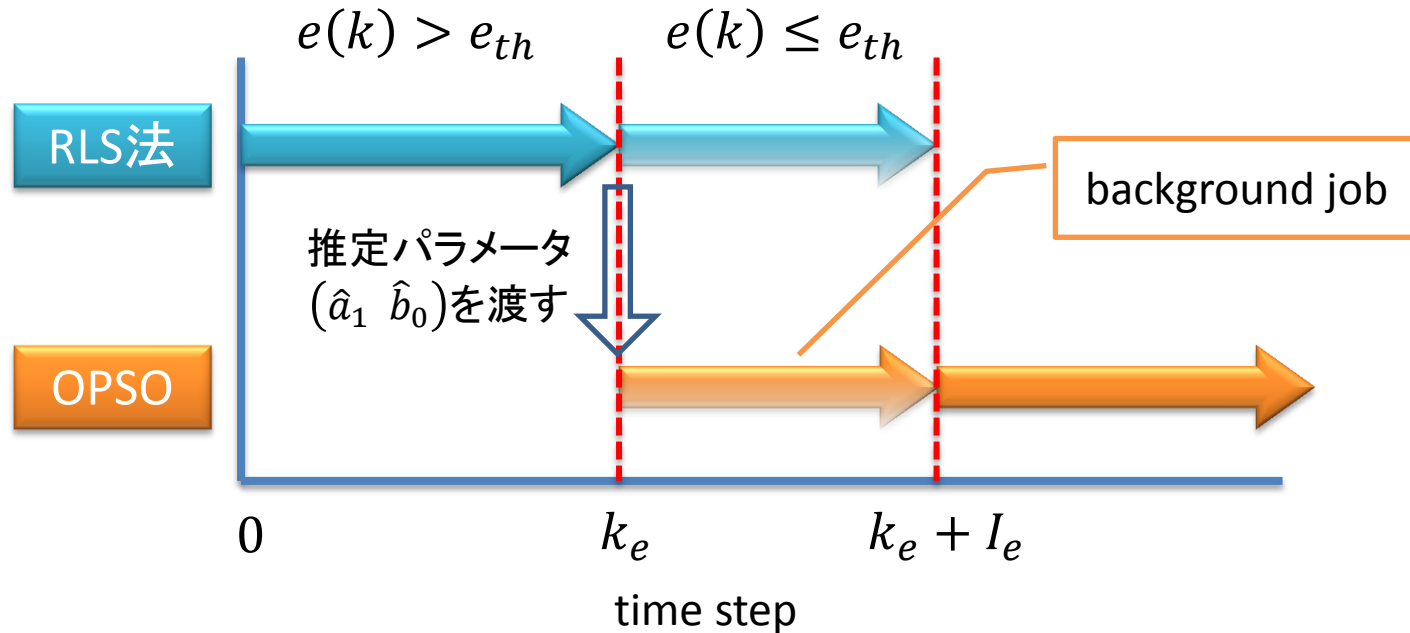
初期値 $\hat{\boldsymbol{\theta}}(0) = \hat{\boldsymbol{\theta}}_0 \quad \mathbf{P}(0) = \gamma \mathbf{I}$



