

ボールねじ駆動ステージに対する 指令値の変化にロバストな反復学習制御

林拓巳*, 藤本博志 (東京大学), 伊佐岡慶浩, 寺田祐貴 (DMG 森精機)

Iterative Learning Control for Ball-screw-driven Stages with Robustness to Reference Trajectory Variation

Takumi HAYASHI*, Hiroshi FUJIMOTO (The University of Tokyo)

Yoshihiro ISAOKA, Yuki TERADA (DMG MORI CO., LTD.)

Ball-screw-driven stages are widely used as feed systems of industrial equipment such as numerically controlled machines. Therefore, high-precision position control of ball-screw-driven stages is required. However, rolling friction deteriorates control performance. In this paper, a friction compensation method based on iterative learning control is proposed. The proposed method can achieve high control performance by using basis functions and data-based friction model. Through simulations and experiments, the proposed method is compared with other friction compensation methods and the effectiveness of the proposed method is verified.

キーワード：ボールねじ駆動ステージ，転がり摩擦，摩擦補償，反復学習制御，基底関数，データベース摩擦モデル。
(Ball-screw-driven stage, rolling friction, friction compensation, iterative learning control, basis function, data-based friction model.)

1. 序論

Fig. 1 に示されるボールねじ駆動ステージは，駆動側モータの回転運動を負荷側ステージの並進運動に変換する機構であり，NC 工作機械など産業用機器の送り系として用いられている。NC 工作機械で高精度な加工を行うためにはボールねじ駆動ステージの精密位置制御が必要となる。

しかし，Fig. 1 のボールねじとリニアガイドで生じる転がり摩擦がボールねじ駆動ステージの制御性能を劣化させる。転がり摩擦の特性を Fig. 2 に示す。Fig. 2 の特性は二つの領域，region 1 (転がり出し領域) と region 2 (転がり領域) に分けて考えることができる。転がり出し領域では，ステージの速度反転直後に転がり摩擦は非線形ばね特性を示す。一方で，転がり領域では，転がり摩擦はほぼ一定の値 (クーロン摩擦 T_c) となる。この転がり摩擦のために，特にステージの速度が反転したときに大きな追従誤差が生じる。したがって，ボールねじ駆動ステージの精密制御には転がり摩擦の補償が必須となる。摩擦補償法はモデルベース摩擦補償法と学習ベース摩擦補償法に大別される。

モデルベース摩擦補償法では，転がり摩擦特性を計測し，厳密にモデリングすることで摩擦補償を行う。現在までにさまざまな転がり摩擦のモデルが考えられてきており，Generalized Maxwell Slip モデル⁽¹⁾，可変自然長ばねモデル⁽²⁾，レオロジーに基づくモデル⁽³⁾などが提案されてきた。これらの摩擦モデルを使って転がり摩擦をモデル化し，得られたモデルを利用して摩擦補償をすることで，追従特性を向上させることに成功している。ただし，周囲の環境やボールねじ駆動ステージの動作状況によって転がり摩擦の特性が変化することが考えられ⁽⁴⁾，得られた転がり摩擦のモデ

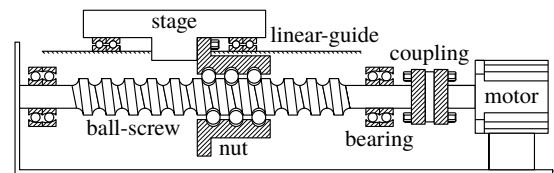


Fig. 1. Schematic view of a ball-screw-driven stage.

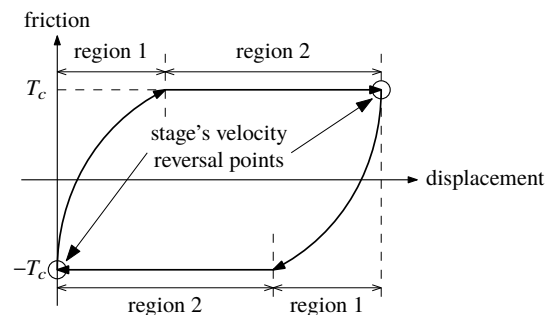


Fig. 2. Characteristics of rolling friction.

