LSTM-Based Model Predictive Control of Piezoelectric Motion Stages for High-speed Autofocus

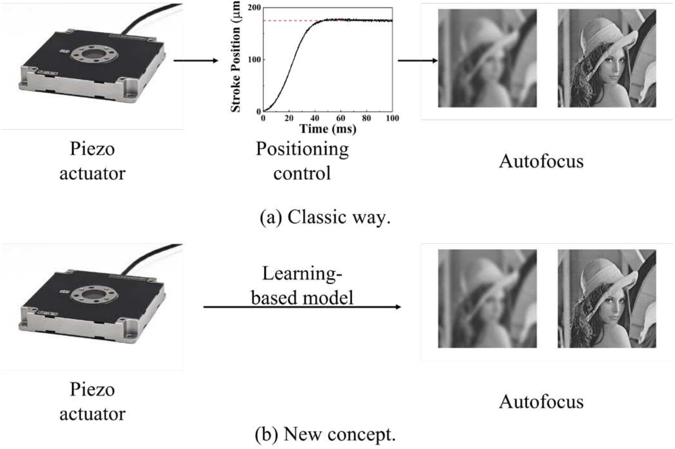
Jingyang Yan, Peter DiMeo, Lu Sun, and Xian Du, *Member, IEEE*

***Abstract*—In this paper, we proposed a neural network-based model predictive control of piezoelectric motion stages for autofocus. Rather than using an internal controller to account for the problematic hysteresis effects of the piezoelectric motion stage, we use the long short-term memory unit to integrate the hysteresis effects and the focus measurement into a single learning-based model. Subsequently, a model predictive control method is developed based on this long short-term memory model that successfully finds the optimal focus position using a series of focus measurements derived from a sequence of images. To further improve the speed of the long short-term based model predictive control, an optimized backpropagation algorithm is proposed that optimizes the model predictive control cost function. Experiments verified our proposed method reduces at minimum 30% regarding autofocus time when compared to well-known ruled-based autofocus methods and other learning-based methods. Videos of the experiments are available at https://youtu.be/AyvnMlq48Vc and https://youtu.be/SQN3ETbuf2g.**

***Index Terms*—Autofocus, model predictive control, neural network modeling, piezo actuator.**

I. Introduction

In-line inspection plays a critical role in the quality control of the roll-to-roll (R2R) flexible electronics printing process [1]. Due to the vibrations of the moving substrates and the micro- and nanoscale size of the printed patterns, real-time imaging of the flexible electronics requires high-speed autofocus (AF). Our preliminary study of AF has shown that a simple step motor-based AF system cannot meet the high-speed requirements due to the slow motor motion control where the step motor movement consumes more than 77% of the overall AF time [2]. Therefore, upgraded hardware is necessary to achieve real-time AF.



(a) Classic way.

(b) New concept.

Fig. 1. Proposed high speed AF (b) compared with the classic AF (a).

Thanks to the discovery of the piezoelectric effect in 1880 by Pierre and Paul-Jacques Curie, the piezoelectric motion has enabled a variety of high-precision positioning technologies. Piezoelectric actuators (PEAs) can achieve high resolution and precision at micro and nanometric scales with very fast responses and bandwidth ranges in the order of kHz [3]. As a result, they are an excellent option for AF applications [4]. In fact, PEAs, or similarly piezoelectric motion stages (PEMSs), are the key devices in micro-optical positioning systems such as the scanning electron microscope, atomic force microscopy, compact two-dimensional single-mirror laser scanner, and the micro focusing mechanism [5].

Despite PEMS being at the forefront of precision, their performance is burdened by nonlinearities which make their applications in AF a challenge. One dominant nonlinear characteristic in the dynamics of PEMSs is hysteresis. This phenomenon is responsible for making the corresponding displacement curves in the voltage lift and return intervals not overlap [6]. The hysteresis seen in piezoelectric materials significantly compromises the control accuracy of PEAs or PEMSs, thus greatly limiting their applications in micro-displacement technology [6]. Thus, traditional PEMS applications for AF first employ an internal controller that overcomes the hysteresis and achieves deterministic

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported in part by NSF Foundation under Grant CMMI-1916866 and CMMI-1942185. The first two authors contributed equally to this work. (*Corresponding author: Xian Du.*)

Jingyang Yan, Peter DiMeo, and Xian du are with the Mechanical and Industrial Engineering Department, Institute for Applied Life Sciences, University of Massachusetts, Amherst, MA, 01003, USA (e-mail: jingyangyan@umass.edu; pdimeo@umass.edu; xiandu@umass.edu).

Lu Sun is with the Department of Electrical and Computer Engineering, Institute for Applied Life Sciences, University of Massachusetts, Amherst, MA, 01003, USA (e-mail: lsun@umass.edu).

positioning control. Subsequently, the PEMS is applied with the embedded controller for further AF applications as conceptually illustrated in Fig. 1(a). The conventional way of designing the internal controller is to automatically adjust the input by utilizing some form of closed-loop control [7]. However, precise closed-loop positioning of a PEMS is far slower than open-loop alternatives and comes with the additional cost of fast and precise displacement sensors for position feedback [8]. For these reasons, open-loop control techniques are preferred.

To cope with the hysteresis effect when using an open-loop control technique, a model is first selected to describe the hysteresis, and then a feedforward controller is designed accordingly to compensate for the hysteresis effect. Although the open-loop feedforward control saves the cost of installing displacement sensors, the design of the feedforward controller requires the inverse of the hysteresis model which is computationally non-trivial. Meanwhile, even though open-loop feedforward control is highly sought after for PEMS positioning control, the nonlinearity of the hysteresis makes it difficult to obtain an accurate invertible physical PEMS model [8], preventing the controller from providing accurate inputs to the PEMS.

Recently, neural-network (NN)-based model predictive control (MPC) emerged with high performance in nonlinear precision motion control fields. This inspired us with the alternative PEMS AF control solution, which is to integrate the defocus measurements and the hysteresis effects of the PEMS accurately in a single learning-based model, as shown in Fig. 1(b). This solution avoids the need for a slower internal position controller before AF control as shown in Fig. 1(a). Our solution is based on the hypothesis that the nonlinear dynamics information, including the hysteresis, is embedded in the sequence of focus/defocus images during the AF process. Therefore, we can learn or infer the dynamic non-linearities and the AF process using the image data from both offline and in situ measurements in an NN-based MPC architecture. The input data to the NN includes focus scores extracted from sequential focus/defocus images. We declare that from the sequential focus scores, we can infer the state of the focus or defocus and ascertain the motion direction and steps needed to bring the lens to the optimal focus position. For PEMS, since most of its non-linearities result from the hysteresis, i.e., the dependence of the state of a system on its history, it is then appropriate to use long short-term memory (LSTM) to model PEMS. The LSTM architecture retains the state information from previous observations thus allowing it to infer the non-linear dynamics of the PEMS. To further improve the speed of the LSTM-based MPC control loop while maintaining sufficient accuracy, we propose an optimized LSTM backpropagation algorithm that reduces the number of redundant calculations during the AF process. Moreover, to cope with the vibrations from the rapid PEMS impulse inputs, a unique S-curve control input profile is designed to reduce the acceleration profile and promote a more stable transition in dynamics and ultimately improve the accuracy of our LSTM-based MPC AF control scheme. Finally, experiments are carried out to test the proposed AF method.

The main contributions of this paper are as follows.

1) Open-loop control of a PEMS using LSTM to identify the system dynamics.
2) LSTM deep learning and MPC for AF using a sequence of focus scores to predict the optimal focus position.
3) Optimized LSTM backpropagation algorithms.
4) S-curve profile for PEMS input to reduce piezoelectric element jitters.
5) Autofocus control loop rate of 40 Hz.

The remainder of the article is organized as follows. In Section II, related work about PEMS control and AF is reviewed. In Section III, the proposed LSTM-based MPC of PEMS for AF is elaborated upon. In Section IV, the proposed method is evaluated by comparing its performance to that of other published AF methods. In Section V, the conclusion and future work are summarized.

