

# Fixed-structure sampled-data feedforward control for multivariable motion systems

Masahiro Mae<sup>a,\*</sup>, Max van Haren<sup>b</sup>, Koen Classens<sup>b</sup>, Wataru Ohnishi<sup>a</sup>, Tom Oomen<sup>b,c</sup>, Hiroshi Fujimoto<sup>a</sup>

<sup>a</sup>The University of Tokyo, Japan

<sup>b</sup>Eindhoven University of Technology, The Netherlands

<sup>c</sup>Delft University of Technology, The Netherlands

---

## Abstract

Increasing performance requirements in high-precision mechatronic systems lead to a situation where both multivariable and sampled-data implementation aspects need to be addressed. The aim of this paper is to develop a design framework for a multi-input multi-output feedforward controller to improve continuous-time tracking performance through learning. The sampled-data feedforward controller is designed with physically interpretable tuning parameters using a multirate zero-order-hold differentiator. The developed approach enables interaction compensation for multi-input multi-output systems and the feedforward controller parameters are updated through learning. The performance improvement is experimentally validated in a multi-input multi-output motion system compared to the conventional feedforward controllers.

*Keywords:* Feedforward control; Reference tracking; Multi-input multi-output system; Sampled-data control; Multirate inversion; Iterative learning control

---

## 1. Introduction

Feedforward control is essential in increasing performance requirements for motion control of high-precision mechatronic systems in industries such as semiconductor lithography systems [1, 2, 3], wire bonders [4], atomic force microscopy [5], machine tools [6], industrial robots [7], magnetic bearing [8], boost converters [9], 2D/3D printers [10, 11, 12], and CT scanners [13]. Iterative Learning Control (ILC) is one of the algorithms to update the feedforward controller by the error data of the previous iteration, and the error is reduced through learning. To overcome the limitation of the interpretability in ILC such as a Finite Impulse Response (FIR) filter-based structure in [1, 11], it is beneficial in industries that the controller consists of physically interpretable tuning parameters to achieve both intuitive tuning and data-driven learning.

Physical interpretability and intuitive tuning of the data-driven feedforward controller are desirable in industrial applications. It is achieved by structure analysis of the controlled system [14, 15], and the controller can be parameterized intuitively by linear combinations with tuning parameters and basis functions [10, 16]. Basis functions typically consist of a reference signal and its derivatives [16] and nonlinear functions such as friction compensation [6]. This structure enables low-complexity parameterization with physical interpretability and flexibility for varying references. In conventional approaches [14, 15, 16], the differentiator for the basis function design is implemented by the backward differentiator, and the sampled-data characteristic is not considered. The gap between the backward differentiator and the zero-order-hold characteristics of

the sampled-data system results in the limitation of the control performance in continuous time.

Sampled-data feedforward control improves the continuous-time tracking performance of high-precision mechatronic systems where the sampling frequency is not sufficiently high compared to the motion profile [17]. In industrial control applications, the controlled system is discretized by sampler and zero-order-hold and these characteristics should be considered in feedforward controller design to improve not only on-sample but also intersample performance [18]. State-tracking feedforward control [19, 20] and ILC [21] with multirate inversion can improve the continuous-time tracking performance in sampled-data systems. These controllers enable on-sample state-tracking and it leads to physically natural intersample behavior. For the application to the complex mechatronic systems, there is no guarantee of a perfect model for model inversion and there must be a modeling error between the identified model and the actual system. The multirate feedforward control also can be extended to the multivariable systems [22, 23]. However, the complex mechatronic systems are represented by the multivariable model in many cases and it results in many tuning parameters for the inverse-based controller design.

Although important contributions have been made to improve the performance and interpretability of feedforward control, the sampled-data characteristics with sampler and zero-order-hold are not taken into account in the basis function design, and the structure of multi-input multi-output (MIMO) feedforward control is not discussed in the context of intuitive tuning and leaning from experimental data. The aim of this paper is to design the MIMO feedforward controller to improve the continuous-time tracking performance through learning. Compared to conventional approaches, the developed basis function

---

\*Corresponding author

Email address: mmae@ieee.org (Masahiro Mae)

design considers the sampled-data characteristics. In this paper, the feedforward controller is parameterized with basis functions for MIMO motion systems and it enables physical interpretation of the feedforward controller parameters and analytic solution of data-driven parameter tuning. The present paper substantially extends the preliminary result in [24], including the comparison between the exact model inversion based on the multirate feedforward control, the generalization for the application in MIMO motion systems, the data-driven tuning algorithm, and the experimental validation.

The main contribution of this paper is the fixed-structure feedforward controller design considering sampled-data characteristics and interactions in MIMO motion systems. The contributions include:

**Contribution 1.** *Discrete-time basis functions are designed for continuous-time reference considering sampled-data characteristics to improve continuous-time tracking performance.*

**Contribution 2.** *ILC with basis functions is formulated with physically interpretable tuning parameters considering the dynamics and interaction of MIMO motion systems.*

The outline is as follows. In Section 2, the problem is formulated. In Section 3, the basis function design considering continuous-time tracking performance is described, constituting Contribution 1. In Section 4, the continuous-time tracking performance of the developed approach is demonstrated by the simulation in SISO motion systems compared to the conventional approaches. In Section 5, ILC with basis functions is formulated in MIMO motion systems, constituting Contribution 2. In Section 6, the performance improvement with interaction compensation and the sampled-data characteristics is experimentally validated. In Section 7, conclusions are presented.

## 2. Problem formulation

In this section, the problem to improve continuous-time tracking performance in MIMO motion systems is formulated. First, the reference tracking problem is defined in continuous time. Second, interaction compensation is investigated for reference tracking in MIMO motion systems. Finally, the problems in this paper are described.

### 2.1. Continuous-time tracking performance in sampled-data control

The considered tracking control configuration in a  $n_u$ -input  $n_y$ -output continuous-time linear time-invariant system  $\mathbf{G}$  is shown in Figure 1, with reference  $\mathbf{r}(t) \in \mathbb{R}^{n_y}$ , control input  $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ , and output  $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ .

The system is controlled by the sampled-data controller that consists of feedforward controller  $\mathbf{F}(\theta)$ , feedback controller  $\mathbf{K}$ , sampler  $\mathcal{S}$ , and zero-order-hold  $\mathcal{H}$ , where sampler and zero-order-hold are defined as follows.

**Definition 1 (Sampler).** *The sampler  $\mathcal{S}$  with sampling time  $T_s$  is defined as*

$$\mathcal{S} : \mathbf{r}(t) \mapsto \mathbf{r}[k], \quad \mathbf{r}[k] = \mathbf{r}(kT_s). \quad (1)$$

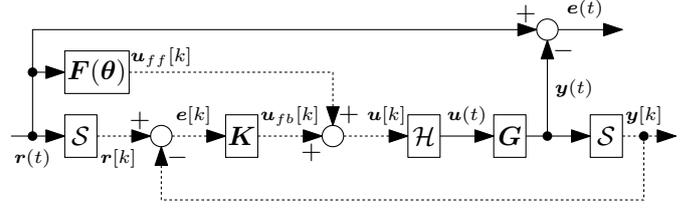


Figure 1: Tracking control diagram. The continuous-time system  $\mathbf{G}$  is controlled by the feedforward controller  $\mathbf{F}(\theta)$  and the discrete-time feedback controller  $\mathbf{K}$  with sampler  $\mathcal{S}$  and zero-order-hold  $\mathcal{H}$ . The objective is to minimize the continuous-time tracking error  $\mathbf{e}(t)$ . The solid and dotted lines denote the continuous-time and discrete-time signals, respectively.

**Definition 2 (Zero-order-hold).** *The zero-order-hold  $\mathcal{H}$  with sampling time  $T_s$  is defined as*

$$\mathcal{H} : \mathbf{u}[k] \mapsto \mathbf{u}(t), \quad \mathbf{u}(kT_s + \tau) = \mathbf{u}[k], \quad \tau = [0, T_s). \quad (2)$$

The control objective in this paper is to minimize the continuous-time tracking error  $\mathbf{e}(t)$ . Traditionally, the conventional discrete-time controller only focuses on the on-sample performance with the discrete-time tracking error  $\mathbf{e}[k]$ . To improve the continuous-time tracking error  $\mathbf{e}(t)$ , not only on-sample but also intersample performance should be considered. The improvement of continuous-time tracking performance is defined as follows.

**Definition 3 (Continuous-time tracking performance).** *The optimization problem to improve the continuous-time tracking performance in the sampled-data motion system is defined as*

$$\underset{\theta}{\text{minimize}} \quad \|\mathbf{e}[k]\|_{\mathbf{W}} \quad (3)$$

$$\text{subject to} \quad \underset{\forall l}{\text{RMS}(e_l[k]) \approx \text{RMS}(e_l(t))}, \quad (4)$$

where  $\theta$  is the tuning parameter,  $\|\bullet\|_{\mathbf{W}}$  is the weighted 2-norm with the weighting matrix  $\mathbf{W}$ ,  $e_l[k]$  and  $e_l(t)$  are the tracking errors of the  $l^{\text{th}}$  axis in discrete time and continuous time, and  $\text{RMS}(\bullet)$  is the operator to calculate the Root Mean Square.

It is the practical constraint that the controller can be tuned using the data of the discrete-time tracking error  $\mathbf{e}[k]$  although the objective of the control problem is the improvement of the continuous-time tracking error  $\mathbf{e}(t)$ . To satisfy approximately the same condition of the discrete-time and continuous-time tracking errors, the controller should be designed not to generate the control inputs that cause the intersample oscillation.

### 2.2. Decoupling control for interaction compensation

In the controller design of MIMO motion systems, the static decoupling is applied by the input decoupling matrix  $\mathbf{T}_u$  and the output decoupling matrix  $\mathbf{T}_y$ . The decoupled system  $\mathbf{T}_y \mathbf{G} \mathbf{T}_u$  should be square and diagonally dominant. In many cases, the single-input single-output (SISO) controller is designed after the static decoupling. Even if the system is statically decoupled, the off-diagonal terms still remain and it results in interaction between inputs and outputs [1]. Therefore, static decoupling is not sufficient in practice and it limits the control

performance of MIMO motion systems. In complex mechatronic systems, interaction analysis is difficult and the model of coupling dynamics always has a modeling error. In this paper, the feedforward controller is designed considering both diagonal and off-diagonal dynamics to compensate for interaction through learning from data with less modeling effort of MIMO motion systems.

### 2.3. Problem description

In this paper, the controller design problem is with respect to the following requirements:

**Requirement 1.** *The sampled-data characteristics with sampler and zero-order-hold should be considered in the discrete-time basis function design to improve continuous-time tracking performance.*

**Requirement 2.** *The basis functions should be parameterized with physically interpretable tuning parameters considering the dynamics and interaction of MIMO motion systems.*

Requirement 1 is dealt with in Section 3 and it results in Contribution 1. Requirement 2 is dealt with in Section 5 and it results in Contribution 2.

