Learning-Based Kinematic Control of a Deployable Manipulator With Long Span and Low Stiffness

Hu Liu, Yi Yang, Yudi Zhao, Yang Yang, Yan Peng, Huayan Pu, and Yang Zhou

***Abstract*—The deployable manipulator features long span and low stiffness during operation, resulting in significant positioning errors. The conventional mechanism control model based on error parameters is complex, lengthy, and challenging to guarantee accuracy in practice. This article proposes a learning-based kinematic control of the deployable manipulator. First, an analysis of kinematic performance and positioning error is presented. Then, the dataset is built by collecting data in the real environment, and the load factor that significantly influences the actual kinematics is taken as an extra feature. We propose a dataset building method based on manipulability according to the kinematic characteristics. A learning-based model consisting of a gated recurrent unit (GRU) and a 1-D convolutional layer is proposed, which is lighter and more effective than existing methods. Trajectory tracking and target grasping experiments are conducted to validate the performance of the kinematic control. The experimental results demonstrate that the proposed learning-based approach can achieve precise control under variable loads. This method could be extended to the kinematic control of similar deployable manipulators or flexible robots.**

***Index Terms*—Deployable manipulator, kinematic control, learning, positioning error.**

Manuscript received 30 September 2022; revised 14 April 2023 and 12 June 2023; accepted 29 June 2023. Recommended by Technical Editor Shaohui Foong and Senior Editor W.J. Chris Zhang. This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102804, in part by the National Natural Science Foundation of China under Grant 51904181, and in part by the Shanghai Outstanding Academic Leaders Plan under Grant 20XD1421700. *(Hu Liu and Yi Yang contributed equally to this work.) (Corresponding author: Yang Zhou.)*

Hu Liu, Yi Yang, Yang Yang, and Huayan Pu are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: liuhu0018@shu.edu.cn; yiyangshu@shu.edu.cn; yangyang_shu@shu.edu.cn; phygood_2001@shu.edu.cn).

Yudi Zhao is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yudizhao@sjtu.edu.cn).

Yan Peng and Yang Zhou are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China, and also with the Shanghai Artificial Intelligence Laboratory, Shanghai 201109, China (e-mail: pengyan@shu.edu.cn; saber_mio@shu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2023.3296698.

Digital Object Identifier 10.1109/TMECH.2023.3296698

I. Introduction

TRANSLATIONAL parallel manipulators (TPMs) have been widely utilized in industrial assembly and food packaging due to their characteristics of low actuation requirements, easy control, and low cost. Many researchers have studied the synthesis [1], [2], performance [3], [4], and structural optimization [5], [6] of TPMs to further improve their capacities. Two-degrees-of-freedom (DOF) TPM is an essential member of the lower-mobility parallel mechanism family synthesized in several types. Liu et al. [7] have designed a 2-DOF parallel manipulator consisting of 2P (Pa) kinematic chains to enable the end effector to move in a constant orientation. A planar version of the Delta robot named Diamond has been proposed by Huang et al. [8]. A 2-DOF TPM with parallel linear motion components has been designed by Yang et al. [9] and applied in an automatic docking device. To enlarge the workspace and save storage space, some studies have introduced the concept of the deployable mechanism into the design of parallel mechanisms. Examples include the scissorlike mechanisms (SLiMs) [10], Wohlharts polyhedral star transformers [11], and Hobermans polyhedral mechanisms [12]. By combining the SLiMs and TPMs, Gonzalez et al. have designed a 6-DOF parallel robot with a triple scissor extender for aircraft assembly [13]. Yang et al. [14], [15] have designed certain types of parallel lower-mobility manipulators with dual scissor-like mechanisms (D-SLiMs).

In this article, we design and fabricate a 2-DOF deployable translational parallel manipulator (DTPM) with D-SLiMs actuated by two rotary motors. It can be collapsed to a small or stowed configuration for storage and expanded to a much larger span or deployed configuration for working. In this complex mechanism, the links are connected by many joints with clearances, which are necessary and cannot be eliminated. Therefore, the deployable manipulator features long span and low stiffness during operation. The motion of the manipulator is greatly influenced by the cumulative clearance and flexible deformation errors. Some methods have been used for clearance and deformation analysis, including the matrix method [16], the interval approach [17], and the direct linearization method [18]. Tsai and Lai [19], [20] used the transmission wrench screw and joint twist screw to solve the problem and explained why multiloop linkage accuracy is difficult to analyze. Zhang et al. [21], [22] analyzed the flexible mechanisms based on the finite element approach. However, the above theoretical analyzes are

applicable primarily to serial manipulators or simple closed-loop linkages. In the case of complex mechanisms involving numbers of closed-loop limbs, links, and joints, the nonlinear equations become lengthy and complicated. They are difficult and unstable to be solved, and the result may be far from the actual situation. Therefore, a learning-based kinematic control of the deployable manipulator is proposed in this article.

Learning-based models, such as neural networks (NN), have emerged as a favorable option due to their ability to approach the nonlinear model and the low computational cost [23]. Researchers have carried out related research, especially for serial robot configurations. The use of NNs in obtaining kinematic models of serial robots has been demonstrated in the paper presented by Jha [24]. Raj et al. [25] have proposed feedforward and radial basis functions (RBF) algorithms to solve the inverse kinematic problem of a 5-DOF robot. Zubizarreta et al. [26] have proposed a methodology that uses NNs to approximate the direct kinematics of the 3PRS robot. Wang et al. [27] have proposed a multilayer NN to solve the inverse kinematics of the continuum manipulator. Toquica et al. [28] have proposed different deep learning models, including long short-term memory network (LSTM) and gated recurrent unit (GRU), to solve the inverse kinematic problem of an industrial parallel robot.

The learning-based kinematic model demonstrates a new way to control the robots and may potentially have benefits in robustness and resilience [29], [30]. A learning-based system minimizes the differences between the desired and actual outputs, thus, enabling kinematic control of the manipulator in a more realistic way [31]. Almusawi et al. [32] have suggested a learning-based motion control scheme of a 6-DOF serial robot, where forward kinematics is used to train the model. Assal et al. [33] have designed a learning-based control system for the problem of online tracking of any arbitrary trajectory by a redundant industrial manipulator. The principle of the control system was based on generating approximate values for the joint angle vector by using the fuzzy neural network (FNN). Jolaei et al. [23] have proposed a learning-based control framework for soft tendon-driven catheters that uses a support vector machine (SVM) classifier to determine which tendon to be driven and four fully connected NNs to determine the length of the tendons.

The main limitations of the above studies include the following. First, the research object is mainly the robots with high stiffness and minor errors, such as industrial manipulators. Second, the analysis of the dataset is inadequate. Third, the effect of external factors, such as load, on the actual kinematics is not considered. The highlight of this article is the use of learning-based kinematics to control the deployable manipulator with long span and low stiffness. The learning-based model consisting of a GRU and a 1-D convolutional layer is proposed, which is lighter and more effective than existing methods. The dataset is built by collecting the moving platform's position and the corresponding joint angles under different loads in the real environment. The load factor, which has a major influence on the actual kinematics, is taken as an extra feature. It would allow for compensation of clearance, deformation, and other factors as the learning-based model is trained by the real data from the motion of the mechanism. A dataset building method is proposed based on manipulability according to the kinematic characteristics, which can characterize the model better in the case of a small sample size. The experimental results show that the proposed control scheme can achieve precise control under variable loads in real time.

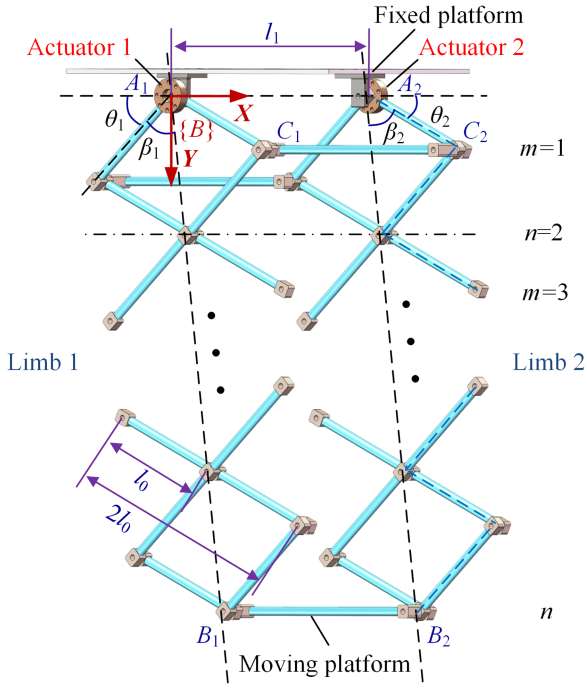


Fig. 1. Structure of the 2-DOF deployable manipulator.