ルを利用して補償しきれないということが考えられる。

他方の学習ベース摩擦補償法は，同じ指令値軌道が繰り返されるときに有効な手法である。反復学習制御 (Iterative Learning Control; ILC)⁽⁵⁾⁻⁽⁸⁾ は繰り返される指令値軌道に出力を追従させたいときに有効な制御手法である。直前の試行を利用して転がり摩擦を補償するため，環境や動作状況による転がり摩擦の特性の変化に強いと考えられる。一方で，一つの指令値軌道しか学習できず，指令値軌道が変わると再学習が必要になるという欠点がある。

本稿では，通常の ILC と基底関数を利用する射影型 ILC⁽⁹⁾

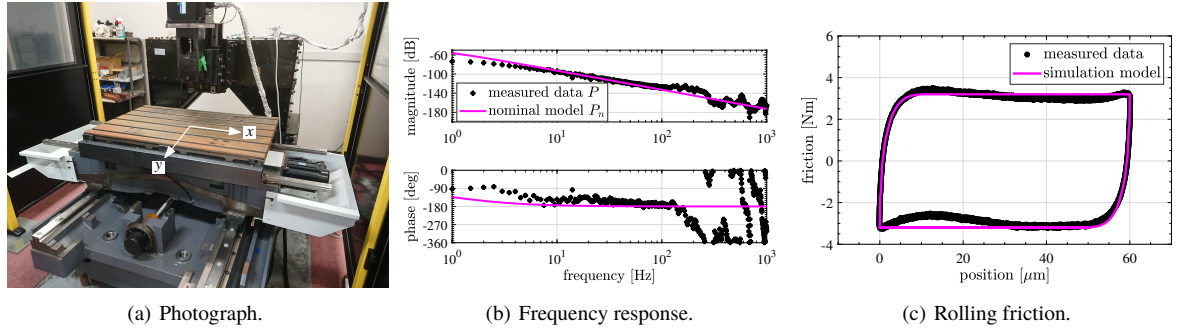


Fig. 3. Experimental setup. Only the x axis is used.

Parameter	Value
Nominal inertia J_n	0.015 kgm ²
Nominal viscosity coefficient D_n	0.1 N m s rad ⁻¹
Torque constant K_T	0.715 N m A ⁻¹
Ball-screw's lead R	1.91 mm rad ⁻¹

をベースに、指令値軌道の変化にロバストな学習ベース摩擦補償法を提案する。シミュレーションと実験を通じて、提案手法の有効性を検証する。

2. 実験機

本稿で扱う実験機を Fig. 3(a) に示す。本実験機は x, y 軸の2つのボールねじを持つが、本稿では x 軸のみを使用する。Fig. 3(b) に示される、本実験機のモータの電流 i [A] からステージの位置 x [m] までの周波数特性から、ノミナルモデル $P_n(s)$ を

$$P_n(s) = \frac{RK_T}{J_n s^2 + D_n s} \dots \dots \dots (1)$$

とする。Eq. (1) の各パラメータの値は Tab. 1 の通りである。

また、本実験機の転がり摩擦を Fig. 3(c) に示す。Fig. 3(c) は極低速駆動時のノミナルモデルを使用した外乱オブザーバで得られた。Fig. 3(c) から、転がり出し領域はおおよそ速度反転から 10 μ m、クーロン摩擦は 3.2 N m である。

3. 反復学習制御

〈3・1〉 通常の反復学習制御 Fig. 4 に示される2自由度制御系を考える。このとき、追従誤差 e は

$$e = r - x = Sr - SP(f - d) \dots \dots \dots (2)$$

となる。 $S = (1 + C_{FB}P)^{-1}$ は感度関数である。

ILC は、同じ指令値軌道を繰り返し入力するときに、繰り返しごとに追従誤差を抑圧する手法である。以下、添字 $j, j+1$ は $j, j+1$ 回目の繰り返し (試行) を示すとする。

今、 j 回目の試行のフィードフォワード入力を f_j 、そのときの追従誤差を e_j とすると、その次の試行のフィードフォワード入力 f_{j+1} は学習フィルタ $L(z)$ とロバストフィルタ $Q(z)$ を利用し、次のように計算される。

$$f_{j+1} = Q(f_j + Le_j) \dots \dots \dots (3)$$

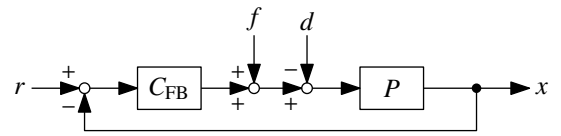


Fig. 4. Block diagram of two-degree-of-freedom control system (r : (position) reference, x : (position) output, f : feedforward input, d : disturbance (rolling friction), P : plant (ball-screw-driven stage), C_{FB} : feedback controller).

