

Contour Error Modeling and Compensation of CNC Machining Based on Deep Learning and Reinforcement Learning

Yakun Jiang

Huazhong University of Science and Technology

Jihong Chen

Huazhong University of Science and Technology

Huicheng Zhou

Huazhong University of Science and Technology

Jianzhong Yang

Huazhong University of Science and Technology

Pengcheng Hu (✉ pchu@hust.edu.cn)

Huazhong University of Science and Technology <https://orcid.org/0000-0002-6828-2038>

Junxiang Wang

Huazhong University of Science and Technology

Research Article

Keywords: Contour error prediction and compensation, NAR-LSTM network, TS-DQN, CNC machining.

Posted Date: April 19th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-404176/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on September 7th, 2021. See the published version at <https://doi.org/10.1007/s00170-021-07895-6>.

Contour error modeling and compensation of CNC machining based on deep learning and reinforcement learning

Yakun Jiang, Jihong Chen, Huicheng Zhou, Jianzhong Yang, Pengcheng Hu*, Junxiang Wang

National Numerical Control System Engineering Research Center, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract

Contour error compensation of the Computer Numerical Control (CNC) machine tool is a vital technology that can improve machining accuracy and quality. To achieve this goal, the tracking error of a feeding axis, which is a dominant issue incurring the contour error, should be firstly modeled and then a proper compensation strategy should be determined. However, building the precise tracking error prediction model is a challenging task because of the nonlinear issues like backlash and friction involved in the feeding axis; besides, the optimal compensation parameter is also difficult to determine because it is sensitive to the machining tool path. In this paper, a set of novel approaches for contour error prediction and compensation is presented based on the technologies of deep learning and reinforcement learning. By utilizing the internal data of the CNC system, the tracking error of the feeding axis is modeled as a Nonlinear Auto-Regressive Long-Short Term Memory (NAR-LSTM) network, considering all the nonlinear issues of the feeding axis. Given the contour error as calculated based on the predicted tracking error of each feeding axis, a compensation strategy is presented with its parameters identified efficiently by a Time-Series Deep Q-Network (TS-DQN) as designed in our work. To validate the feasibility and advantage of the proposed approaches, extensive experiments are conducted, testifying that, our approaches can predict the tracking error and contour error with very good precision (better than about 99% and 90% respectively), and the contour error compensated based on the predicted results and our compensation strategy is significantly reduced (about 70%~85% reduction) with the machining quality improved drastically (machining error reduced about 50%).

Keywords: Contour error prediction and compensation, NAR-LSTM network, TS-DQN, CNC machining.

1. Introduction

In Computer Numerical Control (CNC) machining, the contour error of the feeding system is one of the most fatal factors affecting the machining precision, and the lower the contour error is, the better machining precision it could be. The contour error results from both factors of the tracking error of the machine's feeding axes and the mismatch of multi-axis dynamic characteristics [1], especially at the location of the sharp corner [2][3][4] of the tool path. To improve the machining accuracy, the contour error should be effectively predicted and compensated [5][6].

There have been extensive works on contour error prediction and compensation. In general, they are classified into two types: online and offline approaches [7][8].

As an online method for the contour error estimation and compensation, the concept of Cross-Coupling

* Corresponding author

E-mail: pchu@hust.edu.cn (P. Hu)

Tel:+86-27-8754 0024

Control (CCC) was introduced by Koren in [9], which directly reduces the contour error of the entire feeding system of machine tool simultaneously, rather than modeling and treating the tracking error of each feeding axis individually. Based on the work of [9], an improved concept of CCC is proposed, for which a more effective control law is utilized to build the model of contour error [10]. In the following work of [9], Koren and Lo [11] proposed a variable gain CCC method to estimate the contour error more effectively. For the contour of arbitrary shape, Yeh and Hsu [12] proposed a contouring error vector estimation-based approach to efficiently determine the variable gains for the CCC. Later, the approaches based on the CCC were developed further into the two-layered cross-coupling control [13], cross-coupled fuzzy logic sliding mode control [14], cross-coupled dynamic friction control [15], and the neural-network CCC [16].

Different from CCC algorithms, Huo et al. [17] proposed a generalized Taylor series expansion error compensation method, and this method was proved to enable the elimination of the axial following error via simulation, thereby achieving good contour following results. Zhu et al. [18] presented a real-time contour error estimation method based on the second-order Taylor approximation of the point-to-curve distance function. Erkorkmaz and Altintas [19] presented a contour error estimation method for arbitrary free-form contouring curves by approximating the original continuous trajectory with some discretized positions. By projecting the actual position onto the osculating plane at the nearest reference point, Li et al. [20] proposed a method in which the axial component of contour errors are estimated and compensated into each control loop of the servo motor simultaneously. Hu et al. [21] proposed a numerical computation-based accurate contour error estimation method and the corresponding contour error compensation strategy.

In addition to the online contour error estimation and compensation method, some other works have been done on the offline method [22]. Huo and Poo [23] proposed a Nonlinear Auto-Regressive (NAR) neural network modeling method to build the model of the feeding system by taking the reference position as the input, and the actual position at the next instant can be predicted from the proposed model. Similarly, by designing a NAR neural network with exogenous inputs, Erwinski et al. [24] presented a contour error prediction model, which can directly predict the contour error by using a reference position on the tool path. Zhang et al. [25] proposed an iterative compensation method for contour error of CNC machining, in which the tracking error model is utilized for calculating the optimal compensation value.

For the real-time compensation methods based on the contour error modeling, the accuracy of compensation is not high enough because, on the one hand, the prediction models have limited complexity and thus fair prediction precision; and on the other hand, due to the essential inertia of the electrical and mechanical system, the feeding system of CNC machine tool has the lag-characteristic, i.e., the compensated result cannot work immediately, which could undermine the effect of contour error compensation. For the offline model-based iterative compensation methods, the computational efficiency for determining the compensation value is low because they normally require some iterative processes to calculate the compensation value of each control cycle [25]. Also, the predicted value of the model cannot truly reflect the actual contour error, especially when the error generated by the mechanical transmission elements [26][27] is difficult to be considered and covered by the prediction model.