This rest of this article is organized as follows. Section II presents the analysis of kinematic performance and positioning error. Section III gives a detailed description of the learning-based kinematic control framework. Section IV depicts the experiment process, including trajectory tracking and target grasping, to validate the proposed scheme. Finally, Section V concludes this article.

II. Kinematic Performance and Error

A. Forward and Inverse Kinematics

The proposed 2-DOF DTPM composed of the D-SLiMs is shown in Fig. 1. The intermediate link C_1C_2 links the two corresponding corner joints of the two identical parallel SLiM limbs, i.e., Limb 1 and Limb 2. Based on screw theory, it can be deduced that the moving platform B_1B_2 has two translational mobilities. Two rotating actuators are mounted at the joint of the fixed platform to drive the moving platform to translate in the *XY* plane.

We first consider half of Limb 2, which is marked by the zigzag dashed segments. The origin of the base coordinates $\{B\}$ is set on Point A_1. By default, the coordinates are based on the base coordinate system $\{B\}$. The number of zigzag segments between A_2 and C_2 is m, the number of zigzag segments between A_2 and B_2 is n, the length of the SLiM link is $2l_0$, and the length of links B_1B_2 and C_1C_2 are both l_1. Coordinates of joints B_2 can

be expressed by θ_2 and β_2

$$\begin{cases} X_{B2} = \Lambda_{X\_B2}(\theta_2, \beta_2) = nl_0 \cos\beta_2 \cos(\theta_2 + \beta_2) + l_1 \\ Y_{B2} = \Lambda_{Y\_B2}(\theta_2, \beta_2) = nl_0 \cos\beta_2 \sin(\theta_2 + \beta_2) \end{cases}. \tag{1}$$

Similarly, the coordinates of joints B_1 are

$$\begin{cases} X_{B1} = \Lambda_{X\_B1}(\theta_1, \beta_1) = -nl_0 \cos\beta_1 \cos(\theta_1 + \beta_1) \\ Y_{B1} = \Lambda_{Y\_B1}(\theta_1, \beta_1) = nl_0 \cos\beta_1 \sin(\theta_1 + \beta_1) \end{cases}. \tag{2}$$

As the moving platform has only two pure translational mobilities in the XY plane, Limbs 1 and 2 are always in the same configuration. The corresponding links are either collinear or parallel. Hence, the angles θ_1, θ_2, β_1, and β_2 should satisfy

$$\beta_1 = \beta_2 = \frac{\pi - \theta_1 - \theta_2}{2}. \tag{3}$$

According to (2) and (3), the inverse kinematics can be deduced as follows:

$$\begin{cases} \theta_1 = \arccos \frac{X_{B1}}{-nl_0 \cos\beta_1} - \beta_1 \\ \theta_2 = \pi - \arccos \frac{X_{B1}}{-nl_0 \cos\beta_1} - \beta_1 \end{cases}. \tag{4}$$

B. Analysis of Kinematic Performance

The Jacobian matrix serves as the foundation for kinematic performance analysis. The singularity of the manipulator can be studied by analyzing the rank of the Jacobian matrix, while the dexterity can be assessed by analyzing the manipulability which is also derived from the Jacobian matrix. The Jacobian matrix J_a of the manipulator is

$$J_a = \begin{bmatrix} J_{a1} & J_{a2} & J_{a3} & J_{a4} \end{bmatrix} \tag{5}$$

where

$$J_{a1} = \begin{bmatrix} -n\sin\theta_1 \\ (1-m)\sin\theta_1 \\ 0 \\ 0 \end{bmatrix} \quad J_{a2} = \begin{bmatrix} -n\sin\theta_2 \\ -(1+m)\sin\theta_2 \\ 0 \\ 0 \end{bmatrix}$$

$$J_{a3} = \begin{bmatrix} -n(\sin\theta_1 + \sin\theta_2)/2 \\ (1-m)(\sin\theta_1 - \sin\theta_2)/2 \\ 1 \\ 0 \end{bmatrix}$$

$$J_{a4} = \begin{bmatrix} -n(\sin\theta_1 + \sin\theta_2)/2 \\ (1-m)(\sin\theta_1 - \sin\theta_2)/2 \\ 0 \\ 1 \end{bmatrix}.$$

The determinant of J_a is

$$|J_a| = 2n \sin\theta_1 \sin\theta_2 \ . \tag{6}$$

Equation (6) shows that if $\theta_1 \in \{0, \pi\}$ or $\theta_2 \in \{0, \pi\}$, the determinant of J_a is zero, and the manipulator is in singularity.

Manipulability is a crucial metric for evaluating kinematic performance. It is defined as the determinant of the product of the Jacobian matrix and its transpose, denoted as w

$$w = \sqrt{\det(J_a J_a^T)}. \tag{7}$$

Assuming $l_0 = 114$ mm, $l_1 = 182$ mm, $m = 5$, and $n = 14$, the manipulability is investigated in the entire workspace as shown in Fig. 2. For clarity, the skeleton of the manipulator is incorporated. The input angles θ_1 and θ_2 are limited in the range $\theta_1 \in [5°, 175°]$ and $\theta_2 \in [5°, 175°]$ to avoid singularity. The result shows that the manipulability decreases significantly when the mechanism approaches the singular configuration, indicating poor kinematic performance in these regions. Lower manipulability values signify poorer kinematic performance, while higher values suggest better performance. Since the manipulability can represent the kinematic performance of the mechanism, the dataset built based on manipulability can characterize the kinematic model better in the case of a small sample size, which will be further analyzed in Section III-C.

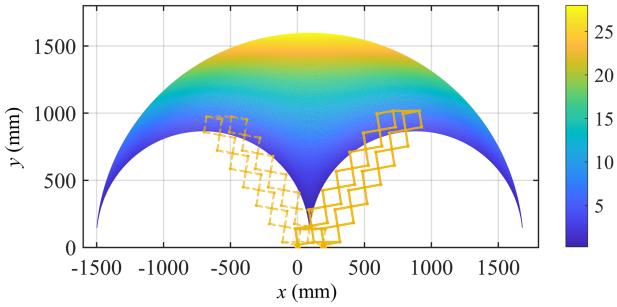


Fig. 2. Manipulability in the entire workspace.

C. Analysis of Positioning Error

As the manipulator features long span and low stiffness, the positioning error between the analytical and actual motion is considerable and significantly influenced by the load factor.

By assigning joint angles, we sampled 144 actual points in the workspace. It is worth noting that due to the actual angular range being within $[5°, 65°]$, the actual workspace is only a portion of Fig. 2. We analyze the errors between the actual points and their corresponding analytical points. Fig. 3(a)–(c) illustrates the distance errors, X-axis errors, and Y-axis errors, respectively, with the colors representing the error values. The figures demonstrate a significant discrepancy between the analytical and actual motion. The results are consistent with the analysis of kinematic performance in Section II-B, indicating that areas with high manipulability exhibit smaller errors while areas with low manipulability exhibit larger errors. The X-axis errors significantly increase when the manipulator is near the singular position. The Y-axis error is reduced as the value of the Y-axis coordinate increases. Generally, the error of the Y-axis is much larger than that of the X-axis. Hence, the errors between the analytical and actual motion are primarily attributable to the Y-axis errors. This is further supported by the similarities between Fig. 3(a) and (c). Fig. 3(d) depicts the relationship between joint angles (θ_1, θ_2) and the Y-axis coordinate of the moving platform under different loads ($M_1 = 0$ g and $M_2 = 300$ g). It is evident that the load factor $\{M_i\}$ greatly influences the Y-axis coordinate. When the two joint angles are relatively small, i.e., the manipulator is in the folded configuration, the Y-axis coordinate is greatly affected by the load. The maximum error is approximately 110 mm. On the contrary, when the two

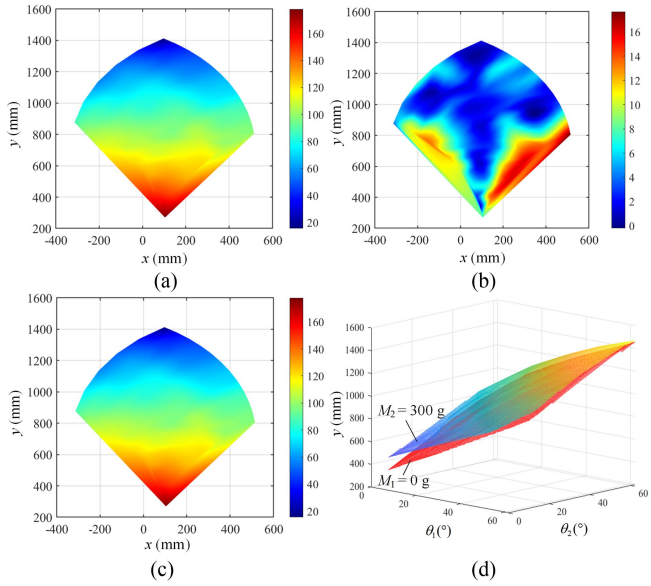


Fig. 3. (a) Distance errors, (b) *X*-axis errors, and (c) *Y*-axis errors between the actual and analytical points. (d) *Y*-axis coordinate of the workspaces under different loads.

joint angles are relatively large, i.e., the manipulator is in the expanded configuration, the *Y*-axis coordinate is less affected by the load.