推定のためにむだ時間や遅れは生じないが、適応能力は低い

3. システムパラメータのオンライン推定 ～RLS法～

RLSとRLSの切り替え



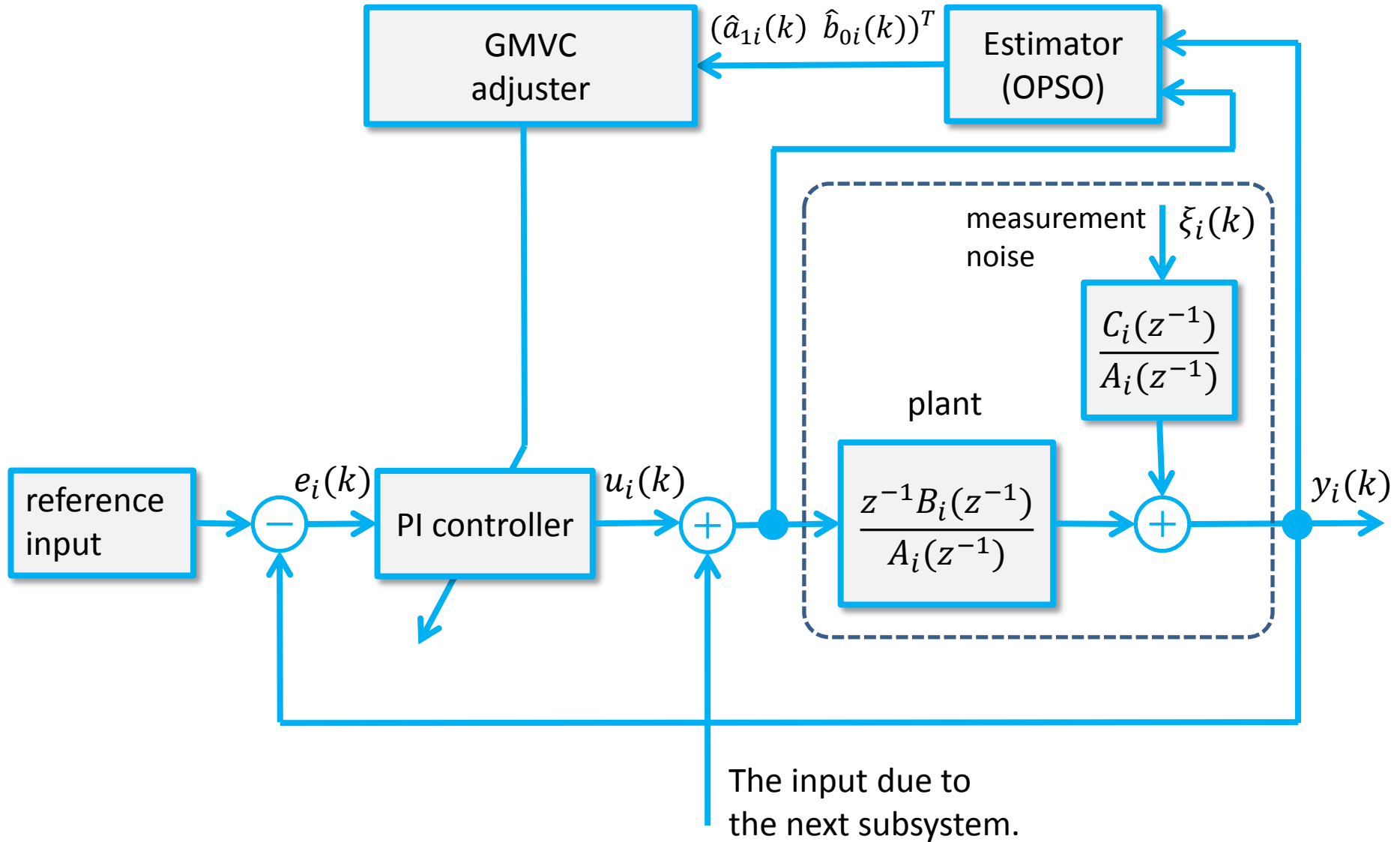
$e(k) \triangleq w(k) - y(k)$: 目標値との誤差

$e_{th} \triangleq r_{th}w(k)$: 目標値への収束を判定するしきい値



RLSの即応性とOPSOの適応性の両立を目指す

4. 制御手法





4. 制御手法

GMVC

$$\Delta u(k) = \frac{R(z^{-1})}{v} w(k) - \frac{F(z^{-1})}{v} y(k)$$

PI controller

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

 $F(z^{-1})/v = C(z^{-1})$ $R(z^{-1}) = F(z^{-1})$ 

GMVCの制御入力 → PIパラメータ

$$k_p = -f_1/v$$

$$T_I = -f_1 T_s / (f_0 + f_1)$$

$$\left\{ \begin{array}{l} f_0 = p_2 + \hat{a}_1(k) + (1 - \hat{a}_1(k))e_1 \\ f_1 = e_1 \hat{a}_1(k) \\ e_1 = p_1 - \hat{a}_1(k) + 1 \\ v = \hat{b}_0(k)(1 + e_1) + \lambda \end{array} \right.$$

