engineering proceedings

MDPI

Proceeding Paper

Data-Driven Long Short-Term Memory Framework for Servo System Modeling and Optimization †

Yong-Zhong Li [1], You-Cheng Chen [1], Xiang-Kai Wang [1] and Ming-Tsung Lin [1,2,*]

[1] Department of Mechanical Design Engineering, National Formosa University, Yunlin 632301, Taiwan; 50733143@gm.nfu.edu.tw (Y.-Z.L.); 50733140@gm.nfu.edu.tw (Y.-C.C.); 41023113@gm.nfu.edu.tw (X.-K.W.)
[2] Smart Machine and Intelligent Manufacturing Research Center, National Formosa University, Yunlin 632301, Taiwan
* Correspondence: mtlin@nfu.edu.tw
† Presented at the 7th Eurasia Conference on IoT, Communication and Engineering 2025 (ECICE 2025), Yunlin, Taiwan, 14–16 November 2025.

Abstract

A novel data-driven modeling framework is developed for servo control using Long Short-Term Memory (LSTM) networks. The framework employs an LSTM model to directly map interpolation commands and feedback signals, such as velocity, acceleration, and jerk, to tracking errors. By adopting end-to-end architecture, the method bypasses complex sequential procedures, including system identification and friction modeling, significantly reducing development time and modeling complexity. Experimental results demonstrate that the LSTM model accurately predicts servo tracking behavior and enables rapid performance evaluation and parameter optimization without performing time-consuming trajectory testing. The proposed framework offers a practical and efficient alternative to traditional model-based techniques in precision motion control.

Keywords: data-driven modeling; LSTM; servo system; tracking error; parameter optimization

1. Introduction

With the rapid development of smart manufacturing and Industry 4.0, modern precision machining demands extremely high levels of accuracy and efficiency in computer numerical control (CNC) machine tools. In high-speed, high-precision contouring and machining, the dynamic response of the servo control system is critical in determining the final geometric accuracy and surface quality of the machined workpiece. However, nonlinear effects such as servo lag, friction, backlash, and structural vibrations cause deviations between the commanded and actual tool paths, leading to servo tracking and contour errors.

To avoid these errors, conventional control strategies depend on precise physical modeling and system identification. For example, feedforward control is widely used to compensate for servo lag, but this requires an accurate inverse dynamic model. However, deriving transfer functions of the servo system is not only tedious and time-consuming, but also numerous uncertainties exist in a practical machining environment. Mechanical components aging, variations in lubrication conditions, and thermal deformation caused by prolonged machining lead to time-varying system parameters, making fixed-parameter physical models inadequate for maintaining long-term prediction accuracy. Therefore, developing an error prediction and compensation method capable of handling complex

Check for updates

Academic Editors: Teen-Hang Meen, Chi-Ting Ho and Cheng-Fu Yang

Published: 3 April 2026

Copyright: © 2026 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.

nonlinear behaviors without extensive system identification and control parameter optimization has become an important research challenge.

Recently, advances in AI have promoted the adoption of data-driven methods in servo control. Compared with traditional methods that rely on physical modeling, data-driven approaches can learn the input-output relationship of servo systems directly from large amounts of machining data, without requiring explicit knowledge of the underlying system dynamics. For parameter optimization, Chen et al. [1] and He et al. [2] employed Deep Reinforcement Learning, specifically Deep Deterministic Policy Gradient and Soft Actor-Critic algorithms, to achieve online automatic tuning of proportional–integral–derivative parameters in servo and hydraulic systems, respectively. Their results demonstrate the effectiveness of AI-based approaches for nonlinear and uncertain systems. Li et al. [3] applied reinforcement learning to trajectory smoothing, improving machining efficiency and illustrating the potential of intelligent algorithms to enhance motion control performance. In time-series prediction and error compensation, Recurrent Neural Networks (RNNs) and their variants, such as LSTM networks and Gated Recurrent Units (GRUs), have been widely adopted for dynamic system modeling due to their ability to effectively capture long-term temporal dependencies.

Jiang et al. [4] proposed a nonlinear autoregressive LSTM model for motion-state classification that uses reference and actual signals to predict the responses of the feed drive system, improving prediction accuracy. Additionally, Hu et al. [5] developed a GRU-based tracking error prediction model and integrated it with a learning adaptive robust control scheme to achieve reference trajectory pre-compensation. Li et al. [6] employed LSTM networks to predict and compensate for contour errors in CNC systems, where linear features were selected using a simplified single-axis model and nonlinear features were extracted from circular tests as network inputs.

