2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Predictive Control of Nano-positioning Stage Using Recurrent-neural-network-based Inversion Model

Shengwen Xie[1] and Juan Ren[1,†]

Abstract— **The system modeling accuracy directly affects the performance of inversion-based control techniques, especially for applications on nonlinear systems, such as piezo actuators. In this paper, we propose to use recurrent neural network (RNN) for modeling the system nonlinearity and thus generating the inversion model for real-time control of piezo actuators. Considering the computation efficiency, one issue of using RNN inversion model is that the low frequency dynamics of the system may not be captured as the length of the training set for training RNN should be kept short to reduce the training time and the number of parameters in RNN. Thus, we propose to use a second order linear system embedded with an error term (LME) to account for the unmodeled low frequency dynamics, and a predictive controller based on LME is designed to improve the tracking performance. Therefore, the proposed approach combines RNN and LME to achieve high precision control. The proposed approach was experimentally demonstrated and compared with other control approaches through implementation on a commercial piezo actuator.**

I. Introduction

Precision motion control of piezo actuators (PEAs) is essential to nanoscale characterizations and fabrication applications, such as scanning probe microscope, micro-forming, and adaptive optics [1]–[5]. However, it is challenging to achieve this goal due to the nonlinearities existing in a PEA system, such as creep effect and hysteresis [6]. For example, broadband control (i.e., control over a large bandwidth in frequency domain) of PEA is challenging as it is difficult to accurately model the rate-dependent hysteresis [6]. Therefore, the system nonlinearities must be accounted for in the PEA control.

To account for the nonlinearities, one efficient approach is to use inversion dynamics. The use of an inversion model simplifies the process of controller design assuming the system nonlinearity can be mostly eliminated by the inversion model. For example, inversion models measured in frequency domain have been used in iterative learning control, such as inversion-based iterative learning control (IIC) and modeling-free inversion-based iterative feedforward control (MIIFC) in high precision PEA control for repetitive tasks [7], [8]. Also, repetitive control tools have been proposed by taking into account the nonlinearities [9]–[11]. Although ILC and repetitive control can achieve high precision tracking of PEAs, their application is restricted by the assumption that the operations are strictly repetitive. To relax this requirement, ILC-based model predictive control approaches that allow small variations to exist between iterations have been developed [12]–[14]. It had been shown that the control accuracy is limited by the variation between iterations which implies that these ILC approaches are still far from real-time tracking. Thus, more accurate inversion models have been developed in real-time PEA control. In [15], an inversion model based on the ferromagnetic material hysteresis was cascaded to the PEA for predictive control. Prandtl-Ishlinskii hysteresis inversion model was used on a dual-stage [10]. Inversion model based on Prandtl-Ishlinskii operator was also applied to reduce the hysteresis effect [16], [17]. However, the highest control frequency achieved by the existing inversion models are reported to be around 100Hz. Both the modeling accuracy and bandwidth limited the existing inversion models for tracking high bandwidth trajectories. Precision inversion model with high bandwidth is still unavailable.

To improve the modeling bandwidth, neural networks have been proposed to model the system dynamics recently [18]–[20]. Feedforward neural network (FNN) has been used for modeling PEA system dynamics [18], [19]. However, it's not trained with time series input and the controller based on FNN may require high-speed hardware to implement when the parameters of the FNN are too much. Therefore, in this work, we propose to use recurrent neural network (RNN) to model the PEA inversion dynamics, called RNNinv later. Compared to FNN, the input to RNNinv is a time series thus the order of input sequence is accounted for. One challenge in using RNNinv is the training part: if the training set contains too many low frequency signals the length of the training set will be too long such that the computation burden is significantly heavy. However, RNNinv may not capture the low frequency inversion dynamics accurately if the length of the training set is limited. To address this issue, the trained RNNinv will be cascaded to the PEA system and a second order linear system embedded with an error term (LME) is designed to handle the PEA low frequency dynamics unmodeled by the RNNinv. Predictive control based on the LME will be implemented to improve the tracking performance in low frequency region. Since the proposed method does not assume any nonlinearity of the system, this framework (RNNinv+LME) can be adapted to control other nonlinear systems as well.

The advantages of the proposed RNNinv+LME framework are threefold. First, compared to the neural network approaches that have been used for system dynamics identification [18], [19], the computation burden is greatly mitigated since the control in low frequency region is handled by

[1]S. Xie and [1]J. Ren are with the Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, USA swxie@iastate.edu,juanren@iastate.edu
† Corresponding author.

the LME and the predictive controller, both of which are linear. Second, the bandwidth of the RNNinv is much higher than existing inversion models. Another advantage of using RNNinv is that the modeling accuracy can be further improved with more complex neural networks without worrying about the computation issue, thus higher controlling performance can be achieved. For demonstration, the proposed RNNinv+LME framework was implemented to control the displacement of a PEA stage, and the control performance was compared with that of a PID controller and even MIIFC to demonstrate that the real-time control accuracy achieved by RNNinv+LME is comparable or even better than that of iterative learning control.