λは設計
パラメータ



一次遅れ系で各サブシステムを近似したので、 T_D は算出されない。

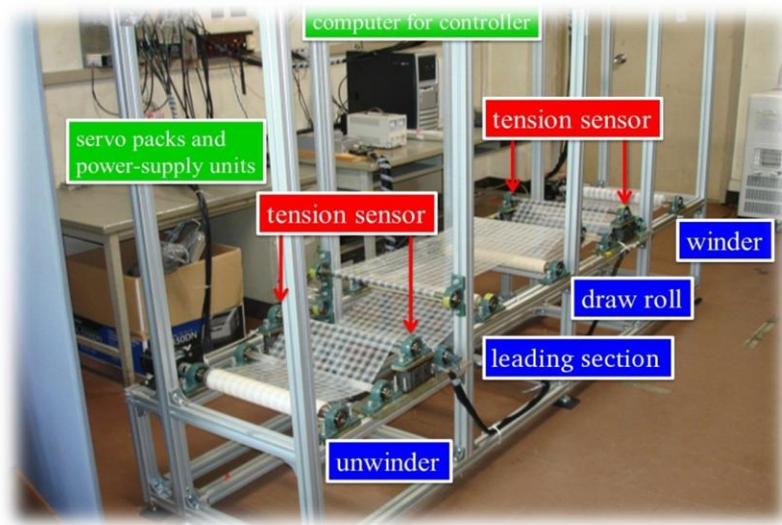
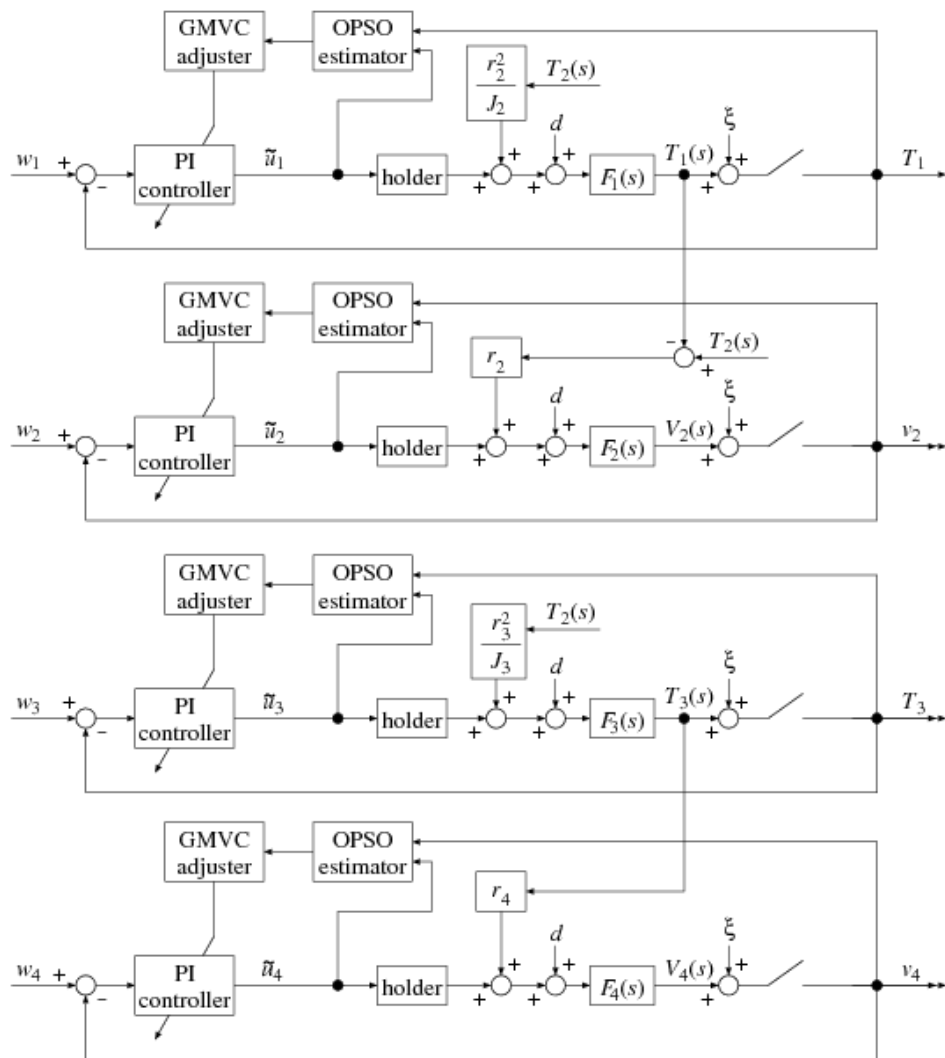


