



Article

# Multivariable Iterative Learning Control Design for Precision Control of Flexible Feed Drives

Yulin Wang<sup>1,2</sup>  and Tesheng Hsiao<sup>1,\*</sup> 

<sup>1</sup> Institute of Electrical and Control Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan; yu.linwang106@gmail.com

<sup>2</sup> School of Mechanical Engineering, Guangdong Ocean University, Zhanjiang 524088, China

\* Correspondence: tshsiao@cn.nctu.edu.tw

**Abstract:** Advancements in machining technology demand higher speeds and precision, necessitating improved control systems in equipment like CNC machine tools. Due to lead errors, structural vibrations, and thermal deformation, commercial CNC controllers commonly use rotary encoders in the motor side to close the position loop, aiming to prevent insufficient stability and premature wear and damage of components. This paper introduces a multivariable iterative learning control (MILC) method tailored for flexible feed drive systems, focusing on enhancing dynamic positioning accuracy. The MILC employs error data from both the motor and table sides, enhancing precision by injecting compensation commands into both the reference trajectory and control command through a norm-optimization process. This method effectively mitigates conflicts between feedback control (FBC) and traditional iterative learning control (ILC) in flexible structures, achieving smaller tracking errors in the table side. The performance and efficacy of the MILC system are experimentally validated on an industrial biaxial CNC machine tool, demonstrating its potential for precision control in modern machining equipment.

**Keywords:** iterative learning control; flexible feed drives; multivariable control; norm-optimal



**Citation:** Wang, Y.; Hsiao, T. Multivariable Iterative Learning Control Design for Precision Control of Flexible Feed Drives. *Sensors* **2024**, *24*, 3536. <https://doi.org/10.3390/s24113536>

Academic Editor: Paolo Mercorelli

Received: 26 April 2024

Revised: 26 May 2024

Accepted: 29 May 2024

Published: 30 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In advanced manufacturing equipment such as manipulators [1], CNC machine tools [2], 3D printers [3], and lithography machines [4], the throughput and accuracy of workpieces are significantly affected by the dynamic positioning accuracy of their feed drive systems. In industrial applications, electromechanical systems are inevitably affected by the structural dynamics of the feed drives and uncertain disturbance, significantly limiting the achievable accuracy of motion control systems. For precision motion systems, the ultimate goal is to use high-performance control algorithms to suppress various disturbances in diverse operational environments. Disturbances affecting control accuracy are typically divided into two main categories: internal and external disturbances.

For CNC machine tool motion systems, cutting forces between the workpiece and the tool are common forms of external disturbances. High-gain feedback controllers are typically utilized to mitigate these disturbances induced by the process. To further enhance performance, more advanced motion controllers have been considered. These include pole placement [5], loop shaping [6], and  $H_\infty$  synthesis strategies [7]. Additionally, feedback control (FBC), using nonlinear control techniques like sliding mode control [8] and model predictive control [9], was adopted to enhance the dynamic positioning accuracy of motion systems.

On the other hand, the presence of internal disturbances, such as lead errors, thermal errors, and weight deformation errors in ball screws, also limits commercial CNC machines to closing the position loop using rotary encoder feedback. These disturbances can lead to premature wear and damage to mechanical components when a linear encoder is used to close the position loop [10]. Recent studies have illustrated that finite element

modeling (FEM) methods can adeptly mitigate these inaccuracies [11]. However, the pre-calibration process is notably time-consuming and requires substantial prior knowledge, obstructing its widespread adoption in engineering applications. Additionally, due to the inherent flexibility of the feed drive system, the accuracy at the motor side does not directly correspond to the required precision at the table side, which is of primary interest. In light of this, the present study proposes a strategy that iteratively refines the reference trajectory commands based on the observed results from the linear encoder on the table side.

Residual vibrations pose another significant challenge to the dynamic accuracy of precision motion stages, primarily stemming from the structural dynamics like resonance damping in flexible feed drive components. These vibrations are induced by inertial forces generated during high-speed and high-acceleration trajectories. However, FBC systems, despite incorporating advanced algorithms, inherently experience delays and transient responses to changes in reference inputs and disturbances. Conversely, a feedforward control signal can be designed to preemptively target future reference commands and disturbances, utilizing an understanding of the system's dynamics and the information at hand. Techniques like notch filtering [12] and input shaping [13] are employed to modify the spectral energy around the resonance frequency in trajectory commands, aiming to reduce vibrations induced by the trajectory. Yet, these approaches may introduce additional delays and can distort trajectory commands, which can compromise the accuracy of contour tracking in multi-axis motion. Furthermore, plant inversion methods such as Zero Phase Error Tracking Control (ZPETC) [14], Zero Magnitude Error Tracking Control (ZMETC), or Nonminimum-Phase Zero Ignore (NPZ-Ignore) [15] can also be used for pre-filtering trajectory commands to prevent vibrations and improve trajectory tracking accuracy. Nonetheless, the effectiveness of these feedforward filtering techniques is dependent on the precise inversion of the model, necessitating an accurate identification of the system's dynamic model.

Iterative Learning Control (ILC) is a feedforward control strategy designed to enhance the performance of systems carrying out repetitive tasks [9,16,17]. This method utilizes error information collected from previous tasks to enhance the performance of current tasks, effectively suppressing the repetitive disturbances mentioned above. The underlying assumptions of ILC is that the reference signal is repetitive. Each repetition is referred to as an iteration, and the initial conditions (ICs) for each iteration are identical.

Notice that these assumptions are not restrictive, because repetitive machining is very common in batch manufacturing. ILC can be effectively used to synthesize error compensation strategies for such scenarios. It is usually integrated into an existing FBC system since ILC itself is incapable of stabilizing an unstable system or compensating for non-repetitive disturbances.

In the literature, there are several mainstream frameworks for designing ILC systems, including model-based ILC, data-driven ILC, and adaptive ILC. Model-based ILC generates control compensation using a known system model and previous error data. This strategy is recognized for its simplicity and ease of implementation, featuring methods like frequency-domain inversion-based and time-domain norm-optimal methods [17]. This technique is widely used in manufacturing systems with linear models, such as CNC machines and wafer stages. However, there is still room for improvement, including the study of flexible structures in this paper. Data-driven ILC [18–21] parameterizes the learning feedforward by introducing basis functions and iteratively obtaining optimal parameters related to the model. This ensures performance under varying reference, but it has limited ability to counter disturbances. Adaptive ILC [22–24] employs Lyapunov-like methods to design learning rules and a parametric updating law to estimate system uncertainties. This addresses uncertainties and disturbances in nonlinear systems. Over the past few years, the ILC field has witnessed significant advancement. For instance, Zhang et al. [25] proposed a data-driven ILC combined with predictive control to address the multivariable tracking problem with actuator constraints. This approach transforms the problem into an iteration-varying quadratic programming problem using only the system's I/O data. Chi

et al. [26] developed a data-driven indirect ILC method for repetitive nonlinear systems. They introduced an adaptive iterative learning rule to update the gains of the indirect iterative learning law for set-points. Li et al. [27] presented an iterative learning-based predictive control method for asynchronous switching of multiphase batch processes with complex characteristics. This method was implemented within the framework of a two-dimensional system.

However, these ILCs are designed for minimizing a single variable, and thus, the majority of existing ILC structures are only suitable for rigid body systems, where the motor position and the table position are identical. A straightforward approach to mitigate table-side errors while minimizing changes to the current rotary encoder feedback control structure is to integrate an additional ILC based on linear encoder measurements. However, conflicts between FBC and ILC might arise due to inconsistency between the motor side and table side errors. Therefore, appropriate modifications to traditional ILC architectures become necessary. Despite this, to the best of the authors' knowledge, only a handful of studies concentrated on ILC design for flexible structures.

Chen et al. [28] proposed a dual-stage ILC for robots with joint elasticity to address model mismatch, but tuning the dual gains could be tedious, and the learning process must be reset whenever the gains are adjusted. Wang et al. [29] introduced a robust  $H_\infty$  synthesis ILC, which integrated torque and motor reference learning to improve error convergence and overcome the limitations of [28]. However, these approaches require more intricate models, making the design process more complex. J. Wallen et al. [30] designed an observer-based ILC, where table side displacements were inferred from real-time measurements of the motor side displacements, and the standard ILC directly updated the compensation command. Nonetheless, J. Bolder et al. [31] pointed out that this method could result in internal instability if the ILC acted directly on performance variables (i.e., the table side error) while the feedback controller acted on measurement variables (i.e., the motor side error). Dumanli et al. [32] developed multiple prefilters based on flexible dynamic systems to iteratively correct reference inputs and control compensation signals, ultimately reducing errors at both the motor and table side. This approach involves designing each filter separately, leading to a substantial design workload. Overall, these works highlight the importance of well-coordinated learning in feedforward and FBC to achieve high-precision motion.

It has been observed that prefiltering trajectory commands for FBC could overcome the contradictions between FBC and ILC [31]. However, this requires an accurate model, and robustness to disturbances is compromised. Motivated by the concept of prefiltering trajectory commands, this paper introduces a multivariable ILC (MILC) method. This aims to reconcile the conflicts between FBC and ILC, achieving precise table side trajectory tracking in flexible drive systems. MILC constructs an error matrix to correct the trajectory command and the control inputs, allowing for simultaneous suppression of errors in both FBC and ILC. Then we carry out experiments to verify the effectiveness of MILC in a biaxial motion stage.