II. Related Work

A. Piezoelectric Motion Stage Control

Like many ferromagnetic and ferroelectric devices, PEMSs exhibit hysteresis. Closed-loop controllers have been introduced to remove these issues by involving another loop that acquires position information, essentially bridging the gap between the input voltage and absolute position. For example, in [11], a robust H_∞ controller was designed to control a large-range nano-positioner actuated by PEAs. In [12], an adaptive fuzzy sliding mode control was developed for piezo-actuated flexure-based micro/nano manipulators. Recently, several NN-based closed-loop control algorithms have also been developed for positioning control of PEAs [13], [14]. Overall, closed-loop control systems are necessary if nm or μm tolerances are required for high precision positioning.

On the other hand, open-loop controllers offer faster rise times and sharper impulse responses for more rapid transitions in dynamics. The most popular approach of open-loop control design for PEMS is the inversion-based feedforward control. In literature, several physics-based models have been proposed to learn the hysteresis effect of PEAs, including the Preisach model [15], the Prandtl-Ishlinskii model [16], etc. However, the calculation of the inverse hysteresis model is either time-consuming or lacks accuracy.

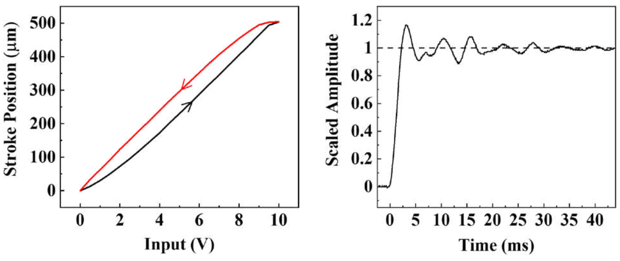
In general, traditional PEMS applications need to first solve the issue of hysteresis with some kind of controller and then apply the PEMS with the embedded controller for the specific application. For applications of PEMS in AF as examples, in [17], the authors proposed an AF method for automated microscopy using a piezoelectric transducer. The piezoelectric transducer used in this method is controlled by an amplifier with built-in servo feedback control for positioning control. The built-in feedback control is part of the hardware system. Meanwhile, the autofocus control algorithm is running on a computer. Therefore, there are two controllers in the system, including a built-in feedback controller for the positioning control of the piezoelectric actuator and a controller for autofocus; in [18], the authors developed a piezoelectric actuator driver integrated circuit (IC) for autofocus in camera module of mobile phone. An embedded self-positioning feedback control algorithm is running in this IC. However, another controller is still needed to run the AF algorithm. In this paper, we propose an end-to-end control algorithm for PEMS which uses LSTM to directly link the open-loop PEMS input voltage to the response variable, and subsequently uses a single model-predictive controller for complete system control. Using an LSTM-based MPC scheme has the potential to reduce the complexity of the control system while still providing fast and accurate responses to the PEMS.

Another difficulty in PEMS applications is the unwanted vibrations of the piezoelectric elements caused by the impulse inputs. In literature, several solutions have been proposed to eliminate the vibrations. In [19], the vibrations are modeled as a linear dynamical model and then a feedforward controller is designed accordingly to compensate for the vibrations. In [8], the vibrations are compensated by applying the Zero Vibration input shaping technique. In this paper, we introduce a simple S-curve input profile capable of reducing the piezoelectric jitter that follows a step input (refer to Section IV.A for the detailed description).

B. Autofocus

Classic AF methods can be broadly divided into two categories, active and passive. Active AF methods use a sensor (e.g., laser) to measure the distance between the camera and target and subsequently adjust the optical system to correct the focus [20]. Active AF methods attempt to remove limitations surrounding focus accuracy that can arise with varying image content (e.g., low-contrast imaging or low-light conditions) [21]. Active AF methods offer superior performance in stationary imaging scenarios, such as microscopy, because the subject always resides in a position where its distance to the camera can be repeatedly measured. If the distance of the subject is unknown or unable to be measured, passive methods are employed. Rather than using additional sensors, the objective of the classic passive AF processes is to autonomously navigate from a defocused image to a focused one by employing various rule-based search algorithms to maximize the measured focus. Within the passive AF category, the phase-based and deep learning-based methods are regarded as state-of-the-art [2]. Phase-based methods are employed on specially designed camera sensors where each pixel consists of two independent photodiodes. Rule-based control methods or deep learning algorithms [22] are implemented to estimate subject depth from the difference in phase of the stereo images, thus enabling fast autofocus or even detailed depth maps for facial recognition [23]. Phase-based methods undoubtedly offer superior AF to traditional passive methods regarding speed; however, these methods can significantly increase the cost and complexity of the image acquisition system (IAS) and thus will not be explored in this paper.

Inspired by the successful applications of deep learning in computer vision tasks, such as image classification, image segmentation, and image super-resolution, various training-based AF methods have been proposed to outperform the phase-based AF methods. In [24], a convolution LSTM (ConvLSTM) framework with a feature detector, object detector, and focus-step predictor was proposed that had the capability of focus tracking. In [25], a convolutional neural network (CNN)-based step estimator and focus discriminator was proposed that was capable of single-shot AF. In [26], a passive AF control system was developed using a trained self-organizing map (SOM) neural network to predict the optimal lens position using the focus score from three initial images.



(a) (b)

Fig. 2. (a)The hysteresis effect. (b) The jitters.

In this paper, we propose the use of LSTM to link the open-loop PEMS input voltage to the level of focus. Rather than using a single image as input to the LSTM, we use a sequence of focus scores as the inputs. There are two advantages of designing the inputs in this way: the first being that the sequential inputs could capture the high-order and delayed-response modes of interaction, and the second using the focus scores as the inputs could remove any image-based dependencies.

III. LSTM-Based Model Predictive Control of Piezoelectric Motion Stages for Autofocus

A. Problem Definition

In this work, we aim to use a learning-based method to directly link the PEMS inputs to the output focus score. Specifically, the control input to the system is denoted as $u(t)$ which defines the input voltage applied to the PEMS at time t. The response variable of the system is a scalar, $y(t)$, which describes the focus measurement of the image. As we aim to achieve high-speed autofocus, the priority when selecting the focusing function is the computational complexity; the accuracy falls second. We tested 28 different focus measure functions and have selected the maximum absolute gradient (MAG) method that has the fastest calculation speed of 6 ms. The MAG method used to calculate the FM score F is defined as

$$F = \frac{1}{NM}\sum_{x=1}^{M}\sum_{y=1}^{N}\max\left(|I_x|, |I_y|, |I_d|\right), \tag{1}$$

where $I(x,y)$ is the image gray value at the pixel coordinate (x,y) in an image of size $M \times N$, I_x, I_y, and I_d are respectively the gradients in the x, y, and diagonal direction defined respectively by $I_x = I(x,y) - I(x+1,y)$, $I_y = I(x,y) - I(x,y+1)$, and $I_d = I(x,y) - I(x+1,y+1)$. Although some functions based on image differentiation are faster than the MAG, MAG has a better performance in removing the inappropriate information in the image as it considers not only the horizontal and vertical gradients but also the diagonal gradient [27].

The non-linearity introduces both position uncertainty and piezoelectric jitter, which leads to large errors in the AF actuator movements. To elaborate, the total stroke of a nano-resolution PEMS (e.g., Piezosystem Jena Inc. nanoSX 400 S CAP) under open-loop control is 503.9 μm with maximum hysteresis of 60.03 μm as shown in Fig. 2(a). This introduces a potential 12% position uncertainty to the medium which controls a high-mass IAS within a depth of field (DOF) range of 3.2 μm. Because the uncertainty is approximately 15-20 times larger than the DOF, AF accuracy of industrial standards in vanilla open-loop control is virtually unattainable. Moreover, the residual vibrations of the piezoelectric elements lengthen the settling time of the open-loop response to greater than 40 ms (Fig. 2(b)), thus removing any possibility of real-time optical metrology. As a result, there exists an unmet need for a solution to the hysteresis and vibration non-linearity of PEMSs to enable fast and accurate optical-positioning-based metrology techniques.