システムパラメータ(\hat{a}_1 \hat{b}_0)の推定に基づいてPIパラメータを自動的に調整する。



PID制御によって運用されているシステムへの導入が容易。

5. 実験



OSPOの粒子数	$M = 100$
----------	-----------

評価ステップ数	$I = 100$
---------	-----------

GMVC-PID の設計パラメータ	$\lambda = 10$
-------------------	----------------

$0 \leq t \leq 15$ [m/s]	$15 < t \leq 30$ [m/s]
--------------------------	------------------------

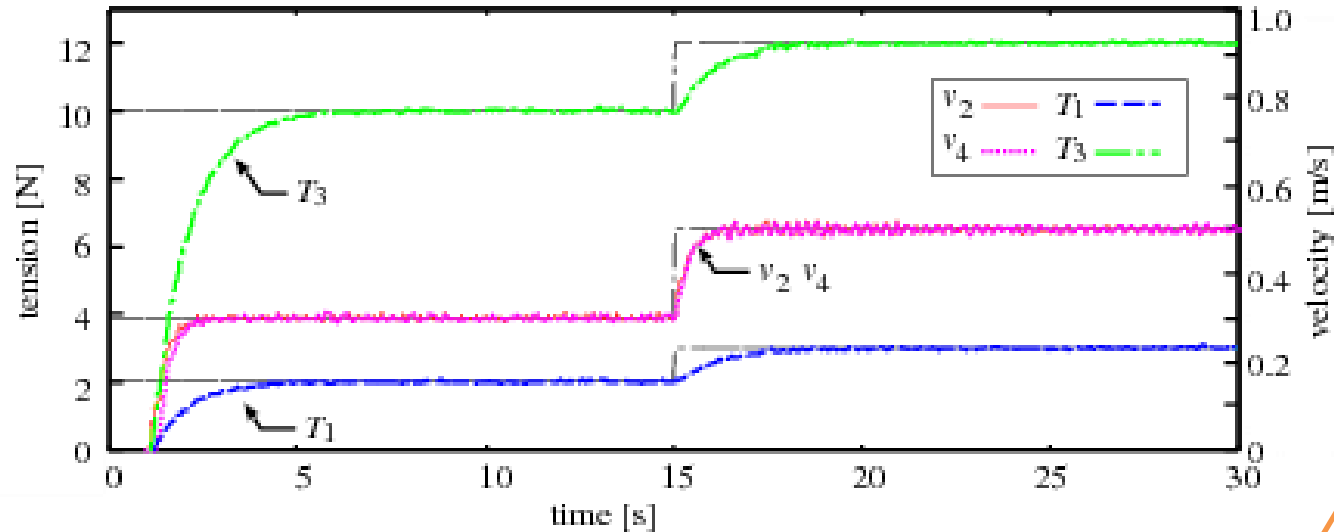
$w_1 = 0.3$ [m/s]	$w_1 = 0.5$ [m/s]
-------------------	-------------------

