

Article

# LSTM-Inversion-Based Feedforward–Feedback Nanopositioning Control

Ruocheng Yin and Juan Ren \*

Department of Mechanical Engineering, Iowa State University, Ames, IA 50014, USA; innocen3@iastate.edu

\* Correspondence: juanren@iastate.edu

**Abstract:** This work proposes a two-degree of freedom (2DOF) controller for motion tracking of nanopositioning devices, such as piezoelectric actuators (PEAs), with a broad bandwidth and high precision. The proposed 2DOF controller consists of an inversion feedforward controller and a real-time feedback controller. The feedforward controller, a sequence-to-sequence LSTM-based inversion model (invLSTMs2s), is used to compensate for the nonlinearity of the PEA, especially at high frequencies, and is collaboratively integrated with a linear MPC feedback controller, which ensures the PEA position tracking performance at low frequencies. Therefore, the proposed 2DOF controller, namely, invLSTMs2s+MPC, is able to achieve high precision over a broad bandwidth. To validate the proposed controller, the uncertainty of invLSTMs2s is checked such that the integration of an inversion model-based feedforward controller has a positive impact on the trajectory tracking performance compared to feedback control only. Experimental validation on a commercial PEA and comparison with existing approaches demonstrate that high tracking accuracies can be achieved by invLSTMs2s+MPC for various reference trajectories. Moreover, invLSTMs2s+MPC is further demonstrated on a multi-dimensional PEA platform for simultaneous multi-direction positioning control.

**Keywords:** LSTM; sequence-to-sequence; PEA; system identification; inversion control; feedforward–feedback control; 2DOF controller



**Citation:** Yin, R.; Ren, J. LSTM-Inversion-Based Feedforward–Feedback Nanopositioning Control. *Machines* **2024**, *12*, 747. <https://doi.org/10.3390/machines12110747>

Academic Editors: Wenjun (Chris) Zhang and Aydin Azizi

Received: 30 September 2024

Revised: 19 October 2024

Accepted: 21 October 2024

Published: 22 October 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

Nanopositioning devices, such as piezoelectric actuators (PEAs), are broadly used in many high-precision industries, products, and systems due to their advantages of fast response and high motion resolution. As one of the most popular nanopositioning devices, PEAs have been implemented in various applications, such as atomic force microscopes (AFMs) [1,2], micro forming [3], and adaptive optics [4]. However, the operation of such a system at high frequency and/or large motion range still remains challenging because of the PEA's nonlinear dynamics (e.g., hysteresis and creep), which is more pronounced when operations are performed under these conditions. As a result, the system operation bandwidth and motion range are rather limited for commercial applications, especially for real-time motion control applications.

To address these issues, efforts in PEA position tracking control have been reported. Feedforward controllers have been proposed to increase the control bandwidth by compensating for the PEA nonlinear dynamics. For example, model-based feedforward control designs have been implemented to improve control performance by compensating for PEA hysteresis, through inverse dynamics approximation [5,6]. To capture the frequency-dependent PEA hysteresis dynamics, rate-independent hysteresis models, such as the improved Preisach model [7], the modified Prandtl–Ishlinskii model [8], and the Bouc–Wen hysteresis model [9], have been implemented in the feedforward controller designs. The feedforward controller can then be combined with a real-time feedback controller to form a two-degree-of-freedom (2DOF) controller to increase the overall tracking

accuracy and robustness [8,10]. In these 2DOF controllers, the feedforward part accounts for the nonlinear and high-frequency PEA dynamics, and the feedback controller takes care of the remaining errors to improve the overall control precision. Theoretically, the 2DOF controller should be able to deliver satisfactory trajectory tracking and positioning performance if both feedforward and feedback parts are well designed. However, due to the complicated PEA nonlinear dynamics, model parameter identification is quite challenging, and the trajectory tracking improvements obtained are rather limited [11], especially when system uncertainties exist.

With the recent development of deep learning algorithms, neural networks (NNs) have been widely proposed for system identification and controller design. With adequate training data, NNs can capture the entire PEA dynamics presented in the data quite well, not just limited to hysteresis. Previous studies [12] show a vanilla recurrent neural network (RNN) performs well at handling the time series input and is increasingly popular in controller design. However, a vanilla RNN still has its limits. For example, an RNN can potentially suffer from gradient vanishing or exploding issues [13–15] and is prone to overfitting. Furthermore, compared to the long short-term memory (LSTM), it cannot consider and understand the system's long-term dependencies subject to sequential input, so the accuracy of a vanilla RNN in system dynamic identification can be limited [16,17]. The advantages of LSTM come from the memory cells in network neurons. The memory cells inherit the time-relevant information from long-term and short-term memory together and help LSTM make future predictions based on important historical data [16,18]. This feature enables LSTM to perform better in handling the complexity of nonlinear systems. But LSTM still can possibly encounter overfitting issues [19–21]. The reason can be originated from the conventional training mechanism. To train an accurate LSTM, it conventionally uses a very long single time series that concatenates many training data samples to cover a broad frequency range [16,18]. As a result, the conventional LSTM training progress could be extremely time-consuming. To address this issue, sequence-to-sequence LSTM learning (namely, LSTMs2s) has been recently implemented for PEA system identification [22] with high accuracy. This provides potential for controller design in high accuracy PEA position tracking.

Leveraging LSTMs2s in system identification, this work proposes a 2DOF controller (invLSTMs2s+MPC) consisting of a feedforward controller based on an LSTMs2s model of the system inverse dynamics (namely, invLSTMs2s) and a model predictive controller (MPC) as a feedback controller. The MPC was chosen primarily owing to its ability to handle constraints and capability for controlling multivariable plants, among other advantages. Considering the frequency and range dependent PEA nonlinear dynamics, the invLSTMs2s training set was generated using sinusoidal and triangle references with various frequency and amplitude components. To reduce the training data size and training time cost, invLSTMs2s was primarily designed to capture the PEA dynamics at mid- and high-frequency ranges (e.g., a range of tens to hundreds of hertz). In this work, the invLSTMs2s model was trained using a parallel training method, in which groups of training datasets were divided into mini-batches. After training, the modeling accuracy of invLSTMs2s was validated by comparing the model hysteresis prediction with the actual PEA response. With the accuracy of invLSTMs2s in nonlinear dynamics identification confirmed, an MPC based on a linear model was integrated as a feedback controller to take care of the low-frequency control and improve the overall tracking accuracy. Weighting factors were incorporated in the feedforward and feedback control paths to generalize the proposed 2DOF controller for tracking various references in different frequency regions. For validation and demonstration, the proposed 2DOF controller was implemented to control the displacement of a commercial PEA stage and compared with other approaches. The experiment results clearly demonstrate the superiority of invLSTMs2s+MPC for PEA trajectory tracking.

## 2. Feedforward–Feedback invLSTMs2s+MPC Controller Design

### 2.1. Feedforward Controller

#### 2.1.1. System Inversion Model

In general, a dynamic system can be modeled in state-space as Equation (1), where  $U = [u_k, u_{k+1}, \dots]$  and  $Y = [y_k, y_{k+1}, \dots]$  are the input and output, respectively. The corresponding inversion model can be modeled as Equation (2). The inversion model's (Equation (2)) output becomes  $U$  when the  $Y$  is fed as the inversion model's input. If an accurate inversion model can be obtained for a time-invariant system and disturbances/uncertainties are not considered, precise control of the system can be achieved by connecting the inversion model and the system in series where the output of the former is the input of the latter and directly feeding the desired system reference as the input of the combination.

$$\begin{aligned}x_{k+1} &= f(x_k, u_k) \\ y_k &= h(x_k, u_k)\end{aligned}\quad (1)$$

$$\begin{aligned}z_{k+1} &= f_{inv}(z_k, y_k) \\ u_k &= h_{inv}(z_k, y_k).\end{aligned}\quad (2)$$

For a linear system, its inversion model can be directly determined through linear system theories. However, for a nonlinear system, such as PEA positioning systems in this work, obtaining an accurate inversion model is not straightforward. One way to solve this problem is to train and regress an inversion machine learning model using the data pair  $\{Y, U\}$  from the original system.

