Strojniški vestnik - Journal of Mechanical Engineering 68(2022)3, 175-184
© 2022 The Authors. CC BY 4.0 Int. Licensee: SV-JME
DOI:10.5545/sv-jme.2021.7455
Original Scientific Paper
Received for review: 2021-11-01
Received revised form: 2022-01-14
Accepted for publication: 2022-02-09

Error Prediction for a Large Optical Mirror Processing Robot Based on Deep Learning

Zujin Jin[1] – Gang Cheng[1,2,*] – Shichang Xu[1] – Dunpeng Yuan[1]
[1] China University of Mining and Technology, School of Mechatronic Engineering, China
[2] Shangdong Zhongheng Optoelectronic Technology Co., China

Predicting the errors of a large optical mirror processing robot (LOMPR) is very important when studying a feedforward control error compensation strategy to improve the motion accuracy of the LOMPR. Therefore, an end trajectory error prediction model of a LOMPR based on a Bayesian optimized long short-term memory (BO-LSTM) was established. First, the batch size, number of hidden neurons and learning rate of LSTM were optimized using a Bayesian method. Then, the established prediction models were used to predict the errors in the X and Y directions of the spiral trajectory of the LOMPR, and the prediction results were compared with those of back-propagation (BP) neural network. The experimental results show that the training time of the BO-LSTM is reduced to 21.4 % and 15.2 %, respectively, in X and Y directions than that of the BP neural network. Moreover, the MSE, RMSE, and MAE of the prediction error in the X direction were reduced to 9.4 %, 30.5 %, and 31.8 %, respectively; the MSE, RMSE, and MAE of the prediction error in the Y direction were reduced to 9.6 %, 30.8 %, and 37.8 %, respectively. It is verified that the BO-LSTM prediction model could improve not only the accuracy of the end trajectory error prediction of the LOMPR but also the prediction efficiency, which provides a research basis for improving the surface accuracy of an optical mirror.

Keywords: Bayesian optimization, BO-LSTM, error prediction, optical mirror processing, hybrid manipulator, hyperparametrics

Highlights

- *An end trajectory error prediction model based on a Bayesian optimized long short-term memory was established.*
- *The established prediction models were used to predict the spiral trajectory errors of the large optical mirror processing robot.*
- *The batch size, number of hidden neurons and learning rate of LSTM were optimized using a Bayesian method.*
- *The experimental results showed that the Bayesian optimized long short-term memory error prediction model could improve not only the accuracy of the end trajectory error prediction of the large optical mirror processing robot but also the prediction efficiency.*

0 INTRODUCTION

With the development of the information age, modern optical systems are rapidly developing towards features including large aperture, high precision, high resolution, and high power. The requirements for LOMPS are also increasing. In order to realize the rapid development of modern optics and keep up with the development trends of advanced optical systems, LOMPS suitable for large-scale, aspherical, efficient and precise optical mirror processing have become a key technology to be solved urgently. Optical mirror processing includes four steps: rough grinding, milling forming, fine grinding, and polishing [1] and [2]. A fixed trajectory is used for the large optical mirror processing robot (LOMPR) in these four steps, with a grid trajectory, concentric circle trajectory, or spiral trajectory commonly used. The dynamic characteristics and control parameters of the LOMPR under a specific trajectory also show periodic changes [3] and [4]. Moreover, the processing of optical mirrors requires multiple iterations. The process of fabricating a large finished optical mirror from a blank can last from several days to several months [5] to [7]. Determining the long-term periodic movement would aid in the mathematical statistical analysis of the non-linear error and predicting the direction and size of the error in the future, which will allow countermeasures to be taken in advance based on the predicted results [8] and [9]. Therefore, it is necessary to predict the trajectory error of the LOMPR based on the previous processing parameters to compensate for the uncertainty of the model and improve the surface accuracy and processing efficiency of the optical mirror [10] to [12].

Deep learning has rapidly developed in recent years, and many algorithms are suitable for non-linear fuzzy data processing. In fact, many scholars have applied deep learning and some intelligent analysis methods to the research field of traditional robots. Dai et al. applied a long short-term memory (LSTM) model to vehicle trajectory prediction [13]. Mici et al. [14] predicted the future instructions of a motor using a neural network structure to compensate for the delay error. To solve the trajectory tracking problem of a wheeled mobile robot under non-holonomic constraints and in the presence of model uncertainties, Mirzaeinejad et al. [15] derived and optimized a control law by minimizing a pointwise quadratic cost function for the predicted tracking errors of a wheeled mobile

*Corr. Author's Address: China University of Mining and Technology, School of Mechatronic Engineering, Nanhu campus, No. 1 University Road, 221116, Xuzhou, China, chg@cumt.edu.cn

robot. Wang et al. [16] proposed an adaptive Jacobian controller based on the prediction error to realize an adaptable kinematic and dynamic parameters driven by the prediction error. Las Casas et al. [17] established a fully connected feedforward artificial neural network with supervised learning to predict the error between the welding parameters of a welding robot. In order to predict the motion direction of a robot's end tool path in real time, Wang et al. [18] proposed that the robot's end tool path be guided by a human in off-line training of the LSTM to generate a trajectory predictor. Zhang et al. [19] predicted the torque of a six-degrees-of-freedom (6-DOF) cooperative robot based on a compensated cooperative robot dynamic model of an LSTM. Yang et al. [20] predicted the fault of a robot in a dynamic working state based on deep learning. Chebbi et al. [21] carried out a sensitivity analysis and positioning error limit prediction for a 3-DOF translational parallel robot. Zatout et al. [22] analysed the optimal output problem of a fuzzy logic controller for quadrotor attitude stabilisation through particle swarm optimization, bat algorithm and cuckoo search, and obtained the optimal performance of bat method. The above scholars applied deep learning to the fields of robot parameter optimization, trajectory prediction, fault diagnosis, and robot positioning, and achieved good results. However, few scholars have conducted relevant research on the error prediction of the end execution point of a hybrid robot with large load and coupling characteristics [23] to [25]. The trajectory error of the end execution point of a robot can be predicted using deep learning, which can provide feedforward compensation data for the servo control system.