The primary contribution of this paper is the development of a multivariable ILC (MILC) framework that combines a two-degrees-of-freedom (DOF) architecture with reference shaping and feedforward compensation. This framework aims to enable simultaneous suppression of errors in both the motor side and the table side for systems with flexible feed drives. The specific contributions of this paper are outlined as follows:

C1: A thorough analysis of prevalent ILCs in controlling flexible structures, highlighting the importance of reference shaping for enhancing ILC performance.

C2: The development of a two-DOF-based MILC framework that effectively reduces feedback errors while boosting ILC performance.

C3: Experimental validation of the MILC approach on a motion system, demonstrating its effectiveness.

This research underlines the need for careful consideration of the balance between ILC and FBC objectives when applying ILC to motion systems with flexible feed drives.

The remainder of this paper is organized as follows. Section 2 introduces the notation and foundational concepts of model-based ILC. The problem under consideration is formulated in Section 3. Section 4 details a combined reference shaping and control feedforward MILC architecture designed for flexible structure control, including stability analysis and considerations for implementation. Section 5 validates the effectiveness of the proposed method by experiments. Lastly, conclusions are drawn in Section 6.

The following conventions of notations are adopted in this paper. A discrete-time single-input, single-output (SISO) linear time-invariant (LTI) system  $P$  is expressed as

$$P = C(zI - A)^{-1}B + D \stackrel{z}{=} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad (1)$$

where  $z$  denotes the complex variable of the  $z$ -transform for a discrete-time system, and  $A$ ,  $B$ ,  $C$ , and  $D$  are matrices of the state space.

Let  $\|x\|_M := x^T M x$ , where  $x \in \mathbb{R}^n$  and  $M \in \mathbb{R}^{n \times n}$  is positive definite ( $M \succ 0$ ), i.e.,  $x^T M x > 0, \forall x \neq 0$ . It is positive semi-definite ( $M \succcurlyeq 0$ ) if and only if  $x^T M x \geq 0, \forall x$ .

In this paper, the design of ILC is approached through the lifted form, which depicts the input and output of a SISO and LTI system  $P$  in the following manner:

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}}_{\underline{y}} = \underbrace{\begin{bmatrix} D & 0 & \cdots & 0 \\ h[1] & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & \cdots & D \end{bmatrix}}_P \underbrace{\begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N-1] \end{bmatrix}}_{\underline{u}} \quad (2)$$

where  $h$  is the impulse response of  $P$ ,  $h[k] = CA^{k-1}B$ ,  $k = 1, 2, \dots$ , and  $N$  is the length of each iteration. The underline denotes the respective lifted matrix form.

## 2. ILC Preliminaries

The MILC is a method that evolves from the model-based ILC. This section analyzes the principles of the model-based ILC and the implementation methods of norm-optimal calculations.

The standard ILC setup can be depicted in Figure 1. In this configuration,  $r_d$  represents the desired reference input;  $y_j^m$  denotes the system trajectory output of the  $j$ -th iteration, with the subscript  $j$  indicating the iteration number and the superscript  $m$  indicating the motor side;  $e_j$  represents the motion error of the  $j$ -th iteration; and  $u_j$  signifies the ILC compensation signal for the same iteration.  $P_M$  stands for the motor side control plant, which includes actuators, mechanics, and sensors, while  $C_P$  is a feedback controller. Additionally, a subscript of 0 indicates that the signal originates from the initial iteration prior to the execution of ILC.

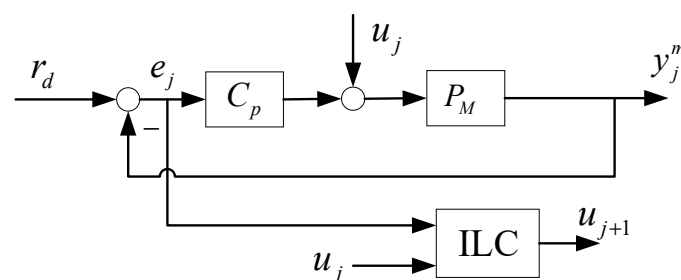


Figure 1. Standard ILC setup.

The feedback error for the  $j$ -th iteration can be expressed as follows:

$$e_j = r_d - y_j^m = Sr_d - Hu_j \quad (3)$$

where  $S = (1 + C_P P_M)^{-1}$  represents the sensitivity function, and  $H = S P_M$  is the learning model that maps  $u_j$  (the learning input) to  $y_j^m$  (the desired output). When ILC feedforward is not applied, as in the initial trial, where  $j = 0$  and  $u_0 = 0$ , the feedback error is expressed as  $e_0 = Sr_d$ . This error can be minimized by reducing the sensitivity function  $S$  at frequencies where the power spectrum of  $r_d$  and disturbances is significant, while also maintaining the stability of the closed-loop system. To achieve this, the paper utilizes a traditional proportional-integral-derivative (PID) type feedback controller.

The ILC compensation signal is calculated as follows:

$$u_{j+1} = \underline{L}_u u_j + \underline{L}_e e_j \quad (4)$$

where  $\underline{L}_u$  and  $\underline{L}_e$  denote the learning functions [17,33], which are determined by minimizing the criterion  $J$  presented in Definition 1.

**Definition 1.** (Performance criterion). A commonly used performance criterion for norm-optimal ILC (NOILC) is defined as follows [14]:

$$J(u_{j+1}, e_{j+1}) = \|e_{j+1}\|_{W_q}^2 + \|u_{j+1} - u_j\|_{W_r}^2 + \|u_{j+1}\|_{W_s}^2 \quad (5)$$

with  $(W_q, W_r, W_s)$  semi-positive definite weighting matrices, which are employed to impose penalties on  $e_{j+1}$ ,  $u_{j+1} - u_j$ , and  $u_{j+1}$ , respectively. A smaller  $W_r$  results in a larger  $u_{j+1} - u_j$ , indicating increased sensitivity of the ILC output to the error information from the previous iteration. As a result, the ILC becomes more vulnerable to non-repetitive disturbances. Conversely, to enhance robust monotonic convergence, a common strategy is to increase  $W_s$ , which, however, slows down the error convergence [34,35]. Thus, the selection of these weighting matrices involves balancing robustness with performance.

Since  $e_{j+1}$  is affine with respect to  $u_{j+1}$ , and  $J(u_{j+1}, e_{j+1})$  is a convex quadratic function of  $u_{j+1}$ , the optimal feedforward control compensation  $u_{j+1}$  can be analytically derived by setting the derivative of the cost function to zero:

$$\frac{dJ(u_{j+1}, e_{j+1})}{du_{j+1}} = 0 \quad (6)$$

thereby identifying the optimal compensation  $u_{j+1}^*$  for the next iteration, as described in [17] and specified by Theorem 1.

**Theorem 1.** (Minimizer of the Performance Criterion in Definition 1) With the model  $H$  and data from the previous iteration, including  $u_j$ ,  $e_j$ , and  $r_d$ , the optimal  $u_{j+1}$  that minimizes the NOILC performance criterion in Definition 1 is:

$$\underline{L}_u = \left( \underline{H}^T W_q \underline{H} + W_s + W_r \right)^{-1} \left( \underline{H}^T W_q \underline{H} + W_r \right) \quad (7)$$

$$\underline{L}_e = \left( \underline{H}^T W_q \underline{H} + W_s + W_r \right)^{-1} \underline{H}^T W_q \quad (8)$$

**Proof of Theorem 1.** By rearranging  $e_j$  and  $e_{j+1}$  in (3), and subsequently eliminating  $r_d$ , we can derive the error dynamics in the iteration domain as follows:

$$e_{j+1} = e_j + \underline{H}(u_j - u_{j+1}) \quad (9)$$

Substitute (9) into (5) and solve (6) to obtain (7) and (8).  $\square$

From Theorem 1, it is evident that NOILC is capable of handling both multivariable scenarios and noncausal computation, the latter of which enables it to effectively deal with non-minimum phase systems. However, calculation of the  $N \times N$  matrix inversion in Equations (7) and (8) is computationally demanding. Consequently, J. van Zundert introduced a more efficient NOILC algorithm [36] that solves Riccati equations to decrease the computational load. For details on this algorithm, please see Appendix A.

### 3. Flexible Feed Drive and Problem Formulation

This section begins by introducing the fundamental two-mass model for flexible ball screw feed drives, which aids in understanding our control objectives. It then proceeds to outline the limitations associated with the direct application of standard ILC to motion systems with flexible drives and feedback from the table side position, which is referred to as Direct ILC (DILC) in this paper.