$w_2 = 2$ [N]	$w_2 = 3$ [N]
---------------	---------------

$w_3 = 10$ [N]	$w_3 = 12$ [N]
----------------	----------------

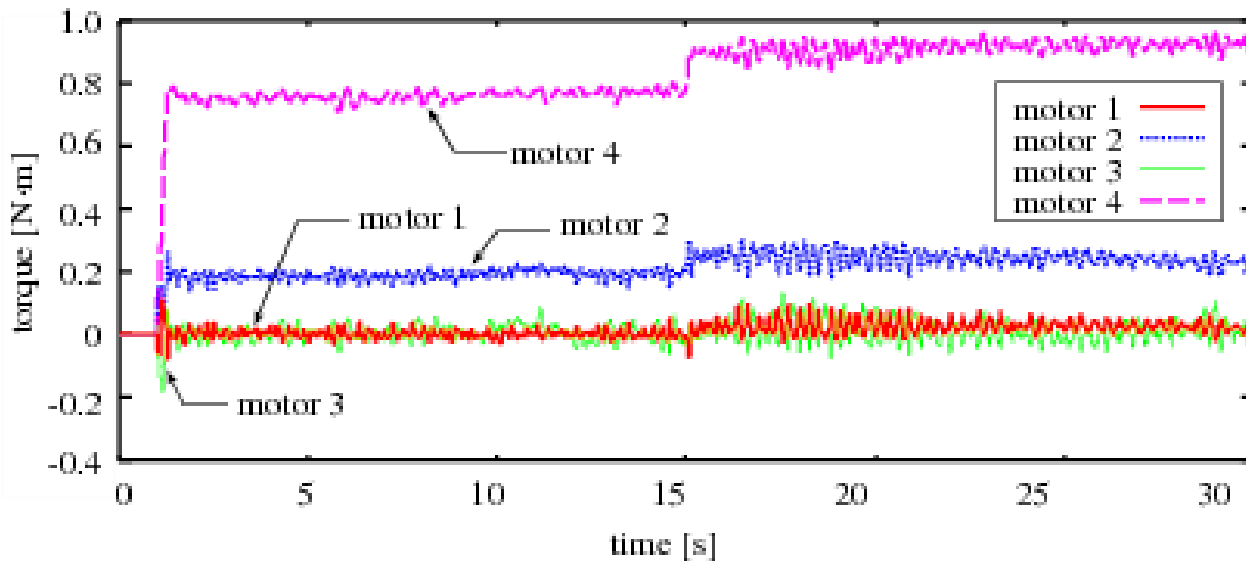
$w_4 = 0.3$ [m/s]	$w_4 = 0.5$ [m/s]
-------------------	-------------------