B. Structure of LSTM

There are various deep learning models with different structures to model an unknown non-linear system. To model the inputs to the PEMS and the output focus scores, as most of the non-linearities result from the hysteresis, i.e., the dependence of the state of a system on its history, it is then appropriate to use a recurrent neural network (RNN). However, the standard RNN models lack long-term dependencies of the state due to the gradient vanishing problem. Hence, we propose to use LSTM to model PEMS for AF.

In an LSTM network, the past information is stored in a memory cell and four gates are used to control the flow of information into or out of the memory cell, namely, the input gate **i**, forget gate **f**, output gate **o**, and the candidate gate **g**. Mathematically, the calculations in the LSTM model at time step t are described as follows,

$$\mathbf{i}(t) = \sigma(\mathbf{W}_{xi}\mathbf{x}(t) + \mathbf{W}_{hi}\mathbf{h}(t-1) + \mathbf{b}_i), \tag{2.a}$$

$$\mathbf{f}(t) = \sigma\big(\mathbf{W}_{xf}\mathbf{x}(t) + \mathbf{W}_{hf}\mathbf{h}(t-1) + \mathbf{b}_f\big), \tag{2.b}$$

$$\mathbf{o}(t) = \sigma(\mathbf{W}_{xo}\mathbf{x}(t) + \mathbf{W}_{ho}\mathbf{h}(t-1) + \mathbf{b}_o), \tag{2.c}$$

$$\mathbf{g}(t) = \tanh\big(\mathbf{W}_{xg}\mathbf{x}(t) + \mathbf{W}_{hg}\mathbf{h}(t-1) + \mathbf{b}_g\big), \tag{2.d}$$

$$\mathbf{c}(t) = \mathbf{f}(t) \odot \mathbf{c}(t-1) + \mathbf{i}(t) \odot \mathbf{g}(t), \tag{2.e}$$

$$\mathbf{h}(t) = \mathbf{o}(t) \odot \sigma\big(\boldsymbol{c}(t)\big), \tag{2.f}$$

$$y_p(t) = \mathbf{W}\mathbf{h}(\mathrm{t}) + b, \tag{2.g}$$

where $\mathbf{x} \in \mathbb{R}^{n_x \times 1}$ is the input vector, $y_p \in \mathbb{R}^{1\times 1}$ is the predicted focus measurement, $\mathbf{h} \in \mathbb{R}^{n_p \times 1}$ is the hidden state, $\mathbf{c} \in \mathbb{R}^{n_p \times 1}$ is the cell state, $\mathbf{W}_{xi} \in \mathbb{R}^{n_p \times n_x}$, $\mathbf{W}_{xf} \in \mathbb{R}^{n_p \times n_x}$, $\mathbf{W}_{xo} \in \mathbb{R}^{n_p \times n_x}$, and $\mathbf{W}_{xg} \in \mathbb{R}^{n_p \times n_x}$ are the weights from the inputs. Similarly, $\mathbf{W}_{hi} \in \mathbb{R}^{n_p \times n_p}$, $\mathbf{W}_{hf} \in \mathbb{R}^{n_p \times n_p}$, $\mathbf{W}_{ho} \in \mathbb{R}^{n_p \times n_p}$, and $\mathbf{W}_{hg} \in \mathbb{R}^{n_p \times n_p}$ are the weights from the hidden states. $\mathbf{b}_i \in \mathbb{R}^{n_p \times 1}$, $\mathbf{b}_f \in \mathbb{R}^{n_p \times 1}$, $\mathbf{b}_o \in \mathbb{R}^{n_p \times 1}$, and $\mathbf{b}_g \in \mathbb{R}^{n_p \times 1}$ are biases for different gate states. $\mathbf{W} \in \mathbb{R}^{1 \times n_p}$ and $b \in \mathbb{R}^{1\times 1}$ are the weight and the bias for the output layer, respectively. ⊙, σ, and tanh are respectively the element-wise multiplication operator, logistic sigmoid function, and hyperbolic tangent function. For the input features to the LSTM, we concatenate both inputs and outputs from the plant to capture the short-term input-output responses. Rather than considering only one single timestep, a sequence of input-output pairs, like $\mathbf{x}(t) = [u(t), u(t-1), \ldots u(t-n_u), y(t-1), y(t-2), \ldots y\big(t-n_y\big)]$ will be used as the input feature vector to capture the delayed-response modes of interaction, where n_u and n_y are respectively the exogenously designated timestep delays of the input and output. The output of the LSTM network is the predicted focus score $y_p(t)$ based on the current input voltage, previous input-output pairs, and the recurrent latent states. In the training process, the LSTM network can be trained by the same algorithm as the vanilla recurrent neural network called backpropagation through time (BPTT) [28].

C. LSTM-Based MPC

MPC calculates control commands by minimizing a cost function over a finite prediction horizon. The cost function is usually formulated by minimizing not only the squared error between the reference signal and the plant's output but also the weighted squared change of the control input. Given the LSTM architecture for the dynamic plant model, the MPC cost function can be formulated by

$$J_{MPC} = \sum_{j=1}^{P} \left\| y_r(t+j) - y_p(t+j) \right\|^2 + \sum_{j=1}^{M} \lambda(j) \|\Delta u(t+j)\|^2, \tag{3}$$

where P is the prediction horizon, M is the control horizon, y_r is the reference signal, $y_p(t+j)$ is the predicted output of the LSTM at the $(t+j\ )^{th}$ period based on the available measurements at the t^{th} sampling period, $\lambda(j)$ is the regulating factor for the control input that maintains the smoothness of the control signal and is typically chosen to be constant or exponential, and $\Delta u(t+j) = u(t+j) - u(t+j-1)$ denotes the change in u.

The objective of the cost function minimization algorithm is to minimize J_{MPC} in (3) with respect to $[u(t+1), u(t+2), \ldots, u(t+M)]$, denoted by $\mathbf{U}$. Since the plant model is an LSTM structure, a gradient descent method can be used to minimize the cost function iteratively. During each iteration, the intermediate value for J_{MPC} is denoted by $J_{MPC}(k)$. Meanwhile, the generated intermediate control input is denoted by $\mathbf{U}(k)$ allowing the control input to be updated according to,

$$\mathbf{U}(k+1) = \mathbf{U}(k) + \Delta\mathbf{U}(k), \tag{4}$$

$$\Delta\mathbf{U}(k) = \eta\left(-\frac{\partial J_{MPC}(k)}{\partial \mathbf{U}(k)}\right), \tag{5}$$

where $\eta > 0$ is the update rate for the control input. The Jacobian matrix is then denoted as

$$\frac{\partial J_{MPC}}{\partial \mathbf{U}}(k) = \begin{bmatrix} \frac{\partial J_{MPC}}{\partial u(t+1)} \\ \cdots \\ \frac{\partial J_{MPC}}{\partial u(t+M)} \end{bmatrix} \tag{6}$$

and the h^{th} element of the Jacobian matrix in (6) is stated as

$$\frac{\partial J_{MPC}}{\partial u(t+h)} = -2\sum_{j=1}^{P} \big[y_r(t+j) - y_p(t+j)\big] \frac{\partial y_p(t+j)}{\partial u(t+h)} + 2\sum_{j=1}^{M} \lambda(j)[\Delta u(t+j)] \frac{\partial \Delta u(t+j)}{\partial u(t+h)} \tag{7}$$