Although existing neural network-based approaches have demonstrated promising performance, most methods still rely on specific training trajectories, such as circles or squares, to construct the surrogate models. When applied to complex machining paths encountered in practical scenarios, including free-form surface machining, the generalization capability of these models remains limited. Moreover, the challenge of constructing a truly end-to-end prediction framework has yet to be fully addressed. The framework eliminates the need for physical parameter identification by directly utilizing interpolation commands and feedback signals.

To address these limitations, we developed a data-driven LSTM-based framework for servo system modeling and optimization. By leveraging the strong temporal modeling capability of LSTM networks, the developed system learns a nonlinear mapping from interpolation commands to servo tracking errors, thereby replacing conventional system identification and model-based tuning procedures. As an end-to-end LSTM-based virtual metrology model, the developed system predicts tracking errors directly from interpolation commands, eliminating the need for physical parameter identification. A hybrid training strategy combining the KAKINO trajectories and random Non-Uniform Rational B-Spline (NURBS) paths is designed to enhance model generalization under diverse dynamic conditions. The system was validated using machining of the Fuji Automatic Numerical Control (FANUC) test mold, demonstrating accurate tracking error prediction for complex and previously unseen trajectories and confirming its effectiveness for practical CNC applications.

2. Data-Driven Deep Learning Framework

A data-driven deep learning framework is designed to address the challenges of physical parameter identification in traditional modeling of the CNC servo system for

Victor Taichung Machinery Works Co., Ltd., Taiwan. The methodology includes the data acquisition strategy, feature selection and preprocessing, and architecture design and training procedure for the LSTM-based prediction model.

2.1. System Overview

The proposed servo system error prediction framework in this study is illustrated in Figure 1.

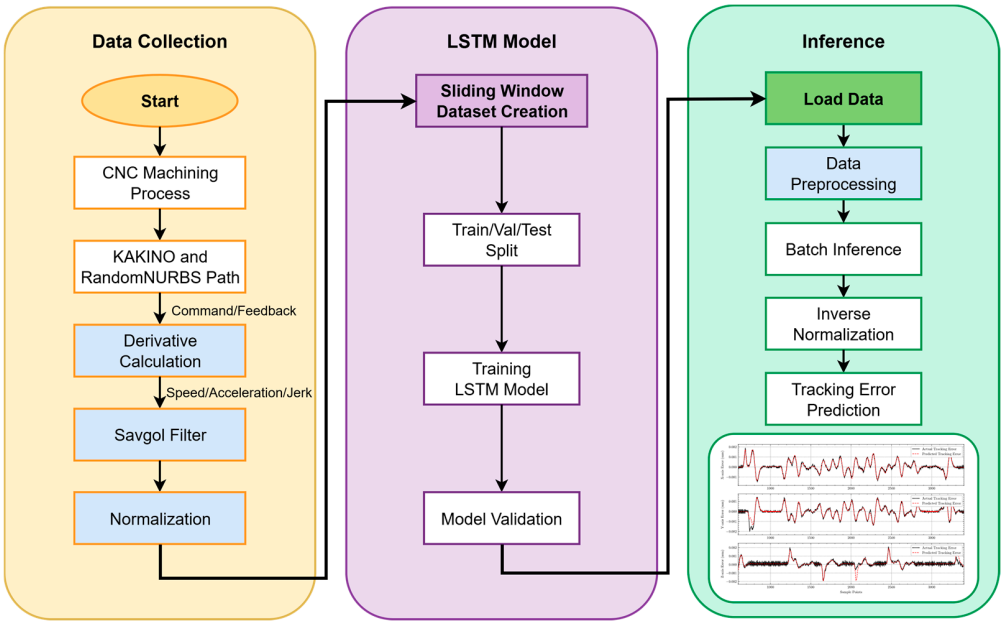


Figure 1. Servo system error prediction framework.