However, not all deep learning models can be applied to different engineering fields. In order to cope with the characteristics of different engineering states and give deep learning algorithms a better fusion effect, many scholars have improved or optimized them. Li et al. [26] proposed an LSTM training method based on evolutionary attention and a competitive random search, which was applied to multivariate time series forecasting. Ahmad et al. [27] proposed a predictive control strategy for a four-degree-of-freedom half-car model in the presence of an active pneumatic surface to improve the attitude control ability of the vehicle. Ullah et al. [28] proposed model predictive control and H∞ control of a 6-DOF manipulator to improve the trajectory accuracy of the robot. Ali et al. [29] used a particle swarm optimization algorithm to optimize the gain of proportional-integral-derivative, the weighting matrices of linear quadratic regulator, and their ratio of contributions, so as to improve the performance of the controller and obtain the accurate trajectory tracking ability of the controller. Bai et al. [30] analysed the performances of linear model predictive control, linear error model predictive control, non-linear model predictive control, and non-linear error model predictive control for mobile robot path tracking. To solve the time delay compensation problem for a non-linear teleoperation system, Shen et al. [31] proposed a motion prediction method based on a cascaded structure state observer. Tang et al. [32] predicted the wind speed range through particle swarm optimization of a deep learning network. Wang et al. [33] designed a high-performance detection model based on a superimposed LSTM to realize unmanned aerial vehicle real-time fault detection using a statistical threshold. Mei et al. [34] proposed a one-dimensional convolutional LSTM based on a vibration terrain classification method to learn both the spatial and temporal characteristics of dampened vibration signals. Wu et al. [35] predicted the health state of an aviation turbofan engine using a vanilla LSTM. Focusing on the unique attributes of the engineering field studied, the above scholars improved the existing deep learning algorithm, optimized the search algorithm, optimized the parameters, adjusted the deep learning framework to predict the data, and achieved ideal results. This background provides a reliable theoretical basis for predicting the error of the end execution point of a LOMPR using a Bayesian optimized long short-term memory (BO-LSTM).

Effectively predicting the end execution trajectory error of a LOMPR is very important when compensating for the robot's feedforward error and improving the surface accuracy of an optical mirror. This study established an end trajectory error prediction model for a LOMPR based on the Bayesian hyperparametric optimization of an LSTM. It can effectively predict the errors of the LOMPR in X and Y directions under different processing trajectories. Based on the above introduction, the structure of this paper is as follows. Section 1 introduces the structure of the LOMPR and the tool path of the robot when a computer controlled optical surfacing (CCOS) grinding system is used. Section 2 shows how the LSTM deep learning model was constructed based on Bayesian optimization. Section 3 shows how the effectiveness of the prediction model under the spiral trajectory was verified using an experimental analysis. Finally, Section 4 gives the conclusion of this paper.

1 LARGE OPTICAL MIRROR PROCESSING ROBOT

Based on the requirements of the optical mirror grinding process, the LOMPR needs at least five degrees of freedom (5-DOF). The LOMPR used in this study was a 5-DOF hybrid robot developed using a 3-DOF parallel (UPS+UP) manipulator combined with a 2-DOF serial manipulator. The topology of the LOMPR is shown in Fig. 1. Among them, the parallel part of the moving platform and fixed platform were connected by three UPS driven branches and one UP constrained branch. A world coordinate system (*O-XYZ*) was established, where *O* was the intersection of the horizontal axis of the fixed platform centre point and the vertical axis of the optical mirror. The *X*-axis pointed in the negative direction of point O_1, the *Z*-axis was vertically downward, and the *Y*-axis was determined according to the right-hand rule. The rotation axis of the primary rotary head was collinear with the UP branch chain, and the rotation axis of the secondary rotary head was perpendicular to the primary rotary head and parallel to the *Y*-axis. A reference coordinate system $O_{\mathrm{A}i}\text{–}x_{\mathrm{A}i}y_{\mathrm{A}i}z_{\mathrm{A}i}$ is established on the A_i, where $O_{\mathrm{A}i}$ is the central hinge point of the Hooke hinge of the fixed platform, the $z_{\mathrm{A}i}$-axis is downward along the driving branch chain, the $y_{\mathrm{A}i}$-axis is perpendicular to the driving branch chain along the internal axis of Hooke hinge, and the $x_{\mathrm{A}i}$-axis follows the right-hand rule. The grinding tool used the CCOS grinding system. The CCOS grinding system consists of a rotating motor and a revolving motor. Among them, a pneumatic suspension device is designed above the rotation axis. When the LOMPR produces a position error in the *Z*-axis direction, it will push the suspension axis up and down to compensate for the error. Therefore, the error caused by the end trajectory of the LOMPR in the *Z*-axis direction will not affect the surface processing accuracy of the optical mirror.

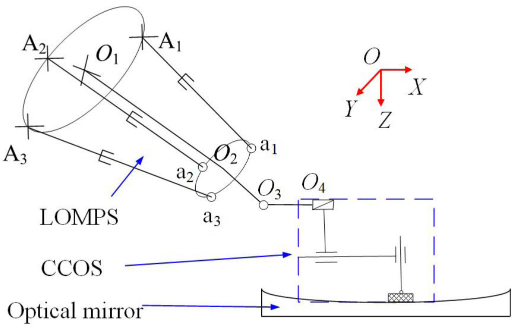

Fig. 1. *Topology of LOMPR*

The position error of the end execution point of the LOMPR is the result of the coupling of multiple influencing factors. The serial manipulator of the LOMPR has a relatively stable structural size, low motion speed and small stroke, so the error caused by the serial manipulator is very small. Here, the position error caused by the parallel module UPS branch chain is considered. The main factors affecting the error of a single UPS branch chain are the friction of moving pair, elastic deformation and servo system error, so the error of a single UPS branch chain can be expressed as:

$$\mathbf{e}_{li} = \mathbf{e}_{fi} + \mathbf{e}_{di} + \mathbf{e}_{ci} + \mathbf{e}_{oi} \quad i = 1, 2, 3, \tag{1}$$

where $\mathbf{e}_{li}$ is the comprehensive error of UPS branch chain *i*, $\mathbf{e}_{fi}$ is the error caused by friction of moving pair, $\mathbf{e}_{di}$ is the error caused by force deformation of branch components, $\mathbf{e}_{ci}$ is the error of servo system, and $\mathbf{e}_{oi}$ is other errors.