II. RNNinv

The structure of RNNinv is same as that used in [21] and is expressed in the following equation.

$$\begin{aligned} x_{k+1} &= \tanh(W_1 x_k + B_2 + B_1 u_{(r),k}) \\ y_{(r),k} &= W_2 x_k + B_3 \end{aligned}, \tag{1}$$

where the sizes of W_1, B_2, B_1, W_2 and B_3 are $N\times N$, $N\times 1$, $N\times 1$, $1\times N$ and 1×1, respectively. Suppose the output (i.e., trajectory) of the PEA system is $Y_{(ts)}$ subject to the drive input $U_{(ts)}$, an ideal inversion model with input $Y_{(ts)}$ should output $Y_{(rts)}$ such as that $||Y_{(rts)} - U_{(ts)}|| < \varepsilon$ for any $\varepsilon > 0$. Therefore, the pair $(Y_{(ts)}, U_{(ts)})$ can be used to train the RNNinv (i.e., to obtain the parameters W_1, B_2, B_1, W_2 and B_3 in Eq.(1)). Next, we present how to construct the training set $(Y_{(ts)}, U_{(ts)})$ of the RNNinv.

To construct the training set $(Y_{(ts)}, U_{(ts)})$, we need to design the time series $Y_{(ts)}$ that considers both the frequency and amplitude dependent behavior of PEAs. In this work, the method developed in [21] is used. Then, the $U_{(ts)}$ can be obtained by applying ILC approaches on the PEA with $Y_{(ts)}$ as the desired trajectory [8], [13].

Once the $(Y_{(ts)}, U_{(ts)})$ pair is determined, the RNN-based inversion model (RNNinv) parameters can be obtained through the training process by solving the following optimization problem.

$$\begin{aligned} \min_{W_1,B_2,B_1,W_2,B_3} \quad & J = ||U_{(ts)} - Y_{(rts)}|| \\ \text{subject to}: \quad & x_{k+1} = \tanh(W_1 x_k + B_2 + B_1 y_{(ts),k}) \\ & y_{(rts),k} = W_2 x_k + B_3 \\ & x_0 = [0,0,\cdots,0]^T,\ k = 1,2,3,\cdots,L \end{aligned}, \tag{2}$$

where $U_{(ts)} = [u_{(ts),1}, u_{(ts),2}, \cdots, u_{(ts),L}]^T$, $Y_{(ts)} = [y_{(ts),1}, y_{(ts),2}, \cdots, y_{(ts),L}]^T$ and $Y_{(rts)} = [y_{(rts),1}, y_{(rts),2}, \cdots, y_{(rts),L}]^T$. L is the length of the time series.

III. LME-based Predictive Control

A. LME

The obtained PEA RNNinv model will be cascaded to the plant forming a new system $\mathbb{H}$ as shown in Fig. 1. As the RNNinv model may not accurately capture the PEA low frequency dynamics when the length of the training set is limited for the concern of computation efficiency, the LME represented by Eq. (3) below is proposed to account for the low frequency dynamics of $\mathbb{H}$,

$$\begin{aligned} \eta_{k+1} &= A_e \eta_k + B_e \hat{u}_k + G_e \hat{e}_k \\ \hat{y}_k &= C_e \eta_k \end{aligned}, \tag{3}$$

where $\hat{u}_k$ is the input to the LME, $\hat{e}_k = \bar{y}_k - \hat{y}_k$ is the model output error with $\bar{y}_k$ as the actual PEA output. The sizes of A_e, B_e, G_e and C_e are 2×2, 2×1, 2×1 and 1×2, respectively. The error term can be regarded as a feedback term. Suppose the output of $\mathbb{H}$ is $\bar{Y}_{(ts)}$ subject to the designed $Y_{(ts)}$, the output of the LME $\hat{Y}_{(ts)}$ should be nearly the same as $\bar{Y}_{(ts)}$. Similar to the RNNinv, the parameters of LME can be obtained through solving the following optimization problem.