## 3. Basis function design for sampled-data motion system

In this section, the basis function design using a sampled-data differentiator is presented. The approach improves the continuous-time tracking performance of the feedforward controller in sampled-data control. First, the conventional approach using a backward differentiator is analyzed in sampled-data control. Second, the single-rate zero-order-hold differentiator is introduced for on-sample performance in sampled-data control. Third, the idea of state compatibility is defined to improve inter-sample performance. Finally, the multirate zero-order-hold differentiator is developed to design the sampled-data basis functions that satisfy state compatibility. It results in Contribution 1.

### 3.1. Challenge in sampled-data basis function design

The continuous-time feedforward controller can be parameterized using the reference signals  $r$  and its derivatives. It results in the combination with the tuning parameters  $\theta$  and the continuous-time basis functions that consist of a continuous-time differentiator  $\frac{d}{dt}$ . For example, the continuous-time acceleration feedforward controller  $F(\theta)$  can be designed for a single-mass motion system  $G(s) = \frac{1}{ms^2}$  as  $F(\theta) = \theta \frac{d^2}{dt^2}$ . In this example, the basis function is  $\Psi = \frac{d^2}{dt^2} r(t)$  and the tuning parameter is  $\theta = m$ . However, for applications of mechatronic systems in industries, the motion controllers are typically implemented in discrete time. Therefore, to design the discrete-time basis function  $\Psi$ , the continuous-time differentiator  $\frac{d}{dt}$  should be replaced by the sampled-data differentiator  $\xi$  defined as follows.

**Definition 4** (Sampled-data differentiator). *The  $n^{\text{th}}$  order sampled-data differentiator  $\xi^n$  with sampling time  $T_s$  is the conversion from the continuous-time signal  $r(t)$  to the discrete-time signal*

$\Psi_n[k]$  that is compatible with the  $n^{\text{th}}$  order derivative of  $r(t)$  and defined as

$$\Psi_n[k] = \xi^n r(t). \quad (5)$$

In the conventional approach [10, 16], the discrete-time basis functions are designed by the continuous-time reference  $r(t)$  and the backward differentiator defined as follows.

**Definition 5** (Backward differentiator). *The  $n^{\text{th}}$  order backward differentiator  $\xi_{bd}^n$  is given by*

$$\xi_{bd}^n = \begin{cases} \left(\frac{1-z^{-1}}{T_s}\right)^n z^{\frac{n}{2}} \mathcal{S} & (n : \text{even}) \\ \left(\frac{1-z^{-1}}{T_s}\right)^n \frac{1+z^{-1}}{2} z^{\frac{n+1}{2}} \mathcal{S} & (n : \text{odd}) \end{cases}, \quad (6)$$

where  $z$  is the discrete-time shift operator with sampling time  $T_s$  defined as  $z^n r[k] = r[k+n]$ .  $z^{\frac{n}{2}}$  denotes the phase compensation. When  $n$  is odd, the phase compensation consists of the half sample shift  $z^{\frac{1}{2}}$  that is a combination of one sample shift and the first order approximation of averaging the current and previous value [16].

The backward differentiator does not take into account the sampled-data characteristics with sampler and zero-order-hold, the performance deteriorates when the sampling frequency is not sufficiently high.

### 3.2. Single-rate zero-order-hold differentiator for on-sample performance

The state-space representation of the continuous-time  $n^{\text{th}}$  order integrator in the controllable canonical form is given by

$$H_{nc} \stackrel{s}{=} \left[ \begin{array}{c|c} \mathbf{A}_{nc} & \mathbf{b}_{nc} \\ \mathbf{c}_{nc} & 0 \end{array} \right] = \left[ \begin{array}{cccc|c} 0 & 1 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & \cdots & 0 & 0 \end{array} \right], \quad (7)$$

where  $\mathbf{A}_{nc} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{b}_{nc} \in \mathbb{R}^{n \times 1}$ , and  $\mathbf{c}_{nc} \in \mathbb{R}^{1 \times n}$ .

To consider the sampled-data characteristics for on-sample tracking performance, the single-rate zero-order-hold differentiator is defined as follows.

**Definition 6.** (Single-rate zero-order-hold differentiator.) *Considering the inverse of the continuous-time  $n^{\text{th}}$  order integrator discretized by sampler  $\mathcal{S}$  and zero-order-hold  $\mathcal{H}$  with one sample shift, the  $n^{\text{th}}$  order single-rate zero-order-hold differentiator  $\xi_{sr}^n$  is given by*

$$\xi_{sr}^n = \{(\mathcal{S}H_{nc}\mathcal{H})z\}^{-1}. \quad (8)$$

Although the discrete-time signal with the continuous-time signal and  $n^{\text{th}}$  order single-rate zero-order-hold differentiator is compatible on-sample with the  $n^{\text{th}}$  order derivative signal of the continuous-time signal, the generated discrete-time signal can be oscillated or diverge. The reason is that the single-rate zero-order-hold differentiator has unstable or oscillating poles because of the inverse of discretization zeros when the

degree of the continuous-time integrator is 2 or more as Euler-Frobenius polynomials [25]. Therefore, even if the single-rate zero-order-hold differentiator is compatible on-sample, there are mismatches in other states and it deteriorates intersample performance.

### 3.3. State compatibility for intersample performance

The sampled-data characteristics with intersample performance can be taken into account by the state-tracking control framework [19, 20, 21]. In the  $n$  samples lifted system, the exact state-tracking can be achieved in every  $n$  sample. In such cases, the states in every  $n$  sample are given by the multirate sampler defined as follows.

**Definition 7** (Multirate sampler). *The multirate sampler  $\mathcal{S}_n$  in every  $n$  sample with sampling time  $T_s$  is defined as*

$$\mathcal{S}_n : r(t) \mapsto r[i_n], \quad r[i_n] = r(i_n T_s). \quad (9)$$

To improve both on-sample and intersample performance in sampled-data systems with zero-order-hold and integrators, the basis functions should satisfy the state compatibility defined as follows.

**Definition 8** (State compatibility). *The discrete-time signal  $\Psi_n[k]$ , that is compatible with the  $n^{\text{th}}$  order derivative signal of the continuous-time signal  $r(t)$ , satisfies state compatibility if the output through the system, that consists of the continuous-time  $(n - m)^{\text{th}}$  order integrator  $H_{(n-m)c}$  and zero-order-hold  $\mathcal{H}$ , is equal to the continuous-time  $m^{\text{th}}$  order derivative signal of  $r(t)$  in every  $n$  sample with multirate sampler  $\mathcal{S}_n$  and defined as*

$$\mathcal{S}_n \frac{d^m}{dt^m} r(t) = \mathcal{S}_n H_{(n-m)c} \mathcal{H} \Psi_n[k], \quad (10)$$

where  $m = 0, 1, \dots, n - 1$ .

The sampled-data differentiator that satisfies the state compatibility enables the feedforward controller parameterization with basis functions to improve continuous-time tracking performance.

### 3.4. Multirate zero-order-hold differentiator with state compatibility

To improve the intersample performance in the discrete-time system, not only the output but also the states of the reference trajectory are considered. The multirate zero-order-hold differentiator is designed by the inverse of the continuous-time integrator discretized by sampler and zero-order-hold to satisfy the state compatibility. In this paper, the reference is assumed to be sufficiently smooth and satisfies the following assumption.

**Assumption 1** (Smoothness of reference). *The continuous-time reference  $r(t)$  for  $n$  states tracking is  $C^{n-1}$  class and differentiable at least  $n - 1$  times.*

To satisfy the  $n$  states compatibility in every  $n$  sample, the lifted signal is considered using the lifting operator defined as follows.

**Definition 9** (Lifting operator). *The lifting operator  $\mathcal{L}_n$  in every  $n$  sample is defined as*

$$\mathcal{L}_n : u[k] \mapsto \underline{u}[i_n], \quad (11)$$

where

$$\underline{u}[i_n] = [u[ni_n] \quad u[ni_n + 1] \quad \dots \quad u[ni_n + (n - 1)]]^T \in \mathbb{R}^n. \quad (12)$$

The  $n$  samples lifted system is defined as follows.

**Definition 10** (Lifted system). *Consider a discrete-time system  $H_d \stackrel{z}{=} \mathcal{C}_d(z\mathbf{I} - \mathbf{A}_d)^{-1} \mathbf{B}_d + \mathbf{D}_d$ . The relation between the input and the output in the  $n$  samples lifted system of  $H_d$  is given by*

$$\underline{y}[i_n] = \mathcal{L}_n y[k] = (\mathcal{L}_n H_d \mathcal{L}_n^{-1})(\mathcal{L}_n u[k]) = \underline{H}_d \underline{u}[i_n], \quad (13)$$

where

$$\underline{y}[i_n] = [y[ni_n] \quad y[ni_n + 1] \quad \dots \quad y[ni_n + (n - 1)]]^T \in \mathbb{R}^n, \quad (14)$$

and the lifted system  $\underline{H}_d$  is defined as

$$\underline{H}_d \stackrel{z^n}{=} \mathcal{L}_n H_d \mathcal{L}_n^{-1} = \left[ \begin{array}{c|ccc} \mathbf{A}_d^n & \mathbf{A}_d^{n-1} \mathbf{B}_d & \mathbf{A}_d^{n-2} \mathbf{B}_d & \dots & \mathbf{A}_d \mathbf{B}_d & \mathbf{B}_d \\ \mathbf{C}_d & \mathbf{D}_d & \mathbf{O} & \dots & \dots & \mathbf{O} \\ \hline \mathbf{C}_d \mathbf{A}_d & \mathbf{C}_d \mathbf{B}_d & \mathbf{D}_d & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{C}_d \mathbf{A}_d^{n-2} & \mathbf{C}_d \mathbf{A}_d^{n-3} \mathbf{B}_d & \mathbf{C}_d \mathbf{A}_d^{n-4} \mathbf{B}_d & \ddots & \mathbf{D}_d & \mathbf{O} \\ \mathbf{C}_d \mathbf{A}_d^{n-1} & \mathbf{C}_d \mathbf{A}_d^{n-2} \mathbf{B}_d & \mathbf{C}_d \mathbf{A}_d^{n-3} \mathbf{B}_d & \dots & \mathbf{C}_d \mathbf{B}_d & \mathbf{D}_d \end{array} \right]. \quad (15)$$

Considering the states in discrete-time, the  $n^{\text{th}}$  order integrator discretized by sampler and zero-order-hold is given by

$$\begin{aligned} H_{nd} &\stackrel{z}{=} \mathcal{S} H_{nc} \mathcal{H} = \left[ \begin{array}{c|c} \mathbf{A}_{nd} & \mathbf{b}_{nd} \\ \mathbf{c}_{nd} & 0 \end{array} \right] \\ &= \left[ \begin{array}{c|c} e^{\mathbf{A}_{nc} T_s} & \mathbf{A}_{nc}^{-1} (e^{\mathbf{A}_{nc} T_s} - \mathbf{I}) \mathbf{b}_{nc} \\ \mathbf{c}_{nc} & 0 \end{array} \right]. \end{aligned} \quad (16)$$