The friction of the moving pair of the mechanism is mainly the friction generated during the movement of the Hooke hinge, composite ball hinge and ball screw of the UPS, resulting in insufficient driving force. Moreover, the posture of the mechanism in the motion space is different, the force state of each motion pair is also different, and the friction of the motion pair of the UPS is also related to its spatial posture. Therefore, the error caused by the friction of the moving pair can be expressed as:

$$\mathbf{e}_{fi} = F_{fi}(\mathbf{M}_{fi}, x, y, z), \tag{2}$$

where F_{fi} is the conversion function between friction and position error of moving pair, $\mathbf{M}_{fi}$ is the friction of the moving pair, and (*x*, *y*, *z*) is the spatial coordinate of the LOMPR. The kinematic pair friction is analysed by introducing viscous friction based on the Coulomb friction model. So, the friction of the UPS can be obtained as:

$$\mathbf{M}_{fi} = \begin{bmatrix} M_{fix} \\ M_{fiy} \\ M_{fiz} \end{bmatrix} = \begin{bmatrix} M_{fix}^{u} + M_{fix}^{s} \\ +F_{fiz}^{p}\left(s\beta_i c\gamma_i + s\alpha_i c\beta_i s\gamma_i\right) \\ M_{fiy}^{u} + M_{fiy}^{s} \\ +F_{fiz}^{p}\left(s\beta_i s\gamma_i - s\alpha_i c\beta_i c\gamma_i\right) \\ M_{fiz}^{p} + M_{fiz}^{s} \\ \\ \end{bmatrix}, \tag{3}$$

where M_{fix}, M_{fiy}, and M_{fiz} are the friction torque around $x_{\mathrm{A}i}$, $y_{\mathrm{A}i}$ and $z_{\mathrm{A}i}$ axis, respectively; M_{fix}^{u}, M_{fiy}^{u}, and M_{fiz}^{u} are the friction torque of Hooke hinge around $x_{\mathrm{A}i}$, $y_{\mathrm{A}i}$, and $z_{\mathrm{A}i}$ axis, respectively; M_{fix}^{s}, M_{fiy}^{s}, and

M^{s}_{fiz} are the friction torque of composite ball hinge around x_{Ai}, y_{Ai}, and z_{Ai} axis, respectively; F^{p}_{fiz} is the friction force of the ball screw along the z_{Ai} axis; M^{p}_{fiz} is the friction torque of ball screw around z_{Ai} axis; α, β, and γ represents the RPY angle of the moving platform relative to the fixed platform.

The position error caused by the elastic deformation of UPS can be expressed by the kinematic constraints of a parallel mechanism under elastic deformation:

$$\mathbf{e}_{di} = \mathbf{J}_{di}\mathbf{q}_{i}\,, \tag{4}$$

where $\mathbf{J}_{di}$ is the kinematic constraint matrix, and $\mathbf{q}_i$ is the displacement and rotation of UPS under elastic deformation.

When the speed feedforward control model is adopted, the servo system error function of UPS can be expressed as:

$$e_{ci} = \lim_{s\to 0}\frac{\begin{bmatrix} Js^4 - \left[B + K_{pv}G_1K_T\left(1-K_{sv}\right)\right]s^3 \\ +K_{pv2}G_1K_T\left(1-K_{sv}\right)s^2 \end{bmatrix}}{b_1s^3 + b_2s^2 + b_3s + b_4}X_i\left(s\right), \tag{5}$$

where J is the load moment of inertia of the motor, B is viscous resistance, K_{pv} is the proportional coefficient of the speed loop, K_T is the torque coefficient, $K_{sv}s$ is the speed feedforward transfer function, $K_{pv2} = K_{pv}/\tau_v$, τ_v is the time constant of the velocity loop, X_i is the input signal, $b_1 = J$, $b_2 = B + K_{pv}G_iK_t$, $b_3 = K_{pv2}G_iK_t + K_pK_{pv}G_iK_t$, $b_4 = K_pK_{pv2}G_iK_t$.

2 BO-LSTM

2.1 LSTM Deep Neural Network

The LSTM was developed using recurrent neural networks (RNNs) and is mainly used to solve the problem of long-term data dependence in RNNs. An LSTM can remember the information and its correlation over a long period of time and can easily be recalled so that the information does not decay. An LSTM consists of three steps: a state forward calculation, error back propagation, and a weight update.

1) State forward calculation

The input value of an LSTM is network input value $\mathbf{X}_t$ at the current time and network state $\mathbf{S}_{t-1}$ at the previous time. These flow through forgetting gate **F**, input gate **I**, hiding unit **H**, and output gate **O**.

(1) Forgetting gate

$$\mathbf{F}_t = g\left(\mathbf{U}^F\mathbf{X}_t + \mathbf{W}^F\mathbf{S}_{t-1} + \mathbf{b}^F\right), \tag{6}$$

where $\mathbf{U}^F$, $\mathbf{W}^F$, and $\mathbf{b}^F$ represent the input connection matrix, feedback connection matrix, and offset term of the forgetting gate, respectively, g is the sigmoid function.

(2) Input gate

$$\mathbf{I}_t = g\left(\mathbf{U}^I\mathbf{X}_t + \mathbf{W}^I\mathbf{S}_{t-1} + \mathbf{b}^I\right), \tag{7}$$

where $\mathbf{U}^I$, $\mathbf{W}^I$, and $\mathbf{b}^I$ represent the input connection matrix, feedback connection matrix, and offset term of the input gate, respectively.

(3) Hiding unit

$$\mathbf{H}_t = \tanh\left(\mathbf{U}^H\mathbf{X}_t + \mathbf{W}^H\mathbf{S}_{t-1} + \mathbf{b}^H\right), \tag{8}$$

where $\mathbf{U}^H$, $\mathbf{W}^H$, and $\mathbf{b}^H$ represent the input connection matrix, feedback connection matrix, and offset term of the hiding gate, respectively.

(4) Storage element status update

$$\mathbf{C}_t = \mathbf{F}_t\mathbf{C}_{t-1} + \mathbf{I}_t\mathbf{H}_t. \tag{9}$$

(5) Output gate

$$\mathbf{O}_t = g\left(\mathbf{U}^O\mathbf{X}_t + \mathbf{W}^O\mathbf{S}_{t-1} + \mathbf{b}^O\right), \tag{10}$$

where $\mathbf{U}^O$, $\mathbf{W}^O$, and $\mathbf{b}^O$ represent the input connection matrix, feedback connection matrix, and offset term of the output gate, respectively.

(6) LSTM hidden layer output

The information, $\mathbf{C}_t$, of the storage element is multiplied by the result of the tanh transfer function and the output gate to obtain the LSTM hidden layer output.

$$\mathbf{S}_t = \mathbf{O}_t\tanh\left(\mathbf{C}_t\right). \tag{11}$$

(7) Network output

The hidden layer output of the LSTM is sent to the softmax output layer through connection matrix **V** to obtain the actual output of the network.

$$\widehat{\mathbf{Y}}_t = f\left(\mathbf{V}\mathbf{S}_t + \mathbf{b}\right). \tag{12}$$

2) Error back propagation

Based on the actual network output obtained by the forward calculation, the error function can be defined as follows:

$$\mathbf{E} = \sum_t \frac{1}{2}\left(\widehat{\mathbf{Y}}_t - \mathbf{Y}_t\right)^2, \tag{13}$$

where $\mathbf{Y}_t$ is the annotation signal from the data. For time step t, the corresponding error is found as follows:

$$e_t = \frac{\partial \mathbf{E}}{\partial \mathbf{S}_t} = \frac{\partial \mathbf{E}_{t+1}}{\partial \mathbf{S}_t} + \frac{\partial \mathbf{E}_t}{\partial \mathbf{S}_t}. \tag{14}$$