Eq. (2) と Eq. (3) から、指令値軌道や外乱が毎試行繰り返されるなら、次の追従誤差に関する関係式が得られる。

$$e_{j+1} = Q(1 - SPL)e_j + (1 - Q)(Sr - SPd) \dots \dots \dots (4)$$

Eq. (4) から、追従誤差が試行ごとに単調減少する条件は

$$\max_{\omega} \left| Q(e^{j\omega}) \left(1 - S(e^{j\omega})L(e^{j\omega})P(e^{j\omega}) \right) \right| < 1 \dots \dots (5)$$

となる。 $L(z), Q(z)$ は Eq. (5) を満たすように設計される。

〈3・2〉 射影型反復学習制御 追従誤差の漸化式 Eq. (4) は同じ指令値軌道が繰り返されるときに成り立つ式であり、指令値軌道が変わると、得られたフィードフォワード入力は意味をなさない。この欠点を解決したのが射影型 ILC (Projection-based ILC; P-ILC) である。P-ILC と区別するために、通常の ILC を S-ILC (Standard ILC) と呼称する。

例えば

$$P(s) = \frac{RK_T}{Js^2 + Ds} \dots \dots \dots (6)$$

なるシステムに対しては、もし外乱 (転がり摩擦) がなければ、フィードフォワード入力を

$$f = \frac{J}{RK_T} \ddot{r} + \frac{D}{RK_T} \dot{r} = \begin{bmatrix} \ddot{r} & \dot{r} \end{bmatrix} \begin{bmatrix} J \\ RK_T \\ D \\ RK_T \end{bmatrix} \dots \dots \dots (7)$$

とすれば、追従誤差を抑圧できる。プラントのパラメータ J, D の真値は一般には分からないが、P-ILC では S-ILC を利用し、毎試行パラメータ J, D を推定する。そして、基底関数 $\Psi(r_j)$ を利用してパラメータ化されたフィードフォワード入力 f_j^p を生成する。

$$f_j^p = \Psi(r_j)\theta_j, \dots\dots\dots (8a)$$

$$f_j^p = \begin{bmatrix} f_j^p[0] & \dots & f_j^p[N] \end{bmatrix}^T, \dots\dots\dots (8b)$$

$$\Psi(r_j) = \begin{bmatrix} \dot{r}_j[0] & \dot{r}_j[0] \\ \vdots & \vdots \\ \dot{r}_j[N] & \dot{r}_j[N] \end{bmatrix}, \dots\dots\dots (8c)$$

$$\theta_j = \begin{bmatrix} \hat{J}_j & \hat{D}_j \\ RK_T & RK_T \end{bmatrix}^T \dots\dots\dots (8d)$$

ここで、長さ $N+1$ のベクトルは ILC の 1 試行の時系列データである。また、 θ_j は推定されたプラントのパラメータを表し、ハット付き文字は推定値を意味している。

〈3・2・1〉 手順 P-ILC では、次の手順で $j+1$ 回目の試行のフィードフォワード入力 f_{j+1}^p が計算される⁹⁾。

(1) $j+1$ 回目の試行でも j 回目の試行の指令値軌道 r_j が繰り返されると仮定し、フィードフォワード入力 f_{j+1}^{np} を S-ILC (Eq. (3)) で計算する。

(2) プラントのパラメータ θ_{j+1} を、S-ILC で計算されたフィードフォワード入力 f_{j+1}^{np} と r_j から得られる基底関数 $\Psi(r_j)$ から推定する。

(3) 指令値軌道を $j+1$ 回目の試行のもの r_{j+1} に更新し、 r_{j+1} から得られる基底関数 $\Psi(r_{j+1})$ と推定されたプラントのパラメータ θ_{j+1} から、Eq. (9) でフィードフォワード入力 f_{j+1}^p を計算する。

$$f_{j+1}^p = \Psi(r_{j+1})\theta_{j+1} \dots\dots\dots (9)$$

〈3・2・2〉 プラントのパラメータの推定 パラメータ化されていないフィードフォワード入力 f_{j+1}^{np} とパラメータ化されたフィードフォワード入力 $\Psi(r_j)\theta_{j+1}$ ($\neq f_{j+1}^p$) を使用したときの指令値軌道 r_j に対する追従誤差 e_{j+1}^{np} , e_{j+1}^p は、外乱を無視するとそれぞれ Eq. (2) から次のようになる。

$$e_{j+1}^{np} = S r_j - S P f_{j+1}^{np}, \dots\dots\dots (10a)$$

$$e_{j+1}^p = S r_j - S P \Psi(r_j)\theta_{j+1}, \dots\dots\dots (10b)$$

ここで、 P と S はそれぞれ P と S から得られる畳み込み行列である。 P と S の真値は知ることができないために Eq. (10) は計算できないが、ノミナルモデルを利用すれば、その推定値を試行前に計算することができる。

$$\hat{e}_{j+1}^{np} = S_n r_j - S_n P_n f_{j+1}^{np}, \dots\dots\dots (11a)$$

$$\hat{e}_{j+1}^p = S_n r_j - S_n P_n \Psi(r_j)\theta_{j+1}, \dots\dots\dots (11b)$$