To design the inverse of the  $n^{\text{th}}$  order integrator discretized by sampler and zero-order-hold, the  $n$  samples lifted system is given by

$$\underline{H}_{nd} \stackrel{z^n}{=} \mathcal{L}_n H_{nd} \mathcal{L}_n^{-1} = \left[ \begin{array}{c|c} \mathbf{A}_{nd} & \mathbf{B}_{nd} \\ \mathbf{C}_{nd} & \mathbf{D}_{nd} \end{array} \right], \quad (17)$$

and in state-space representation defined as

$$\mathbf{x}_n[i_n + 1] = \underline{\mathbf{A}}_{nd} \mathbf{x}_n[i_n] + \underline{\mathbf{B}}_{nd} \underline{u}[i_n] \quad (18)$$

$$\underline{y}[i_n] = \underline{\mathbf{C}}_{nd} \mathbf{x}_n[i_n] + \underline{\mathbf{D}}_{nd} \underline{u}[i_n] \quad (19)$$

where

$$\mathbf{x}_n[i_n] = [x_0[i_n] \quad x_1[i_n] \quad \dots \quad x_{n-1}[i_n]]^T \in \mathbb{R}^n. \quad (20)$$

Satisfying the state compatibility, the relationship between the reference and the states is given by

$$\bar{\mathbf{r}}_n[i_n] = \mathbf{x}_n[i_n], \quad (21)$$

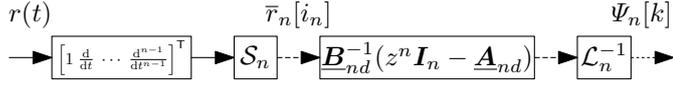


Figure 2: Block diagram of basis function design using multirate zero-order-hold differentiator. The dotted and dashed lines denote the discrete-time signal sampled by  $T_s$  and  $nT_s$ , respectively.

where

$$\begin{aligned} \bar{r}_n[i_n] &= \mathcal{S}_n \left[ 1 \quad \frac{d}{dt} \quad \cdots \quad \frac{d^{n-1}}{dt^{n-1}} \right]^T r(t) \\ &= [r_0[i_n] \quad r_1[i_n] \quad \cdots \quad r_{n-1}[i_n]]^T \in \mathbb{R}^n. \end{aligned} \quad (22)$$

From the discussions above, the multirate zero-order-hold differentiator is defined as follows.

**Definition 11.** (*Multirate zero-order-hold differentiator.*)

From (18) and (21), considering the inverse of the state equation in the continuous-time  $n^{\text{th}}$  order integrator discretized by sampler and zero-order-hold using the multirate feedforward control [19], the  $n^{\text{th}}$  order multirate zero-order-hold differentiator  $\xi_{mr}^n$  that satisfies the state compatibility is given by

$$\xi_{mr}^n = \mathcal{L}_n^{-1} \underline{\mathbf{B}}_n^{-1} (z^n \mathbf{I}_n - \underline{\mathbf{A}}_{nd}) \mathcal{S}_n \left[ 1 \quad \frac{d}{dt} \quad \cdots \quad \frac{d^{n-1}}{dt^{n-1}} \right]^T. \quad (23)$$

The basis function design procedure using the multirate zero-order-hold differentiator is shown in Figure 2. The sampled-data characteristics are not considered in conventional differentiator implementations. The multirate zero-order-hold differentiator has the advantage that it can consider the sampled-data characteristics only replacing the continuous-time differentiator and it results in continuous-time tracking performance improvement.

## 4. Demonstration in SISO sampled-data motion system

In this section, the continuous-time tracking performance of the linearly parameterized feedforward control using the multirate zero-order-hold differentiator is demonstrated. First, the comparison is conducted in the acceleration feedforward control using the backward differentiator, the single-rate zero-order-hold differentiator, and the multirate zero-order-hold differentiator. Second, the comparison is conducted between the linearly parameterized feedforward control using the multirate zero-order-hold differentiator and the multirate feedforward control based on exact model inversion.

### 4.1. Comparison in acceleration feedforward control

The continuous-time tracking performance improvement by considering sampled-data characteristics can be seen clearly in a single-mass motion system example.

#### 4.1.1. Conditions

The controlled system is given as  $G(s) = \frac{1}{ms^2}$  where  $m = 1$  is the mass of the rigid body. The sampling time is set to  $T_s = 5$  ms. The continuous-time reference is the 1<sup>st</sup> order polynomial trajectory. The top of Figure 3 shows control inputs with

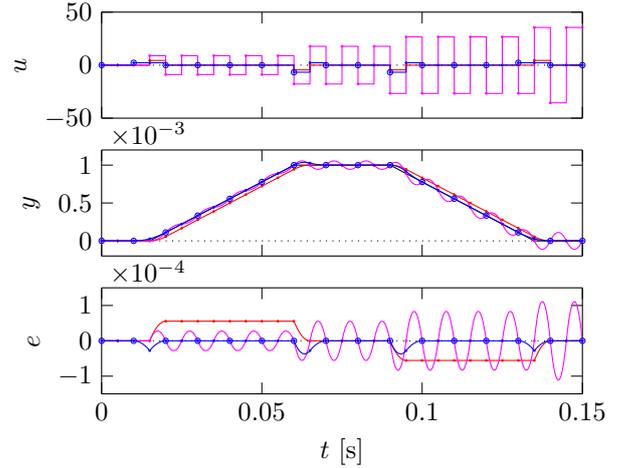


Figure 3: Simulation results of the open-loop tracking in a single-mass motion system with acceleration feedforward control using the backward differentiator (—), that using the single-rate zero-order-hold differentiator (—), and that using the multirate zero-order-hold differentiator (—). Top: control inputs. Center: 1<sup>st</sup> order reference (—) and outputs. Bottom: tracking errors. The acceleration feedforward using multirate zero-order-hold differentiator considers state compatibility and it results in smaller error. (•) and (◦) show the sampling points every  $T_s$  and  $2T_s$ .

the acceleration feedforward control using the backward differentiator  $u[k] = m\xi_{bd}^2 r(t)$ , that using the single-rate zero-order-hold differentiator  $u[k] = m\xi_{sr}^2 r(t)$ , and that using the multirate zero-order-hold differentiator  $u[k] = m\xi_{mr}^2 r(t)$ . The simulation is conducted in an open loop without a feedback controller.

#### 4.1.2. Comparison in continuous-time tracking performance

In Figure 3, the center and the bottom show the comparison of outputs and tracking errors in the open-loop simulation. It shows that the acceleration feedforward control using the multirate zero-order-hold differentiator outperforms because of the state compatibility compensating for the controlled system discretized by sampler and zero-order-hold.

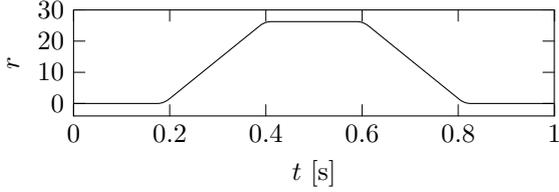
The limitation of the continuous-time tracking performance with the acceleration feedforward control using the backward differentiator is described by the sampled-data analysis that is given by

$$\begin{aligned} y[k] &= \{\mathcal{S}G\mathcal{H}\} u[k] = \left\{ \frac{T_s^2(1+z^{-1})}{2z(1-z^{-1})^2} \right\} \left\{ \frac{(1-z^{-1})^2}{T_s^2} zSr(t) \right\} \\ &= \frac{1+z^{-1}}{2} r[k]. \end{aligned} \quad (24)$$

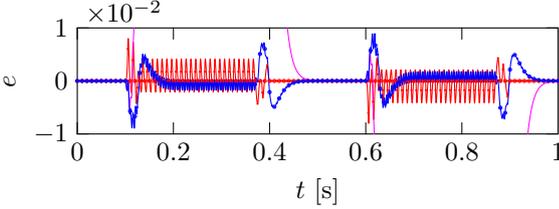
The result shows that the on-sample error appears as the 1<sup>st</sup> order approximated half-sample delay of the reference because of the zero-order-hold. It means that the perfect on-sample tracking can be achieved by a half-sample forward shifted reference with the 1<sup>st</sup> order reference condition but cannot be achieved with higher-order references.

The intersample oscillation in the continuous-time tracking performance with the acceleration feedforward control using the single-rate zero-order-hold differentiator is described by the

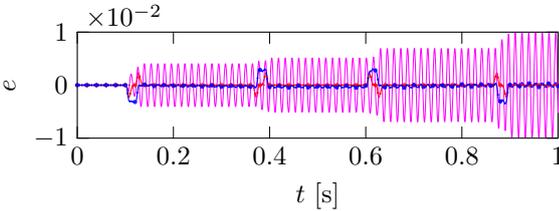




(a) Reference of continuous-time 4<sup>th</sup> order polynomial trajectory.



(b) Tracking error in mass-damper-spring motion system  $G_2$ . (●) and (○) show sampling point every  $T_s$  and  $2T_s$ .



(c) Tracking error in mass-spring-mass motion system  $G_4$ . (●) and (○) show sampling point every  $T_s$  and  $4T_s$ .

Figure 6: Simulation results using multirate feedforward control with model parameters (—), linearly parameterized feedforward control with model parameters (—), and linearly parameterized feedforward control with tuning  $\min\|e[k]\|_2$  (—).

0.5 ms only for evaluation of the continuous-time tracking error  $e(t)$ . The continuous-time tracking error  $e(t)$  is compared to the multirate feedforward control with model parameters, the linearly parameterized feedforward control with model parameters, and the linearly parameterized feedforward control with tuning as  $\min\|e[k]\|_2$ . The simulation is conducted in an open loop without a feedback controller.

#### 4.2.4. Comparison in continuous-time tracking performance

The tracking errors of simulations in the mass-damper-spring motion system are shown in Figure 6(b). It shows that the linearly parameterized feedforward control with the model parameter makes a large error at constant velocity regions because of the mismatch between the model of the feedforward controller and the discretized model of the controlled system. After tuning as  $\min\|e[k]\|_2$ , the linearly parameterized feedforward control provides a smaller error than that of the multirate feedforward control at constant velocity regions. Note that although the multirate feedforward control guarantees the perfect state tracking of position and velocity for every 2 sample in the 2<sup>nd</sup> order motion system, it causes the intersample oscillation because of the mismatch of acceleration and jerk in the continuous-time reference of the 4<sup>th</sup> order polynomial trajectory.

The tracking errors of simulations in the mass-spring-mass

Table 1: Root Mean Square error  $\text{RMS}(e(t))$  with multirate feedforward (MRFF) control and linearly parameterized feedforward (LPFF) control using multirate zero-order-hold differentiator in simulation.