#### 2.1.2. LSTMs2s Inversion Model (invLSTMs2s)

The goal of designing an inversion model is to map the output time series of the original system back to the input time series of the original system. Previous studies have shown that LSTM networks are more potent for this task compared to other forms of RNNs [17,22].

In this work, an advanced development of LSTM, sequence-to-sequence LSTM (LSTMs2s), was selected to generate the nonlinear inversion model of PEA for its efficiency and accuracy [22–24]. As shown in Figure 1, the LSTMs2s contains two LSTM layers expanded timewise: an LSTM encoder and an LSTM decoder. Both LSTM layers have similar structures.

In LSTMs2s (see Figure 1), the encoder LSTM layer reads the entire input time series  $u_{(r)}$  and its final hidden states  $x_{(r)}$  are saved as context vector  $C_v$ . After the encoder processes the entire input time series, the decoder is coupled with a fully connected network to predict the output time series  $y_{(r)}$  based on  $C_v$ . The entire LSTMs2s model can be formulated as follows:

Encoder:

$$x_{e,(k)} = f(x_{e,(k-1)}, u_{(r),k}). \quad (3)$$

Decoder:

$$\begin{aligned}y_{(r),k-1} &= g(x_{d,(k-1)}) \\ x_{d,(k)} &= f(x_{d,(k-1)}, y_{(r),k-1}, C_v) \\ y_{(r),k} &= g(x_{d,(k)}),\end{aligned}\quad (4)$$

where  $x_{d,(k)}$  and  $x_{e,(k)}$  are the hidden states in the encoder and decoder at the sampling instant  $k$ , respectively.  $f(\cdot)$  denotes the LSTM unit function.  $g(\cdot)$  represents the output layer that is implemented by a fully connected network. In this encoder–decoder model, it is clear that the decoder generates output data points  $y_{(r),k}$  at each sampling instant  $k$  and uses it as an input parameter to the decoder for the next output prediction. Note that the context vector  $C_v$  serves as an input parameter to the decoder for each step.

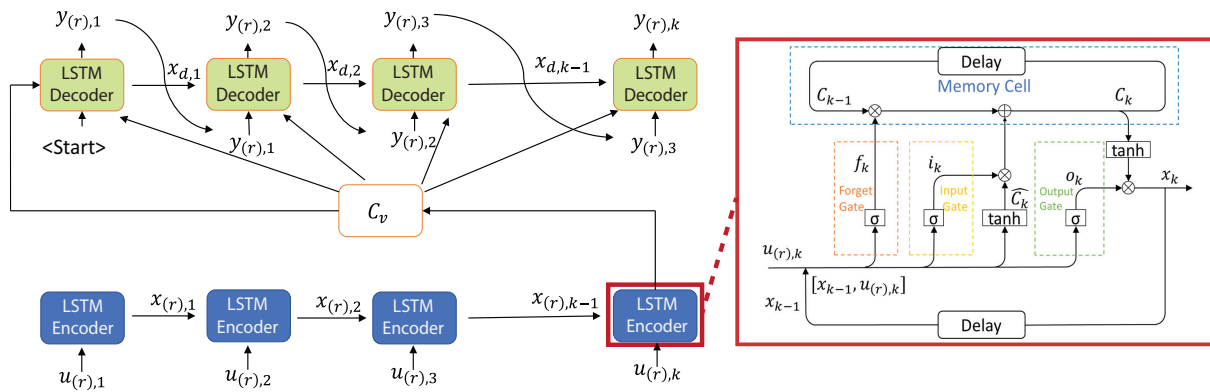


Figure 1. LSTM2s structure.

### 2.1.3. invLSTM2s Training

invLSTM2s aims to map the relation from the PEA output to the PEA input, hence the inverse PEA dynamics. Suppose the measured PEA system output is  $Y_{(ts)}$  subject to the drive input  $U_{(ts)}$ , and  $Y_{(rts)}$  is the inversion model output subject to  $Y_{(ts)}$ ; a perfect inversion model should generate  $Y_{(rts)}$  exactly the same as  $U_{(ts)}$ . In other words, such a perfect inversion model satisfies that the error  $\|U_{(ts)} - Y_{(rts)}\| < \epsilon$  for any  $\epsilon > 0$ . Therefore, the invLSTM2s can be trained using the training set constructed with  $(Y_{ts}, U_{ts})$  by solving the following optimization problem:

$$\begin{aligned}
 \min_{W_f, b_f, W_i, b_i, W_o, b_o, W_C, b_C, W_r, b_r} \quad & J(r) = \|U_{(ts)} - Y_{(rts)}\| \\
 \text{subject to:} \quad & x_{e,(k)} = f(x_{e,(k-1)}, y_{(ts),k}) \\
 & y_{(rts),k-1} = g(x_{d,(k-1)}) \\
 & x_{d,(k)} = f(x_{d,(k-1)}, y_{(r),k-1}, s) \\
 & y_{(rts),k} = g(x_{d,(k)}) \\
 & x_{d,0} = x_{e,L} = s, k = 1, 2, 3, \dots, L
 \end{aligned} \tag{5}$$

where  $f(\cdot)$  and  $g(\cdot)$  are the LSTM unit function and the fully connected output layer, respectively.  $L$  is the length of the time series.

We utilized parallel training to train this structure with multiple input–output time series samples for highly efficient system identification. Considering the frequency- and amplitude-dependent behavior of PEAs, each time series  $U_{(ts),i}$  is generated using the multi-period sinusoidal function  $S(f_i, A_i)$  and triangular function  $T(f_i, A_i)$ , with  $f_i$  as the signal frequency and  $A_i$  as the amplitude. Mathematically, the collection of  $k$  samples of  $U_{(ts),i}$ ,  $U_{(ts)}$  is formulated as

$$\begin{aligned}
 U_{(ts)} &= \{U_{(ts),1}, U_{(ts),2}, \dots, U_{(ts),k}\} \\
 \text{with } U_{(ts),i} &= \begin{cases} S_i, & i = \frac{k+1}{2} \text{ if } k \text{ is odd number} \\ T_i, & i = \frac{k}{2} \text{ if } k \text{ is even number} \end{cases} \\
 S_i &= S(f_i, A_i), \quad T_i = T(f_i, A_i), \quad \text{where } (f_i, A_i) \in \Omega \\
 S(f_i, A_i) &= A_i[\sin(2\pi f_i t + \frac{3\pi}{2}) + 1], t \in [0, \frac{P}{f}] \\
 T(f_i, A_i) &= 4 \cdot A_i |f_i t - \lfloor f_i t + \frac{1}{2} \rfloor|, t \in [0, \frac{P}{f}]
 \end{aligned} \tag{6}$$

where  $\Omega$  is the set of selected  $(f_i, A_i)$  pairs, which can be determined using  $k$ -means clustering [16,25].  $P$  is the number of periods the signal sample contains. The PEA system output time series set  $Y_{(ts)} = \{Y_{(ts),1}, Y_{(ts),2}, \dots, Y_{(ts),k}\}$  can be then measured subject to the

input set  $U_{(ts)} = \{U_{(ts),1}, U_{(ts),2}, \dots, U_{(ts),k}\}$ . Then, the entire training set can be constructed as  $\{Y_{(ts)}, U_{(ts)}\}$ . The samples of the entire training set are then sorted based on their length and divided into mini-batches using a pre-defined batch size  $n$  ( $n$  samples per batch). Then, the inversion model is trained batch by batch, and the total training iteration needed is  $\lambda K_s/n$ , where  $\lambda$  is the epoch number, and  $K_s$  is the total sample size.