添字 n はノミナルモデルを意味している。

プラントのパラメータ θ_{j+1} は Eq. (12) から推定され、これは線形最小二乗法で解くことができる⁹⁾。

$$\begin{aligned} & \min_{\theta_{j+1}} \left\| \hat{e}_{j+1}^{np} - \hat{e}_{j+1}^p \right\|_2 \\ \Rightarrow & \min_{\theta_{j+1}} \left\| S_n P_n f_{j+1}^{np} - S_n P_n \Psi(r_j)\theta_{j+1} \right\|_2 \dots\dots\dots (12) \end{aligned}$$

4. 提案手法

前節で導入した P-ILC は、転がり摩擦を無視した場合に成り立つものである。転がり摩擦を考慮した P-ILC を行う場合、転がり摩擦を推定し補償するための新しい基底関数を加える必要があるが、複雑な挙動を示す転がり摩擦を仮定や近似なしに表現できる基底関数を見つけるのは難しい。本稿では転がり摩擦を、複雑な挙動を示す転がり出し領域と、ほとんど一定となる転がり領域に分けて考える。

〈4・1〉 手順 提案手法では、次の手順で $j+1$ 回目の試行のフィードフォワード入力 f_{j+1}^h が計算される。

(1) $j+1$ 回目の試行でも j 回目の試行の指令値軌道 r_j が繰り返されると仮定し、フィードフォワード入力 f_{j+1}^{np} を S-ILC (Eq. (3)) で計算する。

(2) プラントのパラメータ θ_{j+1} を、S-ILC で計算されたフィードフォワード入力 f_{j+1}^{np} と r_j から得られる基底関数 $\Psi(r_j)$ から推定する。推定の際には、転がり出し領域、すなわち速度反転から近い領域の情報は使わない。

(3) f_{j+1}^{np} と基底関数 $\Psi(r_j)$ 、推定されたプラントのパラメータ θ_{j+1} から、転がり出し領域を含めた転がり摩擦の補償分 $\hat{T}_{rf,j+1}$ を計算し、 r_j と $\hat{T}_{rf,j+1}$ からデータベース摩擦モデル⁽¹⁰⁾ を構築する。

(4) 指令値軌道を $j+1$ 回目の試行のもの r_{j+1} に更新し、 r_{j+1} から得られる基底関数 $\Psi(r_{j+1})$ と推定されたプラントのパラメータ θ_{j+1} 、データベース摩擦モデルから、フィードフォワード入力 f_{j+1}^h を計算する。

〈4・2〉 転がり領域の情報を利用したプラントのパラメータとクーロン摩擦の推定 転がり領域では、転がり摩擦はほとんど一定値(クーロン摩擦 T_c)となるため、基底関数の設定が容易である。そのため、転がり領域の情報からプラントのパラメータを推定する。

クーロン摩擦補償用の基底関数は速度の符号関数とし、これを利用してクーロン摩擦を新たに推定する。

$$\Psi(r_j) = \begin{bmatrix} \dot{r}_j[0] & \dot{r}_j[0] & \text{sign}(\dot{r}_j[0]) \\ \vdots & \vdots & \vdots \\ \dot{r}_j[N] & \dot{r}_j[N] & \text{sign}(\dot{r}_j[N]) \end{bmatrix}, \dots\dots\dots (13a)$$

$$\theta_j = \begin{bmatrix} \hat{J}_j & \hat{D}_j & \hat{T}_{c,j} \\ RK_T & RK_T & K_T \end{bmatrix}^T \dots\dots\dots (13b)$$

フィードフォワード入力の次元が電流のため、 θ_j には \hat{T}_c ではなく、 \hat{T}_c/K_T を追加した。

プラントのパラメータの推定で転がり出し領域の情報を無視するために、Eq. (12) の代わりに Eq. (14) を線形最小二乗法で解く。パラメータ化されていないフィードフォワード入力とパラメータ化されたフィードフォワード入力を使用したときに生じる転がり摩擦は等しいと仮定した。

$$\min_{\theta_{j+1}} \left\| S_n P_n f_{j+1}^{np} - S_n P_n \bar{f}_{j+1}^p(r_j, W_j) \right\|_2, \dots\dots\dots (14a)$$

$$\bar{f}_{j+1}^p(r_j, W_j) = W_j \Psi(r_j)\theta_{j+1} + (I - W_j)f_{j+1}^{np} \dots\dots\dots (14b)$$