$\text{RMS}(e(t))$	$G_2$	$G_4$
MRFF with model parameters	$1.81 \times 10^{-3}$	$4.65 \times 10^{-4}$
LPFF with model parameters	$1.78 \times 10^{-1}$	$4.14 \times 10^{-3}$
LPFF with tuning $\min\ e[k]\ _2$	$2.20 \times 10^{-3}$	$8.35 \times 10^{-4}$

motion system are shown in Figure 6(c). It shows that the linearly parameterized feedforward control with the model parameter makes a large oscillating error because of the mismatch of the resonance frequency between the model of the feedforward controller and the discretized model of the controlled system. The large oscillating error is improved in the linearly parameterized feedforward control with tuning as  $\min\|e[k]\|_2$ .

The Root Mean Square of the continuous-time tracking errors with the multirate feedforward control and the linearly parameterized feedforward control using the multirate zero-order-hold differentiator in simulations is shown in Table 1. It shows that the tracking error of the linearly parameterized feedforward control with tuning as  $\min\|e[k]\|_2$  is around the same scale as that of the multirate feedforward control even though the solution space of the linearly parameterized feedforward control is limited in the linear space. It means that the basis functions using the multirate zero-order-hold differentiator provide a reasonable linear solution space for the sampled-data feedforward control. As a result, it is shown that the linearly parameterized feedforward control can provide around the same performance as the multirate feedforward control through a tuning process using the experimental data.

## 5. ILC with MIMO structured basis functions

In this section, the feedforward controller parameterization and the parameter updating framework using ILC with basis functions are presented. The structured feedforward controller parameterization for MIMO motion systems is formulated with physically interpretable tuning parameters. Parameter update through learning is described with the monotonic convergence condition in MIMO motion systems. It results in Contribution 2.

### 5.1. MIMO fixed-structure feedforward controller parameterization

The dynamics of mechatronic systems are typically dominated by the mechanics assuming that electronics are much faster than mechanics. This results in a situation where rigid body modes dominate the lower frequency and there are several flexible modes at a higher frequency due to limited mechanical stiffness. The  $n_u$ -input  $n_y$ -output continuous-time multi-modal

motion system [26] is defined as

$$\begin{aligned} \mathbf{G}_c(s) &= \mathbf{G}_r(s) + \mathbf{G}_f(s) \\ &= \underbrace{\sum_{k_r=1}^{n_r} \frac{\mathbf{c}_{k_r} \mathbf{b}_{k_r}}{(s^2 + 2\zeta_{k_r} \omega_{k_r} s + \omega_{k_r}^2)}}_{\text{rigid body modes}} + \underbrace{\sum_{k_f=1}^{n_f} \frac{\mathbf{c}_{k_f} \mathbf{b}_{k_f}}{(s^2 + 2\zeta_{k_f} \omega_{k_f} s + \omega_{k_f}^2)}}_{\text{flexible modes}}, \end{aligned} \quad (32)$$

where  $n_r \in \mathbb{N}_+$  and  $n_f \in \mathbb{N}_+$  are the number of rigid body and flexible modes,  $\omega \in \mathbb{R}_+$  and  $\zeta \in \mathbb{R}_+$  are the resonance angular frequency and the damping coefficient. The vectors  $\mathbf{b} \in \mathbb{R}^{1 \times n_u}$  and  $\mathbf{c} \in \mathbb{R}^{n_y \times 1}$  are associated with the inputs, the outputs, and the mode shapes. In this paper, the system is assumed to be square as  $n_u = n_y$  after the static decoupling based on the coordinate transformation.

To compensate for not only the rigid body modes but also the flexible modes in MIMO motion systems, the traditional rigid body feedforward control is extended with the additional snap feedforward control [14] and the ideal feedforward controller  $\mathbf{F}^*(s)$  is defined as

$$\mathbf{F}^*(s) = \mathbf{G}_r^{-1}(s) + \mathbf{D}^*(s)s^4, \quad (33)$$

where  $\mathbf{D}^*(s)$  is the coefficient of the snap feedforward control aiming to compensate for the compliance of the flexible modes.

The objective of the feedforward controller  $\mathbf{F}^*(s)$  is to minimize the closed-loop error given by

$$\mathbf{e}(s) = \mathbf{S}(s)\mathbf{r}(s) - \mathbf{S}(s)\mathbf{G}_c(s)\mathbf{F}^*(s)\mathbf{r}(s), \quad (34)$$

where  $\mathbf{S}(s)$  denotes the sensitivity function matrix that is defined as  $\mathbf{S}(s) = (\mathbf{I} + \mathbf{G}_c(s)\mathbf{K}_c(s))^{-1}$ . It results in  $\mathbf{F}^*(s) = \mathbf{G}_c^{-1}(s)$  and  $\mathbf{D}^*(s)$  is given by

$$\mathbf{D}^*(s) = \frac{1}{s^4}(\mathbf{G}_c^{-1}(s) - \mathbf{G}_r^{-1}(s)). \quad (35)$$

Assuming the reference trajectory in the mechatronic systems mainly contains the low-frequency components and the resonance frequencies of the rigid body modes are enough smaller than that of flexible modes approximated to  $\omega_{k_r} \approx 0$ , the compliance that corresponds to the low-frequency behavior of the flexible modes is given by

$$\begin{aligned} \mathbf{D} &= \lim_{s \rightarrow 0} \mathbf{D}^*(s) = \lim_{s \rightarrow 0} \left\{ \frac{1}{s^4}(\mathbf{G}_c^{-1}(s) - \mathbf{G}_r^{-1}(s)) \right\} \\ &= - \left( \sum_{k_r=1}^{n_r} \mathbf{c}_{k_r} \mathbf{b}_{k_r} \right)^{-1} \left( \sum_{k_f=1}^{n_f} \frac{\mathbf{c}_{k_f} \mathbf{b}_{k_f}}{\omega_{k_f}^2} \right) \left( \sum_{k_r=1}^{n_r} \mathbf{c}_{k_r} \mathbf{b}_{k_r} \right)^{-1}. \end{aligned} \quad (36)$$

Hence, the fixed-structure feedforward controller for MIMO motion systems is parameterized as

$$\mathbf{F}(\boldsymbol{\theta}) = \underbrace{\boldsymbol{\theta}_p + \boldsymbol{\theta}_v \frac{d}{dt} + \boldsymbol{\theta}_a \frac{d^2}{dt^2}}_{\text{rigid body compensation}} + \underbrace{\boldsymbol{\theta}_s \frac{d^4}{dt^4}}_{\text{compliance compensation}}, \quad (37)$$

where  $\boldsymbol{\theta}_p, \boldsymbol{\theta}_v, \boldsymbol{\theta}_a, \boldsymbol{\theta}_s \in \mathbb{R}^{n_u \times n_y}$  are the parameter matrices of the feedforward controller corresponding to the position, velocity,

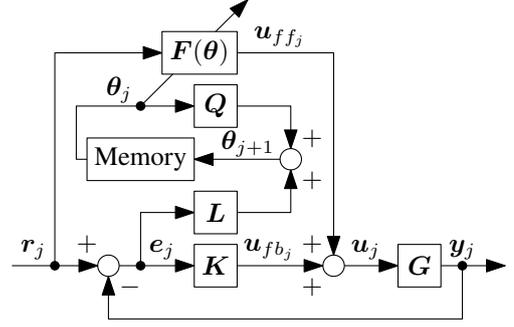


Figure 7: Block diagram of iterative learning control with basis functions.

acceleration and snap basis functions. Finally, the continuous-time differentiator  $\frac{d}{dt}$  is replaced by the sampled-data differentiator  $\xi$ , and the fixed-structure sampled-data feedforward controller for MIMO motion systems is given by

$$\mathbf{F}(\boldsymbol{\theta}) = \boldsymbol{\theta}_p + \boldsymbol{\theta}_v \xi + \boldsymbol{\theta}_a \xi^2 + \boldsymbol{\theta}_s \xi^4. \quad (38)$$

It enables low-complexity parameterization with physical interpretability, flexibility for varying references, and consideration of sampled-data dynamics at the same time. Although using jerk feedforward control as a basis function can improve control performance, in this paper from the viewpoint of interpretability, jerk feedforward control has a less physical meaning, and only snap feedforward control for compliance compensation is included in basis functions in addition to rigid body feedforward control.

In this paper, the developed approach combines the multi-rate zero-order-hold differentiator in (23) and the fixed-structure sampled-data feedforward controller for MIMO motion systems in (38). The developed approach considers both continuous-time tracking performance in Requirement 1 and interaction compensation in Requirement 2.

## 5.2. Norm-optimal ILC with basis functions

Achieving higher performance and ease of tuning for the MIMO feedforward controller, ILC with basis functions is implemented. ILC with basis functions has an advantage in task flexibility compared to traditional ILC. The controller structure is shown in Figure 7. To update the parameters of the feedforward controller through learning, the optimization criterion from the present study [10] is defined as follows.

**Definition 12** (Norm-optimal MIMO ILC with basis functions). *The optimization criterion for norm-optimal ILC with basis functions is given by*

$$\mathcal{J}(\boldsymbol{\theta}_{j+1}) = \|\mathbf{e}_{j+1}\|_{\mathbf{W}_e} + \|\mathbf{u}_{ffj+1}\|_{\mathbf{W}_{ff}} + \|\mathbf{u}_{ffj+1} - \mathbf{u}_{ffj}\|_{\mathbf{W}_{\Delta ff}}, \quad (39)$$

where the weighting matrices are  $\mathbf{W}_e > 0$ ,  $\mathbf{W}_{ff}, \mathbf{W}_{\Delta ff} \geq 0$ , the parameters of the feedforward controller are  $\boldsymbol{\theta}_j \in \mathbb{R}^{n_\theta}$ , and the feedforward control input in next iteration is  $\mathbf{u}_{ffj+1} = \mathbf{F}(\boldsymbol{\theta}_{j+1})\mathbf{r}$ .

Here, the weighting matrices  $\mathbf{W}_e, \mathbf{W}_{ff}, \mathbf{W}_{\Delta ff}$  correspond to optimal performance, robustness for model uncertainty, and robustness for trial varying disturbances including noise, respectively. For the practical tuning procedure of the weighting matrices in the experiment, see [10].

The error in trial  $j + 1$  is given by

$$\mathbf{e}_{j+1} = \mathbf{S}\mathbf{r} - \mathbf{S}\mathbf{G}\mathbf{u}_{ffj+1} \quad (40)$$

$$= \mathbf{e}_j - \mathbf{S}\mathbf{G}(\mathbf{u}_{ffj+1} - \mathbf{u}_{ffj}), \quad (41)$$

where  $\mathbf{S} = (\mathbf{I} + \mathbf{G}\mathbf{K})^{-1}$ .