$$\begin{aligned} \min_{A_e,B_e,G,C_e} \quad & J_1 = ||\hat{Y}_{(ts)} - \bar{Y}_{(ts)}|| \\ \text{subject to}: \quad & \eta_{k+1} = A_e \eta_k + B_e y_{(ts),k} + G\hat{e}_k \\ & \hat{Y}_{(ts),k+1} = C_e \eta_{k+1} \\ & \hat{e}_k = \bar{Y}_{(ts),k} - \hat{Y}_{(ts),k} \\ & \eta_0 = [0,0]^T \\ & \hat{e}_0 = 0,\ k = 0,1,\cdots \end{aligned}. \tag{4}$$

Furthermore, to avoid high frequency disturbance to be fed into the feedback loop, a low pass filter (LPF) is cascaded to the LME to remove the ultra-high frequency dynamics. Suppose the LPF can be represented as

$$\begin{aligned} \beta_{k+1} &= \bar{A}\beta_k + \bar{B}u_k \\ \bar{z}_k &= \bar{C}\beta_k \end{aligned}. \tag{5}$$

Since system Eq. (3) is connected to Eq. (5), we have $\bar{z}_k = \hat{u}_k$. Thus the two models can be regarded as the "plant model" shown next.

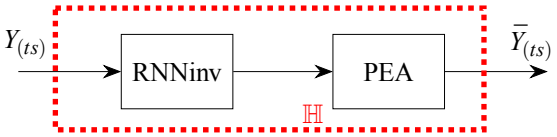

Fig. 1: Generate training set $(Y_{(ts)}, \bar{Y}_{(ts)})$ for LME identification.

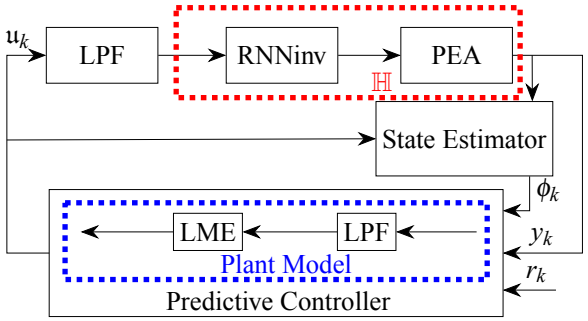

Fig. 2: Schematic block diagram of the proposed RNNinv+LME framework.

$$\begin{aligned}\phi_{k+1} &= \begin{bmatrix}\beta_{k+1}\\ \eta_{k+1}\end{bmatrix} = \begin{bmatrix}\bar{A} & 0\\ B_e\bar{C} & A_e\end{bmatrix}\phi_k \\ &\quad + \begin{bmatrix}\bar{B}\\ 0\end{bmatrix}\mathfrak{u}_k + \begin{bmatrix}0\\ G_e\end{bmatrix}\hat{e}_k \qquad , \\ &= A\phi_k + B\mathfrak{u}_k + G\hat{e}_k \\ \hat{y}_k &= \begin{bmatrix}0 & C_e\end{bmatrix}\phi_k = C\phi_k\end{aligned} \tag{6}$$

where $\mathfrak{u}_k$ is the input. The block diagram of the entire proposed RNNinv+LME framework is schematically shown in Fig. 2.

B. State Estimator

To estimate the state ϕ_k, a Kalman filter is used and in each estimation the error can be computed with the predicted and measured output. The state estimator of LME based on Kalman filter is as follows [22]:

Step 1. Initialize initial state $\hat{\phi}_0^-$, prediction error variance $\hat{P}_0^-$.

Step 2. Update Kalman gain: $K_k = P_k^- C(CP_k^- C^T + R)^{-1}$ where R is the variance of measurement noise.

Step 3. Update state estimate $\hat{\phi}_k = \hat{\phi}_k^- + K_k(\bar{y}_k - C\hat{\phi}_k^-)$, $\hat{e}_k = \bar{y}_k - C\hat{\phi}_k^-$ where $\bar{y}_k$ is the measured output.

Step 4. Error variance $P_k = (I - K_k - C)P_k^-$.

Step 5. State prediction: $\hat{\phi}_{k+1}^- = A\hat{\phi}_k + B\mathfrak{u}_k + G\hat{e}_k$, $P_{k+1}^- = (A - GC)P_k(A - GC)^T + Q$, where Q is the variance of process noise.