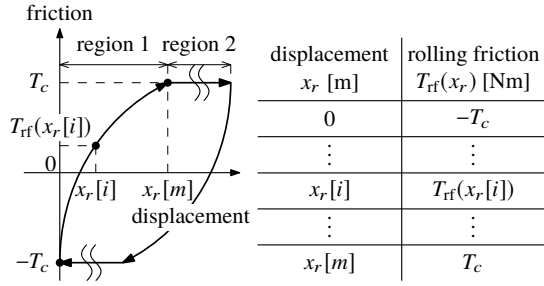


Fig. 5. Friction table for data-based friction model. Nonlinear elastic characteristic of rolling friction (left) is represented as a table (right) without any approximations.

W_j は対角行列であり, j 回目の試行の指令値軌道 r_j から得られる。重み行列 W_j を適切に設定することで, パラメータの推定に用いる情報の取捨選択が可能となる。

〈4・3〉 推定されたプラントのパラメータを利用した転がり摩擦の推定と補償 推定されたプラントのパラメータを利用し, S-ILC で計算されたフィードフォワード入力のうち, Eq. (15) で転がり出し領域を含めた転がり摩擦の補償分 $\hat{T}_{rf,j+1}$ を抜き出す。

$$\hat{T}_{rf,j+1} = \left(f_{j+1}^{np} - \frac{\hat{J}_{j+1}}{RK_T} \dot{r}_j - \frac{\hat{D}_{j+1}}{RK_T} \dot{r}_j \right) K_T \dots \dots \dots (15)$$

得られた $\hat{T}_{rf,j+1}$ と r_j から, Fig. 5 に示すデータベース摩擦モデルを構築する。データベース摩擦モデルは入力が速度反転からの変位, 出力が転がり摩擦のテーブルであり, データベース摩擦モデルを構築する際の速度反転からの変位は r_j から計算される。

$j+1$ 回目の試行のフィードフォワード入力 f_{j+1}^h を計算する際は, 転がり出し領域では基底関数とデータベース摩擦モデルを利用し, 転がり領域では基底関数を利用する。

$$f_{j+1}^h = \left[f_{j+1}^h[0] \quad \dots \quad f_{j+1}^h[N] \right]^T, \dots \dots \dots (16a)$$

$$f_{j+1}^h[i] = \frac{\hat{J}_{j+1}}{RK_T} \dot{r}_{j+1}[i] + \frac{\hat{D}_{j+1}}{RK_T} \dot{r}_{j+1}[i] + \begin{cases} \frac{\hat{T}_{c,j+1}}{K_T} \text{sign}(\dot{r}_{j+1}[i]) \\ \text{(if } r_{j+1}[i] \text{ is in the sliding region),} \\ \frac{\text{Table}_{j+1}(r_{r,j+1}[i])}{K_T} \text{sign}(\dot{r}_{j+1}[i]) \\ \text{(if } r_{j+1}[i] \text{ is in the pre-sliding region).} \end{cases} \dots \dots (16b)$$

ここで $\text{Table}_{j+1}(\cdot)$ は構築したデータベース摩擦モデルから最近傍の値を読み出すことを意味し, その引数の $r_{r,j+1}[i]$ は $j+1$ 回目の試行の指令値軌道 r_{j+1} から計算される, 速度反転からの変位である。

5. シミュレーション

提案手法の有効性を確認するために, シミュレーションを行った。なお, 制御周期 T_s は 1 ms とした。

〈5・1〉 条件

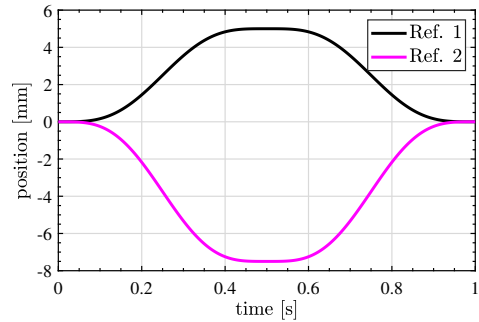


Fig. 6. Reference trajectories (Case 1).