The feedforward controller parameter update is given by

$$\boldsymbol{\theta}_{j+1}^* = \arg \min_{\boldsymbol{\theta}_{j+1}} \mathcal{J}(\boldsymbol{\theta}_{j+1}). \quad (42)$$

When the feedforward control input is linearly parameterized in parameters  $\boldsymbol{\theta}_{j+1}$  and basis functions  $\boldsymbol{\Psi}$ , and is defined as

$$\mathbf{u}_{ffj+1} = \mathbf{F}(\boldsymbol{\theta}_{j+1})\mathbf{r} = \boldsymbol{\Psi}\boldsymbol{\theta}_{j+1}, \quad (43)$$

the optimization criterion (39) is quadratic in  $\boldsymbol{\theta}_{j+1}$  from (41), and an analytic solution to (42) exists [10]. By solving the necessary condition for optimality  $\frac{\partial \mathcal{J}(\boldsymbol{\theta}_{j+1})}{\partial \boldsymbol{\theta}_{j+1}} = 0$  with basis functions  $\boldsymbol{\Psi} = \frac{\partial}{\partial \boldsymbol{\theta}_j} \mathbf{F}(\boldsymbol{\theta}_j)\mathbf{r} \in \mathbb{R}^{n_u \times n_\theta}$  and weighting matrices  $\mathbf{W}_e, \mathbf{W}_{ff}, \mathbf{W}_{\Delta ff}$ , the analytic solution of (42) for the parameter update law that minimizes  $\mathcal{J}(\boldsymbol{\theta}_{j+1})$  is given by

$$\boldsymbol{\theta}_{j+1} = \mathbf{Q}\boldsymbol{\theta}_j + \mathbf{L}\mathbf{e}_j, \quad (44)$$

where the learning filters  $\mathbf{Q}$  and  $\mathbf{L}$  are given by

$$\mathbf{Q} = (\boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{ff} + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi}, \quad (45)$$

$$\mathbf{L} = (\boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{ff} + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top(\mathbf{S}\mathbf{G})^\top \mathbf{W}_e. \quad (46)$$

From (40), (43), and (44), the parameter update law is written as

$$\boldsymbol{\theta}_{j+1} = (\mathbf{Q} - \mathbf{L}\mathbf{S}\mathbf{G}\boldsymbol{\Psi})\boldsymbol{\theta}_j + \mathbf{L}\mathbf{S}\mathbf{r}. \quad (47)$$

The parameter update law (47) leads to the monotonic convergence condition of the parameters  $\boldsymbol{\theta}_j$  if the provided weighting matrices  $\mathbf{W}_e, \mathbf{W}_{ff}, \mathbf{W}_{\Delta ff}$  are selected properly to satisfy

$$\begin{aligned} \bar{\sigma}(\mathbf{Q} - \mathbf{L}\mathbf{S}\mathbf{G}\boldsymbol{\Psi}) &< 1 \Leftrightarrow \\ \bar{\sigma}((\boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{ff} + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \mathbf{W}_{\Delta ff} \boldsymbol{\Psi}) &< 1, \end{aligned} \quad (48)$$

where  $\bar{\sigma}(\cdot)$  is the maximum singular value of the matrix.

The monotonic convergence of the parameters  $\boldsymbol{\theta}_j$  results in the convergence of the feedforward control input  $\mathbf{u}_{ffj}$  from (43) when the basis functions  $\boldsymbol{\Psi}$  from the reference  $\mathbf{r}$  are fixed through iterations. From (40), (43), and (44), the feedforward control input update law is written as

$$\mathbf{u}_{ffj+1} = (\mathbf{Q}' - \boldsymbol{\Psi}\mathbf{L}\mathbf{S}\mathbf{G})\mathbf{u}_{ffj} + \boldsymbol{\Psi}\mathbf{L}\mathbf{S}\mathbf{r}, \quad (49)$$

where the learning filter  $\mathbf{Q}'$  is given by

$$\mathbf{Q}' = \boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{ff} + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi}^{-1} \boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{\Delta ff}). \quad (50)$$

The feedforward control input update law (49) leads to the monotonic convergence condition of the feedforward control input

$\mathbf{u}_{ffj}$  if the provided weighting matrices  $\mathbf{W}_e, \mathbf{W}_{ff}, \mathbf{W}_{\Delta ff}$  are selected properly to satisfy

$$\begin{aligned} \bar{\sigma}(\mathbf{Q}' - \boldsymbol{\Psi}\mathbf{L}\mathbf{S}\mathbf{G}) &< 1 \Leftrightarrow \\ \bar{\sigma}(\boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{ff} + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \mathbf{W}_{\Delta ff} &< 1. \end{aligned} \quad (51)$$

From (48) and (51), the monotonic convergence conditions of both  $\boldsymbol{\theta}_j$  and  $\mathbf{u}_{ffj}$  are guaranteed if

$$\boldsymbol{\Psi}^\top((\mathbf{S}\mathbf{G})^\top \mathbf{W}_e(\mathbf{S}\mathbf{G}) + \mathbf{W}_{ff} + \mathbf{W}_{\Delta ff})\boldsymbol{\Psi} > 0. \quad (52)$$

Note that these monotonic convergence conditions of the parameters  $\boldsymbol{\theta}_j$  and the feedforward control input  $\mathbf{u}_{ffj}$  are derived from that of the norm-optimal ILC [27, 28] that also can be applied to MIMO systems [29, 30, 31]. Specifically, the monotonic convergence condition of the norm-optimal ILC with basis functions is derived in Section 3.1 and Section 4.1 of [32].

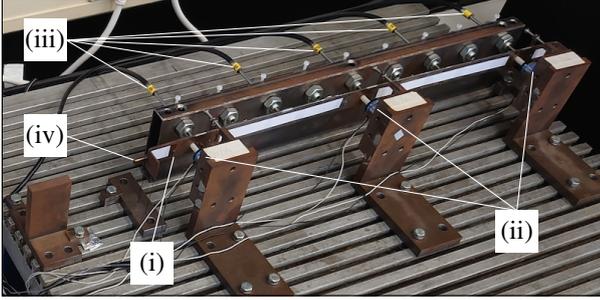
In [10], the basis functions are designed using the backward differentiator that does not take into account the sampled-data characteristics with sampler and zero-order-hold. The theoretical performance limitation is linked to how much the basis functions contain the dynamics of the controlled system, and the lack of the sampled-data characteristics deteriorates the tracking performance. In this paper, the developed approach considers the sampled-data characteristics by using the multirate zero-order-hold differentiator in (23). [10] also does not contain the guidelines for extending to MIMO motion systems. This paper introduces the guidelines with the fixed-structure sampled-data feedforward controller for MIMO motion systems in (38). The performance improvement of the developed approach with the combination of (23) and (38) is validated in Section 6.

## 6. Validation in MIMO sampled-data motion system

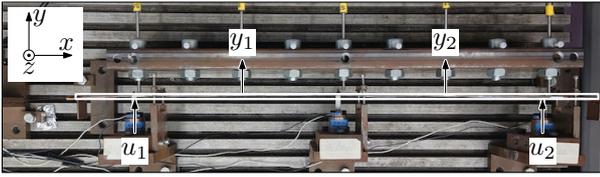
In this section, the developed approach combining Section 3 and Section 5 is applied to a MIMO motion system. The results demonstrate the performance improvement with interaction compensation and sampled-data characteristics in both the simulation and the experiment.

### 6.1. Motion system

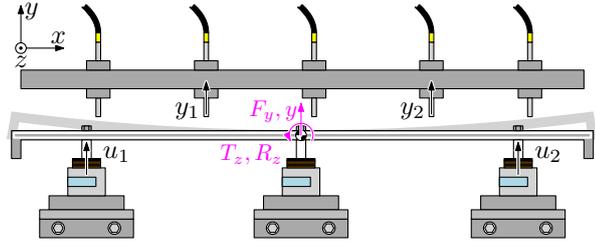
The experimental flexible beam setup of a MIMO motion system is shown in Figure 8. The setup exhibits dominant flexible behavior and coupling dynamics that are expected to arise in high-precision mechatronic systems in industries. Although typical high-precision mechatronic systems operate in six degrees of freedom, the four degrees of freedom are elastically suspended by the leaf spring to facilitate the control design and analysis as shown in Figure 8(a). The real-time controller based on Raspberry Pi with EtherCAT connection is used with the computation frequency 1024 Hz. After the static decoupling of the system with dual-inputs ( $u_1, u_2$ ) and dual-outputs ( $y_1, y_2$ ) based on coordinate transformation as shown in Figure 8(b), the controlled system  $\mathbf{G}$  is given in translation and rotation with dual-inputs ( $F_y, T_z$ ) and dual-outputs ( $y, R_z$ ) as shown in Figure 8(c). The frequency response data obtained by multisine excitation, the continuous-time model  $\mathbf{G}_c$  with the higher-order



(a) Overview photograph of the experimental setup. The system consists of (i) a steel flexible beam (500 mm  $\times$  20 mm  $\times$  2 mm), (ii) three current-driven voice-coil actuators (Akribis Systems: AVM19-5), (iii) five contact-less fiber-optic sensors (Philtec: D64-NQ) with a resolution  $\sim 1 \mu\text{m}$ , and (iv) a leaf spring. The flexible beam is elastically suspended by the leaf spring with a stroke of  $\sim 1$  mm.



(b) Top view photograph of the experimental setup. Two actuators and two sensors are used as dual-inputs ( $u_1, u_2$ ) and dual-outputs ( $y_1, y_2$ ).



(c) Schematic illustration of the experimental setup. After the static decoupling based on coordinate transformation, the controlled system  $\mathbf{G}$  is in translation and rotation with dual-inputs ( $F_y, T_z$ ) and dual-outputs ( $y, R_z$ ). The setup exhibits dominant flexible behavior based on resonance mode and coupling dynamics in translation and rotation.

Figure 8: Experimental flexible beam setup of MIMO motion system.

dynamics for the simulation, and the discrete-time model  $\mathbf{G}_d$  with the only diagonal rigid body dynamics for parameter update and feedback controller design are shown in Figure 9.