To improve the contouring accuracy of the CNC machine tool, this paper presents a set of contour error compensation methods by firstly modeling the contour error based on a Nonlinear Auto-Regressive Long-Short Term Memory (NAR-LSTM) network and then determining the optimal compensation parameters via another machine learning method called Time-Series Deep Q-Network (TS-DQN). There are two major contributions

1 in this paper:

- 2 (1) Based on the internal data of the machine tool, a deep learning network called NAR-LSTM is designed
3 that can precisely predict the tracking error, based on which the contour error can be calculated. For
4 each feeding axis, the nonlinear factor of the tracking error is estimated from the NAR-LSTM network;
5 combine with the linear part as calculated from the steady-state of the feeding axis, the entire tracking
6 error of the feeding axis can be precisely predicted. After that, the contour error is calculated based on
7 the predicted tracking error of two feeding axes.
- 8 (2) To address the lag-characteristic as well as find the best compensation strategy, a reinforcement
9 learning method called TS-DQN is proposed that determines the optimal parameters of the
10 compensation strategy, from which the contour error can be largely reduced and the surface machining
11 quality can be improved significantly.

12 Given the proposed approaches, the contour error for machining a part, no matter it is a new part or an
13 already machined one, can be precisely predicted and compensated without needing any time-consuming
14 iterative calculation for the compensation value and parameters.

15 This paper is organized as follows. Section 2 introduces a NAR-LSTM network-based contour error
16 prediction method. Based on the predicted result, Section 3 presents a TS-DQN-based approach to determine
17 the optimal compensation parameters for a given compensation strategy. In Section 4, some simulation and
18 physical cutting experiments are conducted to verify the performance of the proposed contour error prediction
19 and compensation methods. This paper concludes in Section 5.

21 **2. The NAR-LSTM network-based contour error calculation**

22 In this section, a NAR-LSTM network is firstly designed to model the nonlinear part of the tracking error
23 of a machine's feeding axis, based on which the tracking error (considering both the linear and nonlinear part)
24 can be calculated; and then a contour error (due to the tracking error of multiple axes) calculation approach is
25 presented.

27 **2.1. Tracking error of a feeding axis**

28 For a given machining path, a set of positions will be generated for each feeding axis after it is processed in
29 the CNC system, which is called reference position and denoted by the symbol u . The reference position is
30 then sent to the servo system to drive the feeding axis of the machine tool. As a core component of a machine
31 tool, a feeding axis is composed of mechanical transmission elements, a servo motor, and its controller. For a
32 semi-closed loop control a feeding axis as illustrated in Fig.1, the position feedback w' comes from the rotary
33 encoder of the motor. Compared with w' , the actual position w as measured by using the linear encoder
34 installed on the worktable can reflect the tracking error more precisely because the errors caused by the
35 mechanical transmission elements [6] of the feeding axis are included. In our work, instead of using w' as the
36 position feedback, the actual position w is utilized in our following tracking error modeling approach.

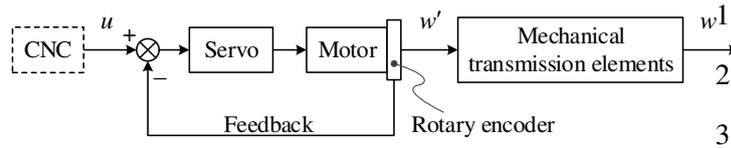


Fig. 1. The general structure of the semi-closed loop control of a feeding axis

For a feeding axis of the machine tool, a typical way of the position control loop is achieved via the classical PID (Proportional-Integral-Derivative) control algorithm [28], i.e., by modulating the proportional, integral, and derivative parameters, the PID controller minimizes the tracking error between the desired position u (i.e., the reference position) and the feedback position w (i.e., the actual position), and

$$E = u - w \quad (1)$$

where, E is defined as the tracking error of a feeding axis as shown in Fig. 2.

To avoid the complexity of determining all the three PID parameters, the P-control mode which only adjusts the proportional parameter is normally utilized in industrial practices, for which the following relationship holds [29],

$$v = K_p \cdot E \quad (2)$$

where, v is the velocity of the velocity loop, and K_p is the position proportional gain and it is a parameter set in the CNC system.

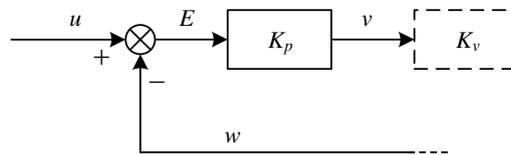


Fig. 2. The position control loop of the feeding system

Based on Eq. (2), the tracking error E can be calculated as

$$E = G(v) = \frac{v}{K_p} \quad (3)$$

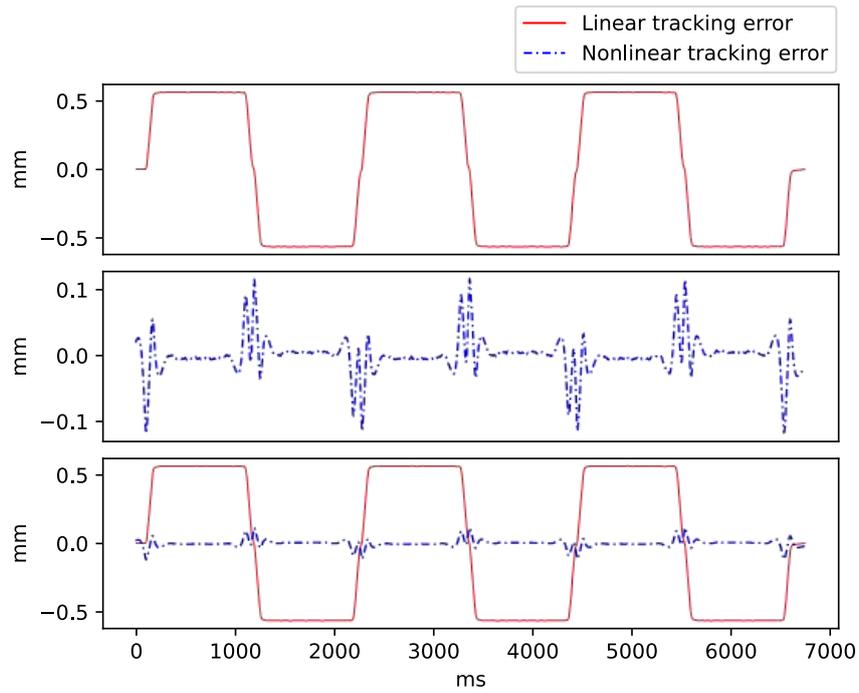
For the E as calculated from Eq. (1) to Eq. (3), it is the tracking error of the steady-state, which is defined as the Linear Tracking Error (LTE) because it is a linear function of v .