C. Predictive Control

With the PEA low frequency dynamics modeled by the LME, we design a predictive controller based on the plant model represented by Eq. (6) for output tracking. Given the current state ϕ_k and input $\mathfrak{u}_k$, the outputs of the system $\mathbb{H}$ in the future N_p steps can be predicted as

$$\bar{Y}^p = G_p\phi_k + HV\mathfrak{U}_p + F\mathfrak{u}_k, \tag{7}$$

with

$$\bar{Y}^p = \begin{bmatrix}\hat{y}_{k+1}\\ \hat{y}_{k+2}\\ \vdots\\ \hat{y}_{k+N_p}\end{bmatrix}_{N_p\times 1},\ G_p = \begin{bmatrix}CA\\ CA^2\\ \dots\\ CA^{N_p}\end{bmatrix}_{N_p\times 1},\ \mathfrak{U}_p = \begin{bmatrix}\mathfrak{u}_{k+1}\\ \mathfrak{u}_{k+2}\\ \vdots\\ \mathfrak{u}_{k+N_c}\end{bmatrix}_{N_c\times 1}$$

$$H = \begin{bmatrix}0 & 0 & \dots & 0\\ CB & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ CA^{N_p-2}B & CA^{N_p-3}B & \dots & 0\end{bmatrix}_{N_p\times N_p}$$

$$V = \begin{bmatrix}I_{N_c\times N_c}\\ \begin{bmatrix}0 & \dots & 0 & 1\\ \vdots & \ddots & \vdots & \vdots\\ 0 & \dots & 0 & 1\end{bmatrix}_{(N_p-N_c)\times N_c}\end{bmatrix},\ F = \begin{bmatrix}CB\\ CAB\\ \vdots\\ CA^{N_p-1}B\end{bmatrix}_{N_p\times 1}$$

where N_c is the control horizon satisfying $1 \le N_c \le N_p$, $I_{N_c\times N_c}$ is an identity matrix. The future inputs $\mathfrak{U}_p$ will be computed in each sample time through minimizing the cost function below,

$$\begin{aligned}J &= (\bar{Y}^p - R_p)^T(\bar{Y}^p - R_p) + \rho\mathfrak{U}_p^T\Gamma^T\Gamma\mathfrak{U}_p \\ &= \mathfrak{U}_p^T(\rho\Gamma^T\Gamma + V^TH^THV)\mathfrak{U}_p + 2U_p^TV^TH^TE + E^TE\end{aligned} \tag{8}$$

with

$$\Gamma = \begin{bmatrix}1 & -1 & \dots & 0\\ 0 & 1 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & 0\end{bmatrix}.$$

In Eq. (8), $R_p = [r_{k+1}, r_{k+2}, \cdots, r_{k+N_c}, \cdots, r_{k+N_c}]^T$ is the reference signal, ρ is the weighting coefficient and $E = G_p\phi_k + F\mathfrak{u}_k - R_p$. Hence, the optimization problem can be written as

$$\begin{aligned}\min_{\mathfrak{U}_p}\quad & J = \frac{1}{2}\mathfrak{U}_p^T\Psi\mathfrak{U}_p + g^T\mathfrak{U}_p \\ \text{s.t.}\quad & |\mathfrak{U}_p| \le \bar{\mathfrak{U}}_p \\ & \Psi = \rho\Gamma^T\Gamma + V^TH^THV \\ & g = V^TH^T(G\phi_k + F\mathfrak{u}_k - R_p)\end{aligned} \tag{9}$$

where $\bar{\mathfrak{U}}_p$ is a constant vector.

IV. Experiment Results and Discussion

The proposed RNNinv+LME framework was implemented on a piezo actuator (Nano-OP30, Mad City Labs) with the maximum displacement of 30μm to track various trajectories and the results were compared with that of a PID feedback controller. Also, to further evaluate the accuracy of the proposed method, the tracking results were compared with one ILC approach—MIIFC—to demonstrate that the proposed real-time control approach was able to reach the control precision as high as that of the off-line control technique [8].

The experiment setup is shown in Fig. 3. All the signals were acquired by the data acquisition system (NI PCIe-6353, National Instruments), which was installed in the workstation (Intel Xeon W-2125, RAM 32GB). The controller was designed in MATLAB Simulink (MathWorks, Inc.) environment. The sampling frequency in the experiment was set to 10kHz.

A. Obtain the RNNinv Model

To generate $Y_{(ts)}$, 100 points in the f-A plane were computed using k-means algorithms [21]. Before implementing the algorithm, 5000 points were randomly generated to cover the space in the given range (frequency: 0-350Hz, amplitude: 0-4.5V). Then the generated $Y_{(ts)}$ was set as the desired trajectory to be tracked by the PEA through MIIFC, and the corresponding $U_{(ts)}$ (i.e., the converged drive input to the PEA calculated by MIIFC) was obtained. A 9^{th} order RNNinv was trained with the generated $(Y_{(ts)}, U_{(ts)})$ by solving the optimization problem in Eq. (2), which has been extensively investigated by the neural network community.