In traditional approaches, the positioning error is primarily addressed by considering the joint clearances. The joint clearance is typically treated as a virtual link to simplify the analysis of the positioning errors of the mechanism, known as the equivalent joint clearance link (EJCL) [19], [34], [35], [36]. However, even a mechanism with few joints and closed loops can result in a large number of constraint equations. Specifically, the deployable manipulator in our article has 44 joints, 41 links, and more than 20 closed loops. If we apply the generalized method to analyze the positioning error, the number of constraint equations would exceed 300, making the calculation difficult and computationally expensive. Therefore, we propose the learning-based kinematic control framework in the next Section.

III. Learning-Based Kinematic Control Framework

A. System Overview

Fig. 4(a) shows the mechanism of the deployable manipulator. Two motors controlled by the controller are installed at the fixed platform to drive the moving platform to translate in the *XY* plane. Eight intermediate links are added as redundant constraints to reduce idle motion during operation. A gravity sensor, a depth camera, and a gripper are mounted on the middle of the moving platform. The base coordinate system and camera coordinate system are denoted as $\{B\}$ and $\{C\}$, respectively.

The detailed hardware configuration is illustrated in Fig. 4(b). The manipulator controller employs a member of the STM32 microcontroller family. The two servo motors (DHPS 380, DHMCU) are powered by a 24V dc source. The motor is equipped with a 12-b magnetic encoder, offering a resolution

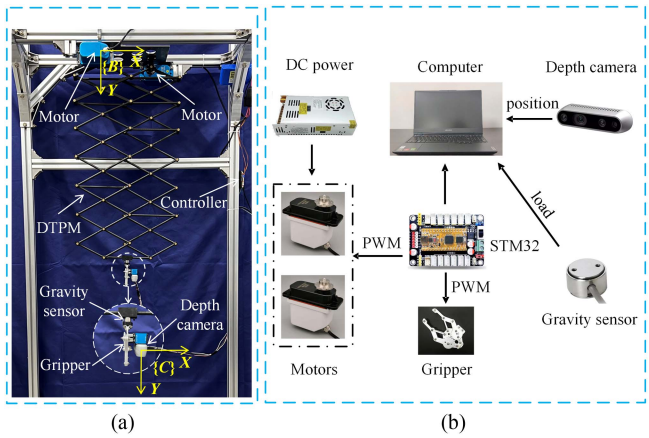


Fig. 4. (a) Mechanism of the deployable manipulator. (b) Hardware configuration.

of 2^{12}, which allows for the detection and precise reporting of 4096 distinct rotational positions. The theoretical accuracy of the motor is 0.087°. The STM32 microcontroller controls the two motors and the gripper via pulsewidth modulation (PWM) signals and communicates with the computer through Python programming. The depth camera (RealSense D435i, Intel) conveys target position information to the system, while the gravity sensor (AR-DN332, ARIZON) supplies feedback on load information. The depth camera is calibrated prior to use, and the depth error in the workspace is less than 2 mm, which has a negligible impact on the motion experiments.

B. Control Framework

To achieve the precise control of the deployable manipulator for vision-based grasping, the learning-based kinematic control framework is proposed and presented in Fig. 5. First, a 5-feature $\{\theta_1, \theta_2, X_i, Y_i, M_i\}$ dataset is built by collecting the moving platform's position and the corresponding joint angles under different loads in the real environment. Then, the generated dataset is used to train the learning-based model. The forward kinematics takes $\{\theta_1, \theta_2, M_i\}$ as the input and $\{X_i, Y_i\}$ as the output. In contrast, inverse kinematics takes $\{X_i, Y_i, M_i\}$ as the input and $\{\theta_1, \theta_2\}$ as the output. It should be noted that the forward and inverse kinematics are trained separately. Since the forward and inverse kinematics are basically similar in the training process, the following study takes inverse kinematics as the representative example. Finally, the kinematic control of the deployable manipulator is carried out based on the learning-based forward and inverse kinematics.

The load information, two actuator angles, and target coordinates are extracted during the control process to facilitate precise control. Real-time load information is acquired through the gravity sensor. According to the current actuator angles and load $(\theta_1', \theta_2', M)$, the coordinates of the moving platform (X, Y) can be determined by the forward kinematics. The target coordinates $({}^{C}X_t, {}^{C}Y_t)$ based on $\{C\}$ are obtained by depth camera through the target detection algorithm You Only Look Once (YOLO) and the depth information in the center of the bounding box.

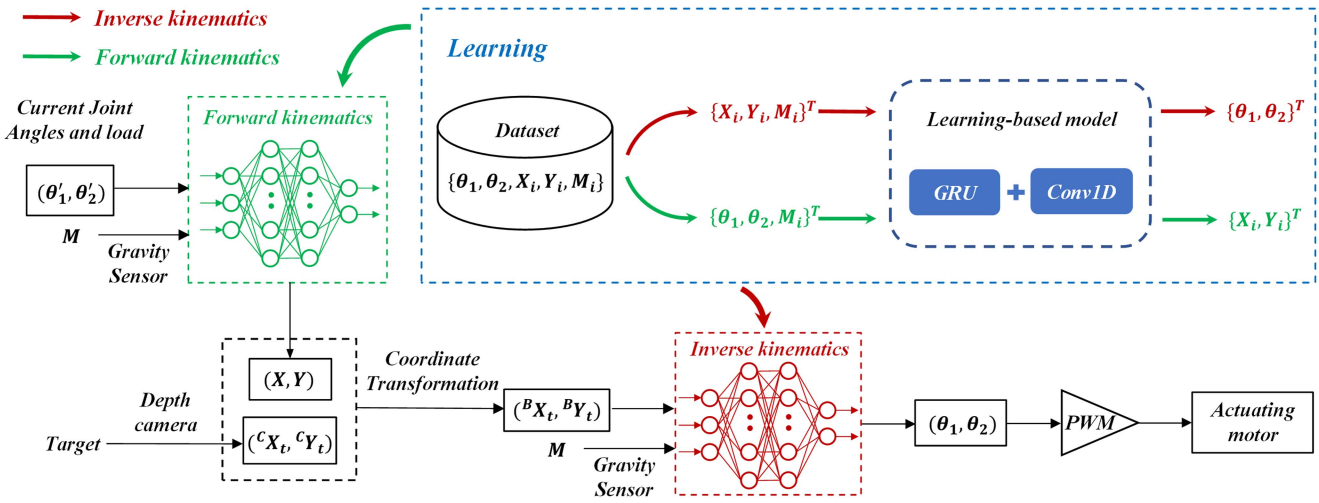


Fig. 5. Learning-based control framework of the deployable manipulator.

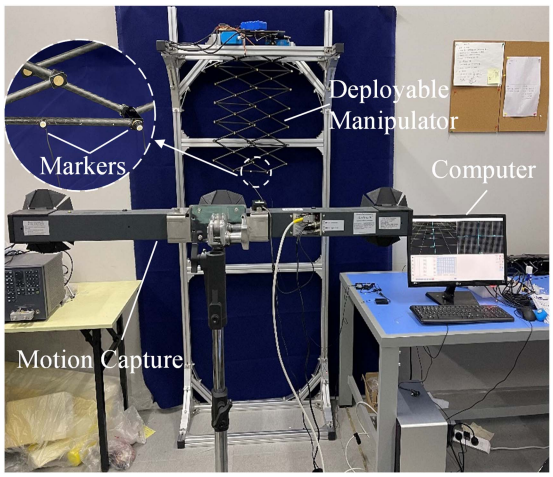


Fig. 6. Configuration of dataset building.

Since the relative position between the camera and the moving platform is fixed, the target coordinates $({}^{B}X_t, {}^{B}Y_t)$ can be calculated through coordinate transformation. After obtaining $({}^{B}X_t, {}^{B}Y_t, M)$, the corresponding joint angles (θ_1, θ_2) can be determined by using inverse kinematics. Then, the two actuator's joint angles are controlled by PWM to reach the target position.