Note that the sample length can be very big for small  $f_{is}$ . This can dramatically increase the training time. Given the PEA's nonlinear dynamics mostly occur for high driving frequencies [22,25], the invLSTMs2s was primarily designed to capture the high-frequency and nonlinear dynamics of the PEA system [22].

## 2.2. Feedback Controller Design

Given the high-frequency dynamics of the PEA can be accurately compensated by the invLSTMs2s, what is left for the PEA positioning control is to design a controller for the PEA's low-frequency dynamics, which is mostly linear. Therefore, a linear MPC is proposed for its efficiency and flexibility in handling control constraints and system disturbances. The MPC works in parallel with the invLSTMs2s feedforward controller. The block diagram of the entire controller design is shown in Figure 2.

The MPC utilizes a state observer and an estimated linear model for the PEA system state estimation and output prediction. Such a state observer, like the Kalman filter, can be expressed as,

$$\begin{aligned} x_{(c),k+1} &= Ax_{(c),k} + Bu_k \\ \hat{y}_k &= Cx_{(c),k}, \end{aligned} \quad (7)$$

where  $x_{(c),k}$  is the estimated state of the state observer, and  $u_k$  is the system input.  $\hat{y}_k$  is the predicted system output.  $A, B, C$  are the estimated state-space matrices that can be obtained using a linear system identification algorithm based on system frequency response fitting, and their dimensions depend on the order of the estimated system chosen by the algorithm and user preference.

**State estimation:** A steady-state Kalman filter [26] was implemented as the state estimator for the MPC. To design the Kalman filter, the process noise covariance and measurement noise covariance were estimated based on the system's linear model identification accuracy, the PEA stage's measurement noise level, and the manufacturer's provided system's noise-to-signal ratio.

For each control interval  $k$ , the system state was estimated as follows:

First, compute the innovation using the current system output measurement  $y_k$

$$e_k = y_k - [Cx_{(c),k|k-1}]. \quad (8)$$

Second, update the current state

$$x_{(c),k|k} = x_{(c),k|k-1} + Me_k, \quad (9)$$

where  $x_{(c),k|k-1}$  is the  $k$  instant controller state estimation at the previous instant,  $k-1$ .  $M$  is the Kalman innovation gain matrix. Then, the MPC uses the updated state  $x_{(c),k|k}$  to predict the system output and to solve for the optimal drive input to the system,  $u_k^{opt}$ , with quadratic programming (QP).

Finally, the state estimator predicts the state for the next control interval  $k+1$

$$x_{(c),k+1|k} = Ax_{(c),k|k-1} + Bu_k^{opt} + Le_k, \quad (10)$$

where  $L$  is the Kalman filter gain matrix.

**Output prediction:** The MPC computes the future system state within the pre-selected prediction horizon  $p$  as follows:

$$x_{(c),k+i|k} = Ax_{(c),k+i-1|k} + Bu_{k+i-1|k}, \quad i = 1, 2, 3, \dots, p. \quad (11)$$

For each step  $i$ , the PEA's output is:

$$\hat{y}_{k+i} = Cx_{(c),k+i|k}. \tag{12}$$

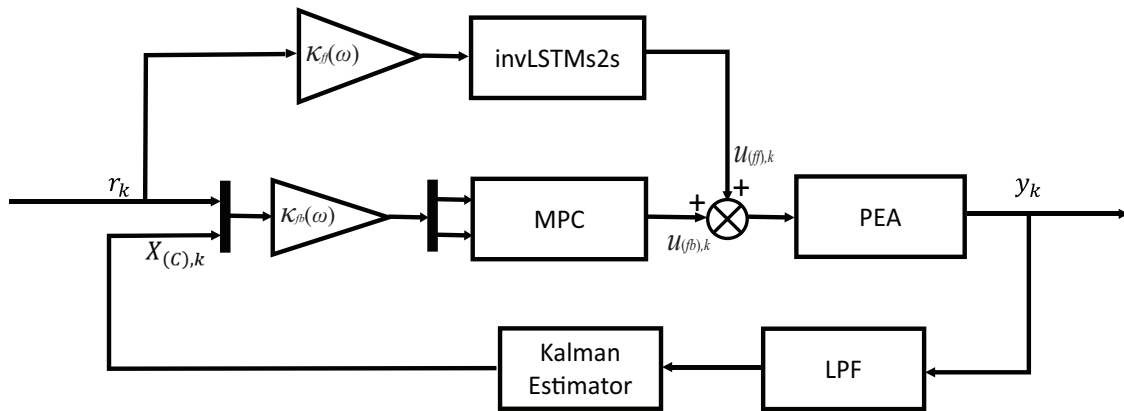


Figure 2. Schematic block diagram of the feedforward–feedback invLSTMs2s+MPC controller.

**QP matrices:** Let  $U = [u_{k+1}, u_{k+1}, \dots, u_{k+m}]$ ,  $\Delta U = [u_{k+1} - u_k, u_{k+2} - u_{k+1}, \dots, u_{k+m} - u_{k+m-1}]$ , and  $1_n = [1, 1, \dots, 1]^T$ , where  $m$  is the control horizon. The aforementioned system output prediction can be rewritten in an alternative form:

$$\begin{aligned} \hat{Y} &= \alpha x_{(c),k|k} + \beta U + F u_k \\ &= \alpha x_{(c),k|k} + \beta (S_1 \Delta U + 1_n u_k) + F u_k \\ &= \alpha x_{(c),k|k} + \beta S_1 \Delta U + (\beta 1_n + F) u_k, \end{aligned} \tag{13}$$

where

$$\hat{Y} = \begin{bmatrix} \hat{y}_{k+1} \\ \hat{y}_{k+2} \\ \vdots \\ \hat{y}_{k+p} \end{bmatrix}_{p \times 1}, \quad U = \begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+m} \end{bmatrix}_{m \times 1}.$$

The terms  $\alpha$  and  $\beta$  are defined as

$$\alpha = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix}_{p \times 1}, \quad \beta = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \cdot & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{p-2}B & CA^{p-3}B & CA^{p-4}B & \cdot & 0 \end{bmatrix}_{p \times m},$$

$$F = \begin{bmatrix} CB \\ CAB \\ CA^2B \\ \vdots \\ CA^{p-1}B \end{bmatrix}, \quad S_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}.$$