#### 3.1. Dynamics of Flexible Ball Screw Drive

Presently, ball screw (BS) feed drives are the predominant choice in most industrial feed drive systems. A typical ball screw feed drive in CNC machine tools and its corresponding two-mass model are illustrated in Figure 2a,b [25]. Due to the flexible coupling and the long shaft between the motor and the table, this mechanism can be modeled as a flexible two-mass system [37]. The achievable motion accuracy of this system is constrained by structural vibrations. In the diagram,  $m_M$  and  $m_T$  represent the mass on the motor side and the table side, respectively. The control input is denoted as  $u$ , while  $c_s$  and  $k_s$  represent the viscous damping and the finite stiffness coefficients between the two masses.  $c_M$  and  $c_T$  signify the viscous friction arising from the guideway and bearing system.

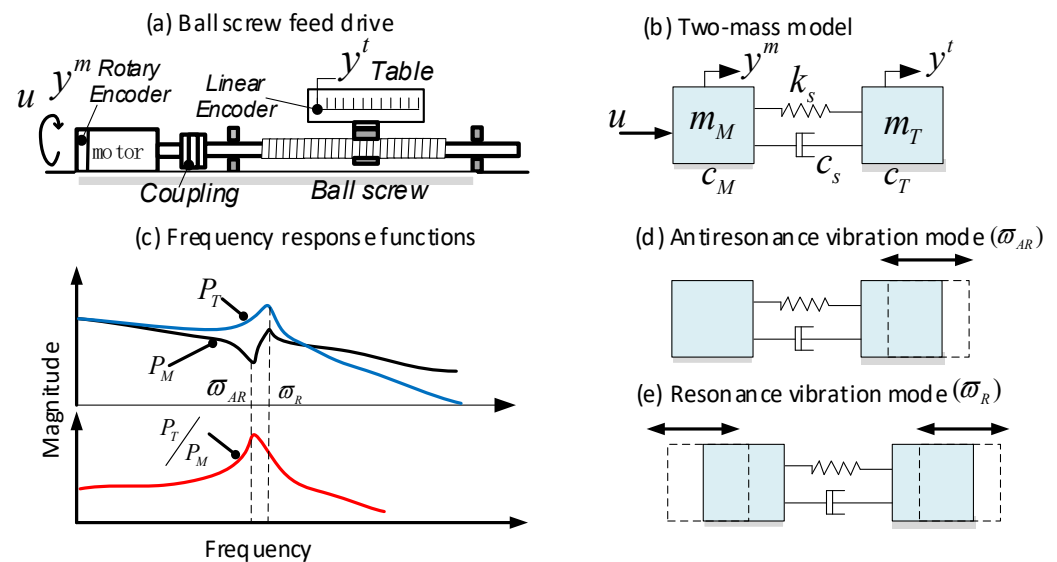


Figure 2. Dynamics of flexible ball screw feed drives. (Redrawn based on Figure 2 of [32]).

The differential equations governing the behavior of this flexible BS drive are formulated as follows:

$$\begin{aligned} m_M \ddot{y}^m &= -c_M \dot{y}^m - k_s (y^m - y^t) - c_s (\dot{y}^m - \dot{y}^t) + u \\ m_T \ddot{y}^t &= -c_T \dot{y}^t + k_s (y^m - y^t) + c_s (\dot{y}^m - \dot{y}^t) \end{aligned} \quad (10)$$

The aforementioned equations can be translated into the Laplace(s) domain as follows:

$$P_M(s) = \frac{m_T s^2 + (c_s + c_T)s + k_s}{q_1 s^4 + q_2 s^3 + q_3 s^2 + q_4 s} \quad (11)$$

$$P_Z(s) = \frac{P_T(s)}{P_M(s)} = \frac{c_s s + k_s}{m_T s^2 + (c_T + c_M)s + k_s} \quad (12)$$

where  $P_M(s)$  and  $P_T(s)$  represent the transfer functions (TF) from the control input  $u$  to the motor side and table side position,  $y^m$  and  $y^t$ , respectively. Then  $P_Z(s)$  defined in (12) is the TF from the motor side position to the table side position. In addition,  $q_1 = m_M m_T$ ,  $q_2 = (m_T + m_M)c_s + m_M c_T + m_T c_M$ ,  $q_3 = (m_T + m_M)k_s + c_M c_T + (c_M + c_T)c_s$ , and  $q_4 = (c_M + c_T)k_s$ .

In the case of collocated control, where the sensor and the actuator are located in close proximity, FBC is designed based on  $P_M(s)$ . This setup leads to an antiresonance vibration mode at  $\omega_{AR} = \sqrt{k_s/m_T}$ , as shown in Figure 2d. On the other hand, in the non-collocated control strategy, where the feedback signal is taken from the table side ( $P_T(s)$ ), the system exhibits the natural resonance vibration mode at  $\omega_R = \sqrt{k_s(m_T + m_M)/m_M m_T}$ , as illustrated in Figure 2e.

Both the resonance and anti-resonance vibration modes can cause relative motion between the motor and the table side, and adversely affect the positioning accuracy of the table. This fact highlights that it is necessary to control both the motor side and the table side simultaneously. A majority of industrial servo closed-loop control systems primarily employ the collocated control approach to guarantee larger stability margins [38]. Additionally, mechanical manufacturing errors such as lead errors, and thermal deformation of the BS, can cause premature wear and damage if a non-collocated control strategy is directly utilized.

Therefore, building upon the arguments presented in Section 1, we propose an MILC method based on collocated control in Section 4. This approach involves iterative tuning to both the control compensation and the reference input, aiming to simultaneously suppress position errors on both the motor side and the table side. To demonstrate the effectiveness of MILC, the following subsection presents the theoretical analysis and proof of traditional ILC directly applied to flexible structures.

### 3.2. Applying Standard ILC to the Flexible Structure

As illustrated in Figure 3 [30], the traditional ILC, when directly applied to flexible structures, is referred to as DILC in this paper. Within the DILC framework, the feedback controller regulates the motor side error, i.e.,  $e_j^{c1} = r_d - y_j^m$ , while the ILC is designed to control the table side error, i.e.,  $e_j^{z1} = r_d - y_j^t$ .

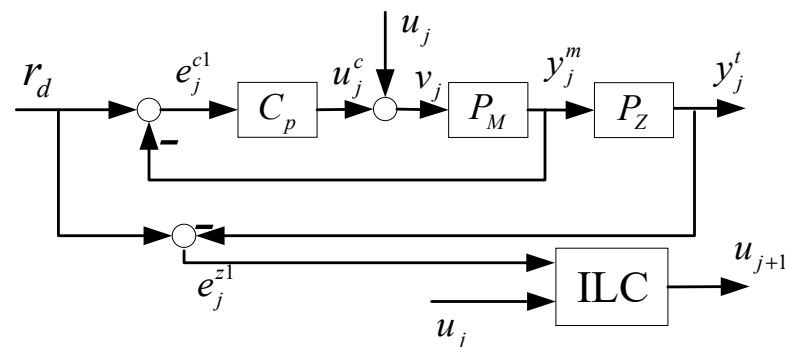


Figure 3. ILC directly applied to flexible structure (DILC).

Instead of analyzing the performance of DILC in the time domain, this paper leverages the frequency domain. Frequency domain analysis, a well-established tool for infinite-

time horizon, is also widely utilized in ILC systems [17], where the repetitive signals have finite lengths. Employing a frequency domain approach sheds light on the system's performance across different frequencies, making the interaction between FBC and DILC more comprehensible.

Define the total control input to the plant as  $v_j = u_j + u_j^c = u_j + C_P(r_d - P_M v_j)$ ; then

$$v_j = S u_j + S C_P r_d \quad (13)$$

The error on the table side is expressed as:

$$\begin{aligned} e_j^{z1} = r_d - y_j^t &= r_d - P_Z P_M v_j = (1 - P_Z P_M S C_P) r_d - P_Z P_M S u_j \\ &= (1 - P_Z (1 - S)) r_d - P_Z P_M S u_j \end{aligned} \quad (14)$$

By comparing (3) and (14), it is observed that the learning model for DILC is represented by  $H_D = S P_M P_Z$ . On the assumption that the ILC stability condition, as detailed in [17], is satisfied, we proceed in our analysis to the steady-state behavior (i.e.,  $j \rightarrow \infty$ ) of DILC. The steady state signals of  $u_j$ ,  $u_j^c$ ,  $e_j^{z1}$ , and  $e_j^{c1}$  are henceforth denoted as  $u_\infty$ ,  $u_\infty^c$ ,  $e_\infty^{z1}$ , and  $e_\infty^{c1}$ , respectively. Employing Equation (4) as a basis, we derive the following equation:

$$u_\infty = (1 - L_u)^{-1} L_e e_\infty^{z1} \quad (15)$$

By substituting (15) into (14), we obtain

$$\begin{aligned} e_\infty^{z1} &= (1 - P_Z (1 - S)) r_d - P_Z P_M S u_\infty \\ &= (1 - L_u) (1 - L_u + P_Z P_M S L_e)^{-1} (1 - P_Z (1 - S)) r_d \end{aligned} \quad (16)$$

Consequently, the steady-state output of DILC is given by

$$u_\infty = L_e (1 - L_u + P_Z P_M S L_e)^{-1} (1 - P_Z (1 - S)) r_d \quad (17)$$

On the other hand, as derived from (13), the error at the motor side is formulated as

$$e_j^{c1} = r_d - y_j^m = r_d - P_M v_j = S r_d - P_M S u_j \quad (18)$$

Based on these expressions for the steady-state errors, we put forward the following proposition. This proposition underscores the equilibrium between errors and control inputs, not only on the table side but also on the motor side.