## 6.2. Conditions

The continuous-time reference of the translation  $y$  is the 4<sup>th</sup> order polynomial trajectory as shown in Figure 10, and that of the rotation  $R_z$  is set to 0 rad for all time. The sampling frequency of the discrete-time controller is  $F_s = 128$  Hz, as Nyquist frequency is enough higher than the first resonance mode, and the sampling time is  $T_s = 1/F_s$ . Although the sampling frequency in industrial applications is typically higher than 128 Hz, sampled-data dynamics are affected by the relative condition between the length of sampling time and the steepness of reference. The continuous-time outputs  $y$  and  $R_z$  are also measured in higher sampling frequency 1024 Hz only for evaluation of the continuous-time tracking errors  $\mathbf{e}(t)$ . To investigate the intersample performance by using the real-time

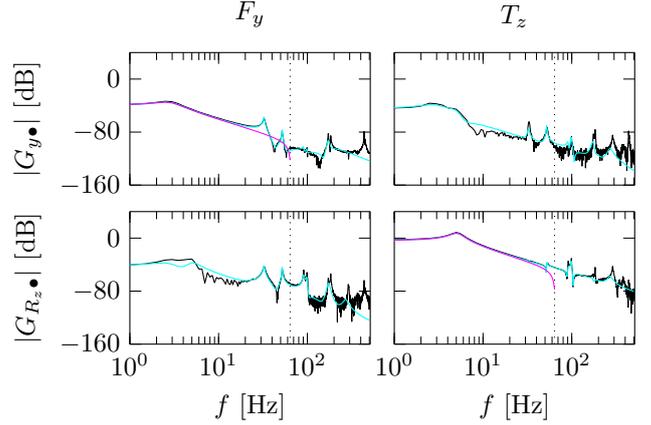


Figure 9: Bode magnitude plot of the experimental setup: frequency response data (—), continuous-time model  $\mathbf{G}_c$  (—) with the higher-order dynamics for the simulation, and discrete-time model  $\mathbf{G}_d$  (—) with the only diagonal rigid body dynamics for parameter update and feedback controller design. Nyquist frequency of the controller is shown in a black dotted line (—).

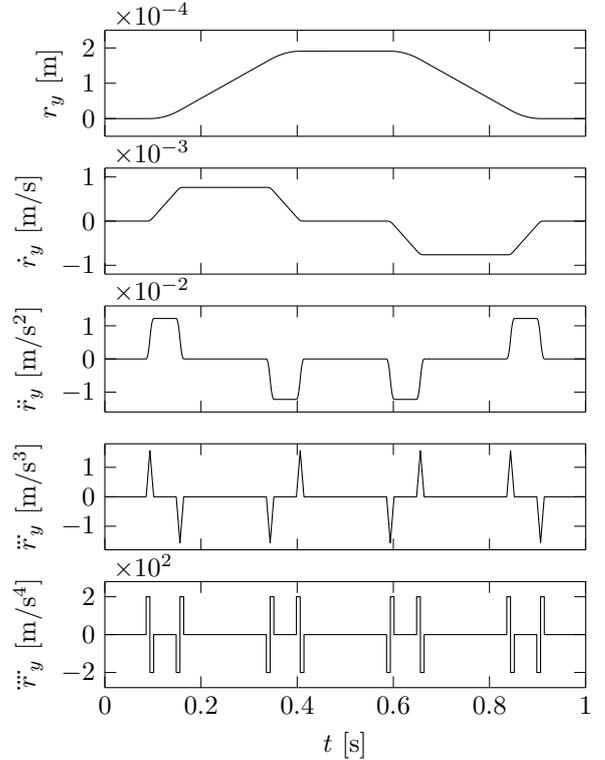


Figure 10: Reference of  $y$ : continuous-time 4<sup>th</sup> order polynomial trajectory and its derivatives. Reference of  $R_z$  is set to 0 rad for all time.

controller with the computation frequency 1024 Hz, the control frequency is set to 128 Hz, and the continuous-time output is measured in 8 times higher sampling frequency 1024 Hz that is equal to the computation frequency of the real-time controller. The feedback controller  $\mathbf{K}$  is designed diagonally with a PD controller and a notch filter as a 5 Hz closed-loop bandwidth and a 6 dB modulus margin for compensation to disturbance and unmodeled dynamics.

### 6.3. Feedforward controller design

From Section 5, the fixed-structure feedforward controller for a MIMO motion system is parameterized as

$$\mathbf{F}(\boldsymbol{\theta}) = \begin{bmatrix} \theta_{p11} & \theta_{p12} \\ \theta_{p21} & \theta_{p22} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \theta_{v11} & \theta_{v12} \\ \theta_{v21} & \theta_{v22} \end{bmatrix} \begin{bmatrix} \xi \\ \xi \end{bmatrix} + \begin{bmatrix} \theta_{a11} & \theta_{a12} \\ \theta_{a21} & \theta_{a22} \end{bmatrix} \begin{bmatrix} \xi^2 \\ \xi^2 \end{bmatrix} + \begin{bmatrix} \theta_{s11} & \theta_{s12} \\ \theta_{s21} & \theta_{s22} \end{bmatrix} \begin{bmatrix} \xi^4 \\ \xi^4 \end{bmatrix}, \quad (53)$$

where  $\xi$  is a sampled-data differentiator. The basis functions of each output are defined as

$$\boldsymbol{\Psi}_y = [r_y \quad \xi r_y \quad \xi^2 r_y \quad \xi^4 r_y], \quad (54)$$

$$\boldsymbol{\Psi}_{R_z} = [r_{R_z} \quad \xi r_{R_z} \quad \xi^2 r_{R_z} \quad \xi^4 r_{R_z}], \quad (55)$$

and the tuning parameter vectors are defined as

$$\begin{aligned} \boldsymbol{\theta}_{11} &= [\theta_{p11} \quad \theta_{v11} \quad \theta_{a11} \quad \theta_{s11}], & \boldsymbol{\theta}_{12} &= [\theta_{p12} \quad \theta_{v12} \quad \theta_{a12} \quad \theta_{s12}], \\ \boldsymbol{\theta}_{21} &= [\theta_{p21} \quad \theta_{v21} \quad \theta_{a21} \quad \theta_{s21}], & \boldsymbol{\theta}_{22} &= [\theta_{p22} \quad \theta_{v22} \quad \theta_{a22} \quad \theta_{s22}]. \end{aligned} \quad (56)$$

In the conventional approach, only the diagonal terms of the feedforward controller are considered. The feedforward control input in the conventional approach is parameterized as

$$\mathbf{u}_{ff} = \mathbf{F}(\boldsymbol{\theta})\mathbf{r} = \boldsymbol{\Psi}\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\Psi}_y & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_{R_z} \end{bmatrix} [\boldsymbol{\theta}_{11} \quad \boldsymbol{\theta}_{22}]^\top. \quad (57)$$

In the developed approach, not only the diagonal terms but also the off-diagonal terms of the feedforward controller are taken into account. The off-diagonal terms also can be obtained for interaction compensation through learning even if only a diagonal model is used for parameter update. The feedforward control input in the developed approach is parameterized as

$$\begin{aligned} \mathbf{u}_{ff} &= \mathbf{F}(\boldsymbol{\theta})\mathbf{r} = \boldsymbol{\Psi}\boldsymbol{\theta} \\ &= \begin{bmatrix} \boldsymbol{\Psi}_y & \boldsymbol{\Psi}_{R_z} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Psi}_y & \boldsymbol{\Psi}_{R_z} \end{bmatrix} [\boldsymbol{\theta}_{11} \quad \boldsymbol{\theta}_{12} \quad \boldsymbol{\theta}_{21} \quad \boldsymbol{\theta}_{22}]^\top. \end{aligned} \quad (58)$$

In both the simulation and the experiment, the weighting matrices are set to  $\mathbf{W}_e = \mathbf{I}$  and  $\mathbf{W}_{ff} = \mathbf{W}_{\Delta ff} = \mathbf{O}$ , and all approaches in this validation satisfy the monotonic convergence condition in (48). The basis functions corresponding to the reference signal and its derivatives are orthogonal and have an analytical solution in the norm-optimal ILC.

### 6.4. Validation with interaction compensation

To validate the performance improvement with interaction compensation, the simulations and the experiments without and with interaction compensation in (57) and (58) are conducted through 20 iterations. In both approaches, the multirate zero-order-hold differentiator  $\xi_{mr}$  in (23) is used as a sampled-data differentiator  $\xi$  to consider sampled-data characteristics. The continuous-time tracking errors, Root Mean Square of tracking errors, and tuning parameters learned through iterations are shown in Figure 11, Figure 12, and Figure 13 for the simulation, and Figure 14, Figure 15, and Figure 16 for the experiment. The

results show that the errors are roughly converged after 10<sup>th</sup> iteration. The translation error  $e_y$  is also improved a little in the simulation and the experiment, but the interaction effect is not serious in the translation  $y$  because the reference of the rotation  $R_z$  is set to 0 rad for all time. The frequency of the residual error  $e_y$  with the feedback controller corresponds to the sensitivity peak around 5 Hz of the closed-loop bandwidth. The rotation error  $e_{R_z}$  is improved significantly with interaction compensation of about factor 100 in the simulation and of about factor 10 in the experiment. Note that the scales of the errors in simulation and experiment are different of about factor 10 in translation  $y$  and of about factor 4 in rotation  $R_z$  because of the dynamics not included in the simulation model, measurement noise, quantization of the sensors and actuators, and communication delay. The controller using the multirate zero-order-hold differentiator does not generate the control inputs that cause the inter-sample oscillation, and Figure 12 and Figure 15 show that Root Mean Square of the discrete-time and continuous-time tracking errors are approximately the same that is the definition of the continuous-time tracking performance improvement. The validation results demonstrate that effective interaction compensation can improve the tracking performance in multivariable motion systems.

### 6.5. Validation with sampled-data characteristics

To validate the performance improvement with sampled-data characteristics, the simulations and the experiments with interaction compensation using the backward differentiator in (6) and the multirate zero-order-hold differentiator in (23) are conducted through 20 iterations. The continuous-time tracking errors, Root Mean Square of tracking errors, and tuning parameters learned through iterations are shown in Figure 17, Figure 18, and Figure 19 for the simulation, and Figure 20, Figure 21, and Figure 22 for the experiment. The translation error  $e_y$  is also improved significantly of about factor 5 in the simulation but is improved a little in the experiment because of the unmodeled dynamics in the simulation such as communication delay. The rotation error  $e_{R_z}$  is improved a little in both the simulation and the experiment. Note that the scales of the errors in simulation and experiment are different about factor 10 in translation  $y$  and about factor 20 in rotation  $R_z$  because of the dynamics not included in the simulation model, measurement noise, and quantization of the sensors and actuators. The validation results demonstrate that considering sampled-data characteristics has the potential to push the envelope of the tracking performance in sampled-data motion systems.

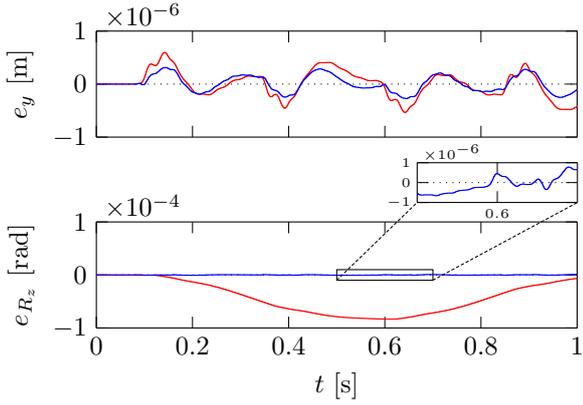


Figure 11: Tracking error  $e(t)$  in simulation using the multirate zero-order-hold differentiator: without (—) and with (—) interaction compensation. Rotation error  $e_{R_z}$  is improved about factor 100.