**Cost function:** With the given QP matrices, the cost function can be mathematically expressed as

$$J = (\hat{Y} - R)^T (\hat{Y} - R) + \rho \Delta U^T \Delta U, \tag{14}$$

where  $R = [r_{k+1}, r_{k+2}, \dots, r_{k+p}]$  is the reference of desired system output,  $\rho$  is the weighting coefficient for the reference tracking error and MV movement suppression, respectively. Let  $E = \alpha x_{(c),k|k} + (\beta 1_n + F)u_k - R$ ; the cost function can be rewritten as

$$\begin{aligned} J &= (\beta S_1 \Delta U + E)^T (\beta S_1 \Delta U + E) + \Delta U^T (\rho I) \Delta U \\ &= \Delta U^T S_1^T \beta^T \beta S_1 \Delta U + E^T \beta S_1 \Delta U + (E^T \beta S_1 \Delta U)^T + E^T E + \Delta U^T (\rho I) \Delta U \\ &= \Delta U^T (\rho I + S_1^T \beta^T \beta S_1) \Delta U + 2E^T \beta S_1 \Delta U + E^T E. \end{aligned} \quad (15)$$

Now, it becomes a convex optimization problem for the QP progress to compute the optimal  $\Delta U$  based on the known value  $u_k$  and  $x_{(c),k|k}$ , and add the first element of  $\Delta U$  with  $u_k$  to obtain  $u_k^{opt}$ . The convex optimization problem can be formulated as follows:

$$\arg \min_{\Delta U} J = \Delta U^T (\rho I + S_1^T \beta^T \beta S_1) \Delta U + 2E^T \beta S_1 \Delta U + E^T E. \quad (16)$$

**Solution and stability analysis:** The solution for  $\Delta U$  that can minimize the cost function  $J$  should be

$$\Delta U = \arg \min_{\Delta U} J = -(\rho I + S_1^T \beta^T \beta S_1)^{-1} E^T \beta S_1. \quad (17)$$

According to the MPC algorithm, only the first step of the manipulated variable would be taken for system input. Setting  $G = [1, 0, 0, \dots, 0]$ , we can compute

$$\begin{aligned} u_{k+1} &= u_k - G(\rho I + S_1^T \beta^T \beta S_1)^{-1} E^T \beta S_1 \\ &= M_1 x_c(k) + M_2 u_k + M_3 R, \end{aligned} \quad (18)$$

where  $M_1 = -G(\rho I + S_1^T \beta^T \beta S_1)^{-1} \alpha^T \beta S_1$ ,  $M_2 = 1 - G(\rho I + S_1^T \beta^T \beta S_1)^{-1} (\beta 1_n)^T \beta S_1$ , and  $M_3 = G(\rho I + S_1^T \beta^T \beta S_1)^{-1} \beta S_1$ . In this case,  $M_1$ ,  $M_2$ , and  $M_3$  are all constant matrices and can be regarded as the coefficient to  $x_{(c),k|k}$ ,  $u_{k-1}$ , and  $R$ , so that the feedback system can be rewritten as

$$\begin{bmatrix} x_{(c),k+1|k} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ M_1 & M_2 \end{bmatrix} \begin{bmatrix} x_{(c),k|k} \\ u_k \end{bmatrix} + \begin{bmatrix} 0 \\ M_3 \end{bmatrix} R. \quad (19)$$

Based on the alternative form above, it is obvious that the feedback system stability condition needs to satisfy the following:

$$\left| \lambda \left( \begin{bmatrix} A & B \\ M_1 & M_2 \end{bmatrix} \right) \right|_{\max} < 1, \quad (20)$$

where  $\lambda(\cdot)$  denotes the eigenvalues of the matrix.

### 2.3. Impact of the invLSTMs2s Uncertainty

Previous studies have shown that the inversion's modeling uncertainty must be small such that the inversion-based feedforward input improves the output tracking performance when compared to the case with feedback control only (regardless of the type of feedback controller used) [27]. Thus, it is necessary to evaluate the uncertainty of invLSTMs2s in this work. According to [27], for the SISO PEA system, the criteria that need to be satisfied such that integrating the inversion feedforward with the MPC improves the tracking performance can be formulated as

$$|\Delta(j\omega)|_2 \leq |G_o(j\omega)|_2, \quad (21)$$

where  $\Delta(j\omega)$  is the modeling uncertainty at different frequencies  $\omega$ , and it is defined as  $\Delta := G_o - G$ , with  $G$  and  $G_o$  denote the PEA's true dynamics and the captured dynamics, respectively. Therefore, the inverse model  $G_o^{-1}(j\omega)$  is exactly the trained inversion model, invLSTMs2s. Then, (21) can be rewritten as

$$\begin{aligned}
\frac{|\Delta(j\omega)|_2}{|G_0(j\omega)|_2} &\leq 1 \\
|\Delta(j\omega)|_2 |G_0^{-1}(j\omega)|_2 &\leq 1 \\
|G_0(j\omega) - G(j\omega)|_2 |G_0^{-1}(j\omega)|_2 &\leq 1 \\
|1 - G(j\omega)G_0^{-1}(j\omega)|_2 &\leq 1.
\end{aligned} \tag{22}$$

Note that  $G(j\omega)G_0^{-1}(j\omega)$  represents the model of the cascaded invLSTMs2s+PEA system, i.e., when the invLSTMs2s feedforward controller is applied to the PEA alone. Thus, the tracking error subject to reference  $R$  in this case is defined as  $E_{ff-only} := R - Y = [1 - G(j\omega)G_0^{-1}(j\omega)]R$ , and (22) becomes

$$\max_{|R(j\omega)|_2 \neq 0} \frac{|E_{ff-only}(j\omega)|_2}{|R(j\omega)|_2} \leq 1, \tag{23}$$

which matches the  $E_{rms}$  in (24) as discussed in next subsection.

Note that  $\frac{|E_{ff-only}(j\omega)|_2}{|R(j\omega)|_2}$  is the RMS tracking error (see (24)). Therefore, as long as the RMS tracking error satisfies the above (23) when only the invLSTMs2s feedforward controller is used, it is guaranteed that the proposed 2DOF controller performs better than the MPC alone.

Due to the training data limit in practice, such as data length, frequency, and amplitude limits, invLSTMs2s may not capture the PEA dynamics outside of the training frequency and/or amplitude ranges. In this case, the modeling uncertainty  $\Delta$  may not satisfy condition (21). To solve this issue, we introduced frequency-dependent weighting factors  $\kappa_{ff}(\omega)$  and  $\kappa_{fb}(\omega)$ , such that the contribution of the inversion feedforward part could be flexibly adjusted based on the frequency range of the reference (see Figure 2). For example, for low-frequency (lower than the invLSTMs2s lower training frequency bounds) tracking,  $\kappa_{ff}$  can be set small so the MPC feedback control is dominant to minimize the uncertainty effect. Also, a low-pass filter (LPF) was used in the MPC part to avoid high-frequency data that could be fed into the feedback loop.

### 3. Experiment Results and Discussion

The proposed invLSTMs2s+MPC 2DOF controller was implemented on a PEA stage (Nano-OP30, Mad City Labs, Madison, WI, USA) for the experimental validation. The performance was compared with that of feedforward invLSTMs2s only, an MPC only, and a traditional PI controller. All the signals were acquired using a data acquisition (DAQ) system (NI PCIe-6353, National Instruments, Austin, TX, USA) installed on a desktop workstation (Intel Xeon W-2125, RAM 32 GB), as shown in Figure 3. All the models and controllers were trained/ designed using MATLAB (R2023a) and Simulink (10.7) (MathWorks, Inc., Natick, MA, USA). For all the experiments, the sampling frequency was set at 20 kHz.