In other words, the downstream error at a certain time, *t*, can be divided into two parts: the error of the backflow in the later time step, *t*+1, and the error of the same time step, *t*. The errors of these two parts are expressed as follows:

$$\begin{cases} \dfrac{\partial \mathbf{E}_t}{\partial \mathbf{S}_t} = \left(\widehat{\mathbf{Y}}_t - \mathbf{Y}_t\right)\dot{\mathbf{f}}\left(\mathbf{V}\mathbf{S}_t + \mathbf{b}\right)\mathbf{V} \\ \dfrac{\partial \mathbf{E}_{t+1}}{\partial \mathbf{S}_t} = \delta_{t+1}^{O}\mathbf{W}^{O} + \delta_{t+1}^{F}\mathbf{W}^{F} \\ \qquad + \delta_{t+1}^{I}\mathbf{W}^{I} + \delta_{t+1}^{H}\mathbf{W}^{H} \end{cases}, \tag{15}$$

where δ is the neural node error. The expressions of the transfer function sigmoid and hyperbolic tangent function are as follows:

$$\begin{cases} y = \dfrac{1}{1+e^{-z}} \\ y = \tanh\left(z\right) = \dfrac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \end{cases}. \tag{16}$$

Further calculation shows that the errors are as follows:

$$\begin{cases} \delta_{t+1}^{O} = e_{t+1}\tanh\left(\mathbf{C}_{t+1}\right)\mathbf{O}_{t+1}\left(1-\mathbf{O}_{t+1}\right) \\ \delta_{t+1}^{F} = e_{t+1}\mathbf{O}_{t+1}\left(1-\tanh^2\left(\mathbf{C}_{t+1}\right)\right)\mathbf{C}_t\mathbf{F}_{t+1}\left(1-\mathbf{F}_{t+1}\right) \\ \delta_{t+1}^{I} = e_{t+1}\mathbf{O}_{t+1}\left(1-\tanh^2\left(\mathbf{C}_{t+1}\right)\right)\mathbf{H}_{t+1}\mathbf{I}_{t+1}\left(1-\mathbf{I}_{t+1}\right) \\ \delta_{t+1}^{H} = e_{t+1}\mathbf{O}_{t+1}\left(1-\tanh^2\left(\mathbf{C}_{t+1}\right)\right)\mathbf{I}_{t+1}\left(1-\mathbf{H}_{t+1}\right) \end{cases}. \tag{17}$$

3) Weight update

Based on the errors of the forgetting gate, input gate, and output gate in each time step, the connection weight parameters associated with these components can be updated and combined with the output values of each neuron obtained in the forward calculation.

The formula for updating the connection parameters of a forgetting gate is as follows:

$$\begin{cases} \mathbf{U}_{n+1}^{F} = \mathbf{U}_{n}^{F} - \beta\sum_t \delta_t^{F}\mathbf{X}_t \\ \mathbf{W}_{n+1}^{F} = \mathbf{W}_{n}^{F} - \beta\sum_t \delta_t^{F}\mathbf{S}_{t-1} \\ \mathbf{V}_{n+1}^{F} = \mathbf{V}_{n}^{F} - \beta\sum_t \delta_t^{F}\mathbf{S}_{t-1} \\ \mathbf{b}_{n+1}^{F} = \mathbf{b}_{n}^{F} - \beta\sum_t \delta_t^{F} \end{cases}, \tag{18}$$

where *β* is the learning rate, and *n* is the number of iterations.

The formula for updating the connection parameters of the input gate is as follows:

$$\begin{cases} \mathbf{U}_{n+1}^{I} = \mathbf{U}_{n}^{I} - \beta\sum_t \delta_t^{I}\mathbf{X}_t \\ \mathbf{W}_{n+1}^{I} = \mathbf{W}_{n}^{I} - \beta\sum_t \delta_t^{I}\mathbf{S}_{t-1} \\ \mathbf{V}_{n+1}^{I} = \mathbf{V}_{n}^{I} - \beta\sum_t \delta_t^{I}\mathbf{S}_{t-1} \\ \mathbf{b}_{n+1}^{I} = \mathbf{b}_{n}^{I} - \beta\sum_t \delta_t^{I} \end{cases}. \tag{19}$$

The formula for updating the connection parameters of the hidden layer is as follows:

$$\begin{cases} \mathbf{U}_{n+1}^{H} = \mathbf{U}_{n}^{H} - \beta\sum_t \delta_t^{H}\mathbf{X}_t \\ \mathbf{W}_{n+1}^{H} = \mathbf{W}_{n}^{H} - \beta\sum_t \delta_t^{H}\mathbf{S}_{t-1} \\ \mathbf{V}_{n+1}^{H} = \mathbf{V}_{n}^{H} - \beta\sum_t \delta_t^{H}\mathbf{S}_{t-1} \\ \mathbf{b}_{n+1}^{H} = \mathbf{b}_{n}^{H} - \beta\sum_t \delta_t^{H} \end{cases}. \tag{20}$$

The formula for updating the connection parameters of the output gate is as follows:

$$\begin{cases} \mathbf{U}_{n+1}^{O} = \mathbf{U}_{n}^{O} - \beta\sum_t \delta_t^{O}\mathbf{X}_t \\ \mathbf{W}_{n+1}^{O} = \mathbf{W}_{n}^{O} - \beta\sum_t \delta_t^{O}\mathbf{S}_{t-1} \\ \mathbf{V}_{n+1}^{O} = \mathbf{V}_{n}^{O} - \beta\sum_t \delta_t^{O}\mathbf{S}_{t-1} \\ \mathbf{b}_{n+1}^{O} = \mathbf{b}_{n}^{O} - \beta\sum_t \delta_t^{O} \end{cases}. \tag{21}$$

2.2 Bayesian Optimized Hyperparameters

During the training of the LSTM, there are many hyperparameters in the model, including the data length, number of iterations, learning rate, number of hidden neural units, and forgetting coefficient, which can be set at will. Appropriate super parameters are vital to improving the performance of the model. Bayesian optimization can not only set continuous hyperparameter values but also has a high search efficiency and accuracy.

Bayesian optimization uses the approximate global optimization algorithm to find the hyperparametric combination that minimizes the objective function:

$$p\left(f\middle|D\right) = \frac{p\left(D\middle|f\right)p\left(f\right)}{p\left(D\right)}, \tag{22}$$

where *f* represents the objective function; $D = \{(x_1, y_1), (x_1, y_1), \ldots (x_n, y_n)\}$ is the data set; $p(f|D)$ and $p(f)$

are the a posteriori probability and a priori probability, respectively; $p(D|f)$ is the likelihood distribution of y; and $p(D)$ is the boundary likelihood distribution of f.