In addition to the LTE, there are a lot of nonlinear factors in the control and transmission process from which some additional tracking error can be produced, especially for the mechanical transmission elements which internally suffer from the nonlinear issues like the backlash and friction force of feeding axis. Tracking error generated due to these nonlinear issues is defined as the Nonlinear Tracking Error (NLTE). The NLTE can significantly affect the contour error of the machine tool's feeding system and surface machining quality, it should be precisely estimated before the contour error can be compensated.

1 For the X-axis of the machine tool (Fig. 11), an example of the LTE and NLTE is shown in Fig. 3. With
 2 considering both the LTE and the NLTE, the Overall Tracking Error (OTE) of a feeding system is the
 3 combination of them,

$$E_a = E + E_{nl} \quad (4)$$

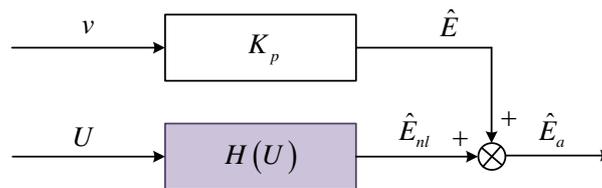
4 where E_a is the OTE and E_{nl} is the NLTE.



5
 6 **Fig. 3.** LTE and NLTE of the X-axis of the machine tool of Fig. 13

7 Based on Eq. (4), building the prediction model for the OTE resorts to the modeling of the LTE and NLTE,
 8 as shown in Fig. 4. Since the LTE can be calculated from Eq. (3), the major challenge for modeling the OTE
 9 lies in the accurate modeling of the NLTE. In this paper, a big data-based model $H(U)$ is utilized to predict
 10 the NLTE, in which the input U is the historical data of the CNC machining process. Details regarding the U
 11 and $H(U)$ will be explained in the next subsection.

12 In the following part of this paper, to distinguish from the real value retrieved from the physical experiment,
 13 we put a hat on the symbol of the estimated data, e.g., the E is the real LTE while the \hat{E} with hat is the
 14 estimated LTE.



15
 16 **Fig. 4.** Prediction model of the OTE

2.2. NAR-LSTM network-based modeling of the NLTE

In this section, the model for predicting the NLTE of the feeding axis is constructed based on the NAR-LSTM network.

Due to the essential nature of time-series for the control signal of the feeding system, the long-short term memory network (LSTM) has direct advantages for modeling of time-series response of the feeding axis. In our early work [30], a network called the NAR-LSTM network is constructed based on the traditional LSTM and the idea of the feedback control of the feeding system, for which the actual position of the feeding axis for the current time instant is predicted by taking the previous instant as the direct feedback. In this work, the idea of NAR-LSTM is utilized for constructing the model for predicting NLTE of the feeding axis.

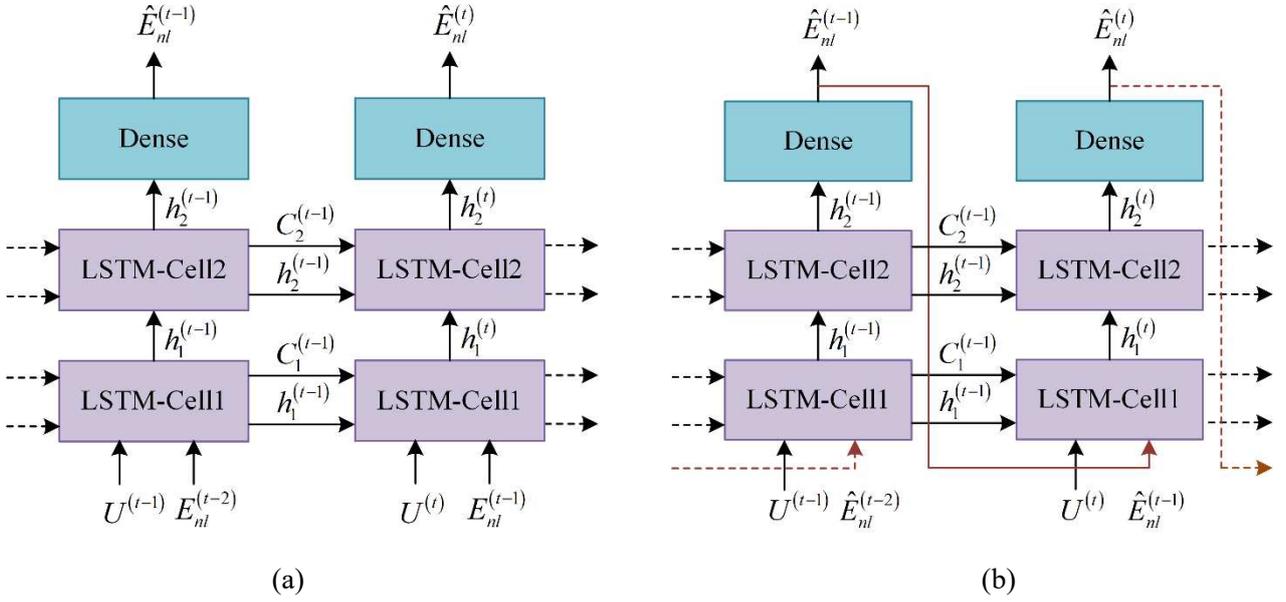


Fig. 5. Architectures of the NAR-LSTM network. (a) Open-loop architecture; (b) Closed-loop architecture

There are two different architectures of the NAR-LSTM network: open-loop and closed-loop as respectively shown in Fig. 5(a) and (b). For the open-loop architecture, the actual nonlinear tracking error of $(t-1)$ -th instant $E_{nl}^{(t-1)}$ is taken as one of the inputs for predicting the $\hat{E}_{nl}^{(t)}$ of t -th instant, while for the closed-loop architecture the previously predicted result $\hat{E}_{nl}^{(t-1)}$ is taken as the input for predicting the $\hat{E}_{nl}^{(t)}$. In this work, the mathematical expression of these two architectures is given by Eq. (5) and Eq. (6), respectively.

$$\hat{E}_{nl}^{(t)} = H \left(E_{nl}^{(t-1)}, E_{nl}^{(t-2)}, \mathbf{L}, E_{nl}^{(t-g-1)}; U^{(t)}, U^{(t-1)}, \mathbf{L}, U^{(t-g)} \right) \quad (5)$$

$$\hat{E}_{nl}^{(t)} = H \left(\hat{E}_{nl}^{(t-1)}, \hat{E}_{nl}^{(t-2)}, \mathbf{L}, \hat{E}_{nl}^{(t-g-1)}; U^{(t)}, U^{(t-1)}, \mathbf{L}, U^{(t-g)} \right) \quad (6)$$

where in Eq. (5) and Eq. (6), H is the nonlinear function for modeling the NLTE; $U^{(t)} = [v^{(t)}, a^{(t)}, J^{(t)}]$ is the input vector of the t -th instant, and $v^{(t)}$, $a^{(t)}$, and $J^{(t)}$ are respectively the reference velocity, acceleration,

1 and jerk as retrieved from the CNC system; g is as a parameter of the network, meaning that the $\hat{E}_{nl}^{(t)}$ is
 2 calculated based on the g -time span of the historical data.