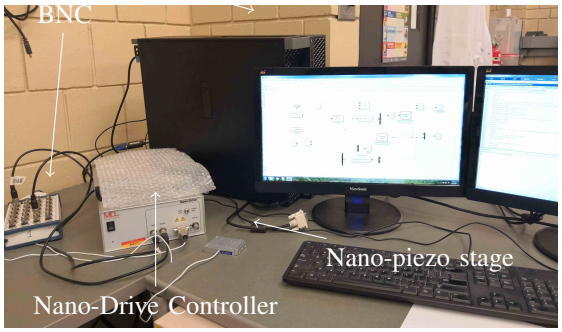

Fig. 3: Experimental setup.

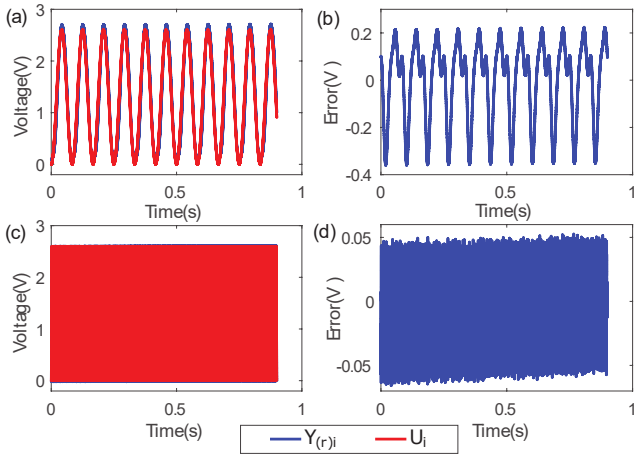

Fig. 4: (a) Comparison of RNNinv output $Y_{(r)1}$ for 12Hz signal and desired output U_1, (b) modeling error. (c) Comparison of RNNinv output $Y_{(r)3}$ for 200Hz signal and desired output U_3, (d) modeling error.

B. Evaluating the Modeling Accuracy of RNNinv

To evaluate the modeling accuracy of the RNNinv, four signals U_i including sinusoidal trajectories (with frequencies of 12Hz, 120Hz and 200Hz, and amplitude of 2.5V) and Γ_1 represented by

$$\Gamma_1(t) = [0.8\sin(2\pi 5t + 1.5\pi) + 0.43\sin(2\pi 50t) + 0.12\sin(2\pi 120t + 1.2\pi) + 0.3\sin(2\pi 180t + \pi)]/1.3 \quad (10)$$

were used to drive the piezo actuator, respectively. Thus four time series pairs, i.e., (U_i, Y_i), $i = 1,2,3,4$ were obtained, where Y_is are the corresponding PEA outputs, respectively. To evaluate the the modeling accuracy, the RNNinv output $Y_{(r)i}$ subject to each input Y_i was compared with U_i (the difference is called modeling error below), respectively.

As an example, Fig. 4 shows the comparison results for sinusoidal signal with frequencies of 12Hz and 200Hz, respectively. It can be seen that the modeling error for 12Hz signal is about four times larger than that of 200Hz. In Table I, the standard deviations for all the four cases are presented, which shows that the RNNinv modeling accuracy is very high for all cases except the lowest frequency one. Since $\Gamma_1(t)$ contains both the high frequency and low frequency signals, it is not surprising that the modeling error is in between of the other cases. Thus, it is clear that when the length of the training set is limited for the concern of computation efficiency, the RNNinv may not accurately capture the PEA low frequency dynamics. To improve the modeling accuracy, we can either use more complex RNN (thus more parameters) and/or include more low frequency sinusoidal signals in $Y_{(ts)}$, both of which will greatly increase the computation burden in the training process. Therefore, using a second order LME to model the PEA low frequency dynamics becomes a much better alternative considering the linear model is quite computationally efficient.

In addition, the effectiveness of the RNN in eliminating system nonlinearity such as hysteresis was verified. The hysteresis curves (see Fig. 5) were measured using different sinusoidal drive voltages with the frequencies of 30Hz, 120Hz and 200Hz, respectively, for the original PEA system and RNNinv cascaded with the PEA. The displacement range generated was about 35% of the total PEA maximum displacement. It can be seen that the PEA hysteresis is both rate-dependent and amplitude-dependent. However, by cascading the proposed RNNinv, hysteresis at all measured frequencies and amplitudes were effectively removed. Therefore, the PEA hysteresis nonlinearity was effectively accounted for by the RNNinv.

C. Tracking Performance Comparison

In this section, the control performance of the proposed RNNinv+LME framework is demonstrated. The desired PEA trajectories used were sinusoidal signals (with the frequencies of 30Hz, 100Hz 200Hz, and amplitude of 6μm) and $\Gamma = 3\Gamma_1$. First, the proposed method is compared with that only using the RNNinv to show the necessity of the LME and the predictive controller. Then it is compared with a PID feedback controller. Since MIIFC has been proven in achieving high precision PEA control, it is chosen as the benchmark to evaluate the accuracy of the proposed framework.