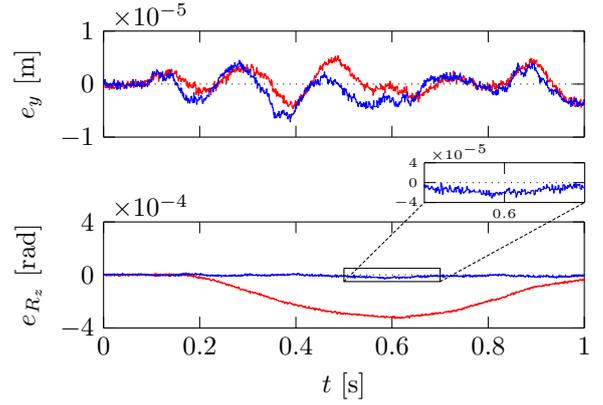


Figure 14: Tracking error  $e(t)$  in experiment using the multirate zero-order-hold differentiator: without (—) and with (—) interaction compensation. Rotation error  $e_{R_z}$  is improved about factor 10.

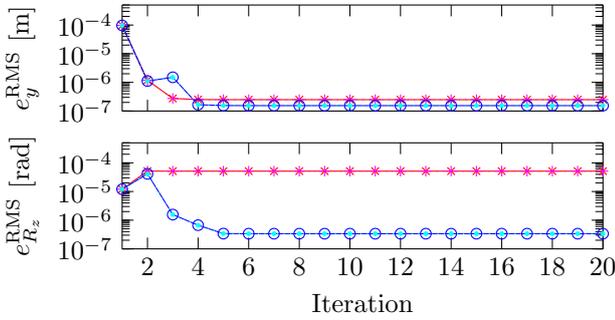


Figure 12: Root Mean Square (RMS) of tracking error in simulation using the multirate zero-order-hold differentiator:  $e[k]$  (—x)  $e(t)$  (—+—) without and  $e[k]$  (—o)  $e(t)$  (—●) with interaction compensation.

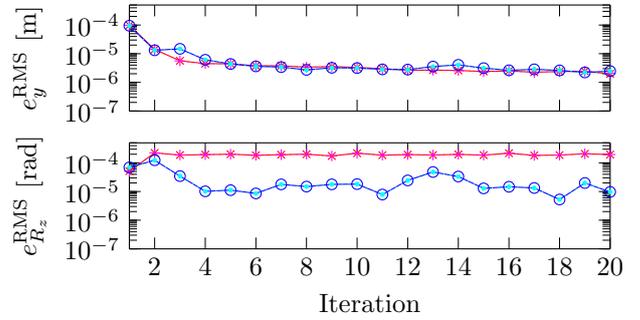


Figure 15: Root Mean Square (RMS) of tracking error in experiment using the multirate zero-order-hold differentiator:  $e[k]$  (—x)  $e(t)$  (—+—) without and  $e[k]$  (—o)  $e(t)$  (—●) with interaction compensation.

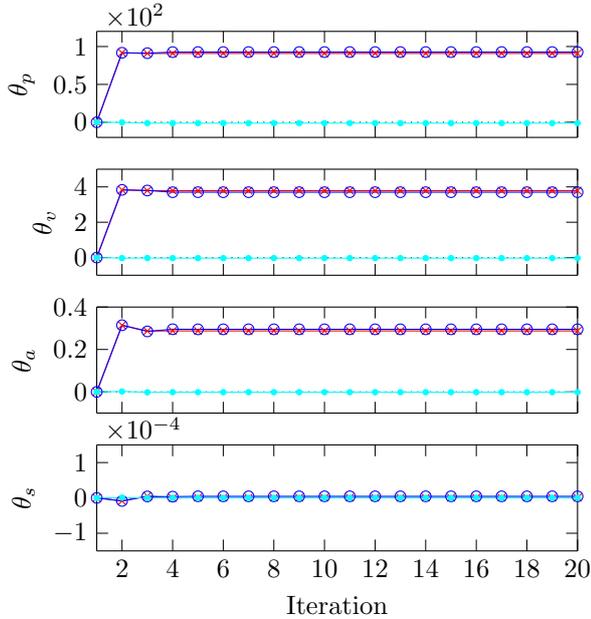


Figure 13: Tuning parameters learned through iterations in simulation using the multirate zero-order-hold differentiator.  $\theta_{\bullet 11}$  (—x) is without interaction compensation.  $\theta_{\bullet 11}$  (—o) and  $\theta_{\bullet 21}$  (—●) are with interaction compensation. Other tuning parameters are  $\theta_{12} = \theta_{22} = \mathbf{0}$  because the reference of the rotation  $R_z$  is set to 0 rad for all time.

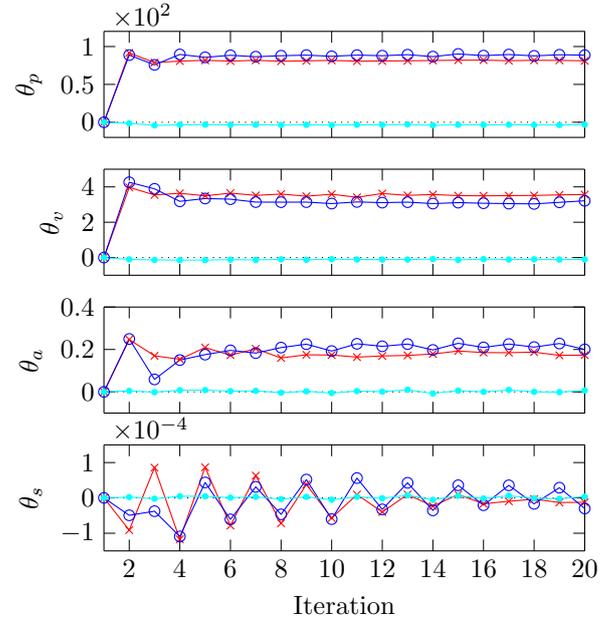


Figure 16: Tuning parameters learned through iterations in experiment using the multirate zero-order-hold differentiator.  $\theta_{\bullet 11}$  (—x) is without interaction compensation.  $\theta_{\bullet 11}$  (—o) and  $\theta_{\bullet 21}$  (—●) are with interaction compensation. Other tuning parameters are  $\theta_{12} = \theta_{22} = \mathbf{0}$  because the reference of the rotation  $R_z$  is set to 0 rad for all time.

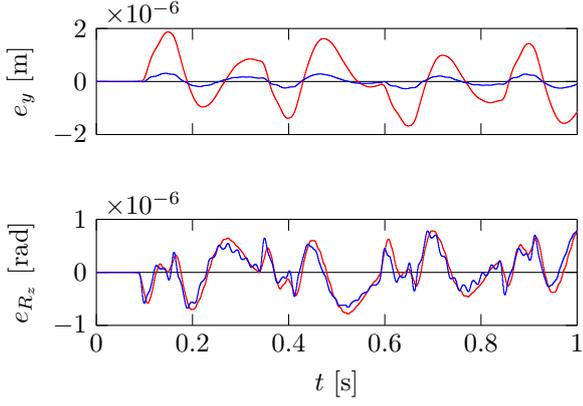


Figure 17: Tracking error  $e(t)$  in simulation with interaction compensation: using the backward differentiator (—) and the multirate zero-order-hold differentiator (—).

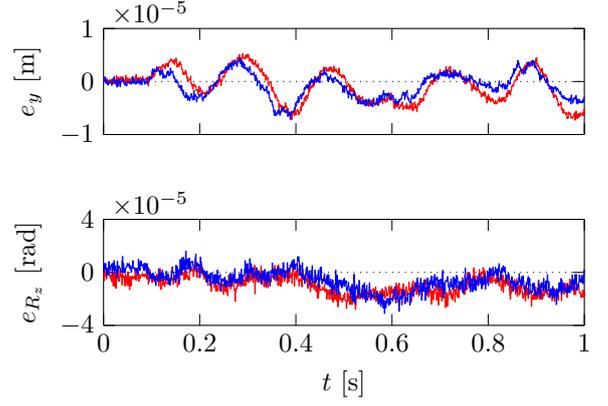


Figure 20: Tracking error  $e(t)$  in experiment with interaction compensation: using the backward differentiator (—) and the multirate zero-order-hold differentiator (—).

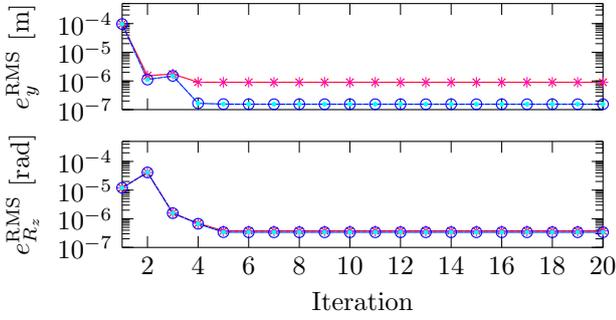


Figure 18: Root Mean Square (RMS) of tracking error in simulation with interaction compensation:  $e[k]$  (—)  $e(t)$  (—) using the backward differentiator and  $e[k]$  (—)  $e(t)$  (—) using the multirate zero-order-hold differentiator.

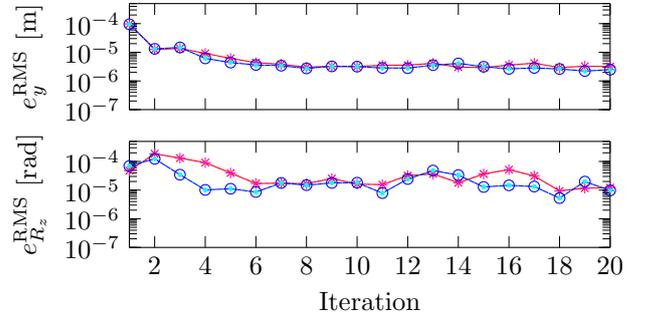


Figure 21: Root Mean Square (RMS) of tracking error in experiment with interaction compensation:  $e[k]$  (—)  $e(t)$  (—) using the backward differentiator and  $e[k]$  (—)  $e(t)$  (—) using the multirate zero-order-hold differentiator.

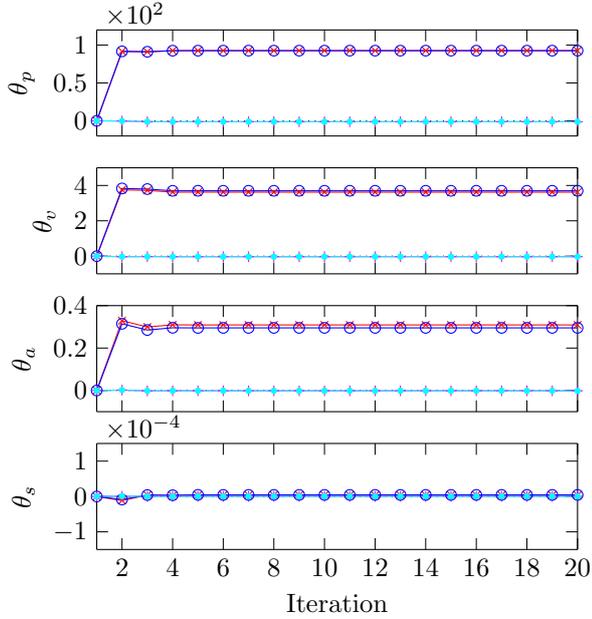


Figure 19: Tuning parameters learned through iterations in simulation with interaction compensation.  $\theta_{*11}$  (—) and  $\theta_{*21}$  (—) are using the multirate zero-order-hold differentiator.  $\theta_{*11}$  (—) and  $\theta_{*21}$  (—) are using the multirate zero-order-hold differentiator. Other tuning parameters are  $\theta_{12} = \theta_{22} = \mathbf{0}$  because the reference of the rotation  $R_z$  is set to 0 rad for all time.