3 For the actual NLTE $E_{nl}^{(t)}$ of Eq. (5), it can be calculated from:

$$E_{nl}^{(t)} = E_a^{(t)} - E^{(t)} \quad (7)$$

4 where, the actual tracking error $E_a^{(t)}$ can be obtained from the CNC system, and $E^{(t)}$ can be calculated from
 5 Eq. (3).

6 To predict the nonlinear tracking error $\hat{E}_{nl}^{(t)}$ of the t -th instant, the data $U^{(t)}$ from the $(t-g)$ -th instant to
 7 the current instant is utilized in the NAR-LSTM structure of Fig. 5. Besides, the historical NLTE data from $(t-$
 8 $g-1)$ -th instant to $(t-1)$ -th instant is also utilized for predicting the current $\hat{E}_{nl}^{(t)}$, but the data source is different
 9 for the training stage and predicting stage per Eq. (5) and Eq. (6): during the training phase, the open-loop
 10 architecture is used because of the availability of the actual NLTE data $E_{nl}^{(t-1)}, E_{nl}^{(t-2)}, \dots, E_{nl}^{(t-g-1)}$; when
 11 predicting the NLTE $\hat{E}_{nl}^{(t)}$, there is no existing NLTE data, so the previous predicted NLTE from the $(t-g-1)$ -th
 12 instant to the $(t-1)$ -th instant is utilized, i.e., the $\hat{E}_{nl}^{(t-1)}, \hat{E}_{nl}^{(t-2)}, \dots, \hat{E}_{nl}^{(t-g-1)}$ as presented in Eq. (6).

13 Based on the proposed NAR-LSTM model, the NLTE of the feeding axis can be predicted from Eq. (6)
 14 and the OTE can be calculated from Eq. (4). Details regarding the prediction results are shown in Section 4.2.
 15 In the next subsection, a contour error prediction model will be presented based on the calculated OTE of each
 16 feeding axis.

18 2.3. Contour error calculation

19 Given the OTE as predicted from the approach presented in Section 2.2, the predicted position of the
 20 feeding axis is

$$\hat{w}^{(t)} = u^{(t)} + \hat{E}_a^{(t)} \quad (8)$$

21 where, $\hat{E}_a^{(t)}$ is the predicted OTE at the t -th instant; $u^{(t)}$ is the reference position of feeding axis; $\hat{w}^{(t)}$ is the
 22 corresponding predicted actual position of the $u^{(t)}$.

23 With the actual position of the two axes (i.e., the X and Y-axis in the 2D contouring machining), the actual
 24 path of the CNC machining can be defined, and the contour error, which is the deviation between the reference
 25 path and the actual path, can be calculated.

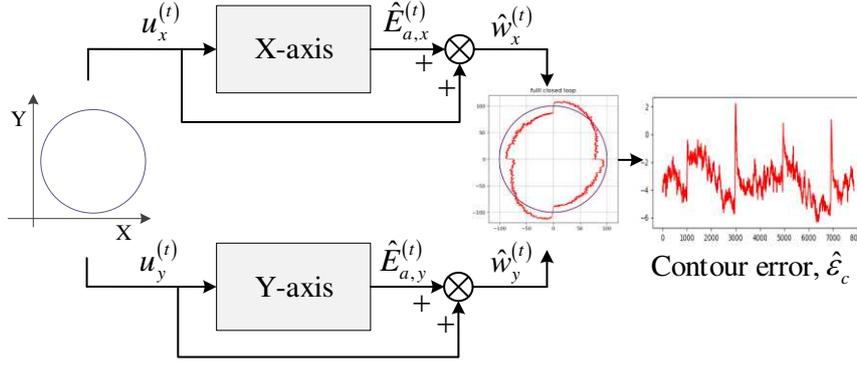


Fig. 6. The procedure of contour error calculation

The general idea of calculating the contour error is shown in Fig. 6. Before the procedure can be explained, the following symbols defined in Fig. 6 are explained at the t -th instant:

$u_x^{(t)}, u_y^{(t)}$ - the reference position based on the given path

$\hat{E}_{a,x}^{(t)}, \hat{E}_{a,y}^{(t)}$ - the predicted OTE of the feeding axis

$\hat{w}_x^{(t)}, \hat{w}_y^{(t)}$ - the predicted position of the feeding axis

$\hat{\varepsilon}_c$ - the predicted contour errors of the feeding system with considering both X and Y-axis

With the above definitions, the proposed procedure for building the Prediction Model of Contour Error (PMCE) is explained as follows.

Step-1: Estimate the position of the X and Y axis at the t -th instant. Taking the X-axis as the example: make $t=g$ (g is the specific time span of the historical data for modeling the NAR-LSTM); given the data of the first g instants, estimated the NLTE $\hat{E}_{nl,x}^{(t)}$ with the model of Eq. (6); calculate the OTE $\hat{E}_{a,x}^{(t)}$ and the predicted position $\hat{w}_x^{(t)}$ per Eq. (4) and Eq. (8), respectively. Likewise, $\hat{w}_y^{(t)}$ of the Y-axis can be predicted.

Step-2: Get the predicted point $P^{(t)} = (\hat{w}_x^{(t)}, \hat{w}_y^{(t)})$ for the t -th instant.

Step-3: Iterate Step-1 and Step-2 by increasing the index of t , until it reaches the last point. The predicted points will form a curve and assume it is $\{\dots, P^{(t)}, P^{(t+1)}, \dots\}$.

Step-4: Calculate the contour error for each predicted point. Taking the point $P^{(t)}$ shown in Fig. 7 as an example, calculate the distances from the $P^{(t)}$ to the all the line segments of $\{u^{(t-g)}u^{(t-g+1)}, u^{(t-g+1)}u^{(t-g+2)}, \dots, u^{(t-1)}u^{(t)}\}$, where the $u^{(i)}$ ($i \in [t-g, t]$) is the point on the reference path corresponding to the i -th instant; the contour error is then selected as the minimal value among these distances, e.g., the distance between the $p_*^{(t)}$ and $P^{(t)}$ as shown in Fig. 7. The contour error of each predicted point is calculated which altogether forms the predicted contour error $\hat{\varepsilon}_c$ of the reference path, as shown in Fig. 6. The detailed process for calculating the contour error could refer to the work [17].