Algorithm 1: Standard BPTT

```
Input: prediction horizon $P$, control horizon $M$, input feature $\mathbf{x}(t+1)$,
       control input sequence $\mathbf{U}$
Output: Jacobian matrix $\frac{\partial y_p(t+i)}{\partial \mathbf{x}(t+j)}$ $i \in [1,P], j \in [1,i]$
Initialization: concatenated input vector to the LSTM
  input $= [], p = 1$;
while $p \leq P$ do
    input $=$ [input; $\mathbf{x}(t+p)$];
    calculate $y_p(t+p)$ through $p$ steps forward propagation;
    calculate $\frac{\partial y_p(t+p)}{\partial \mathbf{input}}$ using BPTT;
    $y(t+p) = y_p(t+p)$;
    if $p \geq M$ then
        $u(t+p) = u(t+M)$;
    end
    $\mathbf{x}(t+p+1) = [u(t+p+1), u(t+p), ..., u(t+p-n_u), y(t+p), y(t+p-1), ..., y(t+p-n_y)]$;
    $p = p+1$;
end
```

The $\frac{\partial \Delta u(t+j)}{\partial u(t+h)}$ term can be expanded and calculated in terms of the Kronecker Delta function

$$\frac{\partial \Delta u(t+j)}{\partial u(t+h)} = \frac{\partial u(t+j)}{\partial u(t+h)} - \frac{\partial u(t+j-1)}{\partial u(t+h)} = \delta(h,j) - \delta(h,j-1)$$

where $\delta(h,j) = \begin{cases} 1 \text{ if } h = j \\ 0 \text{ if } h \neq j \end{cases}$. For the $\frac{\partial y_p(t+j)}{\partial u(t+h)}$ term, as it is modeled by the LSTM, it is then feasible to calculate the derivatives through the backpropagation algorithm. The convergence of the backpropagation algorithm is highly dependent upon the iteration number and the updating rate. Generally, a fast backpropagation algorithm often provides flexibility in tuning the iteration number and the updating rate. The BPTT algorithm has been widely implemented by researchers to solve the Jacobian matrix in the training of LSTM networks. However, the vanilla BPTT algorithm is not suitable for the calculation of (6). If a vanilla BPTT algorithm is used to calculate the gradient following Algorithm 1, we can find that in the forward pass of the 'while' loops, the hidden states and the cell states are calculated multiple times, which affects the computation time in two aspects. First, the size of the hidden and cell states is incrementally increased adversely affecting performance and memory use as repeatedly changing sizes requires the system to allocate more time for seeking larger contiguous blocks of memory. Second, some elements in the hidden and cell states are calculated multiple times. Specifically, the elements in the i^{th} row of the hidden states and the cell states will be calculated $P-i+1$ times. These redundant operations slow down the gradient calculation. To expedite the gradient calculation, we have improved the vanilla BPTT method based on pre-allocation as shown in Algorithm 2 in pseudo-code of MATLAB-style. The improved BPTT algorithm is shown to save 20% of the computation time as detailed in Section IV.

Algorithm 2: The improved BPTT

```
Input: prediction horizon $P$, control horizon $M$, input feature $\mathbf{x}(t+1)$,
       control input sequence $\mathbf{U}$, input vector length $n_x$, number of
       hidden units $n_p$
Output: Jacobian matrix $\frac{\partial y_p(t+i)}{\partial \mathbf{x}(t+j)}$ $i \in [1,P], j \in [1,i]$
Initialization: input $= zeros(P, n_x)$, c_cache $= zeros(P, n_p)$,
  h_cache $= zeros(P, n_p)$, i_cache $= zeros(P, n_p)$,
  f_cache $= zeros(P, n_p)$, o_cache $= zeros(P, n_p)$,
  g_cache $= zeros(P, n_p)$;
while $p \leq P$ do
    input$(p,:) = \mathbf{x}(t+p)$;
    $\mathbf{i}(t+p) = \delta(\mathbf{W}_{xi}\mathbf{x}(t+p) + \mathbf{W}_{hi}\mathbf{h}(t+p-1) + \mathbf{b}_i)$;
    $\mathbf{f}(t+p) = \delta(\mathbf{W}_{xf}\mathbf{x}(t+p) + \mathbf{W}_{hf}\mathbf{h}(t+p-1)) + \mathbf{b}_f)$;
    $\mathbf{o}(t+p) = \delta(\mathbf{W}_{xo}\mathbf{x}(t+p) + \mathbf{W}_{ho}\mathbf{h}(t+p-1)) + \mathbf{b}_o)$;
    $\mathbf{g}(t+p) = \tanh(\mathbf{W}_{xg}\mathbf{x}(t+p) + \mathbf{W}_{hg}\mathbf{h}(t+p-1)) + \mathbf{b}_g)$;
    $\mathbf{c}(t+p) = \delta(\mathbf{f}(t+p) \odot \mathbf{c}(t+p-1) + \mathbf{i}(t+p) \odot \mathbf{g}(t+p))$;
    $\mathbf{h}(t+p) = \mathbf{o}(t+p) \odot \delta(\mathbf{c}(t+p))$;
    c_cache$(p,:) = \mathbf{c}(t+p)$;
    h_cache$(p,:) = \mathbf{h}(t+p)$;
    i_cache$(p,:) = \mathbf{i}(t+p)$;
    f_cache$(p,:) = \mathbf{f}(t+p)$;
    o_cache$(p,:) = \mathbf{g}(t+p)$;
    g_cache$(p,:) = \mathbf{o}(t+p)$;
    calculate $y_p(t+p)$ ;
    calculate $\frac{\partial y_p(t+p)}{\partial \mathbf{input}(1:p,:)}$ regarding c_cache$(1:p,:)$, h_cache$(1:p,:)$,
      i_cache$(1:p,:)$, f_cache$(1:p,:)$, o_cache$(1:p,:)$,
      g_cache$(1:p,:)$;
    $y(t+p) = y_p(t+p)$;
    if $p \geq M$ then
        $u(t+p) = u(t+M)$;
    end
    $\mathbf{x}(t+p+1) = [u(t+p+1), u(t+i), ..., u(t+p-n_u), y(t+p), y(t+p-1), ..., y(t+p-n_y)]$;
    $p = p+1$;
end
```

Algorithm 3: Calculation of $\frac{\partial y_p}{\partial \mathbf{U}}$

```
Input: prediction horizon $P$, control horizon $M$, Jacobian matrix
       $\frac{\partial y_p(t+i)}{\partial \mathbf{x}(t+j)}$ $i \in [1,P], j \in [1,i]$
Output: Jacobian matrix $\frac{\partial y_p(t+i)}{\partial \mathbf{u}(t+j)}$ $i \in [1,P], j \in [1,M]$
while $m \leq M$ do
    while $p \leq P$ do
        if $p \leq M$ then
            $\frac{\partial y_p(t+p)}{\partial \mathbf{u}(t+m)} = \sum_{k=1}^{p}(\frac{\partial y_p(t+p)}{\partial \mathbf{x}(t+k)}(p) + \sum_{l=1}^{n_y} \frac{\partial y_p(t+p)}{\partial \mathbf{x}(t+k)}(n_u+1+l) \times \frac{\partial y_p(t+k-l)}{\partial \mathbf{u}(t+m)})$;
        else
            if $m \neq M$ then
                $\frac{\partial y_p(t+p)}{\partial \mathbf{u}(t+m)} = \sum_{k=1}^{M}(\frac{\partial y_p(t+p)}{\partial \mathbf{x}(t+k)}(p) + \sum_{l=1}^{n_y} \frac{\partial y_p(t+p)}{\partial \mathbf{x}(t+k)}(n_u+1+l) \times \frac{\partial y_p(t+k-l)}{\partial \mathbf{u}(t+m)})$;
            else
                $\frac{\partial y_p(t+p)}{\partial \mathbf{u}(t+m)} = \sum_{k=1}^{M}(\sum_{n=1}^{p-M} \frac{\partial y_p(t+p)}{\partial \mathbf{x}(t+k)}(p+n) + \sum_{l=1}^{n_y} \frac{\partial y_p(t+p)}{\partial \mathbf{x}(t+k)}(n_u+1+l) \times \frac{\partial y_p(t+k-l)}{\partial \mathbf{u}(t+m)})$;
            end
        end
        $p = p+1$;
    end
    $m = m+1$;
end
```

After applying Algorithm 2, the derivatives of $\frac{\partial y_p(t+i)}{\partial \mathrm{x}(t+j)}, i \in [1,P], j \in [1,i]$ are stored in system memory and the next step is to calculate the derivatives of $\frac{\partial y_p}{\partial \mathbf{U}}$. The algorithm for calculating $\frac{\partial y_p}{\partial \mathbf{U}}$ is accomplished by applying the chain rule through time as shown in Algorithm 3. Note that as the stability analysis of the NN-based MPC scheme is well established in published literature, we will not include the stability analysis in this paper. Readers may refer to [29]–[32] for detailed stability analysis for interest.