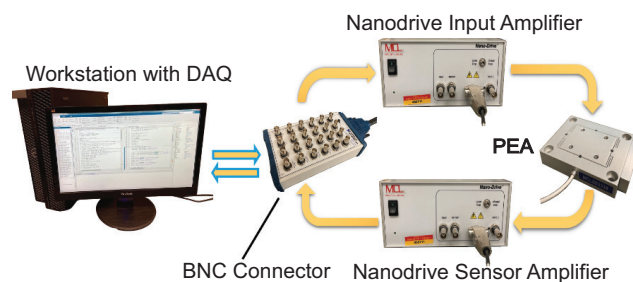
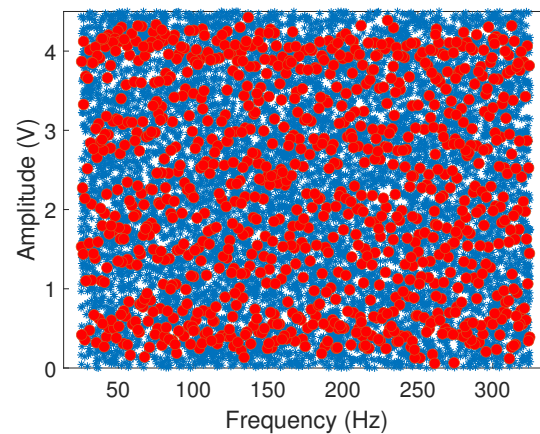


Figure 3. PEA control system setup.



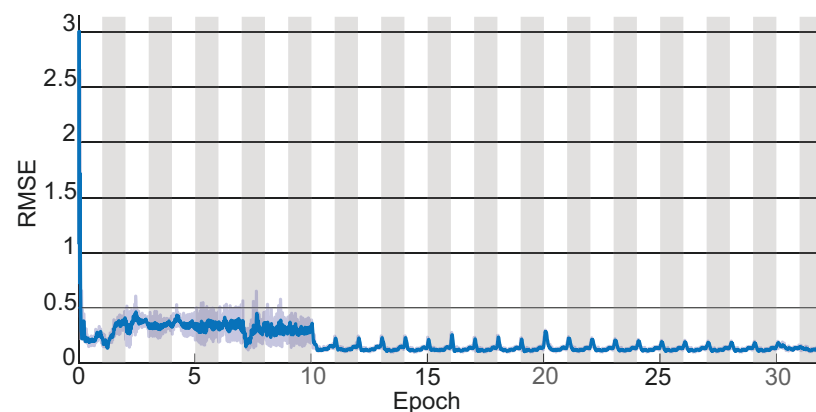
### 3.1. *invLSTMs2s* Training Set Construction and Training Process

The prerequisite step for generating the *invLSTMs2s* training set  $\{Y_{(ts)}, U_{(ts)}\}$  is finding sufficient  $f$ - $A$  pairs. To do so, 10,000 points (blue dots in Figure 4) in the  $f$  range of 25–325 Hz and amplitude range of 0–4.5 V were first randomly generated in the  $f$ - $A$  plane, and the  $k$ -means algorithm [22] picked 1000 points (red points in Figure 4) as  $\Omega$ 's for the training set generation. Then, (6) was used to generate  $U_{(ts)}$  with  $P = 3$ . Using  $U_{(ts)}$  as the drive input of the PEA, the corresponding PEA output  $Y_{(ts)}$  was collected by the DAQ device. Thus, the entire training set  $(Y_{(ts)}, U_{(ts)})$  contained 2000 samples.



**Figure 4.**  $\Omega$  (set of  $(f_i, A_i)$  pairs) selected for generating the training input  $U_{(ts)}$ . Red dots represent the selected  $(f_i, A_i)$  pairs by the  $k$ -means algorithm.

The *invLSTMs2s* model with 40 hidden states was used to model the PEA inverse dynamics and trained with  $\lambda = 32$  epochs using parallel training. A batch size of  $n = 20$  was selected. Therefore, the entire training process took  $\lambda K_s/n = 3200$  iterations. Figure 5 shows the *invLSTMs2s* root-mean-square error (RMSE) learning curve as the epoch number increases. It clearly shows the convergence of the *invLSTMs2s* training process with very low training error.

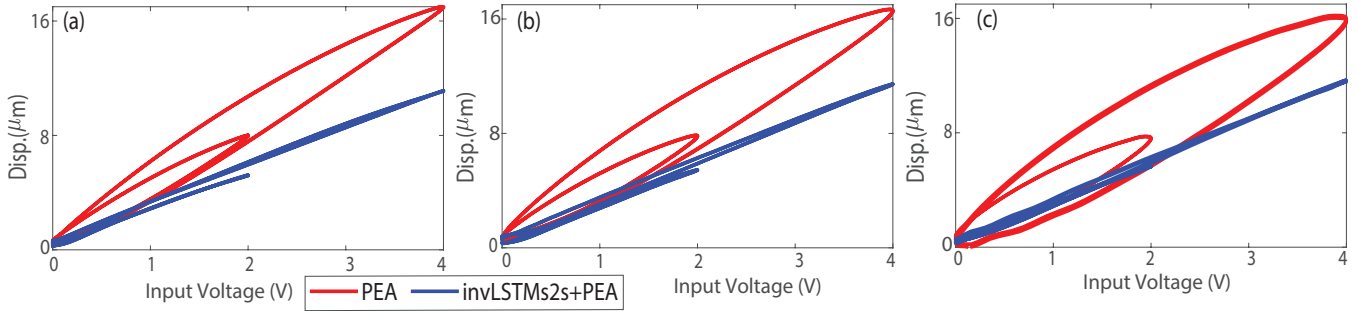


**Figure 5.** The RMSE learning curve of *invLSTMs2s*. With 32 epochs and 100 samples in each mini-batch, the total training iteration number was 3200.

### 3.2. Accuracy of *invLSTMs2s*

The model accuracy was tested by checking how well *invLSTMs2s* could compensate for the PEA hysteresis, i.e., whether the hysteresis could be mostly removed when *invLSTMs2s* was cascaded with the PEA (connected in series). As shown in Figure 6, the original PEA hysteresis curves were measured using different sinusoidal driving voltages with the frequencies 30, 120, and 240 Hz and amplitudes of 2 and 4 V, respectively. Please

note that these frequencies were intentionally selected to avoid overlapping with those in the training set to minimize the effect of data discrepancy. It can be seen that the PEA hysteresis tended to be more pronounced with increasing frequency and/or amplitude. However, the corresponding hysteresis curves of the cascaded invLSTMs2s+PEA system were barely notable for all measured frequencies and amplitudes (see Figure 6). Therefore, the invLSTMs2s inversion model was indeed accurate in accounting for the PEA's non-linear dynamics. This accuracy ensured that the modeling uncertainty stayed small (i.e., (21) was satisfied) and guaranteed that the output tracking performance of the proposed 2DOF invLSTMs2s+MPC controller was better than that of the MPC alone. More detailed comparisons are presented next.



**Figure 6.** PEA displacement vs. input voltage at the frequencies of (a) 30 Hz, (b) 120 Hz, (c) 240 Hz for the comparison between the PEA and the PEA cascaded with inversion model (invLSTMs2s+PEA).

### 3.3. Tracking Performance Comparison

The tracking performance of the proposed invLSTMs2s+MPC was demonstrated as follows. For the MPC, the prediction horizon was set to 60, and the stability condition (20) was satisfied. To design the Kalman state estimator, the covariance for the process noise and measurement noise were selected as  $Q = 10^{-4}$  and  $R = 10^{-6}$ , respectively, based on the system's linear model identification accuracy, the PEA stage's measurement noise level, and the system's noise-to-signal ratio provided by the manufacturer. Also, to avoid high-frequency noise being fed into the feedback loop, a third-order Butterworth LPF with the cut-off frequency of 600 Hz was used to avoid high-frequency noise/disturbance being fed into the feedback loop.

