DEEP LEARNING-BASED PRECISION CONTROL FOR SIX-AXIS COMPLIANT NANOPOSITIONER

[1]Xiangbo Liu, [1]Chenglin Li, [2]Hongsheng Li, [2]Xiaogang Wang, and [1]Shih-Chi Chen*
[1]Department of Mechanical and Automation Engineering
The Chinese University of Hong Kong
[2]Department of Electronic Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR, China
*Email: scchen@mae.cuhk.edu.hk

In this work, we present a new control method for compliant mechanisms based on deep learning models that achieve 100 nm precision in multiple axes with low-lost strain sensor arrays. In our setup, the strain sensor array is applied to the flexing components on a custom-designed compliant six-axis nanopositioner such that it simultaneously measures the mechanical strain (i.e., displacement) and mode shapes for real-time control, i.e., the deep model can learn, predict, and control the motion of the nanopositioner as a time sequence or 2-D images. During the training stage, capacitance sensors are used as references to minimize position errors; the number and location of sensor arrays can be optimized via the deep learning model. In the experiments, we demonstrate that the deep model developed based on the reinforcement learning method [1] can fully replace the classical PID control, realizing 100 nm precision without tuning any control parameters.

INTRODUCTION

Conventionally, precision motion stages and positioners achieve submicron or nanometer level precision by exploiting the precision engineering principle that calls for high repeatability components, e.g., air bearing or flexures, and high resolution sensors and actuators in combination with adequate control methods [2]. However, these solutions are usually of high cost with limited range of applications. To address the issue, we propose a deep learning-based control platform that uses low-cost strain sensor arrays to replace the capacitance probes on a six-axis nanomanipulator, with potentially improved performance and capability, i.e., monitoring the mode shapes and systems dynamics with real-time control.

Deep learning has achieved great success in many fields including speech recognition and synthesis, computer vision, and natural language. Recently, researchers have started to explore integrating deep learning and reinforcement learning into the perception and the control of robots [3], where robots learn by interacting with the environment, leading to promising results. Yet, there is still much room to improve for many industrial applications, e.g., control of precision machines. Generally, deep learning is a powerful tool for treating regression and classification problems based on big data. In our work, given the high repeatability (i.e., nanometer level) of the compliant nanopositioner and high dimensions of the signals, deep learning can be implemented to improve the system performance.

WORKING PRINCIPLE

Figure 1 presents the working principle of the deep learning-based control, realized via low-cost strain sensors installed on the flexing components of a compliant mechanism. Here we use the HexFlex, i.e., a six-axis planar compliant mechanism [4], to illustrate how strain sensors are distributed on the structure.

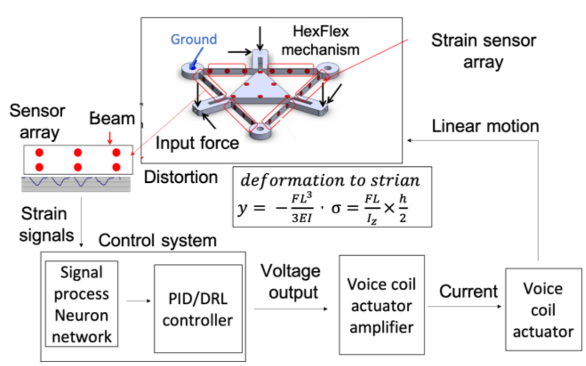


FIGURE 1. Principle of deep-learning-based control for a compliant nanopositioner. The red

dots on the HexFlex mechanism show the initial positions of the strain sensors. DRL: Deep reinforcement learning.

First, strain sensors are uniformly applied to the non-rigid structures on the HexFlex mechanism, where each strain sensor serves a pixel in a 2-D image. The strain sensors should be sufficiently dense to capture basic motions and mode shapes of a structure, e.g., bending and torsion of a double-clamped beam. Knowledge of beam theory can be applied to select high-stress locations and to avoid nodal points in resonant modes. Alternatively, the redundant sensors can also be removed by training the deep learning model. As the motion of the six-axis stage is generated by the deformation of its flexing components, e.g., beam or plates, the measured mechanical strains will have complex yet deterministic relationships with the stage motion. This means that the strain signals can provide the multi-axis motion information and serve as feedback signals to the deep learning-based controller.