〈5・1・1〉 ノミナルモデルとシミュレーションプラント ノミナルモデルは Eq. (1) と Tab. 1 に示されるものとした。また, シミュレーションプラントは Eq. (6) で表されるものとした。ただしパラメータに誤差があるとした ($J = 0.8J_n$, $D = 1.2D_n$)。転がり摩擦に関しては, Fig. 3(c) の “simulation model” を使用した。

〈5・1・2〉 フィードバック制御器の設計 ノミナルモデル Eq. (1) に対して, 閉ループの極が 30 Hz の重根となるよう, 極配置法で設計した後, Tustin 変換で離散化した。

〈5・1・3〉 学習フィルタとロバストフィルタの設計 ILC で使われる 2 つのフィルタについて, 学習フィルタ $L(z)$ はノミナルモデルに対して零位相誤差追従制御器 (Zero-Phase Error Tracking Controller; ZPETC) ⁽¹¹⁾ で設計した。ロバストフィルタ $Q(z)$ は高周波でのモデリング誤差を抑圧するために, 零位相ローパスフィルタとした。

$$Q(z) = \left(\frac{z+2+z^{-1}}{4} \right)^{N_Q}, N_Q = 16. \dots \dots \dots (17)$$

実際には Eq. (17) の実装は不可能なため, Eq. (18) に N_Q サンプル進み補償を加えることで Eq. (17) を実現している。

$$Q_r(z) = Q(z) \cdot z^{-N_Q} \dots \dots \dots (18)$$

〈5・1・4〉 指令値軌道 Fig. 6 に示す 2 種類の指令値軌道にステージの位置を追従させる。ILC のうち, 最初の 5 試行は指令値軌道として “Ref. 1” を用い, 6 回目から 10 回目の試行では “Ref. 2” を用いる。5, 6 回目の試行を境に指令値軌道が変化しており, このときに S-ILC と提案手法 (Prop.) でどのような追従誤差が生まれるかを比較する。

〈5・1・5〉 重み行列の設定と転がり摩擦の補償 提案手法では, プラントのパラメータの推定の際に Eq. (14) を解くが, その際に W_j という重み行列を導入している。 W_j は対角行列であり, 本稿では Eq. (19) のように決めることで, プラントのパラメータの推定の際に, 転がり摩擦が複雑な挙動を示す転がり出し領域の情報を無視することにする。

$$W_j = \text{diag} \left(w_j[0] \quad w_j[1] \quad \dots \quad w_j[N] \right), \dots \dots (19a)$$

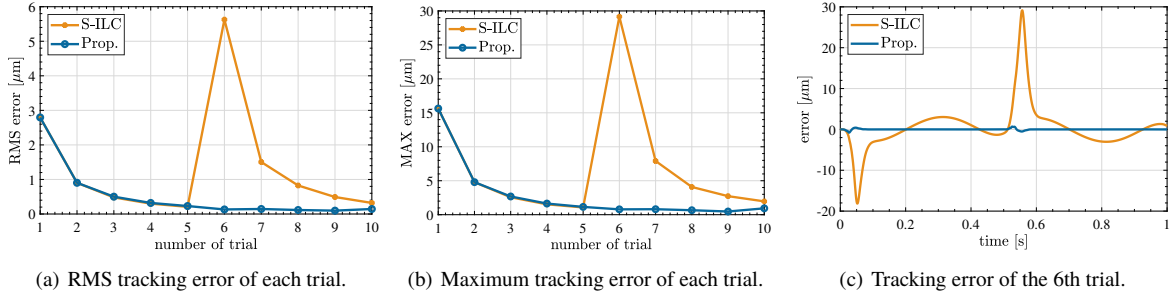


Fig. 7. Simulation results.

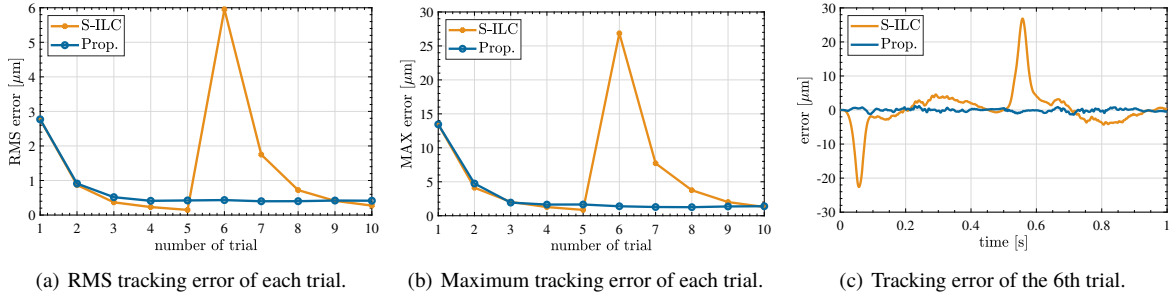


Fig. 8. Experimental results (Case 1).