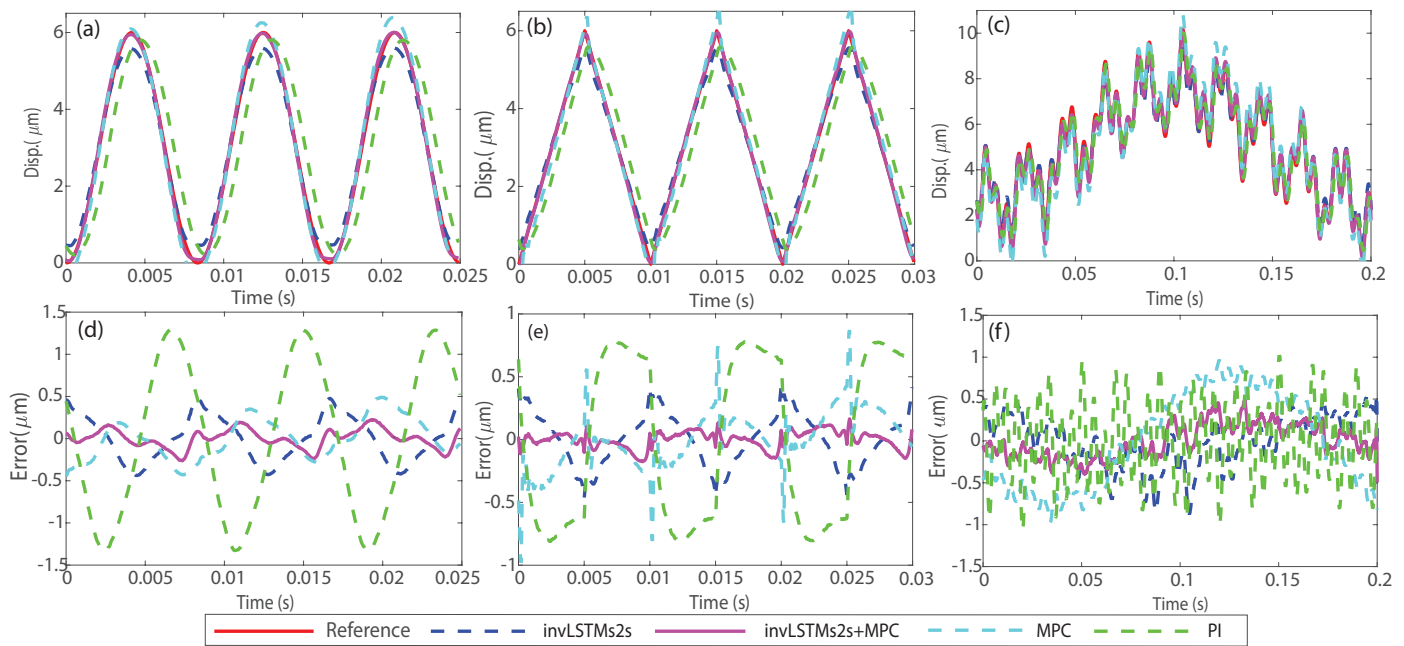
Although the invLSTMs2s was trained up to 300 Hz, the control bandwidth was expected to reach 350 Hz due to the generalizability of LSTM. Based on the PEA behavior observed, whose nonlinear dynamics was highly nonlinear at frequencies close or above 200 Hz, frequencies  $\geq 200$  Hz were considered as high frequencies. To comprehensively test the controller performance over broad frequency/amplitude ranges, the reference trajectories were designed to have low ( $\leq 50$  Hz), mid ( $\sim 100$  Hz), and high frequency ( $\geq 200$  Hz) profiles: sinusoidal signals (at 30, 120, and 360 Hz), triangle signals (at 25, 100, and 300 Hz), and a multi-frequency reference  $\Gamma$ , where  $\Gamma(t) = 0.8 \sin(2\pi 5t + 1.5\pi) + 0.43 \sin(2\pi 50t) + 0.12 \sin(2\pi 120t + 1.2\pi) + 0.3 \sin(2\pi 180t + \pi)$ .

First, the accuracy of invLSTMs2s as the feedforward controller only was evaluated. Then, the proposed invLSTMs2s+MPC 2DOF controller was compared with the cases of using invLSTMs2s, the MPC, and PI feedback. The PI controller parameters were tuned to  $P = 0.312$  and  $I = 1465$  to achieve the best performance possible. The tracking errors were computed as [28]:

$$E_{rms} = \frac{|r(\cdot) - y(\cdot)|_2}{|r(\cdot)|_2} \times 100\%, \quad E_{max} = \frac{|r(\cdot) - y(\cdot)|_\infty}{|r(\cdot)|_\infty} \times 100\%, \quad (24)$$

where  $r(\cdot)$  and  $y(\cdot)$  are the complex vectors of reference and output obtained through discrete Fourier transforms, respectively. Table 1 shows the tracking errors for all the

controllers. Figure 7 shows the detailed tracking performance in the time domain for the selected references of (120 Hz sinusoidal signal, 100 Hz triangular signal, and  $\Gamma$ ).



**Figure 7.** Comparison of the tracking results of invLSTMs2s, invLSTMs2s+MPC, the MPC, and the PI controller for (a) 100 Hz triangular trajectory, (b) 120 Hz sinusoidal trajectory, and (c)  $\Gamma$ . (d–f) are the corresponding tracking errors, respectively.

**Table 1.** Performance comparison.

Controller	Refs.	Sinusoidal			Triangular			$\Gamma$
		30 Hz	120 Hz	360 Hz	25 Hz	100 Hz	300 Hz	
invLSTMs2s	$E_{rms}$	9.35%	7.25%	3.10%	9.04%	6.00%	4.29%	5.64%
	$E_{max}$	7.91%	6.14%	2.30%	7.13%	4.74%	2.51%	3.49%
invLSTMs2s+MPC	$E_{rms}$	3.19%	2.97%	7.53%	4.05%	1.94%	6.32%	3.63%
	$E_{max}$	2.09%	2.07%	6.25%	2.85%	1.05%	4.31%	2.39%
MPC	$E_{rms}$	1.66%	6.57%	27.18%	3.28%	6.80%	21.66%	10.06%
	$E_{max}$	1.12%	4.82%	23.26%	2.63%	3.77%	12.68%	7.57%
PI	$E_{rms}$	6.37%	24.34%	45.89%	5.08%	18.73%	38.01%	8.56%
	$E_{max}$	5.50%	21.04%	39.47%	3.77%	14.88%	30.60%	5.81%

**invLSTMs2s+MPC vs. invLSTMs2s:** First of all, condition (23) was satisfied for the case when only invLSTMs2s was used when tracking all the designed references: the RMS tracking errors were all much smaller than one. Hence, invLSTMs2s could accurately capture the PEA nonlinear dynamics and produced very small errors for tracking high-frequency references; however, relatively higher errors at the low-frequency range were observed. For example, in Table 1, the tracking error of the invLSTMs2s for both sinusoidal and triangular references at 30 and 25 Hz were a few percent higher than those for 360 and 300 Hz, respectively. This was mainly because these frequencies were very close to the lower bound of the training frequency range. Thus, limited knowledge of the system dynamics at these frequencies was available when training invLSTMs2s. This shortage was compensated for by the MPC, which was designed using a linear model that primarily captured the PEA dynamics at low frequencies. The invLSTMs2s+MPC outperformed invLSTMs2s at the low-frequency range. The tracking errors of the invLSTMs2s+MPC for both sinusoidal and triangular signals at 30 and 25 Hz were reduced by more than

50% compared to those of invLSTMs2s. As shown in Figure 7, the low-frequency tracking error was greatly reduced by invLSTMs2s+MPC compared to invLSTMs2s. As the PEA nonlinear dynamics became more pronounced at higher frequencies, the advantage of invLSTMs2s became more significant: invLSTMs2s achieved the highest accuracy for 360 Hz sinusoidal and 300 Hz triangular trajectories. Meanwhile, as the linear MPC was not designed for the PEA's high-frequency control, it negatively impacted the performance of invLSTMs2s. Therefore, bigger errors were observed (see Table 1). However, compared to the errors of MPC, the performance was still greatly improved, and the achieved accuracy was satisfactory.