Assuming a set of super parameter combinations, $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, each super parameter is evaluated, and the evaluation result is $f(x_n)$. Then, the optimal hyperparameter is found as follows:

$$x^* = \arg\min_{x\in X} f(x). \qquad (23)$$

1) Probabilistic agent model

The probability agent model adopts a Gaussian process, and the Gaussian distribution can be expressed as follows:

$$f(x) \sim GP\big(\mu(x), k(x,x)\big), \qquad (24)$$

where $\mu(x)$ is the mean function, $k(x,x)$ is the positive definite covariance function, and $f(x)$ is the average absolute error.

According to the nature of the Gaussian process, f_t and f_{t+1} obey a joint Gaussian distribution:

$$\begin{bmatrix} f_{1:t} \\ f_{t+1} \end{bmatrix} \sim N\left(0 \quad \begin{bmatrix} k & k \\ k^{\mathrm{T}} & k(x_{t+1}, x_t) \end{bmatrix}\right), \qquad (25)$$

where

$$\begin{cases} k = \big[k(x_{t+1}, x_1)\cdot k(x_{t+1}, x_2)\cdots k(x_{t+1}, x_t)\big] \\ P\big(f_{t+1}\,|D_{1:t}, x_{t+1}\big) = N\big(\mu_t(x_{t+1}), \sigma_t^2(x_{t+1})\big) \\ \mu_t(x_{t+1}) = k^{\mathrm{T}} K^{-1} f_{1:t} \\ \sigma_t^2(x_{t+1}) = k(x_{t+1}, x_{t+1}) - k^{\mathrm{T}} K^{-1} k \end{cases}, \qquad (26)$$

where K is the covariance matrix,

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_t) \\ \vdots & \ddots & \vdots \\ k(x_t, x_1) & \ldots & k(x_t, x_t) \end{bmatrix}.$$

2) Acquisition function

The acquisition function can prevent the Bayesian optimization from falling into a local optimal solution. The excepted improvement (EI) acquisition function can find the next sample point with the greatest improvement expectation. The EI sampling function is as follows:

$$EI(x) = \begin{cases} \mu(x) - f(x^+)\Phi(Z) & \sigma(x) > 0 \\ +\sigma(x)\phi(Z) & \\ 0 & \sigma(x) \le 0 \end{cases}, \qquad (27)$$

where Φ is the probability density, φ is the standard normal distribution function, and $f(x^+)$ is the existing maximum. In addition, the following is true:

$$Z = \frac{\mu(x) - f(x^+)}{\sigma(x)}. \qquad (28)$$

Then, the flow of the BO-LSTM is shown in Fig. 2.

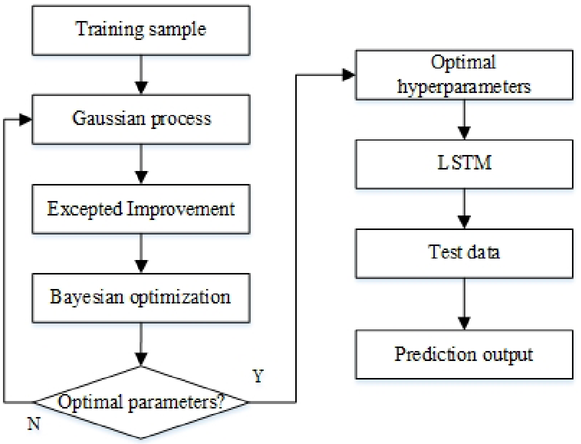

Fig. 2. *Flow of BO-LSTM*

3 EXPERIMENTAL ANALYSIS

The prototype of the LOMPR is shown in Fig. 3. In mirror processing, grid trajectory, concentric circle trajectory, and spiral trajectory are commonly used. The curvature of spiral trajectory constantly changes, which severely tests the dynamic characteristics of LOMPR, and the end trajectory error generated by the processing robot is also relatively large. Therefore, the spiral trajectory is used as a representative to predict the error in the X and Y directions of the end trajectory

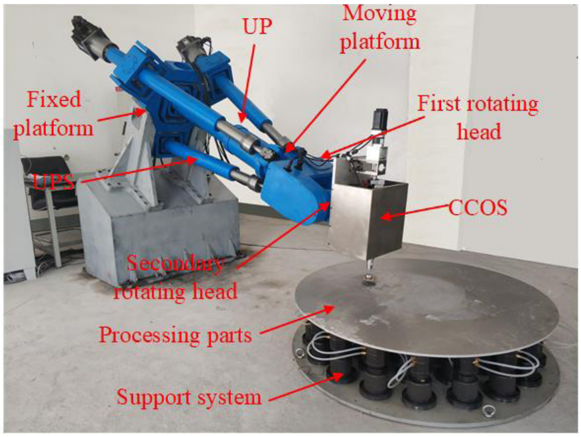

Fig. 3. *Prototype of LOMPR*

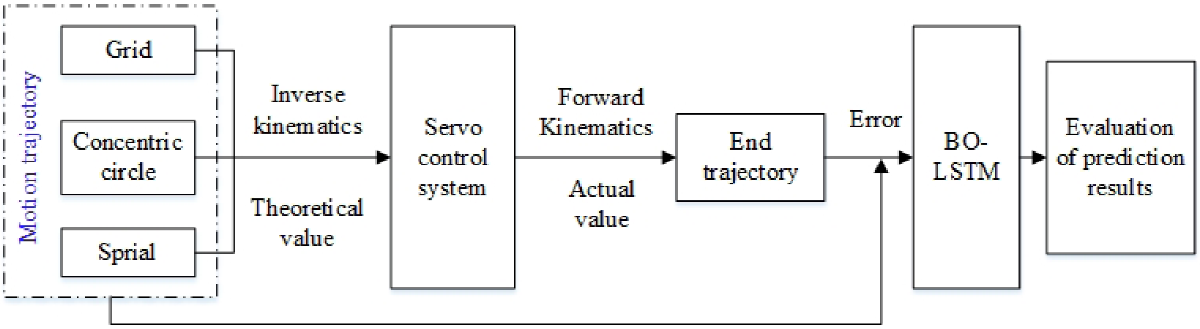

Fig. 4. *Technical route of error prediction*

of the LOMPR. Because the optical mirror needs to complete an iterative cycle before surface detection, it is impossible to collect the error of the mirror surface in real time for prediction. Therefore, the training data used are the end execution point position parameters calculated from the collected motion branch chain parameters into the forward kinematics model. Then the actual trajectory error collected from the experimental mirror is compared with the prediction results. On the one hand, it can verify the effectiveness of the BO-LSTM prediction model; on the other hand, it can verify the accuracy of the forward kinematics model. The technical route is shown in Fig. 4.