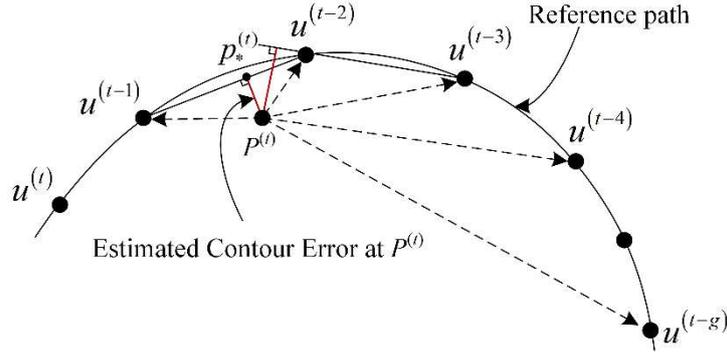


Fig. 7. Contour error calculation of $P^{(t)}$ for a free-form reference path

Given the PMCE as presented in the section, a reinforcement learning-based approach is presented to determine the optimal compensation strategy of contour error, as to be presented in the next section.

3. Reinforcement learning-based contour error compensation

In this section, a contour error compensation strategy is first presented to address the lag-characteristic of the feeding system and ensure good compensation performance, and then a reinforcement learning-based approach is designed to determine the essential parameters of the proposed strategy.

3.1. Contour error compensation strategy

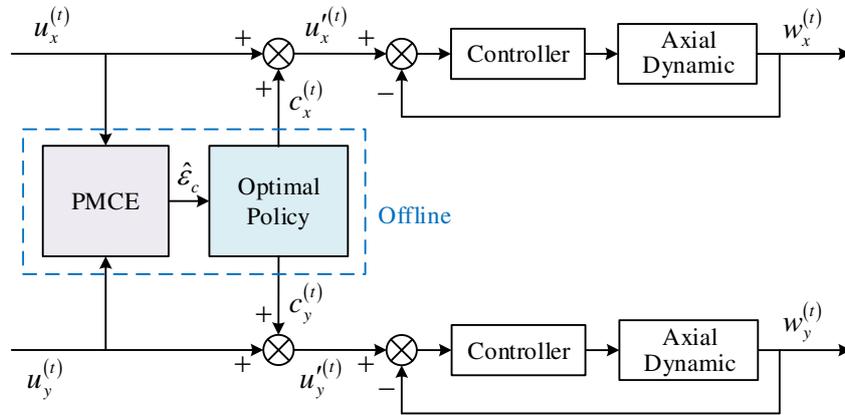


Fig. 8. The general idea of the contour error compensation method

The general idea of the proposed contour error compensation method is shown in Fig. 8. Before the approach is explained, the following definitions are given at the t -th sampling instant:

$u_x^{(t)}$, $u_y^{(t)}$ - the reference position from the given part program

$\hat{\epsilon}_c$ - the contour error calculated from the PMCE of Section 2.3

$u_x'^{(t)}$, $u_y'^{(t)}$ - the compensated position of the corresponding reference path

1 $w_x^{(t)}, w_y^{(t)}$ - the actual position of the feeding axis

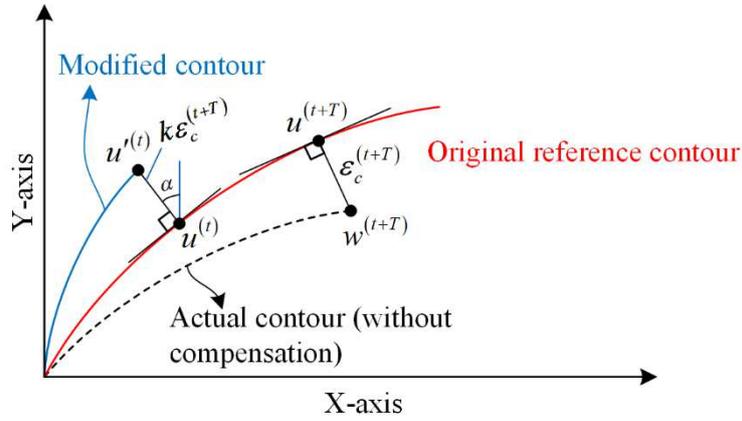
2 $c_x^{(t)}, c_y^{(t)}$ - the compensation values of X- and Y-axis

3 As shown in Fig. 8, the contour error $\hat{\varepsilon}_c$ of a given part program is predicted with the PMCE as proposed
 4 in Section 2 in an offline way. Based on the $\hat{\varepsilon}_c$, an optimal policy is designed to calculate the compensation
 5 value $c_x^{(t)}$ and $c_y^{(t)}$ for the X- and Y-axis, respectively. After that, the compensated position will be sent to the
 6 CNC controller to reduce the contour error and improve the machining accuracy.

7 To achieve good compensation performance, we designed two parameters in the proposed compensation
 8 model, i.e., the forward compensation cycle T and the compensation rate k , which are respectively defined to
 9 avoid the lag-characteristic of the control system and mechanical system of the feeding axis and ensure good
 10 compensation performance. Given a reference position $u^{(t)}$ at the t -th instance, the corresponding compensated
 11 position $u'^{(t)}$ as calculated from the compensation policy can be regarded as a function of the parameter T and
 12 k , and

$$u'^{(t)} = \Phi\left(u^{(t)}, k \cdot \varepsilon_c^{(t+T)}\right) \quad (9)$$

13 In Eq. (9), the $u'^{(t)}$ at the t -th instance is compensated based on the future contour error $\varepsilon_c^{(t+T)}$ at the $(t+T)$ -
 14 th constant, in this way, the lag-characteristic is counterbalanced by the parameter T at the t -th instance, as
 15 shown in Fig. 9.



16
 17 **Fig. 9.** Contour error compensation of the reference contour

18 In the proposed compensation strategy, the compensation values for the X- and Y-axis is denoted as $c_x^{(t)}$
 19 and $c_y^{(t)}$, respectively, and

$$\begin{cases} c_x^{(t)} = k \cdot \hat{\varepsilon}_c^{(t+T)} \cdot \sin(\alpha) \\ c_y^{(t)} = k \cdot \hat{\varepsilon}_c^{(t+T)} \cdot \cos(\alpha) \end{cases} \quad (10)$$

1 where α is the angle between the normal direction at $u^{(t)}$ and the Y-axis.