**invLSTMs2s+MPC vs. MPC:** The proposed invLSTMs2s+MPC was compared with the use of the MPC alone to clearly demonstrate the necessity of integrating the invLSTMs2s feedforward control. As the performance improvement condition (23) was satisfied for all the tracking cases ( $E_{rms} \ll 1$  for the use of invLSTMs2s alone in Table 1), it was expected that the proposed 2DOF invLSTMs2s+MPC controller would outperform the MPC. Table 1 shows that the invLSTMs2s+MPC had less tracking error than the MPC, especially for tracking mid-, high-frequency, and the multi-frequency  $\Gamma$  trajectories. However, it made sense that the tracking errors of invLSTMs2s+MPC at 30 and 25 Hz were a little bit higher than those of the MPC, since the MPC is more accurate than invLSTMs2s at low frequencies, as aforementioned. Note that the slight error increase can be eliminated by setting the feedforward weighting coefficient  $\kappa_{ff}(\omega)$  to a very small value to make sure the 2DOF controller takes full advantage of the MPC at low frequencies.

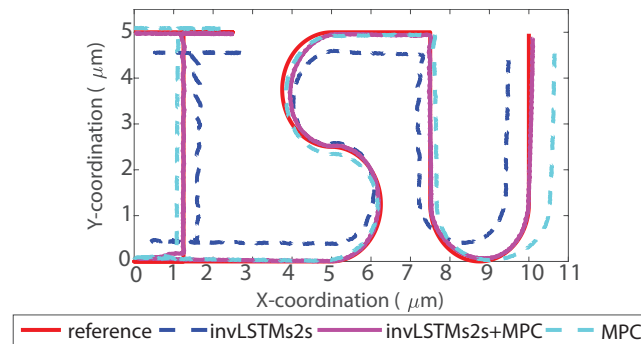
**invLSTMs2s+MPC vs. PI:** To demonstrate the significance of this work, invLSTMs2s+MPC was compared to the case of using a PI controller alone. In Table 1, it can be observed that the tracking error of the PI controller was more pronounced with increasing frequency. For the mid and high frequencies, the PI presented the highest tracking error among all controllers. This is mainly because the PI feedback controller was driven by the errors between references and PEA outputs, and the control bandwidth was low due to the sluggish response of this controller subject to fast error changes.

In summary, invLSTMs2s+MPC is capable of achieving high tracking accuracy for various reference trajectories over a broad bandwidth. The superiority of this 2DOF controller comes from taking the advantages of feedforward invLSTMs2s for the PEA's high-frequency nonlinear dynamics compensation and MPC for low-frequency position tracking precision.

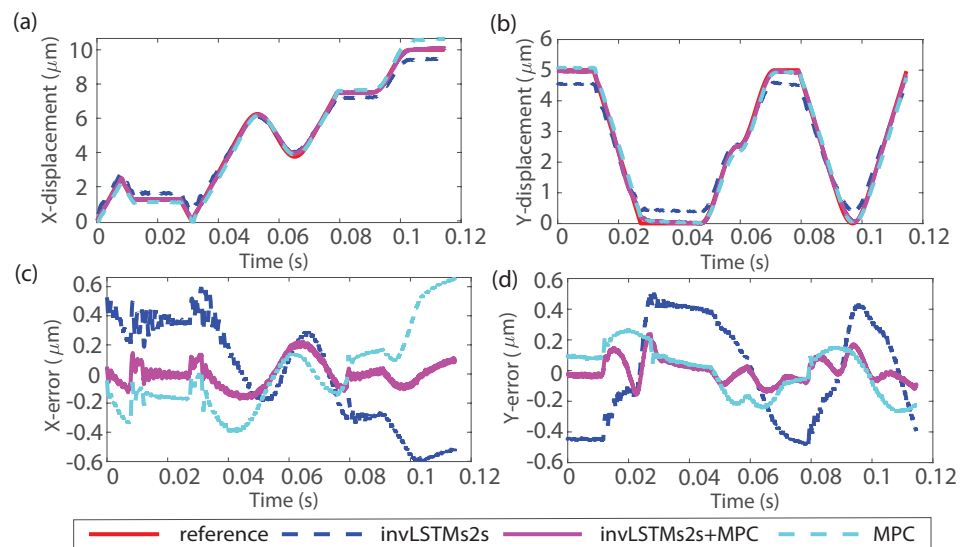
### 3.4. Application Demonstration of the invLSTMs2s+MPC

As an application demonstration, invLSTMs2s+MPC was further adopted to control a two-dimensional ( $X$ - $Y$  lateral plane motion) PEA nanopositioning platform to track the pattern of character string "ISU" along the lateral  $X$ - $Y$  plane. The tracked patterns achieved by different controllers are compared in Figure 8. It was designed to complete the motion within 0.11 s, with the stage average traveling speed of 320  $\mu\text{m}/\text{sec}$ . The designed  $X$  and  $Y$  direction trajectories and tracking results for different controllers are shown in Figure 9a,b, respectively. According to the results, it is clear that the proposed invLSTMs2s+MPC performed better and could achieve very high accuracy. This clearly demonstrated the capability of invLSTMs2s+MPC in two-dimensional positioning tracking control in nanofabrication applications, such as nano/micromanufacturing systems in which the PEA stages are required to track the desired fabrication pattern [29,30].

As invLSTMs2s does not assume any specific forms of nonlinearity, the outstanding performance of invLSTMs2s+MPC directly implies that it can be employed in various nonlinear systems for applications where position tracking is essential, such as precise control of friction force generated by deformable actuators in human-robot reaction [31], AFM applications for material mechanical characterization [32], as well as adaptive optics for high-resolution imaging [33], just to name a few.



**Figure 8.** Comparison of the stage motion pattern achieved using controllers invLSTMs2s, invLSTMs2s+MPC, and MPC.



**Figure 9.** Tracking results of invLSTMs2s, invLSTMs2s+MPC, and MPC for (a) X-direction trajectory and (b) Y-direction trajectory. (c,d) are the corresponding tracking errors, respectively.

#### 4. Conclusions

In this work, we proposed a 2DOF controller, invLSTMs2s+MPC, for nanositioning control of PEA devices with a broad bandwidth, high precision, and fast response. The key novelty of the proposed controller includes (1) the idea of implementing an LSTMs2s structure to model the PEA inverse dynamics for system nonlinearity compensation and (2) integrating invLSTMs2s and a linear MPC to improve the overall tracking performance. As the invLSTMs2s model can be trained offline and a linear MPC is used during real-time control, the proposed controller can achieve broad-bandwidth output tracking with its low computation requirement.