The experiment was based on the deep learning framework Keras library written in the Python language to complete the BO-LSTM model construction, data loading, training, and testing. Firstly, the data is pre-processed, including time stitching, missing data processing, abnormal data processing and data normalization. In the process of data prediction, the learning rate, batch capacity, and the number of hidden neurons directly affected the convergence speed and prediction accuracy of the model. Then, the above hyperparameters were optimized, and the appropriate parameters were input into the model for training. In order to compare the effectiveness of this hyperparametric optimization, a set of default values were used based on engineering experience, including a batch size of 1500, 35 hidden neurons, and a learning rate of 0.01. Considering the influence of the hyperparameters on the model performance and the interaction between hyperparameters, the value range of the batch size was set at (200, 600, 1000, 1500), the number of hidden neurons was (5, 15, 35, 50), the learning rate was (0.1, 0.01, 0.001), the optimizer was “Adam,” and the activation function was “Relu”.

Bayesian optimization was used to search the above hyperparameters, and the following optimal combination of hyperparameters was obtained: a batch size of 1000, 15 hidden neurons, and a learning rate of 0.001. The experimental data were imported into the established LSTM error prediction model using 90 % as training data samples and 10 % as test samples. The end execution point errors of the optical mirror machining robot under different trajectories were then predicted using the BO-LSTM. In order to verify the effectiveness of the BO-LSTM model in the prediction of optical mirror processing trajectory error, the BP neural network prediction model is used for prediction and analysis, and the prediction results are compared with those of BO-LSTM.

The trajectory equation of the robot during spiral processing is as follows:

$$\begin{cases} X = 20t_h \cos\left(\dfrac{\pi}{2}t_h + \dfrac{\pi}{2}\right) \\ Y = 20t_h \sin\left(\dfrac{\pi}{2}t_h + \dfrac{\pi}{2}\right) & t_h = 0:10. \\ Z = 0 \times t_h + 1097 \end{cases} \tag{29}$$

The LOMPR was controlled to process continuously, the motion period is 40 s and the sampling period is 40 ms. And the motion data of each rotating axis were collected, and the trajectory error of the end execution point was calculated. According to the collected data, the prediction results of BO-LSTM prediction model and BP neural network prediction model are shown in Fig. 5.

The data collected during the experiment and the trajectory error prediction results of the spiral trajectory in the *X* and *Y* directions are shown in Fig. 5. As the spiral trajectory is a trajectory with gradually increasing curvature, the errors of the LOMPR in the *X* and *Y* directions accumulates, resulting in the trajectory error in the *X* and *Y* directions gradually increasing amplitude and periodically changing in the form of a sinusoidal function curve. The prediction parameters of the BP neural network are shown in

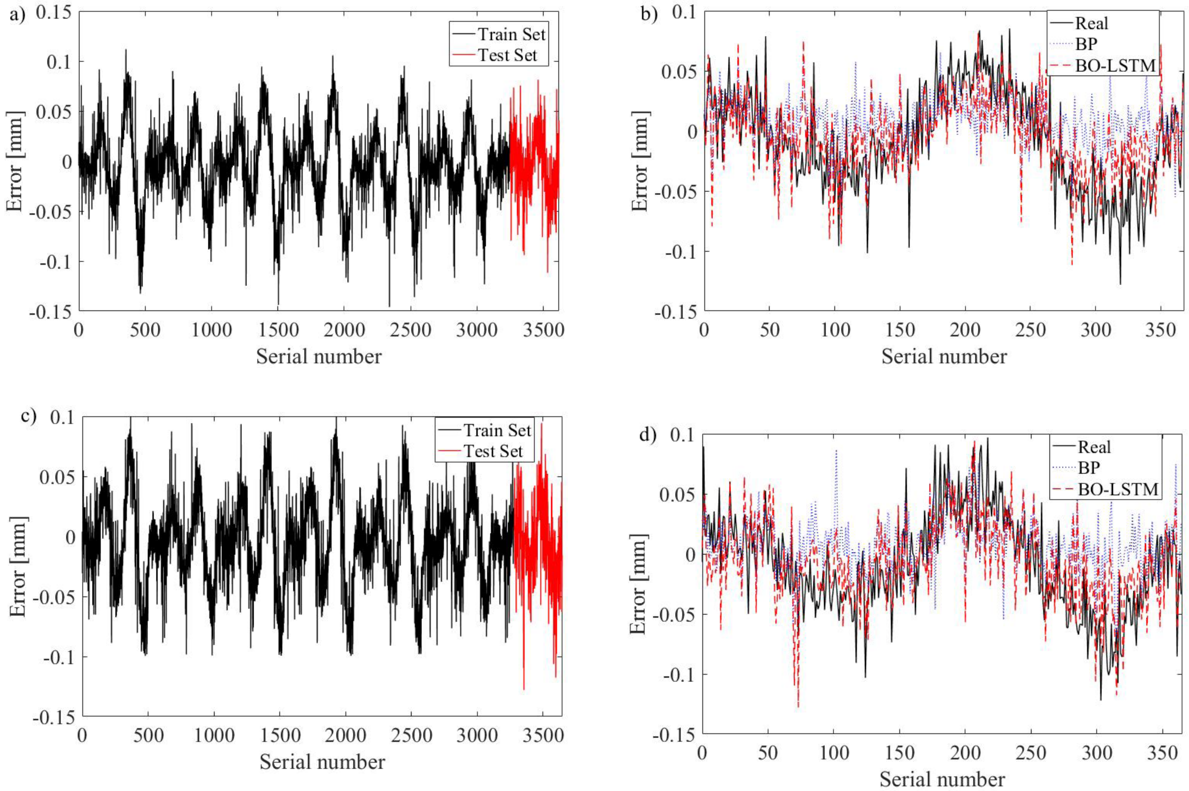

Fig. 5. *Spiral trajectory error prediction; a) data samples on X-axis; b) prediction results on X-axis; c) data samples on Y-axis; d) prediction results on Y-axis*

Table 1. The training time in the *X* and *Y* directions is 42 min and 46 min, respectively. The MSE, RMSE, and MAE of the prediction error in *X* direction are 0.96 %, 9.80 %, and 7.29 %, respectively; the MSE, RMSE and MAE of the prediction errors in *Y* direction are 1.04 %, 10.18 %, and 6.51 %, respectively. When using the BO-LSTM prediction model for trajectory error prediction, the training time in *X* and *Y* directions are reduced to 9 min and 7 min, respectively.