**Proposition 1.** Assume that the DILC system in Figure 3 is stable. Then

1.  $\|(1 - P_Z) r_d\|_2 \leq \|e_j^{z1}\|_2 + \|P_Z\|_2 \|e_j^{c1}\|_2$ , where  $\|P\|_2$  denotes the 2-norm of  $P$ ;
2.  $u_\infty^c = (1 - L_u + P_Z P_M S L_e)^{-1} [C_P (1 - L_u) + L_e] S r_d - u_\infty$ .

**Proof of Proposition 1.** (a) According to (14),

$$\begin{aligned} e_j^{z1} &= (1 - P_Z (1 - S)) r_d - P_Z P_M S u_j \\ &= P_Z (S r_d - P_M S u_j) + (1 - P_Z) r_d \\ &= P_Z e_j^{c1} + (1 - P_Z) r_d \end{aligned} \quad (19)$$

Then the result follows from the triangular inequality.

(b) From Figure 4 and (17), the steady-state control input from the feedback controller is



$$\begin{aligned}
u_{\infty}^c &= C_p S r_d - C_p P_M S u_{\infty} = C_p S r_d - (1 - S) u_{\infty} \\
&= \left[ C_p S - (1 - S) L_e (1 - L_u + P_Z P_M S L_e)^{-1} (1 - P_Z (1 - S)) \right] r_d \\
&= (1 - L_u + P_Z P_M S L_e)^{-1} [C_p S (1 - L_u) + (1 - S) P_Z S L_e - (1 - S) L_e (1 - P_Z (1 - S))] r_d \\
&= (1 - L_u + P_Z P_M S L_e)^{-1} [C_p (1 - L_u) + L_u] S r_d - u_{\infty}
\end{aligned} \tag{20}$$

□

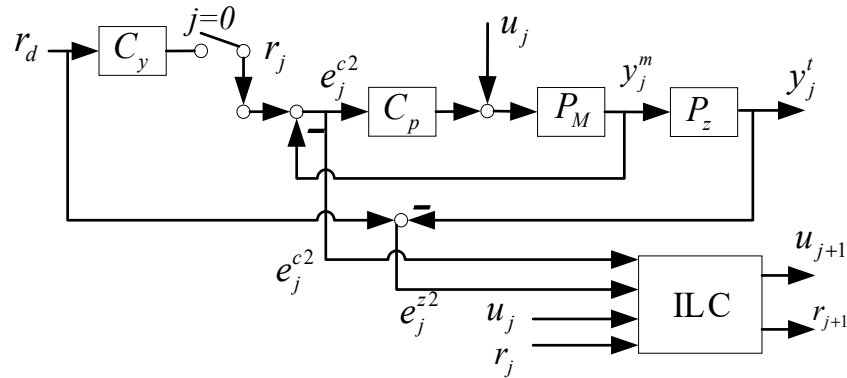


Figure 4. Two-DOF based MILC configuration.

Proposition 1a posits that both  $e_j^{z1}$  and  $e_j^{c1}$  are restricted by a lower bound, which is  $\|(1 - P_Z)r_d\|_2$ . This suggests that for a specified  $r_d$ , they cannot simultaneously diminish to zero. Proposition 1b, on the other hand, illustrates inherent conflict between the feedback controller and DILC. To be specific,  $u_{\infty}^c$  cancels out  $u_{\infty}$ , signifying that they counteract each other in the steady state.

This observation can be understood by examining the resonance modes illustrated in Figure 2. Structural vibrations, induced by large accelerations within the trajectory reference, cause relative motion between the motor and table sides. Concentrating solely on mitigating errors on one side inevitably leads to an amplification of errors on the opposite side, given the interconnected nature of these components. Consequently, the positioning accuracy attained is suboptimal.

In addition, we explore an extreme scenario where  $w_s = 0$  in (5). According to (7),  $L_u = 1$ , and based on (16),  $e_{\infty}^{z1} = 0$ . By employing Equations (17) and (20), it is determined that

$$u_{\infty} = (P_Z P_M S)^{-1} (1 - P_Z (1 - S)) r_d \tag{21}$$

and

$$u_{\infty}^c = (P_Z P_M)^{-1} r_d - u_{\infty} \tag{22}$$

For systems with rigid connections, that is,  $P_Z = 1$ , Equation (22) indicates that  $u_{\infty}^c = 0$ , implying that only DILC contributes to the control input in the steady state. However, when  $P_Z \neq 1$ , there occurs a cancellation effect between  $u_{\infty}^c$  and  $u_{\infty}$ . Moreover, in a well-designed FBC system,  $|S(\omega)| \ll 1$  for the frequencies within the control bandwidth. Consequently, according to (22), both  $u_{\infty}$  and  $u_{\infty}^c$  could become exceedingly large, potentially leading to the saturation of the hardware devices or introducing numerical complexities in the implementation of the control algorithm.

#### 4. Proposed Approach and Analysis

As previously mentioned, direct application of DILC to flexible feed drive systems results in a conflict between the objectives of FBC and ILC. To mitigate this issue, this section aims to suppress both motor side errors and table side errors. This objective is achieved by refining the control compensation and reference inputs, leading to the development of a multivariable ILC (MILC) architecture. This strategy forms contribution C2 of this study. In

addition, comprehensive insights into the stability analysis and the specific implementation of this architecture are thoroughly discussed.

#### 4.1. Multivariable ILC Design

The structure of the MILC is depicted in Figure 4, where  $r_j$  and  $u_j$  represent the reference update and the control compensation update, respectively. It is important to note that during the initial trial,  $r_0 = C_y r_d$  with the prefilter  $C_y = P_Z^{-1}$ , and  $u_0 = 0$ . Within this MILC framework, the calculations of  $y_j^m$  and  $y_j^t$  are as follows:

$$y_j^m = C_P P_M S r_j + P_M S u_j \tag{23}$$

$$y_j^t = C_P P_M P_Z S r_j + P_M P_Z S u_j \tag{24}$$

In this context,  $e_j^{c2}$  signifies the motor side error utilized for the feedback controller within the MILC framework. This is expressed as

$$e_j^{c2} = r_j - y_j^m \tag{25}$$

Furthermore,  $e_j^{z2}$  representing the table side error, is the ultimate interest for control, and expressed as follows:

$$e_j^{z2} = r_d - y_j^t \tag{26}$$

The Equations (23)–(26) can be reorganized into matrix form as follows:

$$\begin{aligned} \begin{bmatrix} e_j^{c2} \\ e_j^{z2} \end{bmatrix} &= \begin{bmatrix} 0 \\ r_d \end{bmatrix} - \begin{bmatrix} y_j^m - r_j \\ y_j^t \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ r_d \end{bmatrix} - \begin{bmatrix} P_M S & -S \\ P_M P_Z S & C_P P_M P_Z S \end{bmatrix} \begin{bmatrix} u_j \\ r_j \end{bmatrix} \end{aligned} \tag{27}$$

Define the learning model of MILC as  $H_M = \begin{bmatrix} P_M S & -S \\ P_M P_Z S & C_P P_M P_Z S \end{bmatrix}$ ; then

$$\begin{bmatrix} e_j^{c2} \\ e_j^{z2} \end{bmatrix} = \begin{bmatrix} 0 \\ r_d \end{bmatrix} - H_M \begin{bmatrix} u_j \\ r_j \end{bmatrix} \tag{28}$$

The MILC update rule in lifted system format is as follows:

$$\begin{bmatrix} u_{j+1} \\ r_{j+1} \end{bmatrix} = \underline{L}_u \begin{bmatrix} u_j \\ r_j \end{bmatrix} + \underline{L}_e \begin{bmatrix} e_j^{c2} \\ e_j^{z2} \end{bmatrix} \tag{29}$$

The purpose of this design is to effectively suppress both motor side FBC error  $e_j^{c2}$  and table side performance error  $e_j^{z2}$ , while avoiding the issues associated with DILC, as highlighted in Proposition 1. Equation (29) implies that the MILC strategy is a two-DOF approach.

This paper uses the following algorithm steps to demonstrate the algorithm process:

- Step 1: Set  $r_j = \frac{1}{P_z} r_d$ ,  $u_j = 0$ . Perform the tracking control task in real time with FBC only.
- Step 2: For  $j \geq 0$ , collect errors  $e_j^{c2}$  and  $e_j^{z2}$ , and use Formula (9) to offline calculate the compensation for the next iteration,  $r_{j+1}$  and  $u_{j+1}$ .
- Step 3: Perform the tracking control task in real time with reference  $r_{j+1}$ , and ILC compensation  $u_{j+1}$ .
- Step 4: Return to Step 2 and repeat until the desired number of iterations is achieved.