2 Given the compensation values for the X and Y axis per Eq. (10), the compensated positions $u_x^{(t)}$ and
 3 $u_y^{(t)}$ of the corresponding reference path are

$$\begin{cases} u_x^{(t)} = u_x^t + c_x^{(t)} \\ u_y^{(t)} = u_y^t + c_y^{(t)} \end{cases} \quad (11)$$

4 The compensated position as calculating from Eq. (11) is then sent to the CNC system to make the modified
 5 contour, which is formed by the compensated positions, closer to the original reference contour.

6 For the proposed compensation strategy, one issue remains that how the two compensation parameters (i.e.,
 7 the T and k of Eq. (10)) can be determined so that a good compensation result can be achieved. In the next
 8 subsection, a reinforcement learning-based approach is applied to determine the two parameters.

9

10 **3.2. TS-DQN based identification of compensation parameters**

11 As a popular and dominant reinforcement learning method, the Deep Q-network (DQN) learns by
 12 interacting with the environment to obtain the optimal policy under a set reward function, and it has been widely
 13 used in the domain of optimal control [31][32]. Considering the time-series feature of the feeding system, a
 14 modified DQN, named Time-Series Deep Q-network (TS-DQN), is presented for identifying the compensation
 15 parameters. The general framework of the TS-DQN is shown in Fig. 10, in which a time-series replay memory
 16 is used to replace the traditional replay memory of the classical DQN [31]. For the replay memory in the
 17 proposed TS-DQN, the rewards in historical steps can be re-updated according to the time-series state of the
 18 feeding system, so as to achieve higher parameters identification efficiency.

19 The TS-DQN based compensation parameters identification scheme is shown in Fig. 10. The PMCE of
 20 Section 2.3 is utilized as the environment for the proposed TS-DQN. Two Q-networks with the same structure
 21 but different parameters are used; and between them, one is the main Q-network for calculating the Q-value of
 22 the agent's action from Eq. (12) and selecting the action under the maximum Q-value, and the other is the target
 23 Q-network for assisting training the network by calculating the target Q-value.

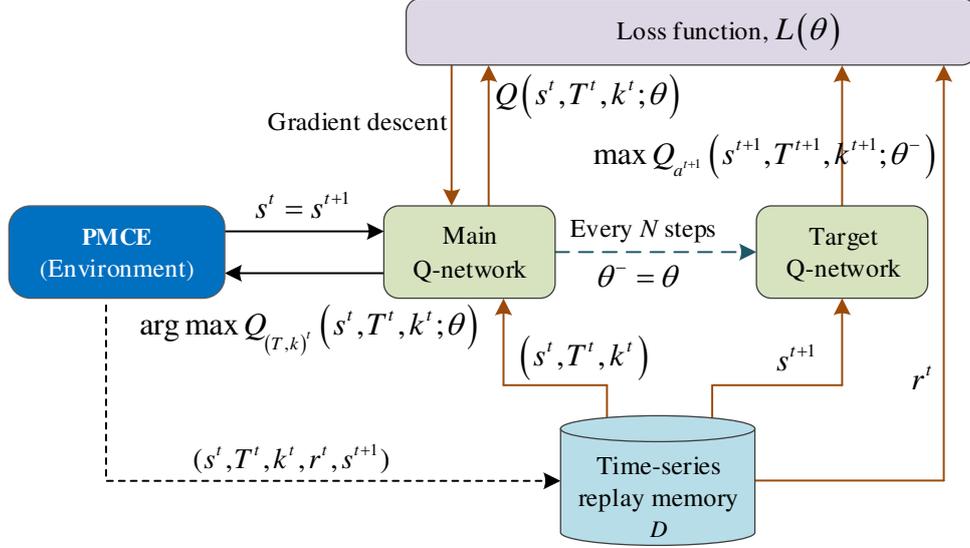


Fig. 10. The general structure of the proposed TS-DQN

The three core components of the TS-DQN algorithm are the prediction performance state s^t (superscript “ t ” represents the data at the t -th instance), decision action of compensation parameter (T^t, k^t) , and the reward for corresponding decision action r^t in a specific state. In addition, the proposed TS-DQN involves an agent, a set of states S , and a set of decision actions for (T, k) . In this paper, there are 8 potential decision actions, as listed in table 1.

Table 1 Decision action set of the compensation parameter

Action	(T_1, k_1)	(T_2, k_2)	(T_3, k_3)	(T_4, k_4)	(T_5, k_5)	(T_6, k_6)	(T_7, k_7)	(T_8, k_8)
Value	(1, 0.1)	(1, 0)	(1, -0.1)	(0, 0.1)	(0, -0.1)	(-1, 0.1)	(-1, 0)	(-1, -0.1)

By selecting and executing a decision action according to the ε -greedy policy, the agent changes from one state to another. In the meantime, the agent will get the reward r when executing a decision action in the specific state.

$$Q(s^t, T^t, k^t) \leftarrow Q(s^t, T^t, k^t) + \alpha \left[r^t + \gamma \max_{a^{t+1}} Q(s^{t+1}, T^{t+1}, k^{t+1}) - Q(s^t, T^t, k^t) \right] \quad (12)$$

where α is the learning rate and γ is the discount factor.

The data set $(s^t, T^t, k^t, r^t, s^{t+1})$ as obtained through the interaction of the agent with the environment at t -instant will be stored in the time-series replay memory. According to the time-series feature of the feeding system of the CNC machining, if the same decision action is selected in adjacent steps and a better compensation performance is obtained, the reward r^{t-1} in the last step will be re-updated in the time-series replay memory by Eq. (13),

$$r^{t-1} = r^{t-1} + p \cdot r^t \quad (13)$$

1 Then, the reward in m previous steps can be updated in the time-series replay memory by

$$r^{t-n} = r^{t-n} + p \cdot (r^{t-n+1} + \dots + r^{t-1} + r^t), n = 1, 2, \dots, m \quad (14)$$

2 where in Eq. (13) and Eq. (14), p is the time-series correlation factor, and m is the number of decision actions
3 that are continuing the same.

4 The Q-network of TS-DQN is trained by using a batch of the data from the time-series replay memory,
5 while the parameters θ of the main Q-network is updated by minimizing the loss function of Eq. (15) via
6 gradient descent algorithm.

$$L(\theta) = E \left[\left(TargetQ - Q(s^t, T^t, k^t; \theta) \right)^2 \right] \quad (15)$$

$$\text{and } TargetQ = r^t + \gamma \max_{T^{t+1}, k^{t+1}} Q(s^{t+1}, T^{t+1}, k^{t+1}; \theta^-)$$

7 where, θ and θ^- are the parameters of the main Q-network and the target Q-network, respectively.