For the predictive controller, N_p and N_c were chosen to be 40 and 20, respectively. The tracking errors E_{rms} and E_{max} were computed as that in [8]. Table II shows tracking errors for all the approaches. Figs. 6-8 show the tracking performance in time domain for the desired trajectories of 100Hz sinusoidal signal and Γ, respectively.

RNNinv+LME vs. RNNinv When tracking trajectories contained low frequency dynamics, RNNinv+LME outperformed RNNinv. For example, in Table II, the tracking errors of RNNinv+LME for 30Hz signal and Γ signal are less that 50% of those of RNNinv. This is because LME can model the low frequency dynamics and the error term could also eliminates the tracking error contributed by the low frequency part, which is clearly shown in Fig. 6(d): part of the low frequency error was removed by RNNinv+LME.

TABLE I: Standard deviations of modeling error for different signals.

Signal	12Hz	120Hz	200Hz	$\Gamma_1(t)$
Std (V)	0.171	0.023	0.022	0.06

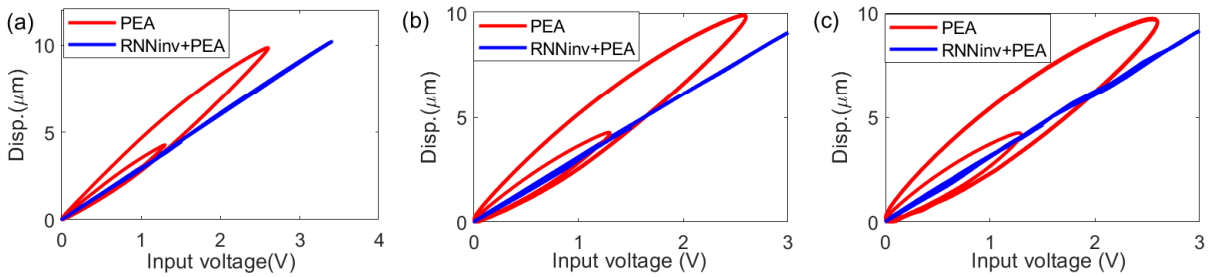

Fig. 5: Measured displacement vs. input voltage curves at the frequency of (a) 30Hz, (b) 120Hz, (c) 200Hz to the PEA alone and the RNNinv+PEA system, respectively.

However, since LME cannot handle the high frequency dynamics, it did not improve the tracking performance when tracking 100Hz and 200Hz signals as can be seen from Table II and Fig. 6(b). Furthermore, the LME could negatively influence the control performance at high frequencies. In Table II, the tracking error of RNNinv for 200Hz signal is very low, which indicates that the RNNinv could model the high frequency inversion dynamics accurately. Thus the tracking errors of RNNinv+LME in this case might be induced by the LME: tracking errors of RNNinv+LME are about three times larger than that of RNNinv for 200Hz signal. Therefore, by incorporating LME, the overall control performance is still improved, especially for tasks covering a large frequency span with somewhat downgrading in high frequency part.

TABLE II: Tracking performance comparison of RNNinv+LME, RNNinv, PID, and MIIFC.

Ref.	Error	RNNinv+LME	RNNinv	PID	MIIFC
30Hz	$E_{rms}(\%)$	0.93	2.41	2.69	2.01
	$E_{max}(\%)$	0.43	1.68	2.29	1.21
100Hz	$E_{rms}(\%)$	1.71	1.64	9.19	1.81
	$E_{max}(\%)$	1.29	1.19	7.93	1.18
200Hz	$E_{rms}(\%)$	4.12	1.37	19.29	1.69
	$E_{max}(\%)$	3.27	0.54	16.61	1.11
Γ	$E_{rmx}(\%)$	2.14	3.92	4.81	2.23
	$E_{max}(\%)$	1.06	1.28	3.30	1.16

RNNinv+LME vs. PID PID as one popular real-time tracking technique works for low frequency tasks but as the frequency increases the control performance gets worse. According to Table II, RNNinv+LME decreased the tracking errors by at least 50% for all the cases compared to PID. The performance difference is more obvious as the frequency increases. Moreover, the tracking errors in time domain in Fig. 7 verifies the superiority of the proposed method over PID.