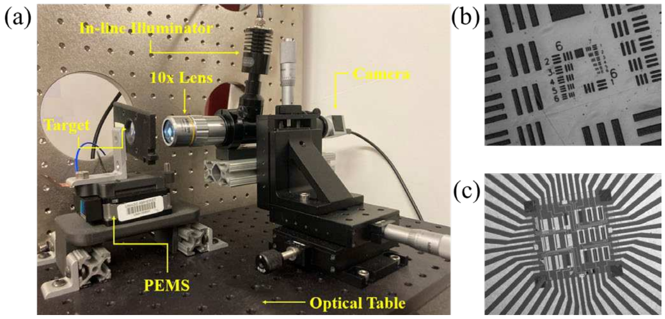


Fig. 3. (a) The image acquisition system in the experimental setup. (b) The US Air Force Hi-Resolution target. (c) The circuit target.

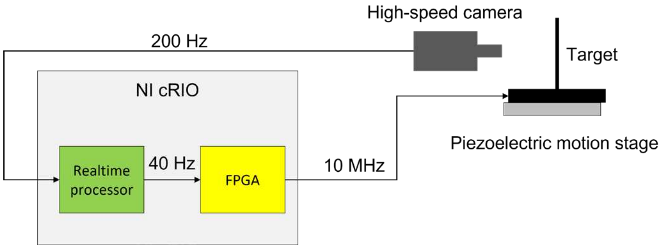


Fig. 4. Experimental schematic of the proposed AF method.

IV. Experimental Results and Discussion

A. Experimental Setup

Our experimental setup is displayed in Fig. 3 which includes the PEMS (piezosystem jena Inc. nanoSX 400 S CAP), AF target, controller, optical table, and stationary 10x lens IAS. The IAS consisted of a high-speed CMOS area-scan camera, a 3D printed adaptor, a lighting source, a coaxial in-line illuminator unit, and a high magnification factor objective lens. Fig. 4 displays the experimental control schematic which includes the high-speed camera (Basler acA1300-200um) connected to a real-time controller (CompactRIO-9049, National Instruments) via USB 3.0. This camera has a maximum resolution of 1280×1024 pixel^2 and a maximum frame rate of 200 fps for the full frame. The LSTM-based MPC algorithm is run in the real-time processor of the controller then the output command is sent to the PEMS via Field Programmable Gate Arrays (FPGA) with a communication speed of 10 MHz.

The PEMS used in our experiment has two operation modes, e.g., the closed-loop mode and the open-loop mode. When operating the PEMS using the closed-loop control mode, the rise time of a full stroke takes 50 ms compared to the open-loop control mode which takes 3 ms. Therefore, for a fast operation speed, the open-loop control mode is preferred. However, in the vanilla open-loop control mode, the S-curve acceleration profile is not followed and motor jerk is accentuated as shown in Fig. 5. To reduce the motor jerk, we introduce a method of generating S-curve motion profiles that abide by the following equation,

$$y = p_1 + (p_2 - p_1)\left(0.5 + 0.5\left(\frac{t}{\sqrt{(1.5+t^2)}}\right)\right), \tag{8}$$

$$p_1 = \frac{V_i}{0.833} - \frac{V_r}{4.992}, \tag{8a}$$

$$p_2 = \frac{V_r}{0.833} - \frac{V_i}{4.992}, \tag{8b}$$

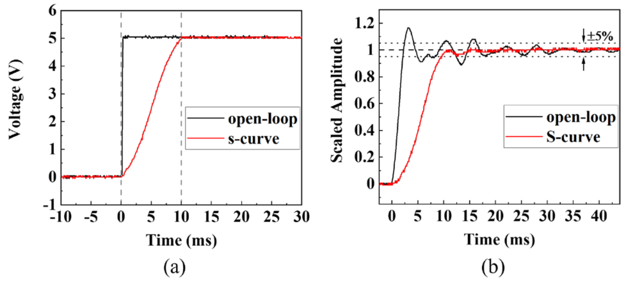


Fig. 5. (a) open-loop step input and the S-curve step input. (b) open-loop step input response and the S-curve step input response.

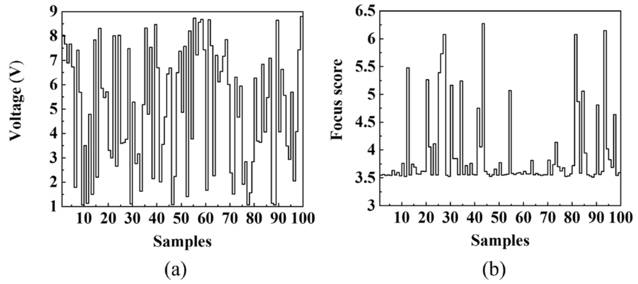


Fig. 6. (a) Rectangular pulse train examples used as the input to the PEMS. (b) Response of the PEMS due to the pulse train inputs.

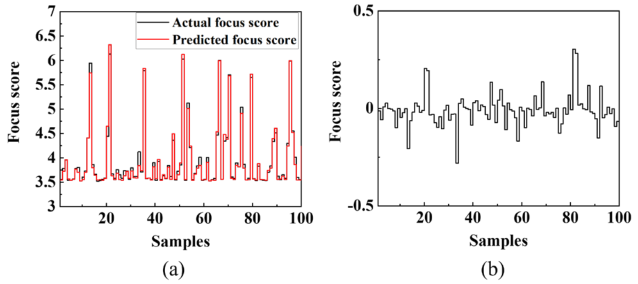


Fig. 7. (a) Actual focus scores and predicted focus scores of the testing dataset. (b) The corresponding error.

TABLE I
Hyperparameters for Training LSTM

Optimizer	Learning rate	Epoch	Batch size
Adam	0.01	1500	64

where V_i is the starting voltage, V_r is the reference voltage, t is the time window of the S-curve profile, and y is the interpolated voltage. In experiments, the time window is selected to be 10 ms. The 10 ms S-curve acceleration profile will be divided into 1000 steps at a resolution of 0.3 mV. The comparison between the open-loop mode step input response versus the S-curve step input response is shown in Fig. 5. Using the S-curve profile the settling time was reduced to 9.6 ms, which is 15 ms less than the vanilla open-loop mode regarding the traditional 5% error band.

B. Training of LSTM

The state-of-the-art literature uses images as the input to the NN [24], [25]; however, the features in each image can vary drastically by scene. For the method to be robust enough to generalize to new scenes, large datasets are needed to train the NN. With the aim of removing any image-based dependencies,

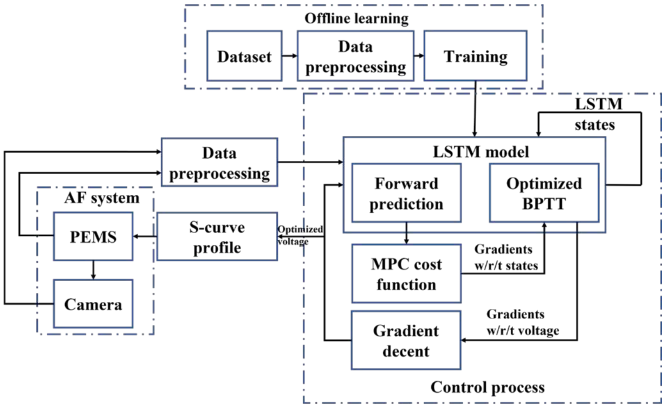


Fig. 8. Structural block diagram of the proposed LSTM-based MPC for AF.