C. Dataset Building and Analysis

Fig. 6 illustrates the configuration of dataset building, primarily comprising the deployable manipulator, a motion capture device (Visualeyez III VZ10K, Phoenix Technologies Inc.), markers, and a computer. The markers are attached to the moving platform and the origin of the base coordinates. The coordinates of the moving platform (X, Y) can be obtained by utilizing the optical motion capture device, which tracks the markers. Since the influence of load on the actual kinematics is evident, the load should be considered as one of the features of the dataset. Therefore, the datasets have 5-D features $\{\theta_1, \theta_2, X_i, Y_i, M_i\}$, i.e., two actuator angles, 2-D coordinates of the moving platform, and load. We collected the data under six load conditions $M_i \in \{0, 50, 100, 150, 200, 300\}$ g. The joint angles (θ_1, θ_2) are represented by pwm $\in [1100, 1700]$ because the two actuators are controlled by PWM. The linear relationship between joint angle and PWM is as follows:

$$\theta = \frac{1750 - \text{pwm}}{10}. \tag{8}$$

The distribution of the dataset plays a crucial role in the performance of the learning-based model. The impact of the dataset distribution is analyzed while disregarding the load feature $\{M_i\}$. Hence, the model only has two inputs $\{X_i, Y_i\}$ and two outputs $\{\theta_1, \theta_2\}$. The dataset contains 144 samples acquired under the load of 0 g. The ratio of the training set to the testing set is approximately 7:3. To ensure an accurate evaluation of the model's generalization ability, the testing set contains not less than 40 samples.

Fig. 7(a) illustrates a scenario where the range of the training set encompasses only a portion of the workspace, with points of the testing set situated outside this area. This type of situation is referred to as a disjoint distribution. As shown in Fig. 7(b), the training set is uniformly distributed in the entire workspace, which is called uniform distribution. The uniform distribution is the most common dataset distribution.

We propose a dataset distribution based on manipulability according to kinematic characteristics. As evident from Section II-B, the manipulability represents the kinematic performance of the mechanism. When the mechanism approaches singular positions, the manipulability decreases significantly, and the kinematic performance becomes poor. It is assumed that the actual kinematic model to be fitted by the learning-based method is P^*. In the region with low manipulability, P^* changes significantly. Conversely, P^* is relatively smooth in the region with high manipulability. For fitting the kinematic model P^*,

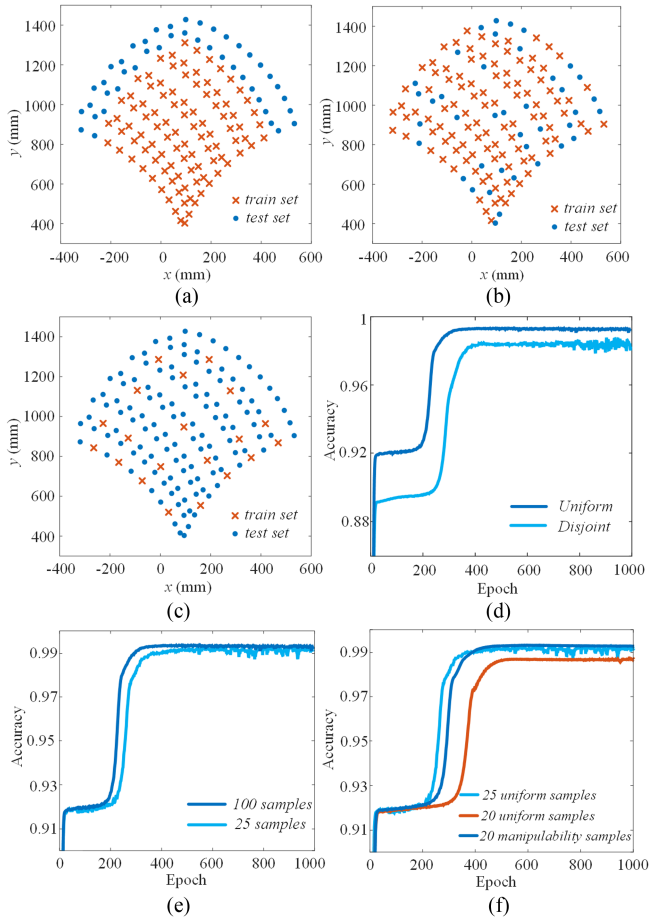


Fig. 7. Effect of the dataset distribution on the learning-based model. (a) Disjoint distribution. (b) Uniform distribution. (c) Manipulability distribution. (d) Comparison between disjoint distribution and uniform distribution with 100 samples. (e) Minimum number of samples required for uniform distribution. (f) Comparison between manipulability distribution and uniform distribution.

the region where P^* changes significantly require more samples, and samples can be appropriately reduced in the smooth region. Therefore, manipulability is utilized as a guide for dataset building. The manipulability dataset is shown in Fig. 7(c), where more sample points are selected as the manipulability decreases. Thus, the manipulability distribution dataset can characterize the kinematic model better in the case of a small sample size.

Fig. 7(d) compares the disjoint distribution and uniform distribution with 100 samples. The model attains an accuracy of only 98.3% when the dataset follows the disjoint distribution. In contrast, the result by uniform distribution can reach 99.4%, which is obviously better. Fig. 7(e) indicates the minimum number of samples required for uniform distribution. It should be noted that the model's accuracy has an upper limit because of measurement error, arrival position error, and other factors. Increasing sample size is not always beneficial because they also accumulate errors simultaneously. When the dataset is uniformly distributed, at least 25 samples are needed to achieve the same accuracy as 100 samples. Fig. 7(f) indicates that 20 manipulability distribution samples can achieve the same accuracy as 25 uniform distribution samples and better than the same number

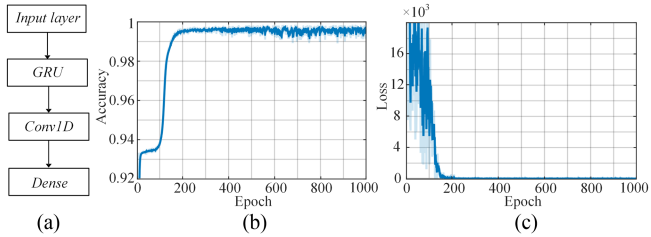


Fig. 8. (a) Configuration, (b) accuracy, and (c) loss for GRU model.

of uniform distribution samples. In general, the results show that the manipulability distribution dataset can characterize the kinematic model better in the case of a small sample size.

Therefore, the dataset should cover the operating workspace as far as possible, and the sample selection should be guided by manipulability. In this way, stable and satisfactory accuracy can be obtained with fewer samples. For planar kinematics modeling with 2-D inputs $\{X_i, Y_i\}$ and 2-D outputs $\{\theta_1, \theta_2\}$, achieving satisfactory accuracy requires only approximately 20 manipulability distribution samples, which is 20% less than the conventional uniform distribution. After further consideration of the feature $\{M_i\}$, it is verified by experiments that the model with 3-D input and 2-D output requires at least 100 samples to obtain satisfactory accuracy.

D. Learning-Based Model

GRU is one kind of deep learning model, which is another changed version of LSTM [37]. In general, GRU has fewer parameters than LSTM. The unique structure in GRU called "gate" is similar to that in the electronic circuit whose state can control the information flow. It enables the GRU to learn the inner complicated and nonlinear relationship between the input and output [37]. The GRU can turn the learning task easier if some samples are identical in certain features [28]. In the generated dataset, some samples are under the same load or identical in kinematic features. Convolution is distinguished by three characteristics, namely, sparse interactions, parameter sharing, and equivariant representations [38]. The convolutional layer is greatly superior to the dense layer in storage requirements and statistical efficiency [39].

The methodology employed for constructing the network is delineated as follows. Initially, the architecture consists of a singular gated unit and a final dense linear layer. Subsequently, the model is tested to find how many outputs of the layers maximize the accuracy and minimize the loss in the existing network configuration. Given the insufficiency of only one gated unit, a 1-D convolutional layer is added to stabilize the model performance and reduce the computational cost. Notably, adding extra GRU layers fails to enhance the model's accuracy.

Consequently, the proposed learning-based model is shown in Fig. 8(a). The model receives three inputs (X_i, Y_i, M_i) that feed the GRU layer, and then the GRU outputs a 1×40 vector. The 1-D convolutional layer is fed by the GRU memory cell and outputs a 1×40 weighted vector. The final dense linear layer is used to predict the target joint angles θ_1 and θ_2.

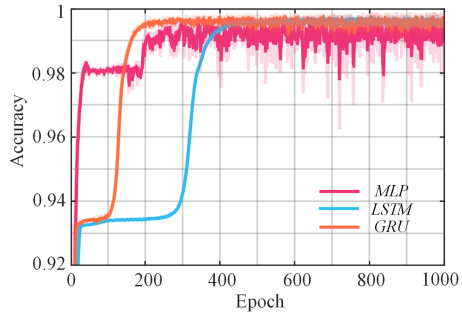


Fig. 9. Comparison of MLP, LSTM, and GRU models.