8 The reward value function corresponding to the specific state and action is given in Eq. (16). By estimating
9 the compensation performance through the prediction model of contour error, the reward r^t at the t -th instant
10 can be given as

$$r^t = c \cdot (\varepsilon_E^t - \tau_E)^3 + 1 \quad (16)$$

$$\text{and } \varepsilon_E^t = E_{MSE}^t * 0.3 + abs(E^t)_{max} * 0.7$$

11 where, E_{MSE}^t is the mean square error (MSE) of the predicted contour error set E^t , and τ_E is the tolerance
12 of contour error and we choose $\tau_E = 2 \mu m$ in this paper, and c is a constant of 1000.

13 For the proposed TS-DQN as shown in Fig. 10, an algorithm called **TS-DQN with Experience Replay** is
14 given below for a clear explanation. As already alluded, a time-series replay memory of the proposed TS-DQN
15 is used to replace the traditional replay memory of DQN, and the re-updated method of rewards is added in the
16 time-series replay memory by setting $r^{t-1} = r^{t-1} + p \cdot r^t$ when $a^{t-1} = a^t$ and $s^{t-1} > s^t$.

Algorithm_ TS-DQN with Experience Replay

Initialize time-series replay memory D

Initialize action-value function Q with random parameters θ

Initialize target action-value function \hat{Q} with random weights $\theta^- = \theta$

for episode = 1, M **do**

Initialize sequence $s^1 = \{x^1\}$ and preprocessed sequenced $\phi^1 = \phi(s^1)$

for $t = 1, T$ **do**

With probability ε select a random compensation parameter decision action (T^t, k^t)

otherwise select $(T^t, k^t) = \arg \max_{T, k} Q(\phi(s^t), T, k; \theta)$

Execute action (T^t, k^t) in emulator and observe reward r^t and image x^{t+1}

Set $s^{t+1} = s^t, T^t, k^t, x^{t+1}$ and preprocess $\phi^{t+1} = \phi(s^{t+1})$

Store transition $(\phi^t, T^t, k^t, r^t, \phi^{t+1})$ in D

Sample random minibatch of transitions $(\phi^j, T^j, k^j, r^j, \phi^{j+1})$ from D

Set $y^j = \begin{cases} r^j & \text{if episode terminates at step } j+1 \\ r^j + \max_{T', k'} \hat{Q}(\phi^{j+1}, T', k'; \theta^-) & \text{otherwise} \end{cases}$

Set $r^{t-1} = r^{t-1} + p \cdot r^t$ if $(T, k)^{t-1} = (T, k)^t$ and $s^{t-1} > s^t$

Perform a gradient descent step on $(y^j - Q(\phi^j, T^j, k^j; \theta))^2$ for the network parameters θ

Every N steps reset $\hat{Q} = Q$

end for

end for

1

2 Based on the above **Algorithm_ TS-DQN with Experience Replay**, the two compensation parameters
 3 (i.e., the T and k of Eq. (10)) at each instance of the machining process are identified which minimizing the
 4 contour error. Results on the identification of the two parameters and the follow-up compensation will be
 5 presented the Section 4.

6

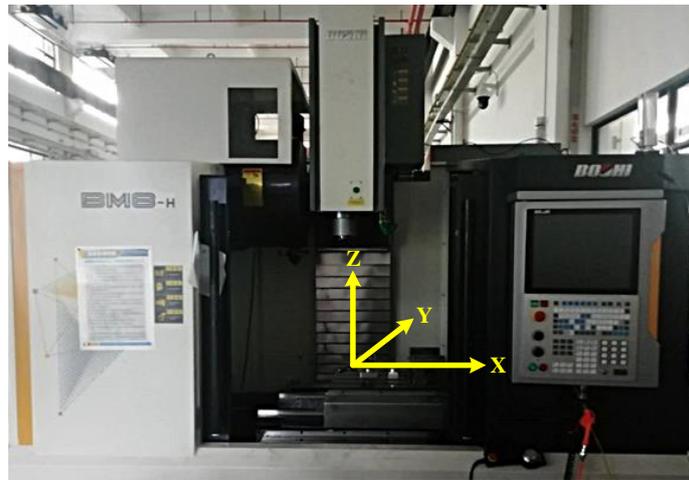
7 4. Experiments

8 In this section, some experiments are conducted to verify the proposed NAR-LSTM network-based contour
 9 error modeling approach and the TS-DQN-based contour error compensation strategy.

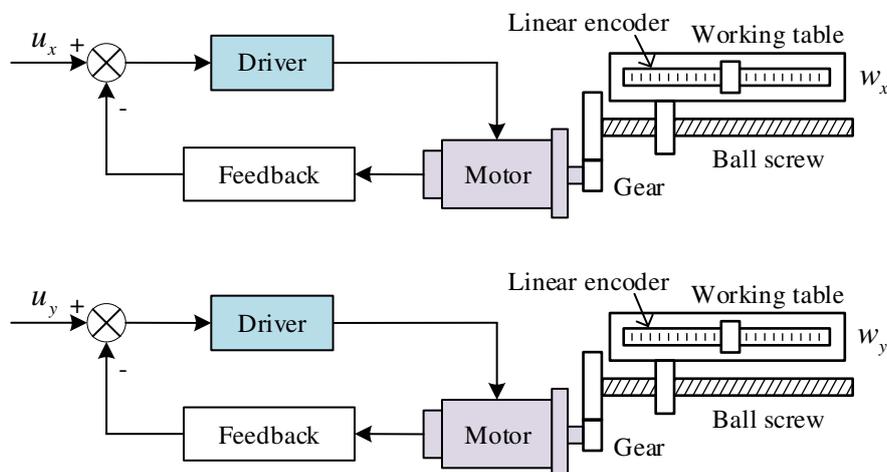
10 4.1. Experimental setup

11 The experimental verifications of the proposed contour error prediction and compensation approaches were

1 conducted toward the BM8-H three-axis CNC milling center equipped with a Huazhong HNC-818B controller,
 2 as shown in Fig. 11. For the machine tool, the 2D contour error resulting from tracking error (both linear and
 3 nonlinear) of X and Y axis is calculated and compensated, where, the reference positions u_x and u_y is
 4 generated and retrieved from the CNC system; and the actual positions of the feeding axis w_x and w_y are
 5 detected by the linear encoder mounting on the working table (shown in Fig. 12); both data is collected via the
 6 data acquisition function of the CNC system with the sampling frequency of 1 kHz.