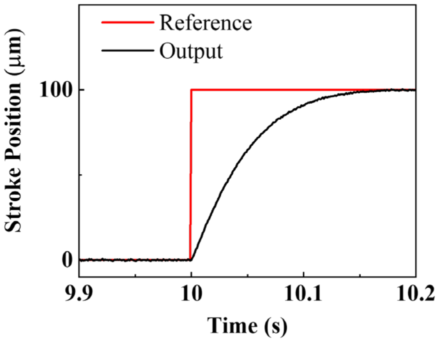


Fig. 9. Performance of the nonlinear observer-based feedback control.

we generalize the proposed method by opting to use a scalar focus score as the input to the LSTM.

To capture the dynamics between the input to the PEMS and the output focus score, we used a pulse-train method to generate the training dataset on the US Air Force Hi-Resolution target as shown in Fig. 3(b). The pulse-train method applies a series of random step inputs to the PEMS, and the resultant focus scores of the captured images are recorded. Fig. 6 shows the series of pulses with random amplitudes and the resultant focus scores from these pulses. Each pulse lasts 15 ms, equal to the same time as the PEMS movement in the control loop. To sufficiently span the working space of the PEMS, 10,000 input-output pairs are generated continuously with the input voltages to the PEMS uniformly randomly selected in the range of [1 V, 9 V]. In this experiment, we use 80% of the data for training and 20% for testing.

The LSTM is trained using the Adam algorithm in the MATLAB R2021a environment. The detailed hyperparameters of the training process are listed in Table I. Considering the tradeoff between the accuracy and computational performance, we found that the delay of time steps for the input and output $n_u = n_y = 2$, and the number of the hidden states in the LSTM structure $n_p = 10$ gave an acceptable performance. Fig. 7(a) shows the focus score prediction performance of the trained LSTM network and Fig. 7(b) is the corresponding prediction error. We conclude that the trained LSTM network adequately learns the nonlinear dynamics of the PEMS and has a high prediction RMSE of 0.19. As the physical system of the PEMS remains the same during the training process and the application process, the learned LSTM can infer the nonlinear dynamics of the PEMS online. The LSTM can be also trained online periodically if unforeseen system condition changes [33]. Finally, the overall structural block diagram of the proposed LSTM-based MPC controller for AF is shown in Fig. 8. The detailed deployment of the LSTM-based MPC controller into the real-time controller can be found in Appendix A.

TABLE II
AF METHOD PERFORMANCE AND EVALUATION METRICS FOR THE AIR FORCE TARGET

Method	Mean No. Images	Std. No. Images	Time (ms)
Random **CL**	20.30	18.84	1015
Fibonacci **CL**	4.56	1.39	228.1
G. Binning **CL**	6.61	3.11	330.3
CNN **CL**	5.84	5.82	777.6
ConvLSTM **CL**	5.01	5.04	275.6
LSTM-MPC **CL**	5.18	1.59	259.0
Random **OL**	15.08	12.14	316.7
Fibonacci **OL**	-	-	-
G. Binning **OL**	-	-	-
CNN **OL**	-	-	-
ConvLSTM **OL**	-	-	-
LSTM-MPC **OL**	6.28	6.25	157.1

***OL** = open-loop ***CL** = closed-loop

TABLE III
RUNTIME FOR KEY ROUTINES IN THE PROPOSED AF METHOD

Routine	Time consumption (ms)
Image acquisition	< 1
Image processing	6
Optimization	3.7
Motor movement	15
Total time	25

C. Autofocus Testing

Proper evaluation of the proposed LSTM-based MPC AF algorithm involves comparison against the commonly used rule-based approaches and state-of-the-art deep learning methods previously mentioned in Section II. Although all methods will be directly compared against one another, the rule-based approaches are included to serve as different baseline comparisons for the performance of the deep learning methods. The random search algorithm is included to demonstrate the absolute lower limit of rule-based approaches.

Experimental evaluation of each method consists of approximately 100 individual AF experiments on the air force target. For each experiment, the IAS was manually moved to a different location such that a wide variety of optimal focus target positions are included in the IAS working distance. An experiment begins with an initial random position of the target before the working distance is searched according to the rule-based and deep learning-based algorithms. For the proposed method, two sequential random positions are selected first as the delay of time steps $n_u = n_y = 2$. The position information and the focus scores of these two images are then used to form the initial input vector $\mathbf{x}$ to the LSTM to predict the focus position. Based on the LSTM-based MPC scheme, a control law is designed based on the case where $P = 7$, $M = 2$, $\lambda = 0.01$, and $\eta = 0.005$. The experiment ends when the focus score presented to the computer is equal to or greater than the minimum considered to reside in the DOF. For our experiments, MAG focus score of 6.1 or greater corresponded to focused positions in the 3.2 µm DOF of the experimental 10× lens.

As shown in Table II, the performance of the AF methods was comparatively evaluated using the three metrics, the mean and standard deviation of the number of images taken, as well as the mean time to completion. Regarding the control of each method, only the proposed and random search methods are capable of running in open-loop (**OL**) due to the black-box learning of the non-linear dynamics and irrelevant absolute position required for a random search. All the other methods require absolute position information, i.e., closed-loop (**CL**) control, to guarantee a successful convergence. The **CL** mode is an embedded electronic closed-loop system that compensates for the hysteresis of the piezo actuator with a settling time of 50 ms. From Table II, we can see that in **CL** mode, learning-based methods have no advantage over rule-based methods, as the positioning control is well managed by the embedded controller. Regarding the mean and standard deviation of images in **CL** mode, the Fibonacci method shows slightly better performance than the proposed method. For the **OL** mode, our proposed method has the fastest AF speed. Compared with the proposed method in **CL** mode, the existence of hysteresis effects adds difficulties in modeling the PEMS which leads to an increase in both the average number and standard deviation of the images needed for AF. However, the searching speed in **OL** mode (25 ms) is 2.4 times faster than the searching speed in **CL** mode (60 ms) which compensates for the increase in searching attempts caused by the hysteresis effects. Additionally, instead of using the embedded closed-loop controller, a different model-based closed-loop controller has been implemented to the PEMS for positioning control. The customized controller is a nonlinear observer-based feedback controller based on the classical Bouc-Wen model [34], [35]. Fig. 9 shows the step setpoint response of the customized controller. The response time for the feedback control is more than 100 ms which is more than 7 times longer than the open-loop control of the PEMS and 2 times longer than the embedded closed-loop controller. Because both the model-based feedback controller and the embedded closed-loop controllers eventually converge to the ground truth position they can be considered to have the same accuracy for positioning control. If the model-based feedback controller were implemented on the method that uses the least number of images (Fibonacci **CL** in Table II), it would take an average of 227.9 ms longer for AF than the previous Fibonacci **CL**. As a result, although it may take fewer images to find the focused position when closed-loop control of PEMS is applied, the overall AF time is slower than the proposed method.

To test the generalization ability of the proposed LSTM-based MPC AF method, a micro-scale circuit was used as the new AF target as shown in Fig. 3(c). As the circuit target was neven seen in the training data, the CNN and the ConvLSTM method cannot find the optimal focus position. However, since the input to the proposed LSTM network is the focus score and is irrelevant to the contents in the target, it can still predict the focus position with a mean number of images of 6.79 and a standard deviation of 5.25. Furthermore, we have tested 5 different static targets and one moving target (videos attached). The detailed implementation of the moving target AF is described in Appendix B. As a result, the proposed method shows superiority in the generalization ability over other image-based AF methods.