TABLE I
PERFORMANCE OF THE MLP, LSTM, AND GRU MODELS

Model	Metrics	
MLP	acc (mean±SD)	99.4%±0.26%
	loss	35.48
LSTM	acc (mean±SD)	99.5%±0.27%
	loss	20.18
GRU	acc (mean±SD)	99.7%±0.21%
	loss	18.95

The training dataset consists of 100 samples, which are 20 samples selected based on manipulability under loads of 0, 50, 100, 200, and 300 g. The testing set contains 70 samples acquired under a load of 150 g. The loss function used in the training process is mean squared error (MSE) loss. The model is compiled by using an Adam optimizer with an initial learning rate of 0.01. The network architecture and optimization framework are written in Python using PyTorch. All training and network computations are accelerated by graphics processing unit (GPU) on Nvidia GTX 1080 and Intel Core i7-10700K central processing unit (CPU).

In Fig. 8(b) and (c), the accuracy and loss for the GRU model are presented. It is noted that the data shown in the figure is smoothed, and the light-shaded part indicates the actual data. The result shows that the model achieves over 93% accuracy within the first 20 epochs. Then, it levels off at approximately 99.7% by about 200 epochs. In terms of the "loss" metric, the value fluctuates around 18.95 after 200 epochs.

We select MLP and LSTM for comparison with the model proposed in this article. The MLP approach is considered a good solution to kinematics problems as it generalizes well [40]. The MLP model has four dense linear layers between the input and output layers, and the neurons in each layer are 128, 64, 32, and 8. The LSTM is a general version of the GRU with similar memory cells. The model has one LSTM layer connected to an additional 1-D convolutional layer followed by a fully connected layer to predict the joint angles (θ_1, θ_2).

The performance results of the MLP, LSTM, and GRU models are shown in Fig. 9 and Table I. GRU and LSTM have similar accuracy, both higher than MLP, while the convergence rate of GRU is faster than LSTM. In terms of model stability, GRU and LSTM perform better than MLP. In conclusion, the GRU network exhibits the most favorable model performance.

IV. Experimental Validation

A. Trajectory Tracking

The system is tested in tracking two desired trajectories to evaluate the performance of the proposed kinematic control framework. The trajectories are triangular and circular, denoted respectively by T_1 and T_2. T_1 is an isosceles triangle with three vertices located at (−3, 1100), (397,1100), and (197,1300), while T_2 is a circle centered at (197, 1100) with a 200 mm radius. The trajectories are defined in the preprocessed 100 intermediate points in the feasible workspace.

Each trajectory is repeated five times with two loads ($M_1 = 0$ g and $M_2 = 250$ g). It should be noted that the load $M_2 = 250$ g does not exist in the training set or testing set. The control loop updates the expected position in line with the temporal sequence of intermediate points in each trajectory in the testing process. Meanwhile, it follows the learning-based model to solve the joint angles (θ_1, θ_2) corresponding to each intermediate point in real time. For comparison, trajectory tracking based on analytical models is also conducted. The optical motion capture device records the position of the moving platform during the tests.

Fig. 10 illustrates the results of the triangular trajectory tracking. In Fig. 10(a), the joint angles (θ_1, θ_2) deduced by the analytical and learning-based models for the triangular trajectory T_1 are represented by dotted and solid lines, respectively. The variation of the three curves is similar, but the error between the angles obtained by the analytical model and the ones by the learning-based model is obvious. The mean errors are approximately 7.9° under the load of 0 g and 10.4° under 250 g. The joint angles θ_1 and θ_2 are used as the input for the control, resulting in the trajectories shown in Fig. 10(b) and (c) under loads of 0 and 250 g, respectively. The trajectory error based on the analytical model is significant, and it further grows with the load increase. The trajectory based on the learning-based model demonstrates high accuracy, outperforming the analytical model, and exhibits excellent repeatability. With the objective of minimizing the root-MSE (RMSE), the trajectories of the analytical model under two loads are compensated in the X-axis and Y-axis directions, as depicted in Fig. 10(d) and (e). It shows that even with the offset, the learning-based model is more accurate to the truth. Fig. 10(f) illustrates that the learning-based model can achieve a high accuracy under different loads and generalize well because the load $M_2 = 250$ g does not exist in the dataset. Another finding is that the accuracy of the trajectory is higher at the beginning of the motion, however, it begins to decline as the motion continues. The plausible reasons for this phenomenon are that the position of the manipulator's motion depends on the previous state and that the errors accumulate as the motion proceeds.

Fig. 11 illustrates the results of the circular trajectory tracking. Most findings are consistent with the triangular trajectory results. In Fig. 11(a), the mean errors are approximately 8.0° and 10.6° under loads of 0 and 250 g, respectively. As shown in Fig. 11(b) and (c), the trajectory error is significant when the value of the Y-axis coordinate is small, corresponding to the state where the manipulator is relatively folded. The error along the Y-axis is much higher compared to the X-axis. Fig. 11(d) and (e) shows that the shape of the trajectory based on the

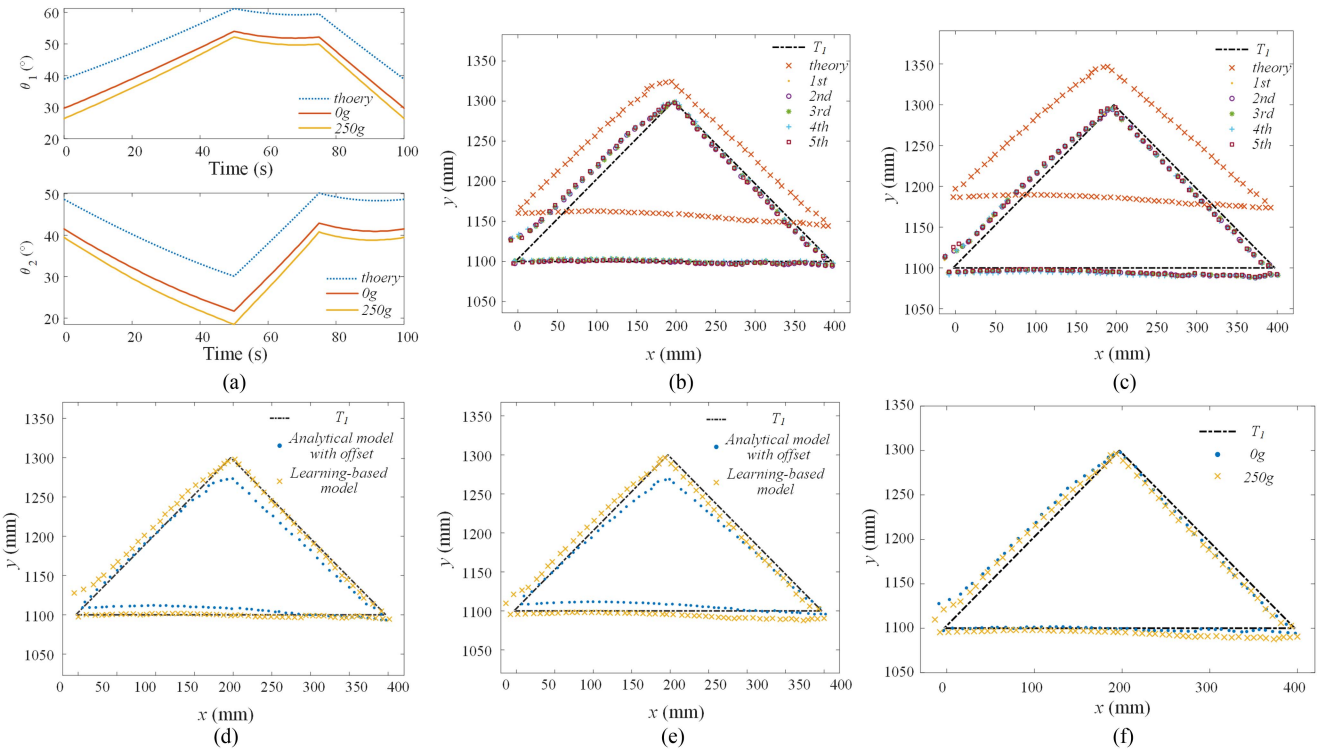


Fig. 10. Results of triangular trajectory tracking based on the analytical and learning-based models. (a) Joint angles (θ_1, θ_2) deduced by the analytical and learning-based models. (b) and (c) Triangular trajectories under loads of 0 and 250 g. (d) and (e) Comparison of trajectories based on the analytical model with offset and learning-based model under loads of 0 and 250 g. (f) Comparison of trajectories under two loads.