Regarding the deep learning model, instead of using a typical deep neural network to predict compensation of position errors in a single step, we have developed a physics-guided deep model that drives the actuators in multiple steps to generate smooth and accurate motion. As the target position error is only measured at the last stage during training, reinforcement learning is used to train the neural networks in the intermediate stages through the design of reward functions, so that the neural networks can predict proper control given input from the strain sensors only at the testing stage. At the training stage, high-precision capacitance sensors are used to detect the positions of the nanopositioner, which are used in the reward functions of intermediate stages in reinforcement learning. As analytic models cannot capture all physical phenomena (e.g., non-ideal boundary conditions, manufacturing errors, thermal errors etc.), the deep learning model can evolve to fill the gap between the engineering model and reference data. Lastly, the collected dataset is used to coach the deep model so that it precisely predicts the position and dynamic behavior of the system. At the testing stage, capacitance sensors are removed and only the strain sensor array is used to predict control.

COMPLIANT SIX-AXIS NANOPOSITIONER

Figure 2 presents the custom-designed parallel six-axis nanopositioner, driven by six voice coil actuators (VCA) (NCC15-24-090-1X, H2W) located below the flexure stage. As shown in Fig. 3A and 3B, this compact configuration is achieved by combining and arranging the flexible beams in unique ways that enable improved dynamic performance [5]. The width of each beam is selected to be 10 mm wide to provide room for strain sensors. Six capacitance probes (C8/CPL190, Lion Precision, resolution = 20 nm) are used to monitor and record motions of the nanopositioner for calibration and training. The performance of the parallel six-axis nanopositioner is summarized in Table 1.

TABLE 1. Performance of the parallel six-axis nanopositioner

	Bandwidth	Range	Resolution
X	121 Hz	±80 μm	30 nm
Y	126 Hz	±60 μm	25 nm
Z	118 Hz	±70 μm	40 nm
θ_X	246 Hz	±900 μrad	1.5 μrad
θ_Y	216 Hz	±1700 μrad	1.4 μrad
θ_Z	188 Hz	±600 μrad	1.0 μrad

STRAIN SENSORS

We install the strain sensors at the high strain locations to increase the system sensitivity and resolution, as shown in Fig. 3B, as in the concept validation stage the nanopositioner is operated at low speeds. Strain sensors are governed by equations (1) and (2):

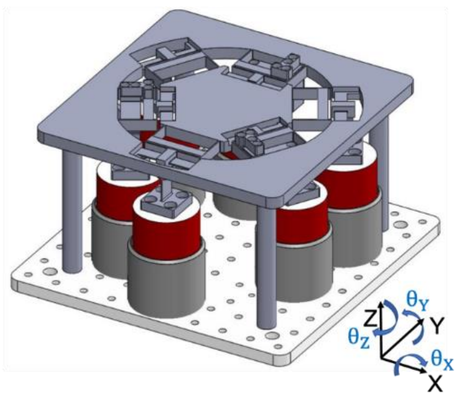
$$V = U\frac{\Delta R}{4R} \tag{1}$$

$$GF = \frac{\Delta R/R}{\varepsilon} \tag{2}$$

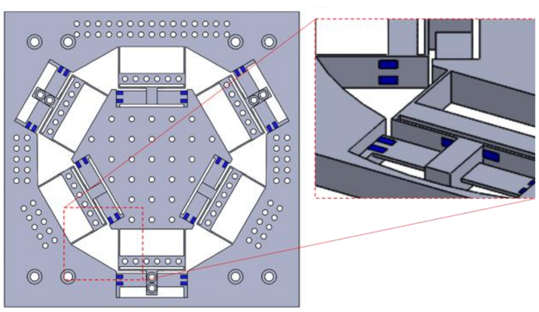
where U = 2.0 V is the Wheatstone bridge active voltage, determined by the data acquisition board (DAQ); GF is the gauge factor; and R is the base resistance of strain sensors, ε is the surface strain. According to Eq. (1) and (2), the larger the GF, the higher the voltage (V). Based on the analysis, self-temperature compensated semiconductor strain sensors (KSN series, KYOWA) are selected for our experiments with GF = 120; R = 120 Ωm. By combining Eq. (1) and (2), the relationship between the mechanical strain and displacement (d) can be derived, as expressed in Eq. (3).