Further analysis of the prediction results shows that the MSE, RMSE, and MAE of the prediction error in the *X* direction were 0.09 %, 2.99 %, and 2.32 %, respectively; the MSE, RMSE, and MAE of the prediction error in the *Y* direction were 0.10 %, 3.14 %, and 2.46 %, respectively. Moreover, through the predicted trajectory error data of 14.6 s, the prediction results of the two models are analysed through the error integration criterion. It can be seen that the IAE, ISE, ITAE, and ITSE of the prediction error of BO-LSTM in the *X* direction are reduced to 56.44 %, 33.33 %, 53.63 %, and 30.47 %, respectively compared with the prediction error of the BP neural network; the IAE, ISE, ITAE, and ITSE of the prediction error of BO-LSTM in the *Y* direction are reduced to 57.87 %, 37.84 %, 54.79 %, and 34.27 % respectively compared with the prediction error of BP neural network. In summary, the prediction accuracy of the established BO-LSTM end trajectory error prediction model is significantly higher than that of the traditional BP neural network, which further verifies the accuracy and effectiveness of the established model.

Table 1. *Analysis of spiral trajectory error prediction results*

Direction	Model	Training time [min]	MSE [%]	RMSE [%]	MAE [%]	IAE	ISE	ITAE	ITSE
X	BP	42	0.96	9.80	7.29	0.6176	0.0402	5.0697	0.3547
	LSTM	9	0.09	2.99	2.32	0.3486	0.0134	2.7187	0.1081
Y	BP	46	1.04	10.18	6.51	0.5920	0.0370	4.7360	0.3131
	LSTM	7	0.10	3.14	2.46	0.3426	0.0140	2.5754	0.1073

4 CONCLUSION

This study investigated a method for predicting the trajectory error of the end execution point of a LOMPR. The hyperparameters were optimized using a Bayesian method, and a BO-LSTM deep neural network was established to optimize the learning rate, batch capacity, and number of hidden neurons to improve the efficiency and accuracy of the model search. Considering the working characteristics of the CCOS grinding system as a tool, the collected kinematic data of the LOMPR were processed, including time stitching, missing data processing, abnormal data processing, and normalization processing, to improve the learning efficiency of the deep learning network. Finally, in end trajectory error prediction experiments using a spiral trajectory, the trajectories in X and Y directions were predicted. At the same time, BP neural network prediction model is used to predict the end trajectory error of LOMPR. The prediction results had high accuracy, and the training time of the model was greatly reduced. Compared with the prediction results of a BP neural network, the training time of BO-LSTM prediction model in X and Y directions are reduced to 21.4 % and 15.2 %, respectively; The MSE, RMSE, and MAE of the prediction error in the X direction are reduced to 9.4 %, 30.5 %, and 31.8 %, respectively; MSE, RMSE, and MAE of the prediction error in Y direction were reduced to 9.6 %, 30.8 %, and 37.8 %, respectively. The validity and accuracy of the model were verified, which provides a reliable theoretical basis and data support for the feedforward compensation of the end trajectory error of a future LOMPR. In addition, the BO-LSTM error prediction model proposed in this paper could be widely used in kinematic error prediction for other parallel and hybrid robots.

6 ACKNOWLEDGEMENT

Financial support for this work, provided by the Priority Academic Program Development of Jiangsu Higher Education Institutions and the National Key R&D Program of China, are gratefully acknowledged.

7 REFERENCES

[1] Jiang, B.C., Zhao, D.W., Wang, B.Q., Zhao, H.J., Liu, Y.H., Lu, X.C. (2020). Flatness maintenance and roughness reduction of silicon mirror in chemical mechanical polishing process. *Science China-Technological Sciences*, vol. 63, no. 1, p. 166-172, **DOI:10.1007/s11431-018-9414-6**.

[2] Kong, Y.X., Cheng, G., Guo, F., Gu, W., Zhang, L.B. (2019). Inertia matching analysis of a 5-DOF hybrid optical machining manipulator. *Journal of Mechanical Science and Technology*, vol. 33, no. 10, p. 4991-5002, **DOI:10.1007/s12206-019-0938-1**.

[3] Erkaya, S. (2018). Effects of joint clearance on the motion accuracy of robotic manipulators. *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 62, no. 2, p. 82-94, **DOI:10.5545/sv-jme.2017.4534**.

[4] Iqbal, J., Ullah, M.I., Khan, A.A., Irfan, M. (2015). Towards sophisticated control of robotic manipulators: an experimental study on a pseudo-industrial arm. *Strojniški vestnik-Journal of Mechanical Engineering*, vol. 61, no. 7, p. 465-470, **DOI:10.5545/sv-jme.2015.2511**.

[5] Jin, Z.J., Cheng, G., Chen, S.B., Guo F. (2021). Human-machine-environment information fusion and control compensation strategy for large optical mirror processing system. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 235, no. 13, p. 2507-2523, **DOI:10.1177/0954406220959689**.

[6] Liao, W.L., Dai, Y.F., Xie, X.H., Zhou, L., Yuan Z. (2012). Corrective capability analysis and machining error control in ion beam figuring of high-precision optical mirrors. *Optical Engineering*, vol. 51, no. 3, **DOI:10.1117/1.OE.51.3.033402**.

[7] Zhong, X.Y., Fan, B., Wu, F. (2017). High-accuracy process based on the corrective calibration of removal function in the magnetorheological finishing. *Optical Engineering*, vol. 56, no. 8, **DOI:10.1117/1.OE.56.8.084109**.

[8] Pan, H.H., He, X.X., Tang, S., Meng, F.M. (2018). An improved bearing fault diagnosis method using one-dimensional CNN and LSTM. *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 64, no. 7-8, p. 443-452, **DOI:10.5545/sv-jme.2018.5249**.

[9] Baizid, K. Cukovic, S. Iqbal, J. Yousnadj, A. Chellali, R. Meddahi, A. Devedzic, G., Ghionea, I. (2016). IRoSim: industrial robotics simulation design planning and optimization platform based on CAD and knowledgeware technologies. *Robotics and Computer-Integrated Manufacturing*, vol. 42, p. 121-134, **DOI:10.1016/j.rcim.2016.06.003**.

[10] Singh A., Pandey P., Nandi G.C. (2021). Effectiveness of multi-gated sequence model for the learning of kinematics and dynamics of an industrial robot. *Industrial Robot - the International Journal of Robotics Research and Application*, vol. 48, no. 1, p. 62-70, **DOI:10.1108/IR-01-2020-0010**.

[11] Wang, S.J., Shao, X.M., Yang, L.S., Liu, N. (2020). Deep learning aided dynamic parameter identification of 6-DOF robot manipulators. *IEEE Access*, vol. 8, p. 138102-138116, **DOI:10.1109/ACCESS.2020.3012196**.

[12] Wang, H.L., Xie, Y.C. (2019). Prediction error based adaptive Jacobian tracking of robots with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control*, vol. 52, no. 12, p. 2889-2894, **DOI:10.1109/TAC.2009.2033764**.