RNNinv+LME vs. MIIFC For the MIIFC, the converged tracking results were chosen. From Table II, it can be seen that RNNinv+LME outperformed MIIFC when tracking low frequency trajectories—the tracking error is about half of that of MIIFC for 30Hz signal, and has comparable accuracy with

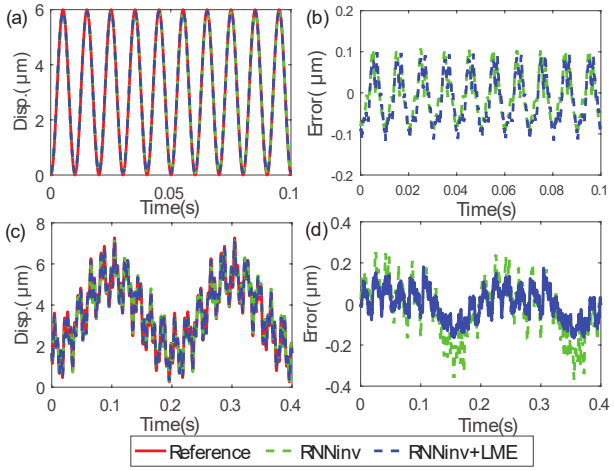

Fig. 6: (a) Comparison of the tracking results for 100Hz sinusoidal signal using RNNinv and RNNinv+LME, (b) the corresponding tracking error. (c) Comparison of the tracking results for Γ signal using RNNinv and RNNinv+LME, and (d) the corresponding tracking error.

MIIFC at "middle" frequency range, i.e., tracking 100Hz signal and Γ signal, which is more straightforward in Fig. 8. Although there is a surge in the tracking error for high frequency trajectory tracking (i.e., 200Hz), RNNinv is still better than MIIFC. Overall, even compared to the MIIFC—an off-line control approach, the proposed method can still achieve similar or even better control accuracy in real-time.

Note that although the RNNinv can model the nonlinearities of the PEA system, it does not assume any nonlinearities of the system in advance, thus it is expected to model any nonlinearities with enough parameters in theory. On the other hand, the LME overcomes the issue caused by the limited length of the training set. Therefore, the RNNinv+LME framework is expected to have broader application in output tracking of other systems. As for future work, the stability of the RNNinv will be investigated.

V. Conclusion

In this paper, we integrated RNNinv and LME to realize precision control of the PEAs. The RNNinv accounts for the high frequency dynamics and nonlinearities of the system. To remedy the inaccuracy in low frequency dynamics modeling, LME-based predictive control is used to improve the

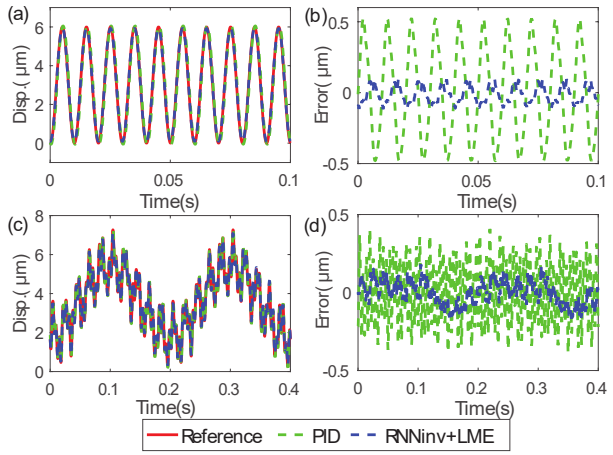

Fig. 7: (a) Comparison of the tracking results for 100Hz sinusoidal signal using PID and RNNinv+LME, (b) the corresponding tracking error. (c) Comparison of the tracking results for Γ signal using PID and RNNinv+LME, and (d) the corresponding tracking error.

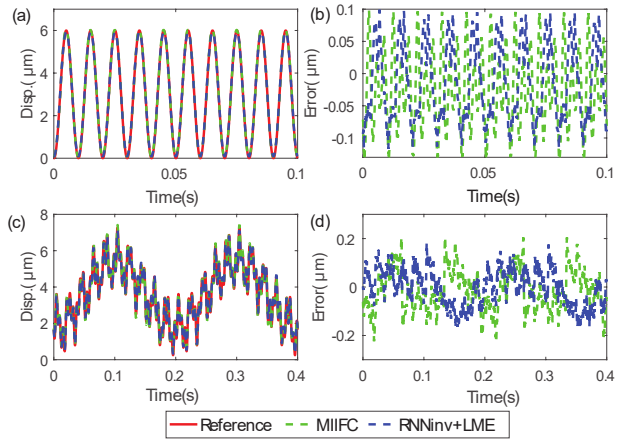

Fig. 8: (a) Comparison of the tracking results for 100Hz sinusoidal signal using MIIFC and RNNinv+LME, (b) the corresponding tracking error. (c) Comparison of the tracking results for Γ signal using MIIFC and RNNinv+LME, and (d) the corresponding tracking error.

tracking accuracy for low frequency tasks. The experiment results have demonstrated the effectiveness of the proposed method. Although the LME introduced tracking error for high frequency trajectory tracking, the overall performance is improved. It is expected the RNNinv+LME framework for output tracking could be extended to solving output tracking problem of other systems with more complicated dynamics.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) (CMMI-1634592 and CMMI-1751503) and Iowa State University.