5. 実験結果 OPSOをパラメータ推定器として利用した場合



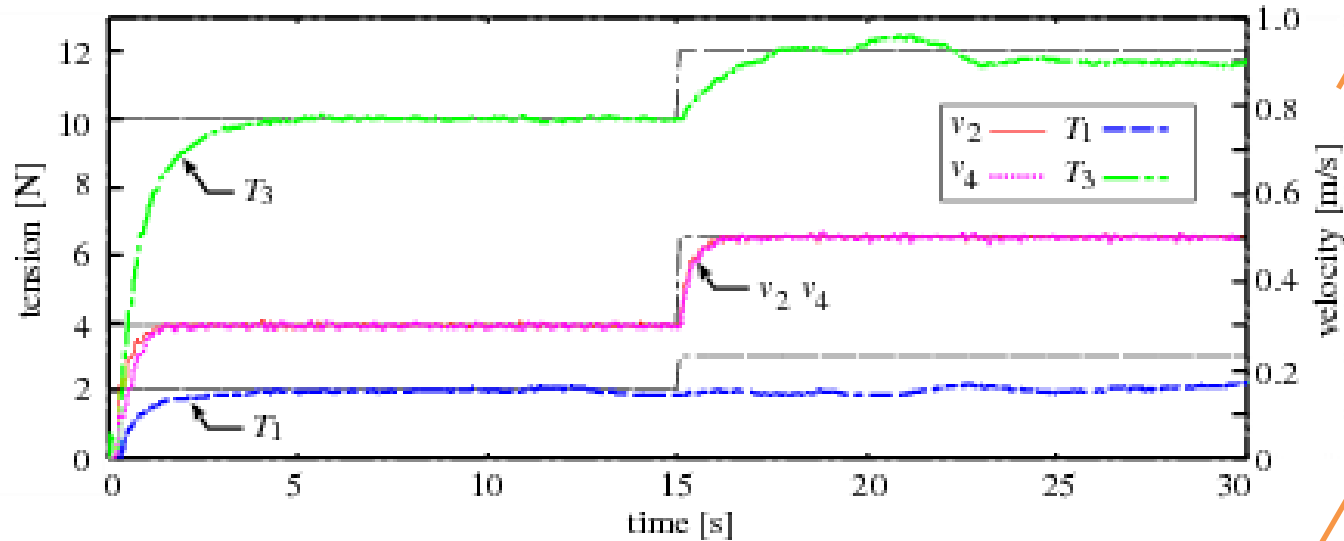
出力の時間推移

入力の時間推移



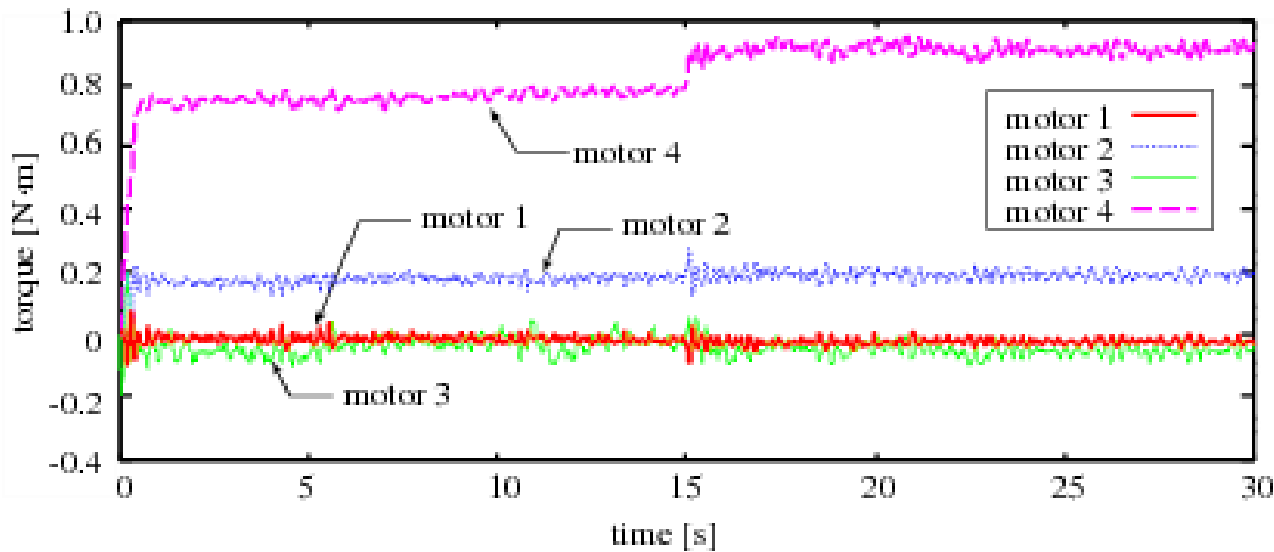
- ✓ オーバーシュート
- ✓ 定常偏差
- ✓ 過渡応答特性
- ✓ 適応性能
- ✗ むだ時間