[13] Dai, S.Z., Li, L., Li, Z.H. (2019). Modeling vehicle interactions via modified LSTM models for trajectory prediction. *IEEE Access*, vol. 7, p. 38287-38296, **DOI:10.1109/ACCESS.2019.2907000**.

[14] Mici, L., Parisi, G.I., Wermter, S. (2018). An incremental self-organizing architecture for sensorimotor learning and prediction. *IEEE Transactions on Cognitive and Developmental*

Systems, vol. 10, no. 4, p. 918-928, **DOI:10.1109/TCDS.2018.2832844**.

[15] Mirzaeinejad, H., Shafei, A.M. (2018). Modeling and trajectory tracking control of a two-wheeled mobile robot: Gibbs-Appell and prediction-based approaches. *Robotica*, vol. 36, no. 10, p. 1551-1570, **DOI:10.1017/S0263574718000565**.

[16] Wang, H.L., Xie, Y.C. (2012). Prediction error based adaptive Jacobian tracking for free-floating space manipulators. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, p. 3207-3221, **DOI:10.1109/TAES.2012.6324694**.

[17] Las-Casas, M.S., Diniz de Avila, T.L.D., Bracarense, A.Q., Lima, E.J. (2018). Weld parameter prediction using artificial neural network: FN and geometric parameter prediction of austenitic stainless steel welds. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, no. 1, **DOI:10.1007/s40430-017-0928-0**.

[18] Wang, Y., Yang, Y.Y., Zhao, B.L., Qi, X.Z., Hu, Y., Li, B., Sun, L.N., Zhang, L.H., Meng, M.Q.H. (2021). Variable admittance control based on trajectory prediction of human hand motion for physical human-robot interaction. *Applied Sciences*, vol. 11, no. 12, **DOI:10.3390/app11125651**.

[19] Zhang, G., Xu, Z., Hou, Z.C., Yang, W.L., Liang, J.M., Yang, G., Wang, J., Wang, H.M., Han, C. (2020). A Systematic error compensation strategy based on an optimized recurrent neural network for collaborative robot dynamics. *Applied Sciences*, vol. 10, no. 19, **DOI:10.3390/app10196743**.

[20] Yang, Q.B., Li, X., Cai, H.S., Hsu, Y.M., Lee, J., Yang, C.H., Li, Z.L., Lin, M.Y. (2021). Fault prognosis of industrial robots in dynamic working regimes: Find degradation in variations. *Measurement*, vol. 173, **DOI:10.1016/j.measurement.2020.108545**.

[21] Chebbi, A.H., Chouaibi, Y., Affi, Z., Romdhane, L. (2018). Sensitivity analysis and prediction of the orientation error of a three translational parallel manipulator. *Proceedings of the Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science*, vol. 232, no. 1, p. 140-161, **DOI:10.1177/0954406216675380**.

[22] Zatout, M.S., Rezoug, A., Rezoug, A., Baizid, K., Iqbal, J. (2022). Optimisation of fuzzy logic quadrotor attitude controller - particle swarm, cuckoo search and BAT algorithms. *International Journal of Systems Science*, vol. 53, no. 4., p. 883-908, **DOI:10.1080/00207721.2021.1978012**.

[23] Peng, G.L., Ji, M.Y., Xue, Y., Sun, Y. (2021). Development of a novel integrated automated assembly system for large volume components in outdoor environment. *Measurement*, vol. 168, **DOI:10.1016/j.measurement.2020.108294**.

[24] You, J.J., Xi, F.F., Shen, H.P., Wang, J.Y., Yang, X.L. (2021). A novel Stewart-type parallel mechanism with topological reconfiguration: Design, kinematics and stiffness evaluation. *Mechanism and Machine Theory*, vol. 162, **DOI:10.1016/j.mechmachtheory.2021.104329**.

[25] He, J., Ding, Q.Y., Gao, F., Zhang, H.B. (2020). Kinematic calibration methodology of hybrid manipulator containing parallel topology with main limb. *Measurement*, vol. 152, **DOI:10.1016/j.measurement.2019.107334**.

[26] Li, Y., Zhu, Z.F., Kong, D.Q., Han, H., Zhao, Y. (2019). EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowledge-Based Systems*, vol. 181, **DOI:10.1016/j.knosys.2019.05.028**.

[27] Ahmad, E., Iqbal, J., Khan, M.A., Liang, W., Youn, I. (2020). Predictive control using active aerodynamic surfaces to improve ride quality of a vehicle. *Electronics*, vol. 16, no. 11, **DOI:10.3390/electronics9091463**.

[28] Ullah, M.I., Ajwad, S.A., Irfan, M., Iqbal, J. (2016). MPC and H-infinity based feedback control of non-linear robotic manipulator. *14th International Conference on Frontiers of Information Technology (FIT)*, p. 136-141, **DOI:10.1109/FIT.2016.033**.

[29] Ali, N., Ayaz Y, Iqbal, J. (2022). Collaborative position control of pantograph robot using particle swarm optimization. *International Journal of Control Automation and Systems*, vol. 20, no. 1, p. 198-207, **DOI:10.1007/s12555-019-0931-6**.

[30] Bai, G.X., Meng, Y., Liu, L., Luo, W.D., Gu, Q. (2019). Review and comparison of path tracking based on model predictive control. *Electronics*, vol. 8, no. 10, **DOI:10.3390/electronics8101077**.

[31] Shen, S.B., Song, A.G., Li, T., Li, H.J. (2019). Time delay compensation for non-linear bilateral teleoperation: A motion prediction approach. *Transactions of the Institute of Measurement and Control*, vol. 41, no. 16, p. 4488-4498, **DOI:10.1177/0142331219860928**.

[32] Tang, G., Wu, Y.F., Li, C.S., Wong, P.K., Xiao, Z.H., An, X.L. (2020). A novel wind speed interval prediction based on error prediction method. *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, p. 6806-6815, **DOI:10.1109/TII.2020.2973413**.

[33] Wang, B.K., Peng, X.Y., Jiang, M., Liu, D.T. (2020). Real-Time fault detection for UAV based on model acceleration engine. *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, p. 9505-9516, **DOI:10.1109/TIM.2020.3001659**.

[34] Mei, M.L., Chang, J., Li, Y.L., Li, Z.R., Li, X.C., Lv, W.J. (2019). Comparative study of different methods in vibration-based terrain classification for wheeled robots with shock absorbers. *Sensors*, vol. 19, no. 5, **DOI:10.3390/s19051137**.

[35] Wu, Y.T., Yuan, M., Dong, S.P., Lin, L., Liu, Y.Q. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, vol. 275, p. 167-179, **DOI:10.1016/j.neucom.2017.05.063**.