REFERENCES

[1] Z. Xu, K. Kim, Q. Zou, and P. Shrotriya, "Broadband measurement of rate-dependent viscoelasticity at nanoscale using scanning probe microscope: Poly (dimethylsiloxane) example," *Applied Physics Letters*, vol. 93, no. 13, p. 133103, 2008.

[2] J. Ren and Q. Zou, "High-speed adaptive contact-mode atomic force microscopy imaging with near-minimum-force," *Review of Scientific Instruments*, vol. 85, no. 7, p. 073706, 2014.

[3] Y. Tian, D. Zhang, and B. Shirinzadeh, "Dynamic modelling of a flexure-based mechanism for ultra-precision grinding operation," *Precision Engineering*, vol. 35, no. 4, pp. 554–565, 2011.

[4] P. Liu, P. Yan, and H. Özbay, "Design and trajectory tracking control of a piezoelectric nano-manipulator with actuator saturations," *Mechanical Systems and Signal Processing*, vol. 111, pp. 529–544, 2018.

[5] F. Qin, D. Zhang, D. Xing, D. Xu, and J. Li, "Laser beam pointing control with piezoelectric actuator model learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[6] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed spm," *Journal of dynamic systems, measurement, and control*, vol. 131, no. 6, p. 061101, 2009.

[7] S. Tien, Q. Zou, and S. Devasia, "Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed afm operation," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 6, pp. 921–931, 2005.

[8] K.-S. Kim and Q. Zou, "A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1767–1777, 2013.

[9] C.-Y. Lin and Y.-C. Liu, "Precision tracking control and constraint handling of mechatronic servo systems using model predictive control," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 4, pp. 593–605, 2012.

[10] Y. Shan and K. K. Leang, "Dual-stage repetitive control with prandtl–ishlinskii hysteresis inversion for piezo-based nanopositioning," *Mechatronics*, vol. 22, no. 3, pp. 271–281, 2012.

[11] Y. Shan and K. K. Leang, "Accounting for hysteresis in repetitive control design: Nanopositioning example," *Automatica*, vol. 48, no. 8, pp. 1751–1758, 2012.

[12] B. Altın, J. Willems, T. Oomen, and K. Barton, "Iterative learning control of iteration-varying systems via robust update laws with experimental implementation," *Control Engineering Practice*, vol. 62, pp. 36–45, 2017.

[13] S. Xie and J. Ren, "Note: Precision control of nano-positioning stage: An iterative learning-based model predictive control approach," *Review of Scientific Instruments*, vol. 89, no. 7, p. 076103, 2018.

[14] S. Xie and J. Ren, "High-speed afm imaging via iterative learning-based model predictive control," *Mechatronics*, vol. 57, pp. 86–94, 2019.

[15] Y. Cao, L. Cheng, X. Chen, and J. Peng, "An inversion-based model predictive control with an integral-of-error state variable for piezoelectric actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 895–904, 2013.

[16] M. Edardar, X. Tan, and H. K. Khalil, "Sliding-mode tracking control of piezo-actuated nanopositioners," in *2012 American Control Conference (ACC)*, pp. 3825–3830, IEEE, 2012.

[17] M. Al Janaideh, S. Rakheja, and C.-Y. Su, "An analytical generalized prandtl–ishlinskii model inversion for hysteresis compensation in micropositioning control," *IEEE/ASME Transactions on mechatronics*, vol. 16, no. 4, pp. 734–744, 2011.

[18] L. Cheng, W. Liu, Z.-G. Hou, J. Yu, and M. Tan, "Neural-network-based nonlinear model predictive control for piezoelectric actuators," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7717–7727, 2015.

[19] W. Liu, L. Cheng, Z.-G. Hou, J. Yu, and M. Tan, "An inversion-free predictive controller for piezoelectric actuators based on a dynamic linearized neural network model," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 1, pp. 214–226, 2016.

[20] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[21] S. Xie and J. Ren, "Recurrent-neural-network-based predictive control of piezo actuators for precision trajectory tracking," in *2019 Annual American Control Conference (ACC)*, pp. 3795–3800, IEEE, 2019.

[22] R. G. Brown, P. Y. Hwang, *et al.*, *Introduction to random signals and applied Kalman filtering*, vol. 3. Wiley New York, 1992.