$$w_j[i] = \begin{cases} 1 & (r_{r,j}[i] > 100 \mu\text{m}) \\ 0 & (\text{otherwise}) \end{cases} \dots\dots\dots (19b)$$

$r_{r,j}[i]$ は j 回目の試行の指令値軌道 r_j から計算される、速度反転からの変位である。極低速駆動時には、転がり出し領域の幅は $10 \mu\text{m}$ であったが、提案手法でデータベース摩擦モデルを作る際は、余裕を持たせ、 $100 \mu\text{m}$ とした。

〈5・2〉 結果 シミュレーション結果を Fig. 7 に示す。Fig. 7(a) は各試行の追従誤差の Root Mean Square (RMS) 値を、Fig. 7(b) は各試行の追従誤差の最大値を示している。また、Fig. 7(c) は指令値軌道が変わった直後の 6 回目の試行の追従誤差の時間波形を示している。Fig. 7 から、提案手法が指令値軌道の変化にロバストであることが示された。

6. 実験

〈6・1〉 Case 1 シミュレーションと同条件で実験を行った。その結果を Fig. 8 に示す。シミュレーション結果 (Fig. 7) と同様の結果となった。提案手法が学習後に S-ILC よりも劣る理由としては、

- ・クーロン摩擦のステージ位置に対するわずかな依存性
- ・基底関数を設定するときに無視した高次のダイナミクスや転がり摩擦以外の外乱

などが考えられる。

〈6・2〉 Case 2 Case 1 の実験では、同じ指令値軌道を繰り返し、途中で別の指令値軌道に変え、それを繰り返すというものであった。Case 2 の実験では、毎試行変わる指令値軌道に対して、提案手法がどのような挙動を示すかを確認する。Case 2 の実験で使う指令値軌道を Fig. 9 に示

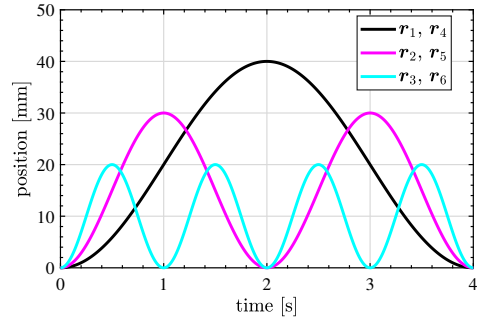
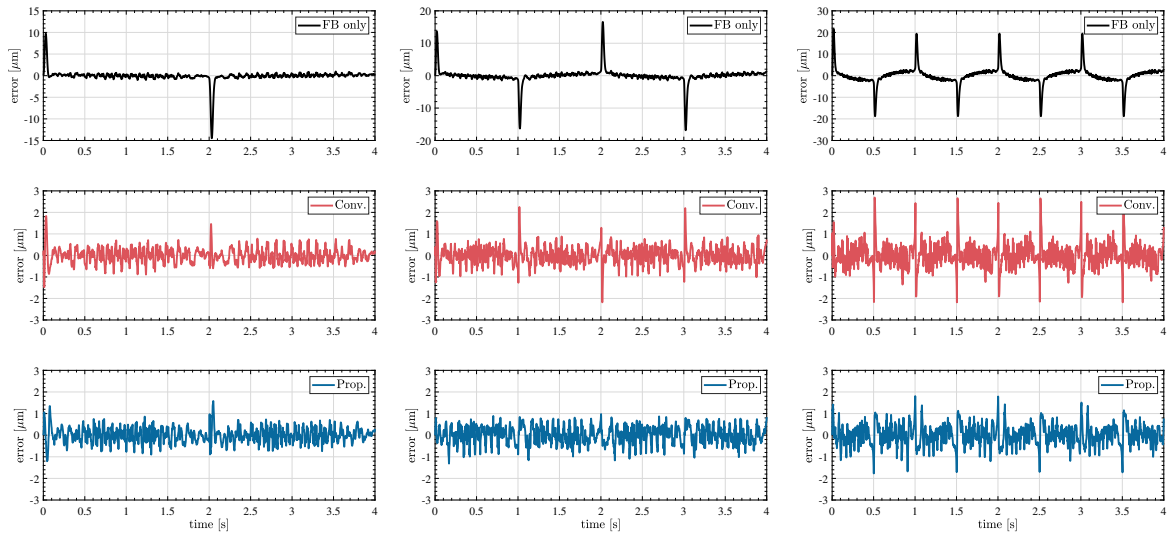


Fig. 9. Reference trajectories (Case 2).