#### 4.2. Implementation Aspects

Fundamentally, MILC functions as a multiple-input, multiple-output (MIMO) system, whose learning rule can be realized through the model-based NOILC detailed in Section 2 and Appendix A. Furthermore, the stability of MILC is guaranteed if the weighting matrices in (5) are chosen appropriately, as shown in Section 4.3. The precondition for efficient implementation of the NOILC method is to possess a state-space model that maps ILC learning inputs to the desired output, as outlined in Appendix A. To obtain this model, the blocks in Figure 4 are expressed in state-space representation, as follows:

$$C_p = \left[ \begin{array}{c|c} A_c & B_c \\ \hline C_c & D_c \end{array} \right] \quad (30)$$

$$P_m = \left[ \begin{array}{c|c} A_m & B_m \\ \hline C_m & 0 \end{array} \right] \quad (31)$$

$$P_z = \left[ \begin{array}{c|c} A_z & B_z \\ \hline C_z & 0 \end{array} \right] \quad (32)$$

and they are assumed to be minimal realizations.

**Lemma 1.** *The state space representation of  $H_M$  as depicted in Equation (28), specifically the transfer matrix from  $(u_j, r_j)$  to  $(-e_j^{c2}, y_j^t)$ , is given by:*

$$H_M = \left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \quad (33)$$

with

$$A = \begin{bmatrix} A_c & -B_c C_m & \mathbf{0} \\ B_m C_c & A_m - B_m D_c C_m & \mathbf{0} \\ \mathbf{0} & B_z C_m & A_z \end{bmatrix}, B = \begin{bmatrix} \mathbf{0} & B_c \\ B_m & B_m D_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, C = \begin{bmatrix} \mathbf{0} & C_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & C_z \end{bmatrix}, \text{ and } D = \begin{bmatrix} \mathbf{0} & -1 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

where boldface  $\mathbf{0}$  denotes the zero matrix with appropriate dimension.

**Proof of Lemma 1.** By systematically substituting the state updates and the output interconnection as outlined in (27), (30), (31), and (32), we arrive at the aforementioned representation.  $\square$

#### 4.3. Stability and Convergence Analysis of MILC

In this subsection, monotonic convergence conditions of MILC are explored for the nominal plant model.

**Theorem 2.** *(Monotonic convergence condition of MILC) In the NOLILC algorithm, where  $W_q \succcurlyeq 0$ ,  $W_s \succcurlyeq 0$ , and  $W_r \succcurlyeq 0$  as defined in Definition 1, monotonic convergence for MILC is guaranteed for MILC learning laws (29) if  $W_q$  and  $W_s$  are symmetric,  $W_s \succ 0$ , and  $W_r = w_r I_N$ , where  $w_r > 0$  and  $I_N$  denotes an  $N \times N$  identity matrix.*

**Proof of Theorem 2.** According to (28) and (29), the trial dynamics of MILC can be expressed as

$$\begin{bmatrix} u_{j+1} \\ r_{j+1} \end{bmatrix} = (\underline{L}_u - \underline{L}_e \underline{H}_M) \begin{bmatrix} u_j \\ r_j \end{bmatrix} + \underline{L}_e \begin{bmatrix} 0 \\ r_d \end{bmatrix} \quad (34)$$

Monotonic convergence [33] of the MILC output requires that

$$\|L_u - L_e H_M\|_{i2} < 1 \quad (35)$$

where  $\|(\bullet)\|_{i2}$  denotes the induced 2-norm of matrix  $(\bullet)$ . For the NOILC design learning in (7) and (8), we can derive

$$\|L_u - L_e H_M\|_{i2} = \left\| \left( H_M^T W_q H_M + W_s + W_r \right)^{-1} W_r \right\|_{i2} < 1 \quad (36)$$

Given the conditions that  $W_q \succcurlyeq 0$ ,  $W_s \succ 0$  and they are symmetric, we have  $X = H_M^T W_q H_M + W_s$  is symmetric and positive definite. In addition,

$$\left( H_M^T W_q H_M + W_s + W_r \right)^{-1} W_r = \left( w_r^{-1} X + I_N \right)^{-1} \quad (37)$$

Let  $\lambda_i(\bullet)$  denote the  $i$ -th eigenvalue of  $(\bullet)$ . Since  $w_r^{-1} X \succ 0$  and is symmetric, we have  $\lambda_i(w_r^{-1} X + I_N) = \lambda_i(w_r^{-1} X) + 1 > 1$  for  $i = 1, \dots, N$ . Therefore, the following result is obtained:

$$\max_i \lambda_i \left( w_r^{-1} X + I_N \right)^{-1} < 1 \quad (38)$$

Moreover,  $w_r^{-1} X + I_N$  is symmetric. Thus

$$\left\| \left( w_r^{-1} X + I_N \right)^{-1} \right\|_{i2} = \max_i \lambda_i \left( w_r^{-1} X + I_N \right)^{-1} < 1 \quad (39)$$

Therefore, inequality (36) holds and monotonic convergence of MILC is guaranteed.  $\square$

**Remark 1.** Typically, the weighting matrices are chosen as  $(W_q, W_r, W_s) \triangleq (w_q I_N, w_r I_N, w_s I_N)$ , and the parameters  $w_q$ ,  $w_s$ , and  $w_r$  are commonly set as  $w_q > 0$ ,  $w_s > 0$ , and  $w_r > 0$ .

**Remark 2.** The sufficient condition for MC outlined in Theorem 2 is a general condition that is also applicable to the NOILC design of DILC.

As stated in Proposition 1a, in DILC,  $e_j^{c1}$  and  $e_j^{z1}$  cannot be zero simultaneously. In contrast, MILC operates differently. Assuming that Theorem 2 holds, the conditions under which  $e_j^{c2}$  and  $e_j^{z2}$  can simultaneously be zero are outlined in the following remark. This demonstrates that MILC can effectively resolve the conflict between FBC and ILC.

**Remark 3.** Given that  $e_j^{z2} = 0$ , it follows that  $y_j^t = r_d$ . Consequently,  $y_j^m = \frac{1}{P_z} r_d$ , and, since  $e_j^{m2} = 0$ , we deduce  $r_j^* = \frac{1}{P_z} r_d$ . Therefore,  $u_j^* = \frac{1}{P_z P_m} r_d$ .

## 5. Experimental Results

### 5.1. Experimental Setup

As shown in Figure 5, a biaxial motion stage, manufactured by Tongtai Machine & Tool Co., Ltd., Kaohsiung, Taiwan, is employed to assess the tracking performance of MILC. This stage is driven by two servo motors from Shihlin Electric & Engineering Co., Taipei, Taiwan. Each motor is equipped with a rotary encoder for a semi-closed position loop, providing a resolution of 52  $\mu$ rad per pulse. The ball screw pitch lengths are 10 mm and 12 mm for the X and Z axes, respectively. Additionally, every axis is equipped with a linear encoder to measure the table position, offering a resolution of 0.25  $\mu$ m per pulse. The actuators of both motors are controlled by a National Instruments cRIO 9035 embedded system with 4 kHz sampling frequency, and it offers a 16-bit resolution of analog output. The National Instruments cRIO 9035, made in Austin, TX, USA, ensures precise control and

measurement. Since the two axes are independent, we apply the control laws to the X axis for experimental verification.

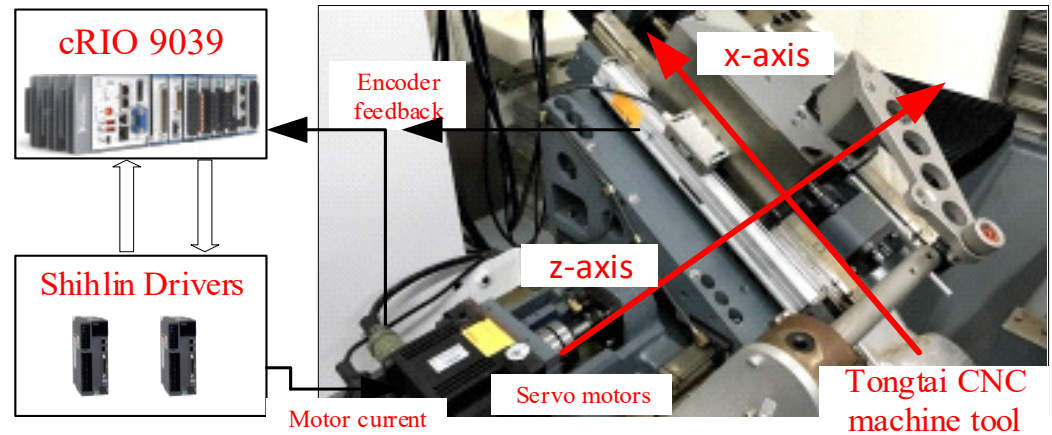


Figure 5. CNC machine tool motion system.

The servo motor for each axis employs the commonly used three-loop control architecture, as depicted in Figure 6. This control structure comprises an outer position loop, an intermediate velocity loop, and an innermost current loop.

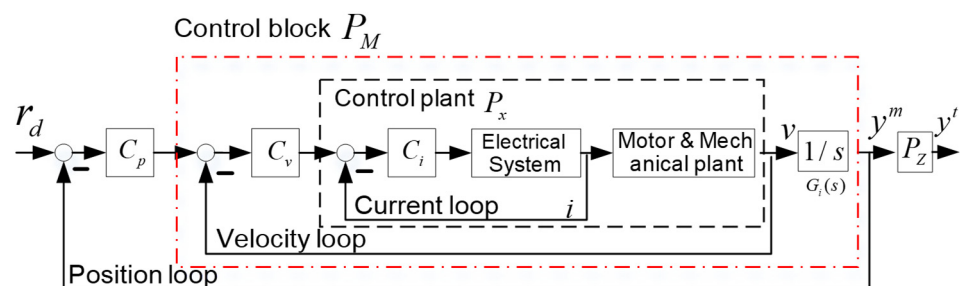


Figure 6. Servo motor cascaded three-loop structure.