TABLE II
SUMMARY OF THE PERFORMANCE OF THE LEARNING-BASED MODEL IN TWO TRAJECTORIES UNDER DIFFERENT LOADS

Trajectory	RMSE (mm)	Error range (mm)		Repeatability (mm)
		x	y	
T_1				
0 g	6.58	(-6.25, 1.66)	(-5.97, 7.00)	± (0.65, 0.91)
250 g	8.86	(-9.94, 9.50)	(-11.85,14.44)	± (0.58, 0.86)
T_2				
0 g	11.46	(-10.22, 3.66)	(-4.79, 30.90)	± (0.98, 1.58)
250 g	10.21	(-10.63, 7.79)	(-18.38, 23.21)	± (0.69, 0.68)
Average				
0 g	9.02	(-8.24, 2.66)	(-5.38, 18.95)	± (0.82, 1.25)
250 g	9.54	(-10.29, 8.65)	(-15.12, 18.83)	± (0.64, 0.77)

TABLE III
SUMMARY OF THE PERFORMANCE OF THE ANALYTICAL MODEL IN TWO TRAJECTORIES UNDER DIFFERENT LOADS

Trajectory	RMSE (mm)		Error range (mm)	
	Original	With offset	x	y
T_1				
0 g	37.34	10.15	(-7.47, 8.78)	(24.27, 68.08)
250 g	55.71	11.64	(-10.89, 7.16)	(46.72, 97.12)
T_2				
0 g	42.13	12.86	(-18.73, 24.01)	(30.16, 83.84)
250 g	62.21	16.04	(-10.45, 10.93)	(51.02, 122.23)
Average				
0 g	39.74	11.51	(-16.84, 16.40)	(27.22, 75.96)
250 g	58.96	13.84	(-10.67, 9.05)	(48.87, 109.68)

learning-based model is closer to the target circular trajectory. Fig. 11(f) demonstrates that the precision of the circular motion is inferior to that of the triangular motion.

Tables II and III present the performance of the kinematic control based on the learning-based model and analytical model in terms of RMSE (average of five repetitions), error range, and repeatability (standard deviation of the five trajectories).

The results indicate that the control scheme is fairly accurate in tracking the desired trajectories, as the average RMSEs for the two trajectories are 9.02 and 9.54 mm under loads of 0 and 250 g, respectively. In contrast, the average RMSEs of the two trajectories based on the analytical model are 39.74 and 58.96 mm. In the case of 0 and 250 g load, the proposed scheme reduces the errors to less than one-fourth and one-sixth of the errors of the analytical model, respectively. The RMSE values of the learning-based model and analytical model with offset illustrate that even with the offset, the learning-based model is more accurate to the truth. By comparing the results of T_1 and T_2, the deployable manipulator performs better when it moves in a straight line. Moreover, the system shows excellent

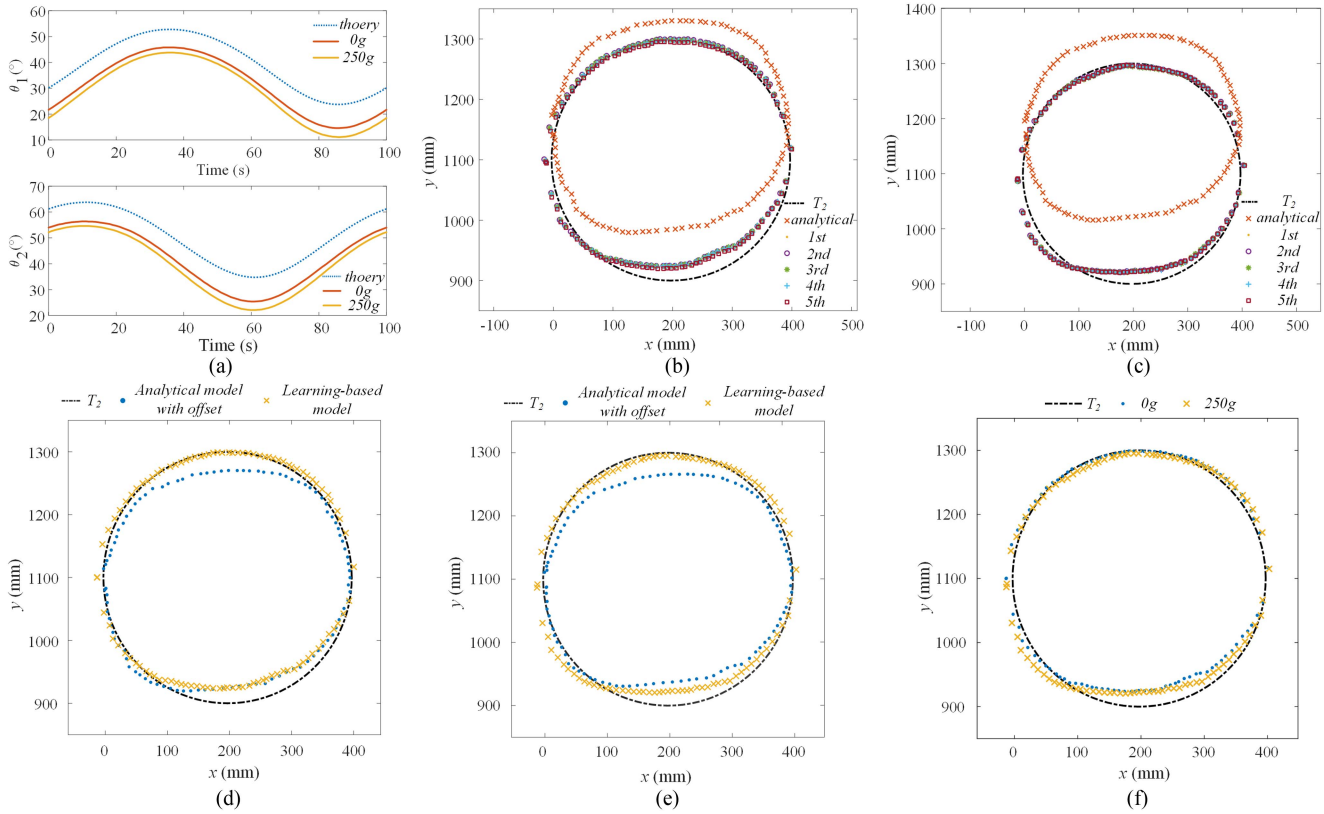


Fig. 11. Results of circular trajectory tracking based on the analytical and learning-based models. (a) Joint angles (θ_1, θ_2) based on the analytical and learning-based models. (b) and (c) Circular trajectories under loads of 0 and 250 g. (d) and (e) Comparison of trajectories based on the analytical model with offset and learning-based model under loads of 0 and 250 g. (f) Comparison of circular trajectories under two loads.

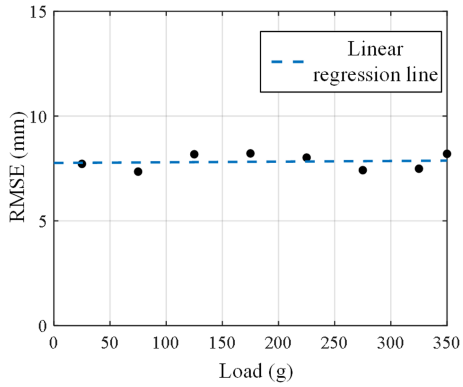


Fig. 12. RMSE of triangular trajectory under different loads.

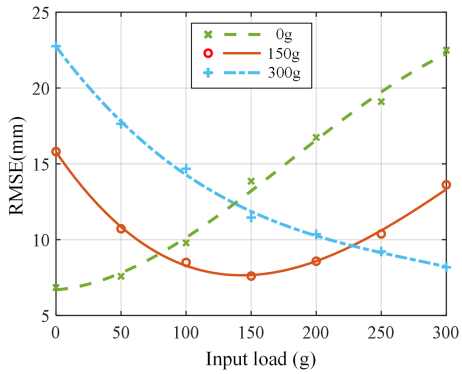


Fig. 13. Impact of input load errors on the trajectory accuracy.

repeatability, evidenced by the standard deviation of the five repetitive trajectories is 0.58–1.58 mm.

B. Generalization and Impact of Input Error on Accuracy

Triangular trajectory tracking experiments are conducted at loads of 25, 75, 125, 175, 225, 275, 325, and 350 g to further validate that the control system can work continuously within the acceptable load range. It is worth mentioning that these loads are not present in the training set or testing set, and the loads of 325 and 350 g exceed the range of the training set. The RMSE for each load is presented in Fig. 12, demonstrating good accuracy over the entire load range. The average RMSE is 7.83 mm. Notably, the precision is almost unaffected by load variations, as indicated by the nearly horizontal trend of the fitted linear regression line with a slope of 0.037. The results show the excellent generalization ability of the proposed control system.