In the future, the work will focus on designing a dynamics separation mechanism to efficiently optimize the collaboration between the feedforward and feedback components. Also, we will explore other modeling network structures to improve the training speed further and increase the modeling bandwidth, robustness, and accuracy.

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

**Funding:** This research was funded by the National Science Foundation (NSF) (CMMI-1751503 and CNS-2409359) and Iowa State University.

**Data Availability Statement:** The original contributions presented in the 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.

### Abbreviations

The following abbreviations are used in this manuscript:

LSTM	Long short-term memory
LSTMs2s	Sequence-to-sequence long short-term memory
invLSTMs2s	Inversive sequence-to-sequence long short-term memory
NN	Neural network
RNN	Recurrent neural network
2DOF	Two-degree of freedom
MPC	Model predictive controller
PEA	Piezoelectric actuator
AFM	Atomic force microscopy
QP	Quadratic programming

### References

- Mollaeian, K.; Liu, Y.; Bi, S.; Ren, J. Atomic force microscopy study revealed velocity-dependence and nonlinearity of nanoscale poroelasticity of eukaryotic cells. *J. Mech. Behav. Biomed. Mater.* **2018**, *78*, 65–73. [[CrossRef](#)] [[PubMed](#)]
- Xie, S.; Ren, J. Iterative Learning-based Model Predictive Control for Precise Trajectory Tracking of Piezo Nanopositioning Stage. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 2922–2927.
- Tian, Y.; Zhang, D.; Shirinzadeh, B. Dynamic modelling of a flexure-based mechanism for ultra-precision grinding operation. *Precis. Eng.* **2011**, *35*, 554–565. [[CrossRef](#)]
- Qin, F.; Zhang, D.; Xing, D.; Xu, D.; Li, J. Laser beam pointing control with piezoelectric actuator model learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *50*, 1024–1034. [[CrossRef](#)]
- Fan, Y.; Tan, U.X. Design of a feedforward-feedback controller for a piezoelectric-driven mechanism to achieve high-frequency nonperiodic motion tracking. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 853–862. [[CrossRef](#)]
- Mao, X.; Wang, Y.; Liu, X.; Guo, Y. A hybrid feedforward-feedback hysteresis compensator in piezoelectric actuators based on least-squares support vector machine. *IEEE Trans. Ind. Electron.* **2017**, *65*, 5704–5711. [[CrossRef](#)]
- Baziyad, A.G.; Ahmad, I.; Salamah, Y.B. Precision Motion Control of a Piezoelectric Actuator via a Modified Preisach Hysteresis Model and Two-Degree-of-Freedom H-Infinity Robust Control. *Micromachines* **2023**, *14*, 1208. [[CrossRef](#)]
- Cheng, L.; Chen, W.; Tian, L.; Xie, Y. A feedforward-feedback controller based on modified Prandtl–Ishlinskii model and submodel for tracking control the trajectories of piezoelectric actuator. *Microsyst. Technol.* **2024**, *30*, 1007–1024. [[CrossRef](#)]
- Ahmad, I. Two degree-of-freedom robust digital controller design with Bouc-Wen hysteresis compensator for piezoelectric positioning stage. *IEEE Access* **2018**, *6*, 17275–17283. [[CrossRef](#)]
- Yuan, J.; Wu, H.; Qin, Y.; Han, J. High-Bandwidth Repetitive Trajectory Tracking Control of Piezoelectric Actuators via Phase-Hysteresis Hybrid Compensation and Feedforward–Feedback Combined Control. *Micromachines* **2023**, *14*, 2009. [[CrossRef](#)]
- Gan, J.; Zhang, X. A review of nonlinear hysteresis modeling and control of piezoelectric actuators. *AIP Adv.* **2019**, *9*, 040702. [[CrossRef](#)]
- Xie, S.; Ren, J. Tracking control using recurrent-neural-network-based inversion model: A case study on a piezo actuator. *IEEE Trans. Ind. Electron.* **2020**, *68*, 11409–11419. [[CrossRef](#)]
- Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Atlanta, GA, USA, 17–19 June 2013; pp. 1310–1318.
- Hochreiter, S. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma Tech. Univ. München* **1991**, *91*, 31.
- Rehmer, A.; Kroll, A. On the vanishing and exploding gradient problem in Gated Recurrent Units. *IFAC-PapersOnLine* **2020**, *53*, 1243–1248. [[CrossRef](#)]
- Patil, M.S.; Charuku, B.; Ren, J. Long Short-term Memory Neural Network-based System Identification and Augmented Predictive Control of Piezoelectric Actuators for Precise Trajectory Tracking. *IFAC-PapersOnLine* **2021**, *54*, 38–45. [[CrossRef](#)]
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
- Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [[CrossRef](#)]
- Wang, Y. A new concept using lstm neural networks for dynamic system identification. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5324–5329.
- Ookura, S.; Mori, H. An efficient method for wind power generation forecasting by LSTM in consideration of overfitting prevention. *IFAC-PapersOnLine* **2020**, *53*, 12169–12174. [[CrossRef](#)]
- Gonzalez, J.; Yu, W. Non-linear system modeling using LSTM neural networks. *IFAC-PapersOnLine* **2018**, *51*, 485–489. [[CrossRef](#)]

22. Yin, R.; Ren, J. Sequence-to-sequence LSTM-based Dynamic System Identification of Piezo-electric Actuators. In Proceedings of the 2023 American Control Conference (ACC), San Diego, CA, USA, 31 May–2 June 2023; pp. 673–678.
23. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
24. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv* **2014**, arXiv:1409.3215.
25. Xie, S.; Ren, J. Recurrent-neural-network-based predictive control of piezo actuators for trajectory tracking. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2885–2896. [[CrossRef](#)]
26. Franklin, G.F.; Powell, J.D.; Workman, M.L. *Digital Control of Dynamic Systems*; Addison-Wesley: Menlo Park, CA, USA, 1998; Volume 3.
27. Devasia, S. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Trans. Autom. Control* **2002**, *47*, 1865–1871. [[CrossRef](#)]
28. Kim, K.S.; Zou, Q. A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems. *IEEE/ASME Trans. Mechatron.* **2012**, *18*, 1767–1777. [[CrossRef](#)]
29. Yan, Y.; Sun, T.; Liang, Y.; Dong, S. Investigation on AFM-based micro/nano-CNC machining system. *Int. J. Mach. Tools Manuf.* **2007**, *47*, 1651–1659. [[CrossRef](#)]
30. Du, P.; Han, L.; Qiu, X.; Chen, W.; Deng, J.; Liu, Y.; Zhang, J. Development of a high-precision piezoelectric ultrasonic milling tool using longitudinal-bending hybrid transducer. *Int. J. Mech. Sci.* **2022**, *222*, 107239. [[CrossRef](#)]
31. Xiong, J.; Chen, J.; Lee, P.S. Functional fibers and fabrics for soft robotics, wearables, and human–robot interface. *Adv. Mater.* **2021**, *33*, 2002640. [[CrossRef](#)]
32. Kathavate, V.; Prasad, K.E.; Kiran, M.S.; Zhu, Y. Mechanical characterization of piezoelectric materials: A perspective on deformation behavior across different microstructural length scales. *J. Appl. Phys.* **2022**, *132*, 121103. [[CrossRef](#)]
33. Ren, K.; Gao, X.; Jin, H.; Qiao, L.; Xia, S.; Li, F. Conformal Ordered Solid–Liquid Coupled Piezoelectric Units for Programmable Adaptive Optics. *Adv. Funct. Mater.* **2024**, 2410173. [[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.