In this study, the closed current control loop, as delineated by the dashed box in Figure 6, is regarded as the plant. Within this setup, the torque command serves as the input and the velocity as the output. Enveloping the plant is the velocity control loop and subsequently the position control loop, both operating at a sampling rate of  $f = 4$  kHz.

The discrete-time TF from the torque command to the velocity of the X-axis is represented by  $P_x(z)$ , while the TF from the motor position to the table side position is denoted as  $P_z(z)$ . Both TFs have been identified through the analysis of experimental data. The magnitude responses of the TFs  $P_x(z)$  and  $P_z(z)$  are depicted in Figure 7.

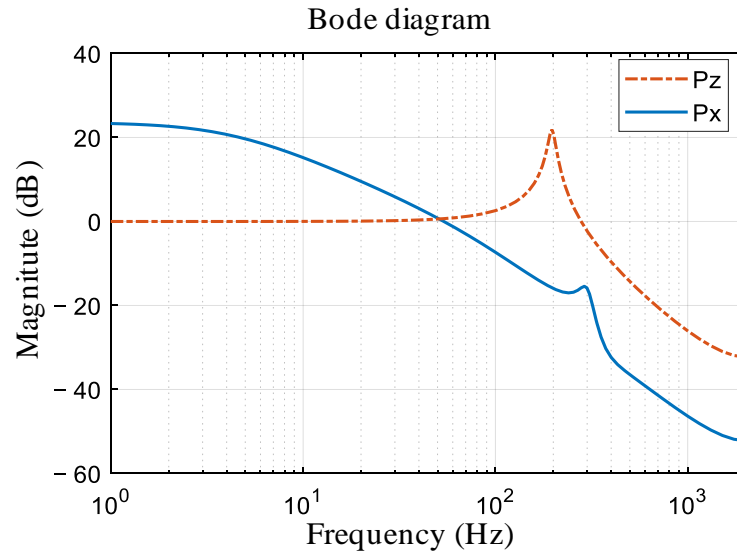


Figure 7. Open loop frequency response functions of nominal  $P_x$  and  $P_z$ .

In this framework,  $C_p(z)$  denotes the position loop feedback controller,  $C_v(z)$  denotes the velocity feedback controller, and  $C_i(z)$  encapsulates the combination of the current loop controller and its corresponding filter [2]. The position loop controller is configured as a proportional ( $P$ ) controller, and the velocity controller is designed as a proportional-integral ( $PI$ ) controller. These controllers are tuned by using the PID tuning toolbox in MATLAB and digitally executed within the CRIO 9035 embedded system. The expressions for  $C_p(z)$ ,  $C_v(z)$ , and a backward type integrator  $G_i(z)$  are as follows:

$$C_p(z) = K_{pp} \quad (40)$$

$$C_v(z) = K_{vp} + K_{vi} \frac{T_s z}{z-1}, \quad G_i(z) = \frac{T_s z}{z-1} \quad (41)$$

where  $K_{vp} = 0.216$ ,  $K_{vi} = 8.18$ ,  $K_{pp} = 180$ ,  $T_s = 1/f$ .

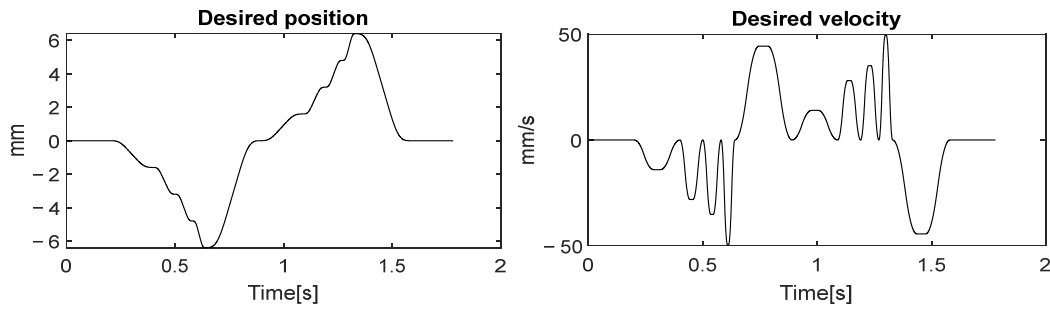
$$P_M(z) = \frac{C_v(z)P_x(z)}{1 + C_v(z)P_x(z)} G_i(z) \quad (42)$$

By applying the controllers (38) and (39), the position closed loop possesses a bandwidth ( $-3$  dB) of 50 Hz.

To demonstrate the significant performance improvement of MILC, DILC is also implemented by the norm-optimal approach for comparison purposes. The state-space representation for DILC implementations is provided in Appendix B.

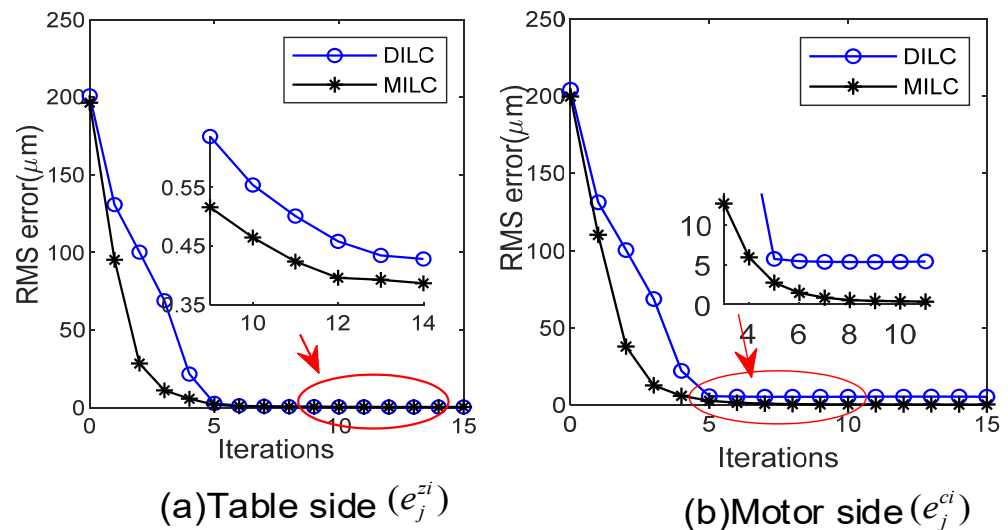
## 5.2. Experimental Results

The effectiveness of the proposed MILC is evaluated in comparison with DILC through experiments on the motion stage. The desired trajectory  $r_d$  (illustrated in Figure 8) consists of a sequence of smoothed step references, and every step reference is designed using an S-curve velocity profile. This trajectory, encompassing a wide range of accelerations, is apt for evaluating the characteristics of a dynamic system in high-throughput manufacturing scenarios. Due to practical considerations of model mismatch [34], the performance weights in (5) are set as  $w_q = 2 \times 10^6$ ,  $w_r = 10^{-4}$ , and  $w_s = 0.2 \times 10^2$  for both MILC and DILC schemes. Furthermore, feedforward friction compensation is effectively implemented by a look-up table, which is based on the result gathered from preliminary friction identified experiments. To prevent instantaneous fluctuations in friction compensation due to measurement noise in low-speed regions, the reference velocity is utilized as the input for the look-up table.



**Figure 8.** Desired trajectory is a forward and backward movement.

The experimental results are depicted in Figures 9 and 10. The performance is evaluated using the root mean squared (RMS) and maximum (MAX) errors in every iteration. After repeated iterations, the relationship between iteration index and RMS of the tracking error is shown in Figure 9. With DILC, once the motor side error decreases to a certain level, further reductions are not observed despite an increase in iterations. In contrast, the proposed MILC successfully reduces errors at both the motor and table sides to a much smaller range. It should be noted that the proposed scheme is different from DILC because it iteratively corrects the reference input and control compensation, aligning the objectives of FBC and ILC for more rapid and effective convergence of table side errors.



**Figure 9.** RMS of tracking error by DILC and MILC.

Additionally, a comparison of the two approaches is made using the table side tracking error ( $e_j^{zi}, i = 1, 2$ ) in the 10-th iteration, as shown in Figure 10 and summarized in Table 1. In sections with high acceleration, such as near 0.57 s, 0.59 s, and 0.61 s as highlighted in the magnified view, the tracking performance of DILC significantly worsens, but MILC demonstrates a notable performance improvement. For the proposed MILC scheme, errors at both the motor side and table side can be simultaneously suppressed, and outstanding compensation performance can be achieved, i.e., MILC achieves a MAX error of 2.06  $\mu\text{m}$ , which is approximately a 42% improvement compared to DILC. As inferred from Figures 2 and 7, due to the presence of large accelerations, resonance modes of  $P_x$  and  $P_z$ , are triggered, resulting in relative motion between the motor side and the table side. Controlling both sides is necessary to achieve high-speed and high-precision motion control.