To analyze how the system's accuracy is affected when the payload is entered incorrectly, triangular trajectory experiments are conducted under actual loads of 0, 150, and 300 g, while varying the input loads from 0 to 300 g in 50 g increments. Fig. 13 illustrates the experimental results, with points representing the average values of three trials and curves representing the fitted

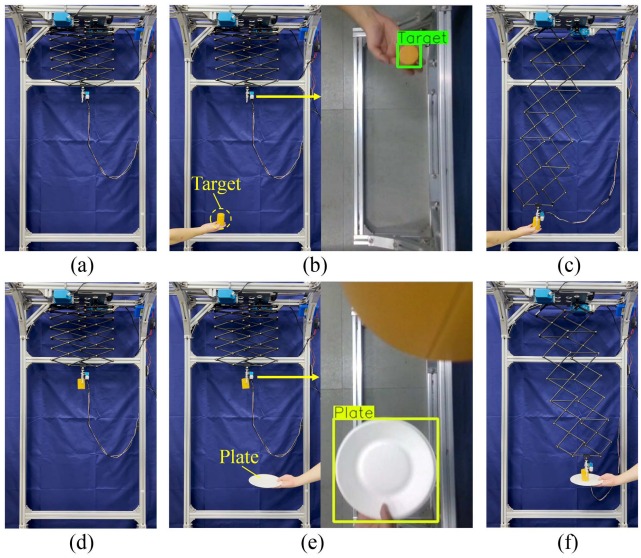


Fig. 14. Target grasping experiment. (a) Initial state. (b) Detecting target. (c) Grasping target. (d) Retracting to original. (e) Detecting plate. (f) Releasing target.

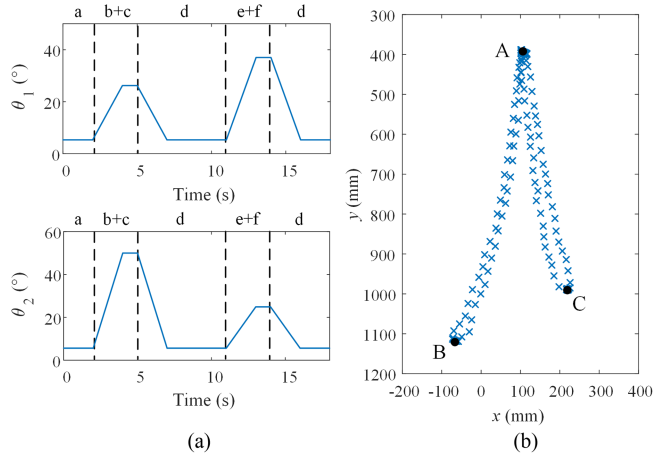


Fig. 15. (a) Variation of two joint angles during the grasping. (b) Trajectory of the experiment.

lines. The results show that the error increases as the discrepancy in input load grows, with an increasing rate of change. Within a 50 g input error, the RMSE difference remains relatively small, approximately 1 mm. However, when the input error reaches 300 g, the error increases to approximately 22 mm, nearly three times the RMSE of the correct input. Notably, when the input load is set at 150 g, the system achieves relatively small errors regardless of the actual load value within the acceptable range.

Thus, the control system can still operate effectively even when the gravity sensor has a certain degree of error (± 50 g). The system exhibits insensitivity to noise within a certain range, reflecting its good robustness. Furthermore, in the event of a malfunctioning gravity sensor that cannot provide load feedback, the system can still operate with relatively small errors by using an input load of 150 g. This demonstrates that our system possesses a degree of resilience.

C. Target Grasping

As stated in the Introduction, conventional methods face significant challenges in achieving successful target grasping due to the complexity of the manipulator. The target grasping experiment is conducted to demonstrate that the proposed learning-based approach can achieve precise control under varying loads in real time. Fig. 14(a) illustrates the initial state of the manipulator under a load of $M_0 = 260$ g (mass of gravity sensor, depth camera, and gripper). When the depth camera detects the target [see Fig. 14(b)], the manipulator reaches the target position based on the proposed control framework and grasps the target [see Fig. 14(c)]. Then, the load feature changes to $M_1 = 295$ g because the target is 35 g which is obtained by the gravity sensor. Next, the manipulator retracts to the original folded configuration, as shown in Fig. 14(d). When the depth camera detects the plate [see Fig. 14(e)], the manipulator reaches the position of the plate and places the target on the plate [see Fig. 14(f)]. Afterward, the load feature returns to $M_0 = 260$ g based on the gravity sensor, and the manipulator returns to the initial state to prepare for the next capture.

Fig. 15 demonstrates the variation of two joint angles and the trajectory of the target grasping experiment. Fig. 15(a) is divided into five parts by the dotted line, representing the different stages of the grasping experiment: (a) initial state, (b) detecting target, (c) grasping target, (d) retracting to original, (e) detecting plate, and (f) releasing target. In Fig. 15(b), points A, B, and C represent the initial position, target position, and position of the plate, respectively. The experimental results show that the proposed learning-based kinematic control scheme can accurately grasp and place targets of arbitrary mass over long distances.

V. Conclusion

This article proposes a learning-based kinematic control framework to address the issue of positioning error in the deployable manipulator with long span and low stiffness. The dataset is built by collecting data in the real environment, and the load factor, which significantly influences the actual kinematics, is taken as an extra feature. We propose a dataset building method based on manipulability according to the kinematic characteristics. In this way, the learning-based model only needs 100 samples, 20% less than the conventional uniform distribution, and can achieve 99.7% accuracy. A learning-based model consisting of GRU and 1-D convolutional layer is proposed, which is lighter and more effective than existing methods. The trajectory tracking experiments show the system's high accuracy, with an average RMSE of 9.02 mm under a load of 0 g and 9.54 mm under 250 g. The experimental results prove that our approach possesses strong generalization capabilities and may offer benefits in terms of robustness and resilience. Target grasping experiments are further conducted to validate the feasibility and practicality of this kinematic control. In conclusion, the proposed control scheme can achieve precise control under variable loads in real time with low computational costs. In addition, it is highly portable and easy to implement.

It is noted that the proposed approach would be preferred over the conventional approach if both of the following circumstances are satisfied.

1) The manipulator is complex or flexible, making it challenging to develop an accurate analytical model.
2) With the same factors considered in the learning-based model, the motion of the manipulator needs high repeatability.

This article mainly focuses on the static issues related to kinematic control. Future research can concentrate on dynamic problems, such as dynamic grasping and addressing oscillation during operation, to achieve more precise motion control for manipulators with large span and low stiffness.

REFERENCES

[1] M. Carricato and V. Parenti-Castelli, "A family of 3-DOF translational parallel manipulators," *J. Mech. Des., Trans. ASME*, vol. 125, no. 2, pp. 302–307, Jun. 2003.

[2] S. Yang, T. Sun, T. Huang, Q. Li, and D. Gu, "A finite screw approach to type synthesis of three-DOF translational parallel mechanisms," *Mechanism Mach. Theory*, vol. 104, pp. 405–419, Oct. 2016.

[3] C. Wu, X.-J. Liu, L. Wang, and J. Wang, "Optimal design of spherical 5R parallel manipulators considering the motion/force transmissibility," *Trans. ASME*, vol. 132, no. 3, pp. 0310021–03100210, 2010.

[4] B. Li, Y. Li, and X. Zhao, "Kinematics analysis of a novel over-constrained three degree-of-freedom spatial parallel manipulator," *Mechanism Mach. Theory*, vol. 104, pp. 222–233, Oct. 2016.

[5] G. Wu, S. Bai, and P. Hjørnet, "Architecture optimization of a parallel Schönflies-motion robot for pick-and-place applications in a predefined workspace," *Mechanism Mach. Theory*, vol. 106, pp. 148–165, Dec. 2016.

[6] L.-T. Schreiber and C. Gosselin, "Schönflies motion parallel robot (SPARA): A kinematically redundant parallel robot with unlimited rotation capabilities," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 5, pp. 2273–2281, Oct. 2019.

[7] X. J. Liu, Q. I. M. Wang, and J. Wang, "Kinematics, dynamics and dimensional synthesis of a novel 2-DoF translational manipulator," *J. Intell. Robot. Syst.: Theory Appl.*, vol. 41, no. 4, pp. 205–224, Jan. 2005.

[8] T. Huang, S. Liu, and D. G. Chetwynd, "Optimal design of a 2-DOF pick-and-place parallel robot using dynamic performance indices and angular constraints," *Mechanism Mach. Theory*, vol. 70, pp. 246–253, Dec. 2013.

[9] Y. Yang, Y. Peng, H. Pu, and Q. Cheng, "Design of 2-degrees-of-freedom (DOF) planar translational mechanisms with parallel linear motion elements for an automatic docking device," *Mechanism Mach. Theory*, vol. 121, pp. 398–424, Mar. 2018.

[10] Z. You and S. Pellegrino, "Foldable bar structures," *Int. J. Solids Struct.*, vol. 34, no. 15, pp. 1825–1847, 1997.