5. 実験結果 RSLをパラメータ推定器として利用した場合



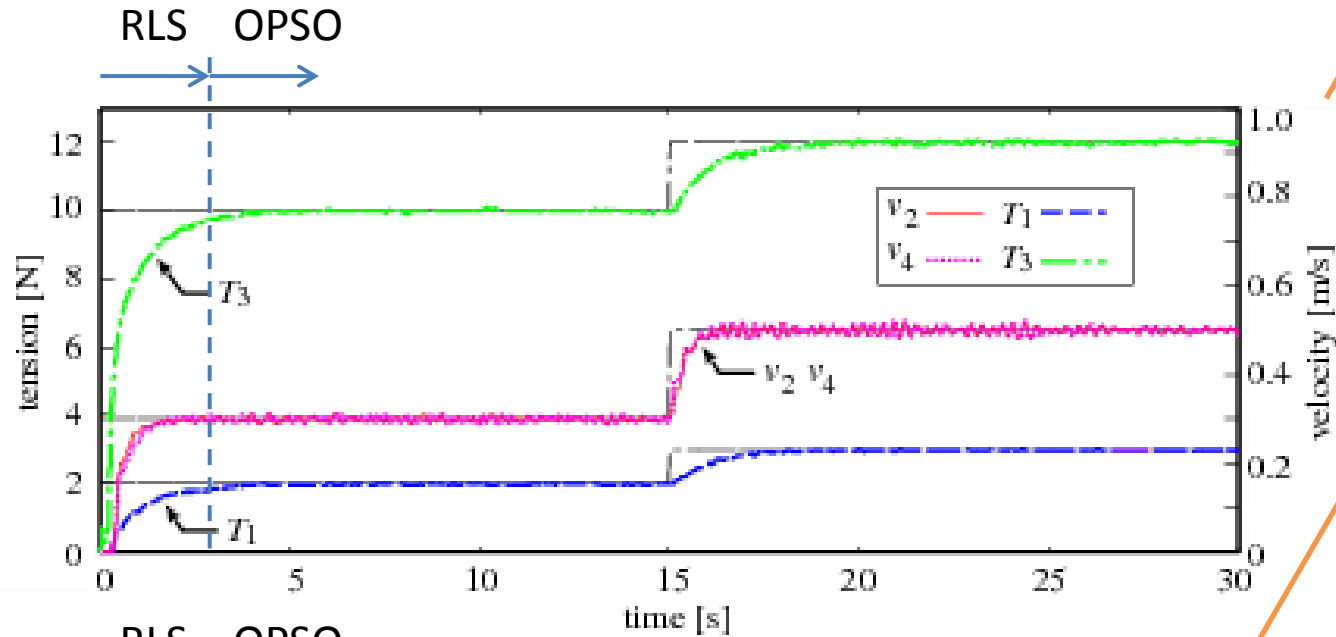
出力の時間推移

入力の時間推移



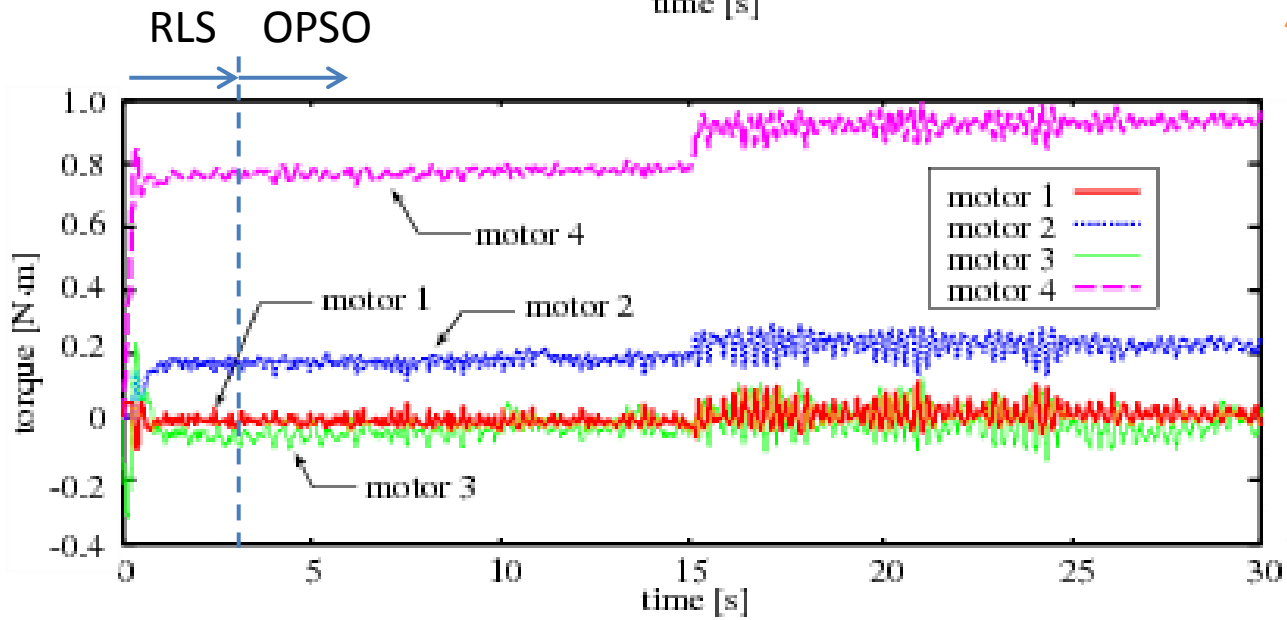
- ✓ オーバーシュート
- ✓ 定常偏差
- ✓ 過渡応答特性
- ✓ 無駄時間
- ✗ 適応性能

5. 実験結果 RSLとOPSOを切り替えて利用した場合



出力の時間推移

入力の時間推移



- ✓ オーバーシュート
- ✓ 定常偏差
- ✓ 過渡応答特性
- ✓ 無駄時間
- ✓ 適応性能

6. まとめ

result

オンラインシステムパラメータ推定のためのRLSとOPSOを切り替えて用いるGMVCに基づくセルフチューニングPIコントローラの構成手法を提案した

result

提案制御手法を重複分割分散制御法と組み合わせてウェブ搬送実験機に組み込むことで、ウェブの速度と張力の高精度な制御が可能であることを確かめた。

Future work

搬送速度と精度のさらに向上させることが可能な制御系の構築