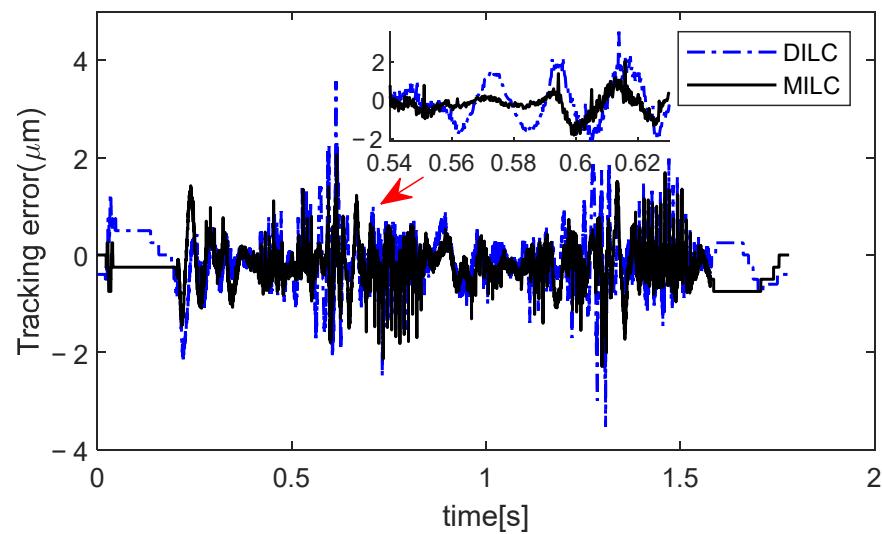


Figure 10. Comparison of  $e_j^{zi}$  in 10-th trial ( $j = 10$ ) for different ILCs.

Table 1. Performance in different approaches ( $\mu\text{m}$ , [The symbol “\*” indicates the best result under the current metrics]).

Method	RMS ( $e_0^{zi}$ )	RMS ( $e_0^{ci}$ )	RMS ( $e_{10}^{zi}$ )	RMS ( $e_{10}^{ci}$ )	MAX ( $e_{10}^{zi}$ )
DILC <sub><math>i=1</math></sub>	201.00	204.21	0.55	5.39	3.57
MILC <sub><math>i=2</math></sub>	196.46 *	199.81 *	0.46 *	0.40 *	2.06 *

Figure 11 displays the control input from the feedback controller in the 10th trial for DILC and MILC. Due to the cancellation between  $u_j^c$  and  $u_j$ , DILC exhibits a substantial control input from the feedback controller. Conversely, MILC, by simultaneously tuning the reference input and the control compensation, maintains a small control input from the feedback controller in the steady state. This result is attributed to the consistent objectives between FBC and MILC to jointly suppress errors at both the motor and table sides, while there are conflicts in control objectives between FBC and DILC as illustrated in Proposition 1b.

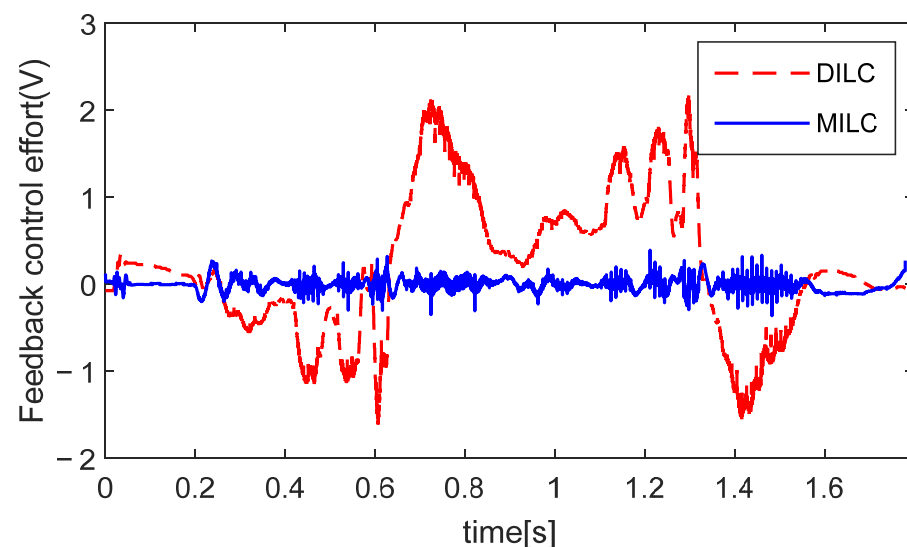


Figure 11. FBC effort  $u^c$  in 10-th trial for different ILCs.



## 6. Conclusions

This paper presents a novel multivariable ILC strategy designed to improve the dynamic positioning accuracy of flexible feed drive motion systems by simultaneously reducing errors at both the motor and table sides. The approach utilizes a two-DOF control framework, combining iterative reference shaping with control compensation updates, effectively balancing the objectives of FBC and ILC. This development is driven by the need to address relative movements caused by the flexible couplings between the motor and table sides. These movements are prompted by various disturbances, including closed-loop dynamics, lead error, structural vibrations, and thermal error. The proposed method has been validated on an industrial CNC machine tool, demonstrating improvements in dynamic tracking errors of up to 42% (as seen in Figure 10 and Table 1). Overall, this MILC can be implemented without the need for altering the servo controller. It provides a convenient approach by modifying the reference trajectory and control compensation. However, injecting the control compensation into the internal controller may encounter some difficulties in certain closed commercial controllers. The study in this paper shows that MILC has appreciable application prospects for precision/ultra-precision systems with flexible feed drives to meet extreme motion accuracy requirements. Future research could further explore the real-time implementation of these advanced control schemes on standard industrial controllers.

**Author Contributions:** Conceptualization, Y.W. and T.H.; data curation, Y.W.; formal analysis, Y.W. and T.H.; funding acquisition, T.H.; investigation, Y.W.; methodology, Y.W. and T.H.; project administration, T.H.; resources, T.H.; software, Y.W.; supervision, T.H.; validation, Y.W. and T.H.; visualization, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, T.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by: National Science and Technology Council, Taiwan.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

**Theorem A1.** (Efficient norm-optimal ILC) In the context of the performance criterion outlined in Definition 1 and the linear time-invariant (LTI) model presented in Section 1 (from  $u_j[k]$  to  $y_j[k]$ ), consider the model in state-space form as follows:

$$H = \left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] \quad (\text{A1})$$

The optimal  $u_{j+1}[k]$  in (5) can be determined through the following steps [36]:

Step 1.

Solve the discrete Riccati equation backward:

$$P[k] = (A^T - C^T w_q D \varphi^{-1} B^T) P[k+1] \times (I - B(\varphi + B^T P[k+1] B)^{-1} B^T P[k+1]) \times (A - B \varphi^{-1} D^T w_q C) - C^T w_q D \varphi^{-1} D^T w_q C + C^T w_q C \quad (\text{A2})$$

with  $P[N] = 0$ , and

$$L[k] = (\varphi + B^T P[k+1] B)^{-1} (D^T w_q C + B^T P[k+1] A) \quad (\text{A3})$$

$$L_u[k] = (\varphi + B^T P[k+1] B)^{-1} w_r \quad (\text{A4})$$

$$L_e[k] = (\varphi + B^T P[k+1] B)^{-1} D^T w_q \quad (\text{A5})$$

$$L_g[k] = (\varphi + B^T P[k+1]B)^{-1} B^T \quad (\text{A6})$$

$$\varphi = D^T w_q D + w_r + w_s \quad (\text{A7})$$

Step 2.

Solve the vector difference backward:

$$g_{j+1}[k] = (A^T - K_g[k]B^T)g_{j+1}[k+1] + Cw_q e_j[k] + K_g[k]w_s u_j[k] \quad (\text{A8})$$

with  $g_{j+1}[N] = 0$ , and

$$K_g[k] = (A^T - C^T w_q D \varphi^{-1} B^T) P[k+1] (I + B \varphi^{-1} B^T P[k+1])^{-1} B \varphi^{-1} \quad (\text{A9})$$

Step 3.

Solve the optimal state as follows:

$$\Delta x^*[k+1] = (B \varphi^{-1} B^T P[k+1] + I)^{-1} (A \Delta x^*[k] + B B^T g[k+1] - B \varphi^{-1} w_r u_j[k]) \quad (\text{A10})$$

Step 4.