$$d = \frac{2L^2\varepsilon}{3h} \quad (3)$$

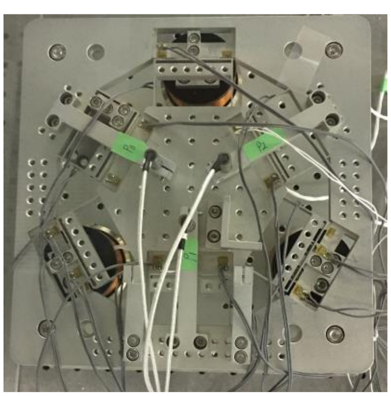
According to the system design, L = 24 mm. The DAQ board (NI9235) has a minimum measurable voltage of $V_0 \sim 10^{-7}$ V. Accordingly, from Eq. (3), the theoretical minimum detectable displacement of each strain sensor is calculated to be $d_{min} \sim 5$ nm. This shows nanometer level precision control can be achieved with low-cost strain sensors with proper control strategy. In the calibration experiments, we have experimentally confirm the strain sensors have a resolution of 100 nm, limited by electronic noises.



A



B



C

FIGURE 2. Parallel six-axis nanopositioner. A. CAD model; B. Location of strains sensors; C. Completed stage.

EXPERIMENTS

In this section, we devise and perform positioning experiments to validate the deep learning-based control method can achieve 100 nm level precision. All net models in this work are developed on the pytorch-gpu. Data acquisition programs are developed on the LabVIEW 2018-pro with the Enthought Python Integration Toolkit.

DESIGN AND TRAINING OF CNN MODEL

As shown in Fig. 1, the stage displacement (or beam deformation) and mechanical strain have a linear relationship based on the beam theory. Considering this linear relationship and the non-linear nature of the strain sensors, we have developed a convolutional neural network (CNN) as illustrated in Fig. 3, where the linear and nonlinear layers correspond to the aforementioned two cases; and each fully connected layer contains 12,000 neurons.

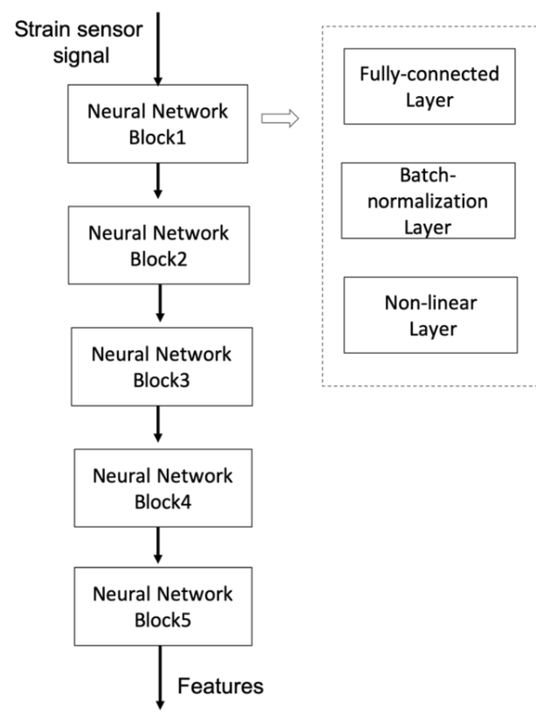
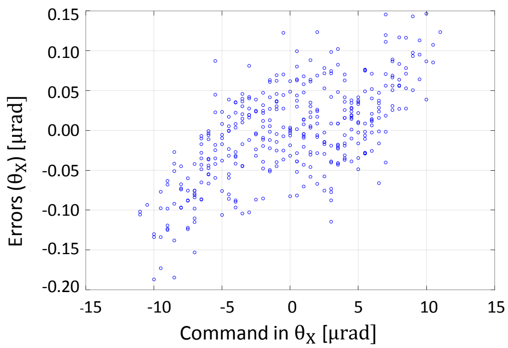


FIGURE 3. Structure of the deep learning model with 12,000 neurons in each fully-connected layer.

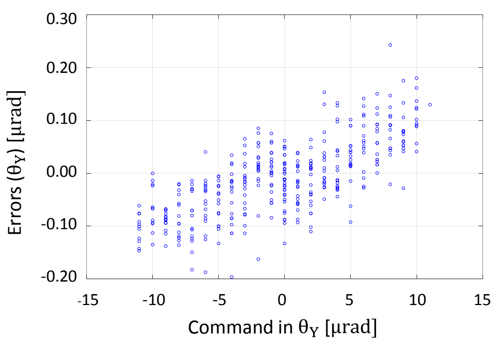
To train the deep learning model, the nanopositioner is commanded to perform designed positioning tasks with a variety of ranges (0 -15 µm) with a minimum step size of 50 nm. During the training process, capacitance probes are used as the reference data; and via the probes, the PID control is implemented to achieve a precision of 10 nm. During each positioning task, the DAQ records signals from the strain sensors for 2 seconds at a rate of 1 kHz.