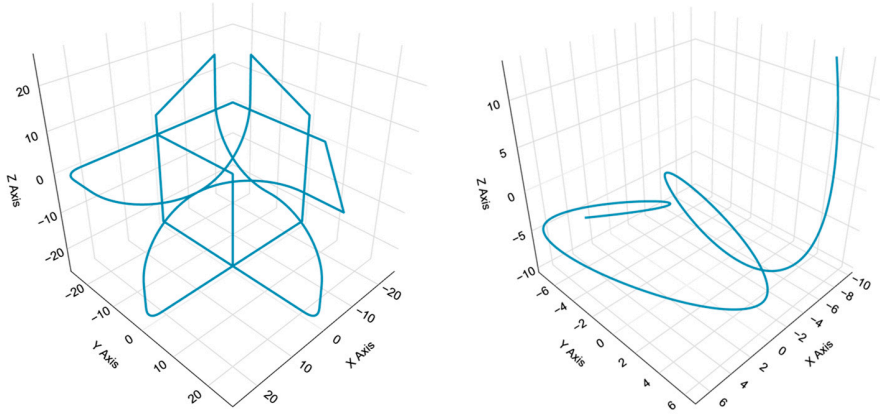
The framework consists of three main stages: (1) data collection and feature generation, (2) data preprocessing, and (3) model training and inference. First, real-time machining data, including position commands and feedback signals, is acquired via the FANUC FOCAS (Fanuc Open CNC Application Programming Interface Specifications) communication interface of the CNC controller. To capture servo dynamics under diverse operating conditions, KAKINO trajectories and randomly generated NURBS paths are employed as training data. Second, the collected raw data undergo feature engineering, in which velocity, acceleration, and jerk are computed via numerical differentiation, and measurement noise is suppressed using a Savitzky–Golay (Savgol) filter. Subsequently, the data are normalized and processed using a sliding window to form time-series signals suitable for model input. Finally, an LSTM-based prediction model is developed to estimate the current servo tracking error from historical interpolation command features. For a three-axis machine tool, independent LSTM models are constructed for the X-, Y-, and Z-axes to account for axis-specific friction and inertia characteristics.

2.2. Data Acquisition Strategy

The experiments are conducted on a 3-axis machining center equipped with a FANUC controller. Data acquisition is performed via the FANUC FOCAS library, which enables an external PC to access internal CNC registers directly at a high sampling rate. The collected key variables include the command position P_{cmd} generated by the interpolator and the actual position P_{fbk} measured by the servo encoder. To ensure sufficient generalization capability of the deep learning model, careful design of the training dataset is essential.

We adopted two complementary trajectory strategies. The KAKINO trajectory (Figure 2a) is a standard benchmark for evaluating dynamic errors in CNC machine tools.

It comprises straight segments, circular arcs, and sharp corners, effectively exciting servo dynamic behaviors such as reversal errors and high-curvature tracking deviations. In contrast, traditional standard test paths (circles or squares) exhibit regular geometries and limited frequency content, which may lead to model overfitting. To better represent more practical machining conditions involving complex free-form surfaces, randomly generated NURBS trajectories are introduced and illustrated in Figure 2b. By randomizing the control point distributions and weighting factors, the path produces diverse curvature variations and acceleration/deceleration profiles in interpolation, enabling the model to learn servo dynamic responses across a broad frequency range.



(**a**) (**b**)

Figure 2. Standard test path: (**a**) KAKINO; (**b**) random NURBS.

2.3. Data Preprocessing

The servo tracking error $e(t)$ is defined as the difference between the command P_{cmd} and the feedback position P_{fbk} (Equation (1)).

$$e(t) = P_{cmd}(t) - P_{fdb}(t) \tag{1}$$

Although the position command represents the reference input, the dynamic response of the servo system is influenced by its higher-order kinematic profiles, including the velocity, acceleration, and jerk. Since the higher-order kinematic profiles cannot be acquired via FANUC FOCAS, these kinematic features are derived from the position commands using the finite difference method. Assuming a constant sampling interval Δt, the velocity $V(t)$, acceleration $A(t)$, and jerk $J(t)$ at time t are calculated as follows.

$$\begin{aligned} V(t) &= \frac{P_{cmd}(t) - P_{cmd}(t-1)}{\Delta t} \\ A(t) &= \frac{V(t) - V(t-1)}{\Delta t} \\ J(t) &= \frac{A(t) - A(t-1)}{\Delta t} \end{aligned} \tag{2}$$