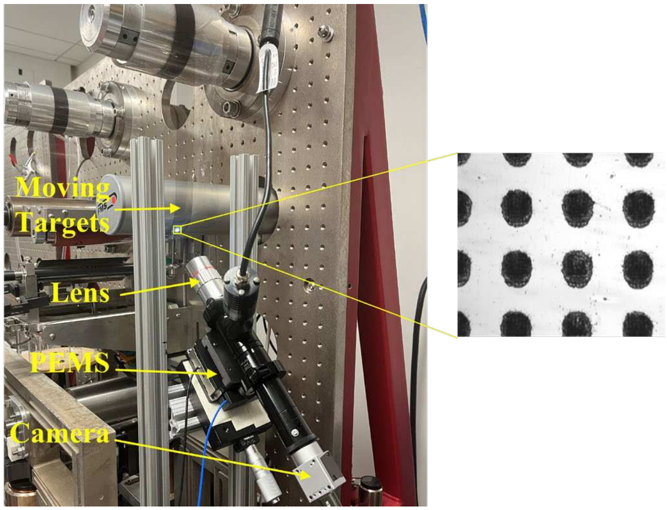


Fig. 10. Experimental setup for moving target AF.

Lastly, the specific time cost of each routine in the proposed method is shown in Table III. The cost of the proposed LSTM-based MPC AF method can be broken down into 4 separate steps: image acquisition, image processing to get the focus score, optimization of the MPC cost function, and motor movement. The focus score is extracted using the MAG method from the obtained image. The time complexity of this method is O(n) in the big-O notation where n is the number of image pixels, which means mathematically the image processing time is only related to the size of the obtained image and it is not related to the surface appearance. For the camera used in our project, the ROI is 1280×1024 pixel^2 so the processing time is 6 ms for any images captured by this camera. Inside the optimization of the MPC cost function, the convergence of the U in (4) needs 50 iterations. Since the improved BPTT method was applied in the optimization of the MPC cost function, the optimization process only takes 14.8% of the overall AF time.

V. Conclusion

In this paper, we proposed a new control concept of PEMS in AF applications. Compared with traditional applications of PEMS which require an internal controller to account for the hysteresis effects, we integrate the focus measurement and the hysteresis effects in a single learning-based model. We then infer the optimal focus position using the focus scores from a sequence of images in an NN-based MPC architecture. To learn the hysteresis effect, the NN architecture is chosen to be LSTM due to its superior ability to draw inferences from learned time sequence data. To improve the speed of the LSTM-based MPC control, we also propose an optimized LSTM backpropagation algorithm. Additionally, to reduce the vibrations from the rapid PEMS impulse inputs, a unique S-curve control input profile is designed to reduce the acceleration profile. This promotes a more stable transition in dynamics and ultimately improves the accuracy of the proposed LSTM-based MPC AF control scheme. Compared with baseline ruled-based AF methods and other deep learning-based methods, our proposed method demonstrates significant advantages regarding the AF time.

Finally, it should be noted that the proposed control concept of PEMS is not limited to the sole application of AF. As long as sufficient pairs of PEMS and response variable training inputs can be acquired, the same control concept can be applied to other PEMS or PEA applications to save the efforts in designing the internal controller.

APPENDIX A

To ensure the real-time performance of the algorithm in LabVIEW, we have converted the LSTM-based MPC controller from the MATLAB code into an executable in LabVIEW. The conversion includes four steps. First, the LSTM network is trained with the Deep Learning toolbox in MATLAB. Second, we code and debug the LSTM-based MPC controller in the MATLAB environment. Third, the LSTM-based MPC controller is converted to C code with the help of the MATLAB Coder. Finally, an executable that can be run in the LabVIEW environment is built from the C project using C&C++ Development Tools for NI Linux Real-Time. By the above four steps, the LSTM-based MPC controller can be run in the LabVIEW environment independently.

APPENDIX B

Fig. 10 shows the experimental setup of the AF system for moving targets. The PEMS-based AF system is assembled on a roll-to-roll system. The moving target is a preprinted dot array on flexible substrates as shown in Fig. 10. The target is moving at 1 mm/s which is a common printing speed for roll-to-roll microcontact printing of micro- and nano-scale patterns [36]. In the uploaded video, it can be found that it takes 6 images for the AF process to get the focused image of the moving objects which is about 150 ms.

REFERENCES

[1] X. Du, D. Hardt, and B. Anthony, “Real-Time Imaging of Invisible Micron-Scale Monolayer Patterns on a Moving Web Using Condensation Figures,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 4077–4087, May 2020, doi: 10.1109/TIE.2019.2914632.

[2] P. DiMeo, L. Sun, X. Du, and X. Du, “Fast and accurate autofocus control using Gaussian standard deviation and gradient-based binning,” *Opt. Express, OE*, vol. 29, no. 13, pp. 19862–19878, Jun. 2021, doi: 10.1364/OE.425118.

[3] A. Aabid *et al.*, “A Systematic Review of Piezoelectric Materials and Energy Harvesters for Industrial Applications,” *Sensors (Basel)*, vol. 21, no. 12, p. 4145, Jun. 2021, doi: 10.3390/s21124145.

[4] S. Mohith, A. R. Upadhya, K. P. Navin, S. M. Kulkarni, and M. Rao, “Recent trends in piezoelectric actuators for precision motion and their applications: a review,” *Smart Mater. Struct.*, vol. 30, no. 1, p. 013002, Dec. 2020, doi: 10.1088/1361-665X/abc6b9.

[5] C. Li, K. Liang, W. Zhong, J. Fang, L. Sun, and Y. Zhu, “Electrochemical Coupled Analysis of a Micro Piezo-Driven Focusing Mechanism,” *Micromachines (Basel)*, vol. 11, no. 2, p. 216, Feb. 2020, doi: 10.3390/mi11020216.

[6] H. Li, Y. Xu, M. Shao, L. Guo, and D. An, “Analysis for hysteresis of piezoelectric actuator based on microscopic mechanism,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 399, p. 012031, Sep. 2018, doi: 10.1088/1757-899X/399/1/012031.

[7] C. Rusu, S. Besoiu, and M. Tatar, “Design and closed-loop control of a piezoelectric actuator,” *IOP Conference Series: Materials Science and Engineering*, vol. 1018, p. 012002, Jan. 2021, doi: 10.1088/1757-899X/1018/1/012002.

[8] M. Rakotondrabe, C. Clevy, and P. Lutz, “Complete Open Loop Control of Hysteretic, Creeped, and Oscillating Piezoelectric Cantilevers,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 440–450, Jul. 2010, doi: 10.1109/TASE.2009.2028617.

[9] J. M. Rodriguez-Fortun, J. Orus, J. Alfonso, F. Buil, and J. A. Castellanos, “Hysteresis in Piezoelectric Actuators: Modeling and Compensation,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 5237–5242, Jan. 2011, doi: 10.3182/20110828-6-IT-1002.01063.

[10] C. Ru and L. Sun, “Study of polarization control model for piezoelectric actuator,” *Ultrasonics*, vol. 44, pp. e731–e735, Dec. 2006, doi: 10.1016/j.ultras.2006.05.199.

[11] A. Sebastian and S. M. Salapaka, “Design methodologies for robust nano-positioning,” *IEEE Transactions on Control Systems Technology*, vol. 13, no. 6, pp. 868–876, Nov. 2005, doi: 10.1109/TCST.2005.854336.

[12] M. Ghafarian, B. Shirinzadeh, A. Al-Jodah, and T. K. Das, “Adaptive Fuzzy Sliding Mode Control for High-Precision Motion Tracking of a Multi-DOF Micro/Nano Manipulator,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4313–4320, Jul. 2020, doi: 10.1109/LRA.2020.2996065.

[13] S. Xie and J. Ren, “Recurrent-Neural-Network-Based Predictive Control of Piezo Actuators for Trajectory Tracking,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2885–2896, Dec. 2019, doi: 10.1109/TMECH.2019.2946344.

[14] 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, Dec. 2015, doi: 10.1109/TIE.2015.2455026.

[15] P. Ge and M. Jouaneh, “Modeling hysteresis in piezoceramic actuators,” *Precision Engineering*, vol. 17, no. 3, pp. 211–221, Jul. 1995, doi: 10.1016/0141-6359(95)00002-U.