STATIC POSITION CONTROL

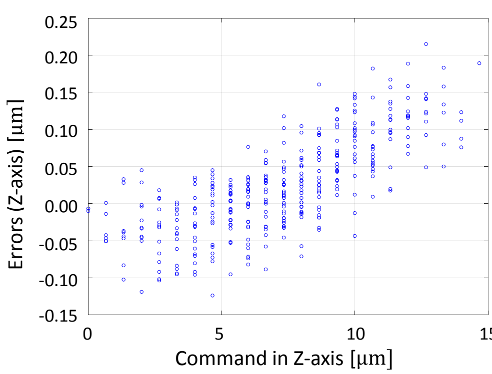
For validation, 90% of the data from the strain sensor are used to train the deep model; the model is then used to predict the stage motion and compare with the rest 10% data. After 200 epochs trained, the test error is ~200 nm. The required training time is less than 1 hour. The test results along the θ_X, θ_Y, Z axes are presented in Fig. 4A, 4B, and 4C respectively. From Fig. 4, one can confirm that the deep learning model can achieve precision static displacement control.



A



B



C

FIGURE 4. Positioning errors in A. θ_X; B. θ_Y; and C. Z-axis

DYNAMIC POSITIONING

In this section, we present the development of a deep reinforcement learning (DRL) model, shown in Fig. 5, for dynamic position control, where the optimal control strategy is "self-learned" through training.

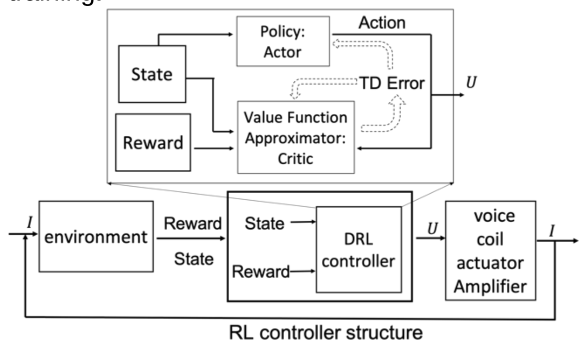


FIGURE 5. Deep reinforcement learning (DRL) controller for the parallel six-axis nanopositioner. Displacement $\tilde{P} = [\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, \tilde{p}_4, \tilde{p}_5, \tilde{p}_6]$; U = Output actions.

The basic deep reinforcement learning (DRL) problem can be considered as a Markov decision process (MDP), whose aim is to find an action strategy to maximize the specified rewards. Considering the control system as a simple environment, the controller functions as an agent that interacts with the environment by sending and receiving inputs/feedback to/from the system. Based on the characteristics of the system including speed, overshoot and stability, we can define rewards to the agent. The controller self-optimization is also a DRL problem that has a continuous action space.

In Fig. 5, system state is represented by the displacement $\tilde{P}$, the first-order derivative $\dot{\tilde{P}}$, the target position P^* and the input voltage $\tilde{U}$. The displacement in each axis are measured by the capacitance sensors whose output is expressed as $[\tilde{p}_1, \tilde{p}_2, \tilde{p}_3, \tilde{p}_4, \tilde{p}_5, \tilde{p}_6]$. The output action is the output voltage U, which is used to control the VCAs. Output voltages have a continuous value from -2.5 V to 2.5 V. To address the issue, we use deep deterministic policy gradient (DDPG) algorithm [6] as our DRL model. The DDPG algorithm is known for its effectiveness for continuous-action-space systems. The DRL model consists of two networks, i.e., critic and actor network. The actor network receives the current state as the input and outputs an action. With this predicted action, the real environment then returns the next state and rewards the DRL model. Meanwhile, the critic network receives both the current state and predicted action as the input, and outputs a Q value which judges the selected action that corresponds to current state. With the predictions and feedback, the two networks are updated. In our controller, the actor network consists of an input layer with 24 neurons; three hidden fully-connected layers with 300 neurons in each layer; and an output layer with six neurons, generating the predicted actions. The critic network consists of an input layer with 30 neurons, 5 hidden fully-connected layers with 300 neurons in each layer; and an output layer with one neuron, generating Q-values that reveal the performance of the selected actions.