Because numerical differentiation inherently amplifies quantization noise in raw signals, directly using differentiated data may degrade training stability. To mitigate this issue, a Savitzky–Golay filter is applied to smooth the signal. Unlike moving average filtering, the Savitzky–Golay filter performs local polynomial least-squares fitting, effectively suppressing high-frequency noise while preserving signal peaks and waveform characteristics, and minimizing signal distortion. Moreover, input features such as position, velocity, and acceleration possess different physical units and numerical scales. Directly feeding these features into a neural network may hinder gradient convergence or make the model

overly sensitive to features with large magnitudes. To eliminate scale effects and accelerate training convergence, Z-score standardization is adopted. This normalization transforms each feature to have zero mean and unit variance. For a given feature x, the standardized form is expressed as follows.

$$x_{std} = \frac{x - \mu_x}{\sigma_x} \tag{3}$$

where μ_x and σ_x denote the mean and standard deviation of the feature computed from the training set, respectively. After model inference, the inverse normalization is applied to the standardized output y_{std} to recover the predicted tracking error in physical units. It yields the following.

$$y = y_{std} \cdot \sigma_y + \mu_y \tag{4}$$

where μ_y and σ_y represent the mean and standard deviation of the target variable (tracking error) computed from the training set, respectively. To accommodate time-series characteristics, a sliding window technique is used to transform continuous signals into supervised learning samples. Given a window length W, the input vector X_t at time step t is constructed using feature values from the previous W time steps. It becomes the following.

$$X_t = [F_{t-W+1} \quad F_{t-W+2} \quad \dots \quad F_t] \tag{5}$$

where F_t denotes the feature vector at time t, which includes the velocity, acceleration, jerk, and other relevant kinematic features. The objective of the model is to predict the servo tracking error $e(t)$ at time step t.

3. Architecture of LSTM Model

3.1. Principles of LSTM Units

LSTM is a specialized form of RNN designed to address the vanishing gradient problem encountered by conventional RNNs when modeling long-term dependencies. Figure 3 illustrates the architecture of an LSTM cell. By introducing gating mechanisms, LSTM cells regulate information flow and retention within the network. A standard LSTM unit consists of a forget gate, an input gate, and an output gate. The mathematical operations of an LSTM unit at time step t are given as follows.

- Forget gate (f_t): Decides which information to discard from the cell state.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6}$$

- Input gate (i_t) and candidate cell state ($\widetilde{C}_t$): Decide which new information to update the cell state.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{7}$$

- Cell state update (C_t):

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{8}$$

- Output gate (o_t) and hidden state (h_t):

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{9}$$

https://doi.org/10.3390/engproc2026134027

where $\sigma(\cdot)$ denotes the sigmoid activation function, W and b represent the corresponding weight matrices and bias terms, respectively. * represents element-wise multiplication.

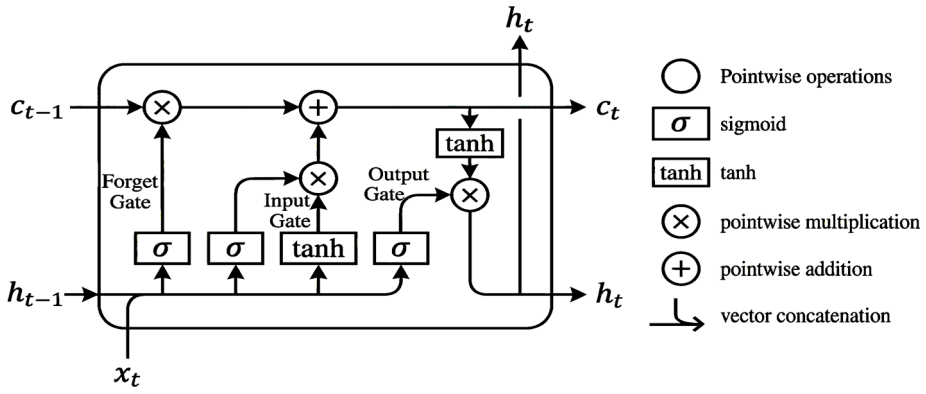


Figure 3. LSTM cell structure.

3.2. Network Structure Design

The architecture of the developed error prediction model inn this study is shown in Figure 4. The input layer receives a time-series feature matrix with dimensions [Window_Size, Feature_Dim]. Stacked LSTM layers are employed to extract deep temporal features from the input sequence, where a multi-layer configuration (e.g., two to three LSTM layers) is adopted, and the full sequence output of each layer is propagated to the subsequent layer. The output of the final LSTM layer is then fed into a fully connected (dense) layer, which maps the extracted high-dimensional temporal features to a single scalar output. The output layer provides the predicted servo tracking error.