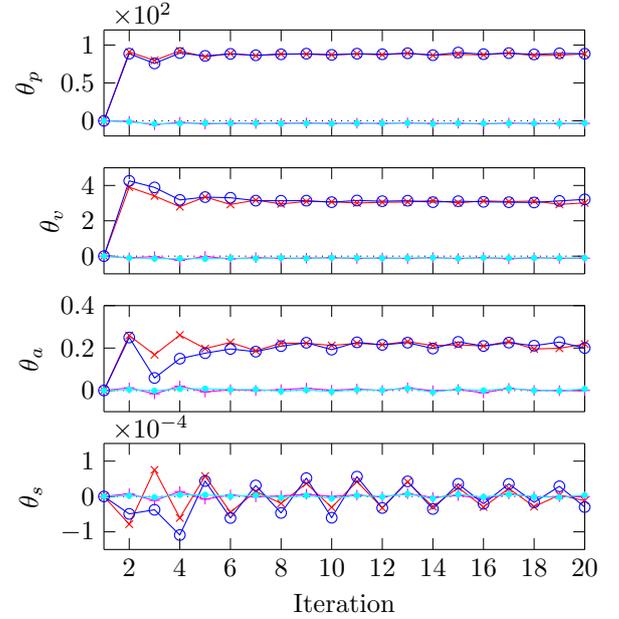


Figure 22: Tuning parameters learned through iterations in experiment with interaction compensation.  $\theta_{*11}$  (—) and  $\theta_{*21}$  (—) are using the multirate zero-order-hold differentiator.  $\theta_{*11}$  (—) and  $\theta_{*21}$  (—) are using the multirate zero-order-hold differentiator. Other tuning parameters are  $\theta_{12} = \theta_{22} = \mathbf{0}$  because the reference of the rotation  $R_z$  is set to 0 rad for all time.

## 7. Conclusion

Fixed-structure feedforward control considering sampled-data characteristics and interactions in MIMO motion systems is developed. The feedforward controller that is parameterized by MIMO sampled-data basis functions and physically interpretable tuning parameters are updated through learning. Application to the sampled-data MIMO motion system demonstrates a significant improvement in tracking performance with interaction compensation compared to the conventional diagonal approach in both the simulation and the experiment. In engineering practice, the discrete-time basis functions that correspond to the continuous-time reference are designed using the multirate zero-order-hold differentiator. The feedforward control signal is generated by the MIMO fixed-structure feedforward controller parameterization using the basis functions. The tuning parameters of the feedforward controller are updated through iterative learning control on batch-to-batch. Ongoing research focuses on ILC with rational sampled-data basis functions and basis function design with higher-order dynamics.

## Acknowledgement

This research was partly supported by JSPS KAKENHI Grant Number 21J13196 and 23K19116.

## References

- [1] M. Heertjes, D. Hennekens, M. Steinbuch, MIMO feed-forward design in wafer scanners using a gradient approximation-based algorithm, *Control engineering practice* 18 (5) (2010) 495–506.
- [2] M. Steinbuch, T. Oomen, H. Vermeulen, Motion Control, *Mechatronics Design*, and Moore's Law, *IEEE Journal of Industry Applications* 2 (4) (2021) 21006010.
- [3] M. Li, T. Chen, R. Cheng, K. Yang, Y. Zhu, C. Mao, Dual-Loop Iterative Learning Control With Application to an Ultraprecision Wafer Stage, *IEEE Transactions on Industrial Electronics* 69 (11) (2022) 11590–11599.
- [4] M. Poot, J. Portegies, N. Mooren, M. van Haren, M. van Meer, T. Oomen, Gaussian Processes for Advanced Motion Control, *IEEE Journal of Industry Applications* 11 (3) (2022) 396–407.
- [5] N. Nikooienejad, M. Maroufi, S. O. R. Moheimani, Iterative Learning Control for Video-Rate Atomic Force Microscopy, *IEEE/ASME Transactions on Mechatronics* 26 (4) (2021) 2127–2138.
- [6] T. Hayashi, H. Fujimoto, Y. Isaoka, Y. Terada, Projection-based Iterative Learning Control for Ball-screw-driven Stage with Consideration of Rolling Friction Compensation, *IEEE Journal of Industry Applications* 9 (2) (2020) 132–139.
- [7] Y.-H. Lee, S.-C. Hsu, T.-Y. Chi, Y.-Y. Du, J.-S. Hu, T.-C. Tsao, Industrial robot accurate trajectory generation by nested loop iterative learning control, *Mechatronics: the science of intelligent machines* 74 (2021) 102487.
- [8] C.-W. Chen, S. Rai, T.-C. Tsao, Iterative Learning of Dynamic Inverse Filters for Feedforward Tracking Control, *IEEE/ASME Transactions on Mechatronics* 25 (1) (2020) 349–359.
- [9] S. Miyoshi, W. Ohnishi, T. Koseki, M. Sato, Output Voltage Precise Tracking Control for Boost Converters Based on Noncausal and Nonlinear Feedforward Control, *IEEE Journal of Industry Applications* 12 (6) (2023) 1114–1126.
- [10] J. Bolder, T. Oomen, S. Koekebakker, M. Steinbuch, Using iterative learning control with basis functions to compensate medium deformation in a wide-format inkjet printer, *Mechatronics: the science of intelligent machines* 24 (8) (2014) 944–953.
- [11] L. Blanken, S. Koekebakker, T. Oomen, Data-driven feedforward tuning using non-causal rational basis functions: With application to an industrial flatbed printer, *Mechatronics: the science of intelligent machines* 71 (2020) 102424.
- [12] Z. Wang, C. P. Pannier, K. Barton, D. J. Hoelzle, Application of robust monotonically convergent spatial iterative learning control to microscale additive manufacturing, *Mechatronics: the science of intelligent machines* 56 (2018) 157–165.
- [13] A. Wache, H. Aschemann, B. J. Krause, J. Kurth, Iterative learning control of a pneumatically actuated lung tumour mimic model for an improvement of PET/CT-imaging, *Mechatronics: the science of intelligent machines* 88 (2022) 102915.
- [14] M. Boerlage, MIMO jerk derivative feedforward for motion systems, in: 2006 American Control Conference, IEEE, 2006, pp. 3892–3897.
- [15] S. H. van der Meulen, R. L. Tousain, O. H. Bosgra, Fixed Structure Feedforward Controller Design Exploiting Iterative Trials: Application to a Wafer Stage and a Desktop Printer, *Journal of dynamic systems, measurement, and control* 130 (5) (2008) 051006.
- [16] P. Lambrechts, M. Boerlage, M. Steinbuch, Trajectory planning and feedforward design for electromechanical motion systems, *Control engineering practice* 13 (2) (2005) 145–157.
- [17] T. Chen, B. A. Francis, *Optimal Sampled-Data Control Systems*, Springer London, 1995.
- [18] J. van Zundert, T. Oomen, On inversion-based approaches for feedforward and ILC, *Mechatronics: the science of intelligent machines* 50 (2018) 282–291.
- [19] H. Fujimoto, Y. Hori, A. Kawamura, Perfect tracking control based on multirate feedforward control with generalized sampling periods, *IEEE Transactions on Industrial Electronics* 48 (3) (2001) 636–644.
- [20] J. van Zundert, W. Ohnishi, H. Fujimoto, T. Oomen, Improving Intersample Behavior in Discrete-Time System Inversion: With Application to LTI and LPTV Systems, *IEEE/ASME Transactions on Mechatronics* 25 (1) (2020) 55–65.
- [21] W. Ohnishi, N. Stribosch, T. Oomen, State-tracking iterative learning control in frequency domain design for improved intersample behavior, *International Journal of Robust and Nonlinear Control* 33 (7) (2023) 4009–4027.
- [22] M. Mae, W. Ohnishi, H. Fujimoto, MIMO multirate feedforward controller design with selection of input multiplicities and intersample behavior analysis, *Mechatronics: the science of intelligent machines* 71 (2020) 102442.
- [23] M. Mae, W. Ohnishi, H. Fujimoto, Multirate feedforward control with mode decomposition for intersample performance in multivariable motion systems, *Control engineering practice* 141 (2023) 105694.
- [24] M. Mae, M. van Haren, W. Ohnishi, T. Oomen, H. Fujimoto, Feedforward with Acceleration and Snap using Sampled-Data Differentiator for a Multi-Modal Motion System, in: *The 2nd Modeling, Estimation and Control Conference*, Vol. 55, Elsevier Ltd, 2022, pp. 253–258.
- [25] K. J. Åström, P. Hagander, J. Sternby, Zeros of sampled systems, *Automatica: the journal of IFAC*, the International Federation of Automatic Control 20 (1) (1984) 31–38.
- [26] W. Gawronski, *Advanced Structural Dynamics and Active Control of Structures*, Springer Science & Business Media, 2004.
- [27] S. Gunnarsson, M. Norrlöf, On the design of ILC algorithms using optimization, *Automatica: the journal of IFAC*, the International Federation of Automatic Control 37 (12) (2001) 2011–2016.
- [28] M. Norrlöf, S. Gunnarsson, Time and frequency domain convergence properties in iterative learning control, *International journal of control* 75 (14) (2002) 1114–1126.
- [29] K. L. Barton, A. G. Alleyne, A Norm Optimal Approach to Time-Varying ILC With Application to a Multi-Axis Robotic Testbed, *IEEE transactions on control systems technology: a publication of the IEEE Control Systems Society* 19 (1) (2011) 166–180.
- [30] J. van Zundert, J. Bolder, S. Koekebakker, T. Oomen, Resource-efficient ILC for LTI/LTV systems through LQ tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer, *Mechatronics: the science of intelligent machines* 38 (2016) 76–90.
- [31] Y. Wang, T. Hsiao, Multivariable Iterative Learning Control Design for Precision Control of Flexible Feed Drives, *Sensors (Basel, Switzerland)* 24 (11) (2024) 3536.
- [32] J. van de Wijdeven, O. H. Bosgra, Using basis functions in iterative learning control: analysis and design theory, *International journal of control* 83 (4) (2010) 661–675.