[11] G. Kiper, E. Söylemez, and A. U. O. Kişisel, "A family of deployable polygons and polyhedra," *Mechanism Mach. Theory*, vol. 43, no. 5, pp. 627–640, 2008.

[12] X. Ding and J. S. Dai, "Design and kinematic analysis of a novel prism deployable mechanism," *Mechanism Mach. Theory*, vol. 63, pp. 35–49, 2013.

[13] D. J. Gonzalez and H. H. Asada, "Design and analysis of 6-DOF triple scissor extender robots with applications in aircraft assembly," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1420–1427, Jul. 2017.

[14] Y. Yang, Y. Tian, Y. Peng, and H. Pu, "A novel 2-DOF planar translational mechanism composed by scissor-like elements," *Mech. Sci.*, vol. 8, no. 1, pp. 179–193, 2017.

[15] Y. Yang, L. Tang, H. Zheng, Y. Zhou, Y. Peng, and S. Lyu, "Kinematic stability of a 2-DOF deployable translational parallel manipulator," *Mechanism Mach. Theory*, vol. 160, Jun. 2021, Art. no. 104261.

[16] P. D. Lin and J. F. Chen, "Accuracy analysis of planar linkages by the matrix method," *Mechanism Mach. Theory*, vol. 27, no. 5, pp. 507–516, 1992.

[17] W. Wu and S. S. Rao, "Interval approach for the modeling of tolerances and clearances in mechanism analysis," *J. Mech. Des.*, vol. 126, no. 4, pp. 581–592, 2004.

[18] J. W. Wittwer, K. W. Chase, and L. L. Howell, "The direct linearization method applied to position error in kinematic linkages," *Mechanism Mach. Theory*, vol. 39, no. 7, pp. 681–693, 2004.

[19] M. J. Tsai and T. H. Lai, "Accuracy analysis of a multi-loop linkage with joint clearances," *Mechanism Mach. Theory*, vol. 43, no. 9, pp. 1141–1157, 2008.

[20] M. J. Tsai and T. H. Lai, "Kinematic sensitivity analysis of linkage with joint clearance based on transmission quality," *Mechanism Mach. Theory*, vol. 39, no. 11, pp. 1189–1206, 2004.

[21] W. J. Zhang and K. V. D. Werff, "Automatic communication from a neutral object model of mechanism to mechanism analysis programs based on a finite element approach in a software environment for CADCAM of mechanisms," *Finite Elements Anal. Des.*, vol. 28, no. 3, pp. 209–239, 1998.

[22] X. Hu, L. Cao, Y. Luo, A. Chen, E. Zhang, and W. J. Zhang, "A novel methodology for comprehensive modeling of the kinetic behavior of steerable catheters," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 4, pp. 1785–1797, Aug. 2019.

[23] M. Jolaei, A. Hooshiar, and M. Packirisamy, "Toward task autonomy in robotic cardiac ablation: Learning-based kinematic control of soft tendon-driven catheters," *Soft Robot.*, vol. 8, no. 3, pp. 340–351, 2021.

[24] P. Jha, "Inverse kinematic analysis of robot manipulators," Ph.D. dissertation, Nat. Inst. Technol. Rourkela, India., 2015.

[25] D. R. Raj, I. J. Raglend, and M. D. Anand, "Inverse kinematics solution of a five joint robot using feed forward and radial basis function neural network," in *Proc. Int. Conf. Comput. Power, Energy, Inf. Commun.*, 2015, pp. 117–122.

[26] A. Zubizarreta, M. Larrea, E. Irigoyen, I. Cabanes, and E. Portillo, "Real time direct kinematic problem computation of the 3PRS robot using neural networks," *Neurocomputing*, vol. 271, pp. 104–114, Jan. 2018.

[27] H. Wang, X. Wang, W. Yang, Z. Du, and Z. Yan, "Construction of controller model of notch continuum manipulator for laryngeal surgery based on hybrid method," *IEEE/ASME Trans. Mechatron.*, vol. 26, no. 2, pp. 1022–1032, Apr. 2021.

[28] J. S. Toquica, P. S. Oliveira, W. S. R. Souza, J. M. S. T. Motta, and D. L. Borges, "An analytical and a deep learning model for solving the inverse kinematic problem of an industrial parallel robot," *Comput. Ind. Eng.*, vol. 151, Jul. 2020, Art. no. 106682.

[29] F. Wang, Z. Qian, Z. Yan, C. Yuan, and W. Zhang, "A novel resilient robot: Kinematic analysis and experimentation," *IEEE Access*, vol. 8, pp. 2885–2892, 2020.

[30] T. Zhang and M. M. Gupta, "Resilient robots: Concept, review, and future directions," *Robotics*, vol. 6, no. 4, Sep. 2017, Art. no. 22.

[31] L. Jin, S. Li, J. Yu, and J. He, "Robot manipulator control using neural networks: A survey," *Neurocomputing*, vol. 285, pp. 23–34, Apr. 2018.

[32] A. R. J. Almusawi, L. C. Dülger, and S. Kapucu, "A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm (Denso VP6242)," *Comput. Intell. Neurosci.*, vol. 2016, 2016, Art. no. 5720163.

[33] S. F. M. Assal, K. Watanabe, and K. Izumi, "Neural network-based kinematic inversion of industrial redundant robots using cooperative fuzzy hint for the joint limits avoidance," *IEEE/ASME Trans. Mechatron.*, vol. 11, no. 5, pp. 593–603, Oct. 2006.

[34] X. Li, X. Ding, and G. S. Chirikjian, "Analysis of angular-error uncertainty in planar multiple-loop structures with joint clearances," *Mechanism Mach. Theory*, vol. 91, pp. 69–85, Sep. 2015.

[35] Q. Zhao and Z. Liu, "Analysis of angular errors of the planar multi-closed-loop deployable mechanism with link deviations and revolute joint clearances," *Aerosp. Sci. Technol.*, vol. 87, pp. 25–36, Apr. 2019.

[36] K. L. Ting, K. L. Hsu, Z. Yu, and J. Wang, "Clearance-induced output position uncertainty of planar linkages with revolute and prismatic joints," *Mechanism Mach. Theory*, vol. 111, pp. 66–75, May 2017.

[37] C. Hu, T. Ou, H. Chang, Y. Zhu, and L. Zhu, "Deep GRU neural network prediction and feedforward compensation for precision multiaxis motion control systems," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 3, pp. 1377–1388, Jun. 2020.

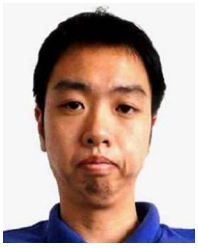
[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[39] C. Cheng et al., "A deep learning-based remaining useful life prediction approach for bearings," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 3, pp. 1243–1254, Jun. 2020.

[40] O. Lakhal, A. Melingui, and R. Merzouki, "Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 3, pp. 1326–1335, Jun. 2016.

Hu Liu received the B.Eng. and M.Eng. degrees in naval architecture and ocean engineering from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2014 and 2017, respectively. He is currently working toward the doctoral degree in mechanical engineering with Shanghai University, Shanghai, China.

His research interests include deployable mechanisms and machine learning.



Yi Yang received the B.Eng. degree in mechanical engineering from Harbin Engineering University, Harbin, China, in 2004, and the M.Eng. and Dr. Eng. degrees from Beihang University, Beijing, China, in 2007 and 2012, respectively, both in mechanical engineering.

He is currently a Professor with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China. His research interests include robotics and mechanism design.

Yudi Zhao received the Ph.D. degree in control science and engineering from Donghua University, Shanghai, China, in 2022.

She is a Postdoctoral Researcher with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai. Her scientific interests include deep learning and computer vision.

Yang Yang received the Ph.D. degree in mechatronics from Ritsumeikan University, Kyoto, Japan, in 2015.

From 2015 to 2016, he was a Postdoctoral Researcher with the Department of Mechanical and Aerospace Engineering, Tokyo Institute of Technology, Tokyo, Japan. He is a Lecturer with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China. His research interests include soft robots and mobile robots.

Yan Peng received the Ph.D. degree in pattern recognition and intelligent Systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2009.

She is a Professor with Shanghai University, Shanghai, China, and also a Dean of the Research Institute, USV Engineering, Shanghai. Her current research interests include modeling and control of USV, field robotics, and locomotion system.

Huayan Pu received the M.Sc. and Ph.D. degrees in mechatronics engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2011, respectively.

She is currently a Professor with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China. Her current research interests include vibration controlling and robotics.

Yang Zhou received the B.S. degree in mechanical engineering from Southwest Petroleum University, Chengdu, China, in 2013, the Ph.D. degree in fluid dynamics from Fudan University, Shanghai, China, in 2018.

He is a Research Associate with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai. His current research interests include understanding the representation of computer vision and image processing with theoretical description.