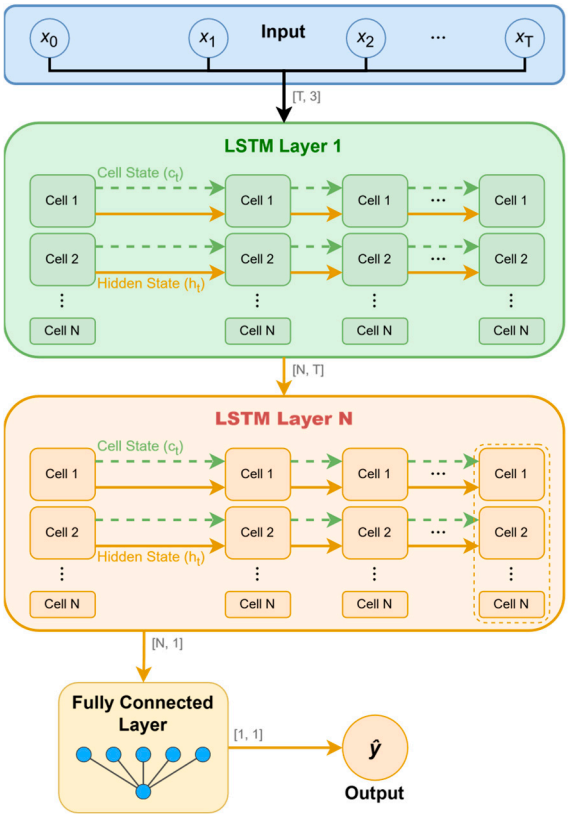


Figure 4. The architecture of the LSTM error prediction model.

For a three-axis machine tool, independent LSTM models are constructed for the *X*-, *Y*-, and *Z*-axes. This decoupled modeling strategy enables each network to learn axis-specific physical characteristics, such as the effects of moment of inertia on the *X*-axis and gravitational influences on the *Z*-axis, improving overall prediction accuracy.

4. Experiment Verification

To validate the effectiveness of the proposed data-driven LSTM framework for servo system modeling and tracking error prediction, we designed the hardware configuration of the test platform, the dataset composition and partitioning strategy, and the training hyperparameter settings of the neural network model.

4.1. Experimental Setup

The experimental platform is a no-load test machine designed to eliminate external disturbances, such as cutting forces, and to focus on the intrinsic dynamic behavior of the servo control system. CNC programs for the training trajectories and automated acquisition of servo signals are performed via the FAUNC FOCAS library. The experimental architecture is shown in Figure 5. The machine tool is equipped with industrial-level controllers and servo drive systems. A FANUC 0i-MF Plus CNC controller is adopted in this study, providing high computational performance, stable communication interfaces, and enabling high-frequency access to position commands and servo state data via the FOCAS interface. Each axis is driven by a FANUC βiS2/4000-B AC servo motor equipped with high-resolution encoders, which provide precise position feedback for high-accuracy motion control experiments.

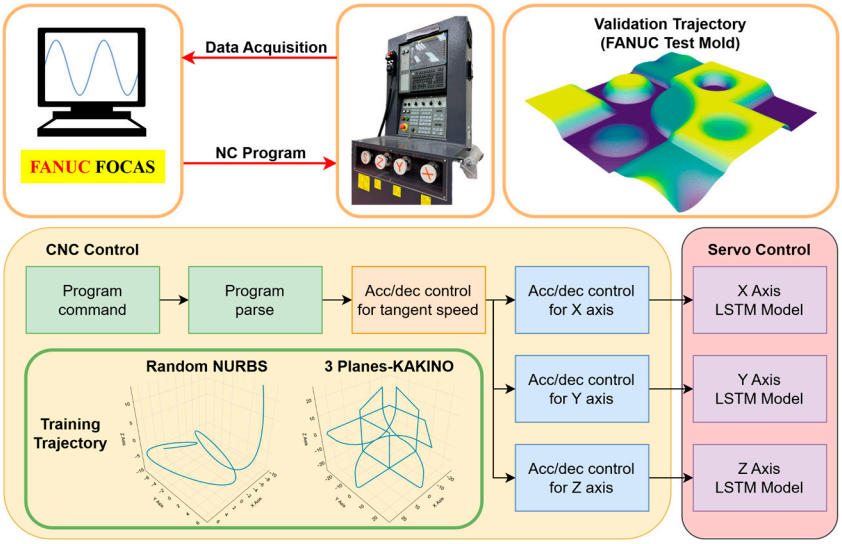


Figure 5. Experimental platform.