Obtain the optimal ILC updates:

$$\begin{aligned} u_{j+1}^*[k] &= \Delta u_{j+1}^*[k] + u_j[k] \\ &= -L[k] \Delta x^*[k] + L_g[k] g[k+1] + (I - L_f[k]) u_j[k] \end{aligned} \quad (\text{A11})$$

**Proof of Theorem A1.** Omitted here, see [36]. □

## Appendix B

**Lemma A1.** The state space equation from  $u_j$  to  $y_j^t$  in Figure 3 is given as follows:

$$H_D = \left[ \begin{array}{c|c} A & B \\ \hline C & \mathbf{0} \end{array} \right] \quad (\text{A12})$$

with

$$A = \begin{bmatrix} A_c & -B_c C_m & 0 \\ B_m C_c & A_m - B_m D_c C_m & 0 \\ 0 & B_z C_m & A_z \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ B_m \\ 0 \end{bmatrix}, \quad C = [0 \quad 0 \quad C_z]$$

**Proof of Lemma A1.** By systematically substituting the state updates and the output interconnection as outlined in (14), (30), (31), and (32) yields the representation. □

## References

1. Hsiao, T.; Huang, P.-H. Iterative learning control for trajectory tracking of robot manipulators. *Int. J. Autom. Smart Technol.* **2017**, *7*, 133–139. [CrossRef]
2. Wang, Y.; Hsiao, T. Fast-Update Iterative Learning Control for Performance Enhancement with Application to Motion Systems. *IEEE Access* **2022**, *10*, 79458–79468. [CrossRef]
3. Ertay, D.S.; Yuen, A.; Altintas, Y. Synchronized material deposition rate control with path velocity on fused filament fabrication machines. *Addit. Manuf.* **2018**, *19*, 205–213. [CrossRef]
4. Song, F.; Liu, Y.; Xu, J.; Yang, X.; Zhu, Q. Data-Driven Iterative Feedforward Tuning for a Wafer Stage: A High-Order Approach Based on Instrumental Variables. *IEEE Trans. Ind. Electron.* **2019**, *66*, 3106–3116. [CrossRef]
5. Gordon, D.J.; Erkorkmaz, K. Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter. *Precis Eng.* **2013**, *37*, 308–322. [CrossRef]
6. Hsiao, T.; Jhu, J.-H. Design of Precision Motion Controllers Based on Frequency Constraints and Time-Domain Optimization. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 933–944. [CrossRef]

7. Mizrachi, E.; Basovich, S.; Arogeti, S. Robust time-delayed  $H_{\infty}$  synthesis for active control of chatter in internal turning. *Int. J. Mach. Tools Manuf.* **2020**, *158*, 103612. [[CrossRef](#)]
8. Zhang, T.; Li, X.; Gai, H.; Zhu, Y. Integrated Controller Design and Application for CNC Machine Tool Servo Systems Based on Model Reference Adaptive Control and Adaptive Sliding Mode Control. *Sensors* **2023**, *23*, 9755. [[CrossRef](#)]
9. Zhang, H.-T.; Wu, Y.; He, D.; Zhao, H. Model predictive control to mitigate chatters in milling processes with input constraints. *Int. J. Mach. Tools Manuf.* **2015**, *91*, 54–61. [[CrossRef](#)]
10. Zhu, M.; Yang, Y.; Feng, X.; Du, Z.; Yang, J. Robust modeling method for thermal error of CNC machine tools based on random forest algorithm. *J. Intell. Manuf.* **2023**, *34*, 2013–2026. [[CrossRef](#)]
11. Li, Z.; Zhao, C.; Lu, Z. Thermal error modeling method for ball screw feed system of CNC machine tools in x-axis. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 5383–5392. [[CrossRef](#)]
12. Huang, W.-W.; Zhang, X.; Zhu, L.-M. Band-stop-filter-based repetitive control of fast tool servos for diamond turning of micro-structured functional surfaces. *Precis Eng.* **2023**, *83*, 124–133. [[CrossRef](#)]
13. Altintas, Y.; Khoshdarregi, M.R. Contour error control of CNC machine tools with vibration avoidance. *CIRP Ann.* **2012**, *61*, 335–338. [[CrossRef](#)]
14. Tomizuka, M. Zero phase error tracking algorithm for digital control. *J. Dyn. Syst. Meas. Control* **1987**, *109*, 65–68. [[CrossRef](#)]
15. Butterworth, J.A.; Pao, L.Y.; Abramovitch, D.Y. Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems. *Mechatronics* **2012**, *22*, 577–587. [[CrossRef](#)]
16. Ahn, H.-S.; Chen, Y.; Moore, K.L. Iterative Learning Control: Brief Survey and Categorization. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 1099–1121. [[CrossRef](#)]
17. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A survey of iterative learning control. *IEEE Control Syst. Mag.* **2006**, *26*, 96–114.
18. Blanken, L.; Isil, G.; Koekebakker, S.; Oomen, T. Flexible ILC: Towards a Convex Approach for Non-Causal Rational Basis Functions. *IFAC-Pap.* **2017**, *50*, 12107–12112. [[CrossRef](#)]
19. van Zundert, J.; Bolder, J.; Oomen, T. Iterative Learning Control for varying tasks: Achieving optimality for rational basis functions. In Proceedings of the 2015 American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015.
20. Bolder, J.; Kleinendorst, S.; Oomen, T. Data-driven multivariable ILC: Enhanced performance by eliminating L and Q filters. *Int. J. Robust Nonlinear Control* **2018**, *28*, 3728–3751. [[CrossRef](#)]
21. Yang, L.; Zhang, H. Data-Driven Feedforward Parameter Tuning Optimization Method Under Actuator Constraints. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 3429–3439. [[CrossRef](#)]
22. Li, X.; Huang, D. Adaptive Iterative Learning Control for Non-Square Nonlinear Systems with Various Nonrepetitive Uncertainties: A Unified Approach. *IEEE Trans. Autom. Control* **2023**, *69*, 1736–1743. [[CrossRef](#)]
23. Sang, S.; Zhang, R.; Lin, X. Model-Free Adaptive Iterative Learning Bipartite Containment Control for Multi-Agent Systems. *Sensors* **2022**, *22*, 7115. [[CrossRef](#)] [[PubMed](#)]
24. Tayebi, A.; Chien, C.-J. A Unified Adaptive Iterative Learning Control Framework for Uncertain Nonlinear Systems. *IEEE Trans. Autom. Control* **2007**, *52*, 1907–1913. [[CrossRef](#)]
25. Zhang, C.; Hu, Y.; Xiao, L.; Gong, X.; Chen, H. Data-Driven Robust Iterative Learning Predictive Control for MIMO Nonaffine Nonlinear Systems with Actuator Constraints. *IEEE Trans. Ind. Inform.* **2024**, 1–11. [[CrossRef](#)]
26. Chi, R.; Li, H.; Lin, N.; Huang, B. Data-Driven Indirect Iterative Learning Control. *IEEE Trans. Cybern.* **2024**, *54*, 1650–1660. [[CrossRef](#)]
27. Li, H.; Wang, S.; Shi, H.; Su, C.; Li, P. Two-Dimensional Iterative Learning Robust Asynchronous Switching Predictive Control for Multiphase Batch Processes with Time-Varying Delays. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 6488–6502. [[CrossRef](#)]
28. Chen, W.; Tomizuka, M. Dual-Stage Iterative Learning Control for MIMO Mismatched System with Application to Robots With Joint Elasticity. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 1350–1361.
29. Wang, C.; Zheng, M.; Wang, Z.; Peng, C.; Tomizuka, M. Robust Iterative Learning Control for Vibration Suppression of Industrial Robot Manipulators. *J. Dyn. Syst. Meas. Control* **2017**, *140*, 011003. [[CrossRef](#)]
30. Wallen, J.; Norrlof, M.; Gunnarsson, S. A framework for analysis of observer-based ILC. *Asian J. Control* **2011**, *13*, 3–14. [[CrossRef](#)]
31. Bolder, J.; Oomen, T. Inferential Iterative Learning Control: A 2D-system approach. *Automatica* **2016**, *71*, 247–253. [[CrossRef](#)]
32. Dumanli, A.; Sencer, B. Data-Driven Iterative Trajectory Shaping for Precision Control of Flexible Feed Drives. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 2735–2746.
33. Barton, K.L.; Alleyne, A.G. A Norm Optimal Approach to Time-Varying ILC With Application to a Multi-Axis Robotic Testbed. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 166–180. [[CrossRef](#)]
34. van Zundert, J.; Oomen, T. On inversion-based approaches for feedforward and ILC. *Mechatronics* **2018**, *50*, 282–291. [[CrossRef](#)]
35. Ge, X.; Stein, J.L.; Ersal, T. A frequency domain approach for designing filters for Norm-Optimal Iterative Learning Control and its fundamental tradeoff between robustness, convergence speed and steady state error. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; IEEE: New York, NY, USA, 2016; pp. 384–391.
36. van Zundert, J.; Bolder, J.; Koekebakker, S.; Oomen, T. Resource-efficient ILC for LTI/LTV systems through LQ, tracking and stable inversion: Enabling large feedforward tasks on a position-dependent printer. *Mechatronics* **2016**, *38*, 76–90. [[CrossRef](#)]

37. Zhang, J.-X.; Yeh, S.-S. Modelization and Identification of Feed Drive Axis in CNC Machine Tools Using Two-Mass Model and Particle Swarm Optimization. In Proceedings of the 2023 IEEE International Conference on Industrial Technology (ICIT), Orlando, FL, USA, 4–6 April 2023.
38. Erkorkmaz, K.; Hosseinkhani, Y. Control of ball screw drives based on disturbance response optimization. *CIRP Ann.* **2013**, *62*, 387–390. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.