[16] K. Kuhnen, “Modeling, Identification and Compensation of Complex Hysteretic Nonlinearities: A Modified Prandtl-Ishlinskii Approach,” *European Journal of Control*, vol. 9, no. 4, pp. 407–418, Jan. 2003, doi: 10.3166/ejc.9.407-418.

[17] F. Shen, L. Hodgson, and K. Hahn, “[32] - Digital Autofocus Methods for Automated Microscopy,” in *Methods in Enzymology*, vol. 414, J. Inglese, Ed. Academic Press, 2006, pp. 620–632. doi: 10.1016/S0076-6879(06)14032-X.

[18] C. Park *et al.*, “A highly accurate piezoelectric actuator driver IC for auto-focus in camera module of mobile phone,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1284–1287. doi: 10.1109/ISCAS.2010.5537267.

[19] D. Croft, G. Shed, and S. Devasia, “Creep, Hysteresis, and Vibration Compensation for Piezoactuators: Atomic Force Microscopy Application,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 1, pp. 35–43, Nov. 1999, doi: 10.1115/1.1341197.

[20] Z. Zhou *et al.*, “Facile large-area autofocusing Raman mapping system for 2D material characterization,” *Opt. Express, OE*, vol. 26, no. 7, pp. 9071–9080, Apr. 2018, doi: 10.1364/OE.26.009071.

[21] C.-C. Gu, K.-J. Wu, J. Hu, C. Hao, and X.-P. Guan, “Region sampling for robust and rapid autofocus in microscope,” *Microscopy Research and Technique*, vol. 78, no. 5, pp. 382–390, 2015, doi: 10.1002/jemt.22484.

[22] Y. Zhang, N. Wadhwa, S. Orts-Escolano, C. Häne, S. Fanello, and R. Garg, “Du2Net: Learning Depth Estimation from Dual-Cameras and Dual-Pixels,” in *Computer Vision – ECCV 2020*, Cham, 2020, pp. 582–598. doi: 10.1007/978-3-030-58452-8_34.

[23] X. Wu, J. Zhou, J. Liu, F. Ni, and H. Fan, “Single-Shot Face Anti-Spoofing for Dual Pixel Camera,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1440–1451, 2021, doi: 10.1109/TIFS.2020.3035879.

[24] L. Wang, K. Zhang, Y. Wang, and Z. Sun, “An End-to-End Autofocus Camera for Iris on the Move,” in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, Aug. 2021, pp. 1–8. doi: 10.1109/IJCB52358.2021.9484340.

[25] C. Wang, Q. Huang, M. Cheng, Z. Ma, and D. J. Brady, “Deep Learning for Camera Autofocus,” *IEEE Transactions on Computational Imaging*, vol. 7, pp. 258–271, 2021, doi: 10.1109/TCI.2021.3059497.

[26] C.-Y. Chen, R.-C. Hwang, and Y.-J. Chen, “A passive auto-focus camera control system,” *Applied Soft Computing*, vol. 10, no. 1, pp. 296–303, Jan. 2010, doi: 10.1016/j.asoc.2009.07.007.

[27] A. Santos, C. Ortiz De Solórzano, J. J. Vaquero, J. M. Peña, N. Malpica, and F. Del Pozo, "Evaluation of autofocus functions in molecular cytogenetic analysis," *Journal of Microscopy*, vol. 188, no. 3, pp. 264–272, 1997, doi: 10.1046/j.1365-2818.1997.2630819.x.

[28] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Networks*, vol. 1, no. 4, pp. 339–356, Jan. 1988, doi: 10.1016/0893-6080(88)90007-X.

[29] J. Yan, L. Jin, Z. Yuan, and Z. Liu, "RNN for Receding Horizon Control of Redundant Robot Manipulators," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1608–1619, Feb. 2022, doi: 10.1109/TIE.2021.3062257.

[30] J.-F. Qiao and H.-G. Han, "Identification and modeling of nonlinear dynamical systems using a novel self-organizing RBF-based approach," *Automatica*, vol. 48, no. 8, pp. 1729–1734, Aug. 2012, doi: 10.1016/j.automatica.2012.05.034.

[31] K. Patan and P. Witczak, "Robust model predictive control using neural networks," in *2014 IEEE International Symposium on Intelligent Control (ISIC)*, Oct. 2014, pp. 1107–1112. doi: 10.1109/ISIC.2014.6967615.

[32] K. Patan, "Neural Network-Based Model Predictive Control: Fault Tolerance and Stability," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 1147–1155, May 2015, doi: 10.1109/TCST.2014.2354981.

[33] S. Wen, Y. Wang, Y. Tang, Y. Xu, P. Li, and T. Zhao, "Real-Time Identification of Power Fluctuations Based on LSTM Recurrent Neural Network: A Case Study on Singapore Power System," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5266–5275, Sep. 2019, doi: 10.1109/TII.2019.2910416.

[34] G. Flores and M. Rakotondrabe, "Robust Nonlinear Control for a Piezoelectric Actuator in a Robotic Hand Using Only Position Measurements," *IEEE Control Systems Letters*, vol. 6, pp. 872–877, 2022, doi: 10.1109/LCSYS.2021.3087102.

[35] G. Flores, N. Aldana, and M. Rakotondrabe, "Model Predictive Control Based on the Generalized Bouc-Wen Model for Piezoelectric Actuators in Robotic Hand With Only Position Measurements," *IEEE Control Systems Letters*, vol. 6, pp. 2186–2191, 2022, doi: 10.1109/LCSYS.2021.3136456.

[36] X. Zhou, D. Wang, J. Wang, and S.-C. Chen, "Precision design and control of a flexure-based roll-to-roll printing system," *Precision Engineering*, vol. 45, pp. 332–341, Jul. 2016, doi: 10.1016/j.precisioneng.2016.03.010.

Jingyang Yan received the B.S. degree in automation from Zhejiang University, Hangzhou, China in 2014, and the M.S. degree in control engineering from the University of Science and Technology of China, Hefei, China in 2017. He is currently working toward a Ph.D. degree in mechanical engineering at the University of Massachusetts, Amherst, MA, USA. His research interests include deep learning, flexible electronics manufacturing, and nonlinear control.

Peter DiMeo is working toward his B.S and M.S in mechanical engineering at the University of Massachusetts Amherst. Peter was the recipient of the First and Third place NSF Research awards in the 2020 and 2021 ASME IMECE Conference for his work in autofocus. He is also the first author of a peer-reviewed autofocus publication (2021). Peter is a teaching assistant, former NCAA D1 athlete, and club chairman. His research areas include mechanical design, control, and computer vision.

Lu Sun received the B.S. degree in Physics from Qufu Normal University, Qufu, China, in 1995, the M.S. degree in Condensed Matter Physics from Peking University, Beijing, China, in 1998, and the Ph.D. degrees in Microelectronics from Nanyang Technological University, Singapore, in 2005.

Dr. Sun is currently working as a Senior Research Fellow in the Electrical and Computer Engineering Department at the University of Massachusetts, Amherst.

Xian Du received the B.S. degree in mechanical engineering from Tianjin University, Tianjin, China, in 1994, the M.S. degree in mechanical engineering from Shanghai Jiaotong University, Shanghai, China, in 2001, and the M.S. and Ph.D. degrees in the program of innovation of manufacturing systems and technology from the Singapore-MIT Alliance at the National University of Singapore, Singapore, in 2002 and 2007, respectively.

Since 2018, he has been an Assistant Professor with the Mechanical and Industrial Engineering Department and Institute for Applied Life Sciences at the University of Massachusetts, Amherst, MA, USA. His current research topic is the innovation of high-resolution, large-area, and fast-speed machine vision and pattern recognition technologies for manufacturing and medical devices. His research interests include pattern recognition, intelligent imaging and vision, flexible electronics manufacturing, robotics, and medical device realization.

Dr. Du was the recipient of the NSF CAREER award in 2020. He is a member of the IEEE, ASME and OSA.