4.2. Dataset Specifications

To ensure that the model effectively learns the dynamic behavior of the servo system across diverse geometric features and achieves strong generalization capability, a hybrid dataset comprising standard KAKINO paths and random NURBS paths is generated in this study. A total of 729 datasets for the KAKINO trajectory were collected using combinations of various FAUNC interpolation parameters and feed rates. The trajectories contain rich geometric features, including circles of different radii, sharp corners, and straight segments. They are used to capture the error characteristics of the servo system under specific geometric transformations. In addition, 1000 datasets of random NURBS trajectories were collected. By randomly generating control points and weights, the paths

with diverse curvature and velocity profiles were produced. This method enables the model to learn dynamic behaviors of servo systems rather than merely memorizing specific geometric patterns. To evaluate model performance, all collected samples were processed using sliding windows, randomly shuffled, and then divided into training, validation, and test subsets in a ratio of 80:10:10.

4.3. Hyperparameter Tuning

The training process of the LSTM model was conducted on a deep learning workstation with graphics processing unit acceleration. After performing the trial-and-error method, the optimal hyperparameter configuration was determined (Table 1). Mean squared error (MSE) is selected as the loss function. It effectively penalizes larger prediction errors, making the model minimize the deviation of the entire trajectory.

Table 1. Hyperparameter configuration.

Hyperparameter	Value
Input size	3
Window size	128
LSTM layers	2
Hidden size	64
Optimizer	Adam
Loss function	MSE
Learning rate	0.001
Batch size	1024
Epochs	10

4.4. Analysis of Model Convergence

The LSTM model demonstrates a robust convergence trend during training. Taking the X-axis model as an example, Figure 6 depicts the evolution of training and validation losses. The training loss exhibits a steep descent in the initial epoch, indicating the model can rapidly extract the dominant features of the servo dynamics. As training proceeds, the loss stabilizes with minor oscillations, exhibiting asymptotic behavior below 0.1. Moreover, the validation loss, representing performance on unseen data, follows a consistent downward trajectory from 0.095 to approximately 0.082. As shown in Figure 6, the validation curve follows the training loss without significant divergence. This absence of a generalization gap corroborates that the model avoids overfitting, validating the efficacy of the proposed hybrid dataset strategy in enhancing model robustness. The experiment results for the three motion axes are given in Table 2. All motion axes can achieve good results for the training, validation, and test loss functions. This demonstrates the prediction accuracy of the LSTM model.