The state space and action space are complex, and requires a lot of time to learn from a randomly initialized model. To shorten the learning time and avoid unnecessary damages to the real systems, we pre-train the DRL model in a virtual environment before applying it to the real system. First, we build the virtual environment by simulating the real environment with neural networks. According to control theory, we know that the next system state is determined by the current velocity, position and input voltage. So, a neural network can be designed to model this relationship. The input of the network consists of the current position, velocity and input voltage; and the output is the next position. The parameters of this network are updated using the collected real system data. With the trained virtual environment network, we can pre-train the DRL controller.

To pre-train the controller, the required training data are automatically collected over 50 hours by randomly setting the target positions from 0 – 25 µm, while the PID controller drives the nanopositioner to the target positions. During the time, the DAQ records the reference positions from the capacitance probes and the voltage output from the PID controller. First, the actor network and critic network are initialized randomly. Then, by using the collected data, the virtual environment model is trained. Under the virtual environment, we pre-train the controller to function as the PID controller. The pre-training step can help rapidly optimize the model for real operation. This is particular efficient for systems with a large state space. After pre-training, we replace the simulated virtual environment with the parallel nanopositioner. Lastly, the controller is trained with the dynamic system.

Figure 6 presents the three-axis positioning results based on the DRL model. In the experiment, the stage is commanded to move to the 0 rad, 0 rad, and 5 µm positions in the θ_X, θ_Y, Z axes respectively, where the initial positions in each axis are randomly selected. The results show that the stage moves to the target position with less than 100 nm translational errors and 10 µrad angular errors within 5 seconds, which achieves our target precision. The DRL controller model used in this test is trained for 2000 epochs with a pre-training time of 2 hours. (The total training time can be substantially reduced by pre-training.) With more training time and data, the controller can be further improved both in terms of speed and precision.

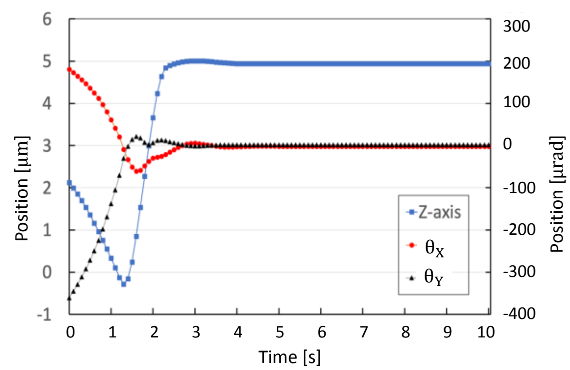


FIGURE 6. Positioning experiments along the θ_X, θ_Y, and Z axes based on the DRL controller.

CONCLUSION

We have presented a new deep-learning based control method to a custom-designed parallel six-axis nanopositioner, achieving a static and dynamic positioning accuracy of 200 nm and 100 nm respectively. The deep model is realized by applying an array of low-cost strain sensors to the nanopositioner. The training of the deep model is performed automatically, where the reference signals are obtained from six capacitance sensors. The experimental results can be further improved with more training time and training data. The new control method can be easily applied to other precision machines to improve performance with reduced cost.

ACKNOWLEDGMENT

This work is supported by the HKSAR Innovation and Technology Commission (ITC) Innovation and Technology Fund (ITF), ITS/077/17FP: Deep Learning for Next-generation Precision Machine Tools and Manufacturing.

REFERENCES

[1] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. Journal of Artificial Intelligence Research. 1996; 4: 237-285.

[2] Slocum AH. Precision Machine Design, Society of Manufacturing Engineers. 1992.

[3] Levine S, Finn C, Darrell T, Abbeel P. End-to-end Training of Deep Visuomotor Policies. Journal of Machine Learning Research. 2016; 17: 1 - 40.

[4] Culpepper ML, Anderson G. Design of a Low-cost Nano-manipulator which Utilizes a Monolithic, Spatial Compliant Mechanism. Precision Engineering. 2004; 28(4): 469 - 482.

[5] Li C, Liu X, Chen S. Design of a Six-axis Parallel Nanopositioner using Decoupled Flexural Legs. Proc. of the Annual Meeting of the ASPE, Pittsburgh, PA, USA, Oct. 2019.

[6] LILLICRAP, Timothy P., et al. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.