す。 $r_1, r_2, r_3, \dots, r_6$ と毎試行、指令値軌道を変える。指令値軌道が毎試行変わる場合、S-ILC ではもはや追従誤差を抑圧できない。そこで本稿では提案手法の比較相手として、従来手法 (Conv.) をノミナルモデルに対するマルチレートフィードフォワード制御⁽¹²⁾と極低速駆動で得られたデータベース摩擦モデルによる転がり摩擦補償とする。従来手法はモデルベース摩擦補償法の一つであり、プラントのパラメータやデータベース摩擦モデルは試行ごとに更新されないことに注意されたい。従来手法のデータベース摩擦モデルは、 $0.1 \mu\text{m}$ 刻みで速度反転から $10 \mu\text{m}$ までのデータがテーブル化されている。

4, 5, 6 回目の試行の指令値軌道 r_4, r_5, r_6 に対する追従誤差を Fig. 10 に示す。参考までに、転がり摩擦の補償を行わない、フィードバック制御のみの結果も示している。転がり摩擦の補償を行うことで、従来手法と提案手法ともに追従誤差を抑圧できている。提案手法では試行ごとにプラ



(a) Tracking error of the 4th trial.

(b) Tracking error of the 5th trial.

(c) Tracking error of the 6th trial.

Fig. 10. Experimental results (Case 2).

ントのパラメータとデータベース摩擦モデルを更新しているため、従来手法よりも追従誤差が小さくなっている。また、提案手法は指令値軌道が毎試行変わっているにも関わらず、学習できていることが確認された。

7. 結論

本稿では、一つの指令値軌道しか学習できない S-ILC や、複数の指令値軌道に対応するために基底関数を利用した P-ILC に対し、基底関数では上手く表現できない転がり摩擦の非線形ばね特性を S-ILC を利用してデータベース摩擦モデル化することで、指令値軌道が変わったとしてもプラントのパラメータと転がり摩擦を学習し、補償する手法を提案した。提案手法を S-ILC や従来のモデルベース摩擦補償法とシミュレーションおよび実験と比較し、その有効性を検証した。今後の研究課題としては、収束性の議論や計算量・使用メモリの低減手法などが挙げられる。

文 献

- (1) F. Al-Bender, V. Lampaert, and J. Swevers, "The Generalized Maxwell-Slip Model: A Novel Model for Friction Simulation and Compensation," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1883–1887, 2005.
- (2) H. Asaumi and H. Fujimoto, "Proposal on nonlinear friction compensation based on variable natural length spring model," in *SICE Annual Conference*, 2008, pp. 2393–2398.
- (3) Y. Maeda and M. Iwasaki, "Feedforward Friction Compensation Using the Rolling Friction Model for Micrometer-stroke Point-to-point Positioning Motion," *IEEJ Journal of Industry Applications*, vol. 7, no. 2, pp. 141–149, 2018.
- (4) T. Beauvain and H. Fujimoto, "Distributed and Parameter-Varying Friction Compensation for Ball-Screw Feed Drive

Systems," in *The 3rd IEEE International Workshop on Sensing, Actuation, Motion Control, and Optimization*, 2017.

- (5) D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A Survey of Iterative Learning Control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- (6) M. Yamamoto, M. Iwasaki, K. Ito, and N. Matsui, "Precise Disturbance Modeling for Improvement of Positioning Performance," *IEEJ Transactions on Industry Applications*, vol. 128, no. 6, pp. 742–749, 2008, (in Japanese).
- (7) F. Sakai, "A Continuous-time System Identification for Systems with Nonlinear Friction Using Iterative Learning Control," *Transactions of the Society of Instrument and Control Engineers*, vol. 53, no. 8, pp. 448–454, 2017, (in Japanese).
- (8) T. Oomen, "Advanced Motion Control for Precision Mechatronics: Control, Identification, and Learning of Complex Systems," *IEEJ Journal of Industry Applications*, vol. 7, no. 2, pp. 127–140, 2018.
- (9) F. Boeren, A. Bareja, T. Kok, and T. Oomen, "Frequency-Domain ILC Approach for Repeating and Varying Tasks: With Application to Semiconductor Bonding Equipment," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 6, pp. 2716–2727, 2016.
- (10) T. Takemura and H. Fujimoto, "Proposal of novel rolling friction compensation with data-based friction model for ball screw driven stage," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, vol. 1, 2010, pp. 1932–1937.
- (11) M. Tomizuka, "Zero Phase Error Tracking Algorithm for Digital Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 1, pp. 65–68, 1987.
- (12) H. Fujimoto, Y. Hori, and A. Kawamura, "Perfect tracking control based on multirate feedforward control with generalized sampling periods," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 3, pp. 636–644, 2001.