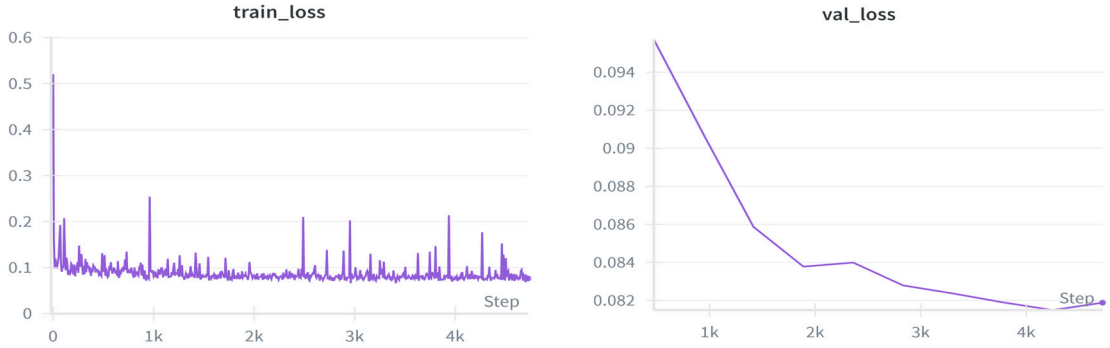


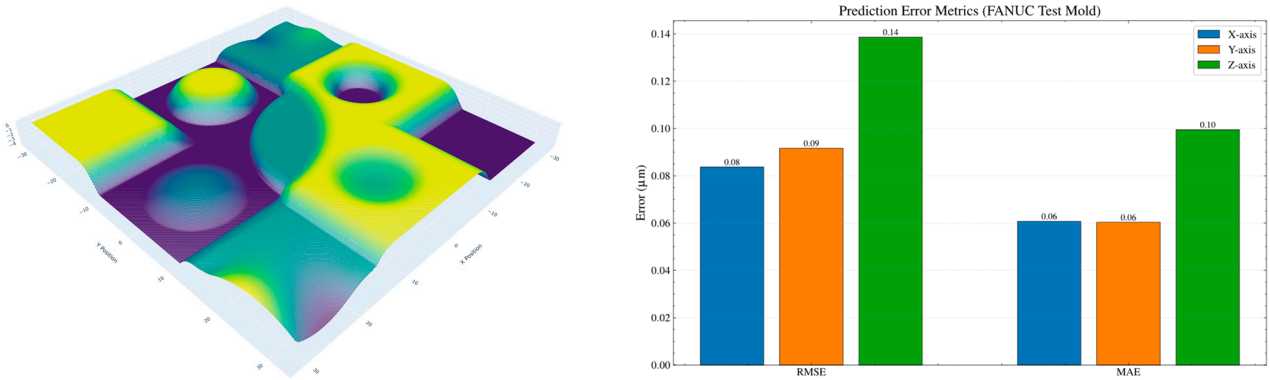
Figure 6. Training loss and validation loss.

Table 2. Training results of the LSTM model.

Axis	Training Loss	Validation Loss	Test Loss
X	0.07642	0.08187	0.08217
Y	0.05904	0.06143	0.06128
Z	0.26132	0.26529	0.26538

4.5. Prediction Accuracy and Generalization Evaluation

To assess generalization capability under realistic industrial scenarios, the FANUC test mold trajectory shown in Figure 7a was employed as an independent benchmark. Unlike the KAKINO or NURBS paths used for training, this standard trajectory poses significant challenges due to its complex free-form surfaces, sharp corners, and frequent quadrant transitions. Figure 7b summarizes the prediction error metrics for the *X*-, *Y*-, and *Z*-axes. The results show that both the RMSE and MAE are maintained at the micron level across all axes. This high-accuracy performance implies that the LSTM model achieves high-fidelity reconstruction of the actual servo dynamic response. These findings confirm that, despite being trained on fundamental geometric features, the data-driven framework possesses the intrinsic capability to generalize to complex, unseen machining paths, proving its reliability for practical applications.



(**a**) (**b**)

Figure 7. FANUC test mold: (**a**) 3D profile; (**b**) Prediction error metrics.

5. Conclusions

We developed an end-to-end data-driven framework for CNC servo error prediction and compensation, effectively addressing the inherent complexities of high-fidelity physical modeling. By integrating LSTM networks with a hybrid trajectory training strategy, the proposed model achieves robust and high-precision prediction of servo tracking errors. While the current results are promising, future research is required to (1) incorporate the LSTM model as a virtual surrogate environment within a reinforcement learning framework to automate the optimization of interpolator parameters; (2) develop adaptive learning mechanisms to maintain predictive accuracy amidst parameter drifts, such as dynamic load variations; and (3) implement the prediction model into edge-based CNC controllers to realize real-time, prediction-based error compensation.

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

Funding: This research was funded by the National Science and Technology Council (NSTC) of Taiwan, ROC, under grant numbers NSTC 114-2221-E-150-051- and 114-2218-E-002-007-.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Chen, P.; He, Z.; Chen, C.; Xu, J. Control strategy of speed servo systems based on deep reinforcement learning. *Algorithms* **2018**, *11*, 65. [CrossRef]
2. He, J.; Su, S.; Wang, H.; Chen, F.; Yin, B. Online PID tuning strategy for hydraulic servo control systems via SAC-based deep reinforcement learning. *Machines* **2023**, *11*, 593. [CrossRef]
3. Li, B.; Zhang, H.; Ye, P.; Wang, J. Trajectory smoothing method using reinforcement learning for computer numerical control machine tools. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101847. [CrossRef]
4. Jiang, Y.; Chen, J.; Zhou, H.; Yang, J.; Xu, G. Nonlinear time-series modeling of feed drive system based on motion states classification. *J. Intell. Manuf.* **2020**, *31*, 1935–1948. [CrossRef]
5. Hu, C.; Ou, T.; Zhu, Y.; Zhu, L. GRU-type LARC strategy for precision motion control with accurate tracking error prediction. *IEEE Trans. Ind. Electron.* **2021**, *68*, 812–820. [CrossRef]
6. Li, J.; Qi, C.; Li, Y.; Wu, Z. Prediction and compensation of contour error of CNC systems based on LSTM neural network. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 572–581. [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.