7
8 **Fig. 11.** The BM8-H three-axis CNC milling center



9
10 **Fig. 12.** Data retrieved for the X and Y-axis from the controller

12 To testify the proposed approaches in terms of contour error prediction and compensation, one elementary
 13 contour (i.e., circle shown in Fig. 13(a)) and two more freeform contours (i.e., heart contour and goggles contour
 14 shown in Fig. 13(b) and Fig. 13(c), respectively) are chosen as test examples. For each contour, experiments are
 15 conducted on four feedrates of 1200 mm/min, 2400 mm/min, 3600 mm/min, and 4800 mm/min. Among the
 16 three contours, two of them (i.e., the circle, and heart) are taken as the training and validation samples for the
 17 proposed NAR-LSTM network and TS-DQN, while the goggles contour serves as the test sample. Details about
 18 the three contours are listed below.

- Circular contour: a circle centered at the origin with a radius of 50 mm.
- Heart contour: a NURBS curve with degree=4 and 7 control points [33], details on its geometric parameters are listed in Appendix B.
- Goggles contour: a NURBS curve with degree=4 and 9 control points [34], details on its geometric parameters are also listed in Appendix B

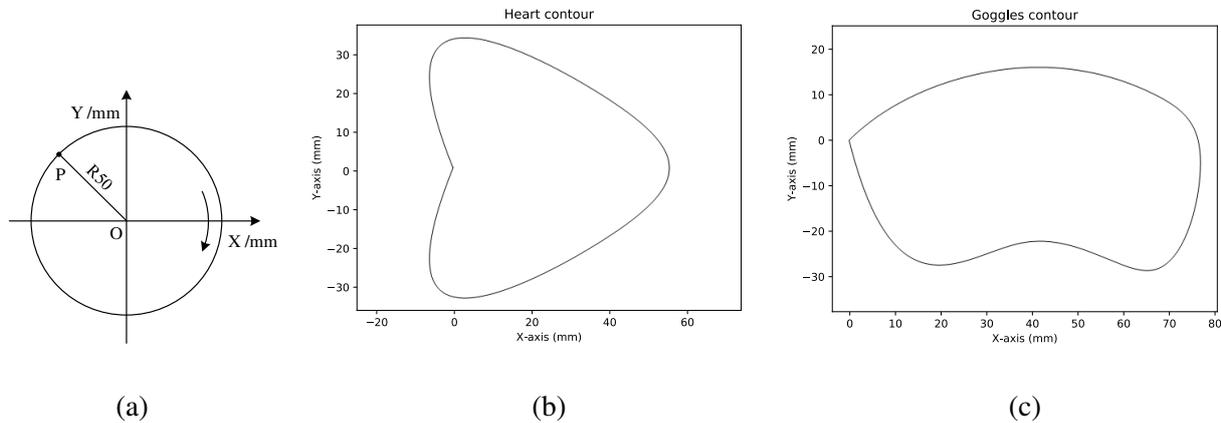


Fig. 13. Three test contours. (a) circular contour; (b) heart contour; (c) goggles contour.

For the proposed NAR-LSTM network in Section 2, its structure is presented in section 2.2. In each time-series block of the NAR-LSTM network, two basic LSTM-cell and one fully connected layer were used. The number of nodes in each cell and the parameter g were found by trial and error approach, and they are 60 and 14 in this paper, respectively. The dimensions of the input and output are 4 and 1 respectively. Moreover, as a typical recurrent neural network, the structure and parameters of the NAR-LSTM are reused in each control cycle. For the proposed TS-DQN in Section 3, its Q-network is a three-layer fully connected neural network with 10 nodes in the hidden layer, the input dimension being 30, and the output dimension being the total number of the decision actions 8.

4.2. Experimental results

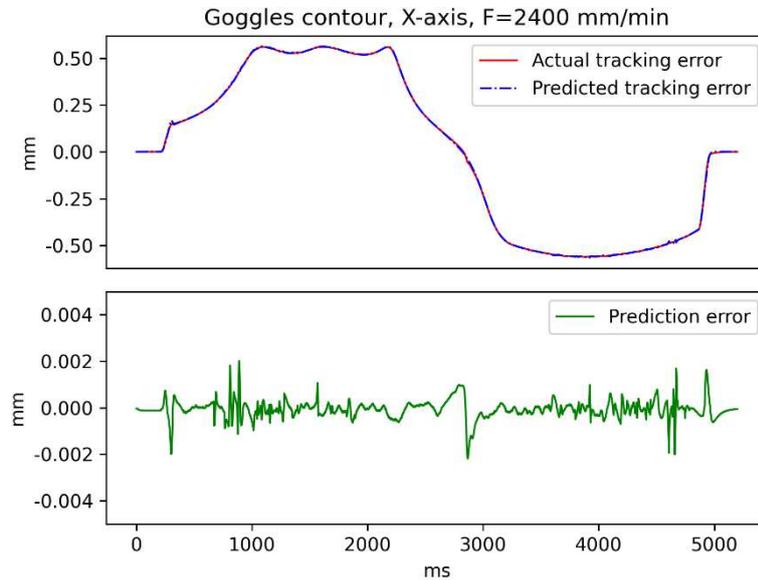
In this section, the experimental results on the NAR-LSTM network-based contour error prediction are first reported, and then the feasibility and advantage of the proposed TS-DQN-based contour error compensation approach are testified by simulation and physical cutting experiments.

4.2.1. Contour error prediction

The detailed results on the contour error prediction are presented for machining goggles contour with the feedrate of 2400 mm/min. Results on the other contours and the other feedrates are similar, with their details presented in Section 4.2.2.

The prediction results for the tracking error of the X and Y-axis from the approach of Section 2.1 and Section 2.2 are shown in Fig. 14 and Fig. 15, respectively. Results show that the predicted results match with the actual tracking error, which is retrieved from the controller of the CNC machine tools, very well. For the X-axis with the maximum actual tracking error of 0.5659 mm (as shown in Fig. 14), the prediction error, which is the deviation between the actual position and the predicted position, has the maximum value of 0.0019 mm,

1 which is only 0.34% of the corresponding actual tracking error. For the Y-axis which has the largest actual
 2 tracking error of 0.5661 mm, the maximum prediction error is 0.0021 mm, which is only 0.37% of the actual
 3 tracking error at this location. Results shown in Fig. 14 and Fig. 15 validated that, the proposed NAR-LSTM
 4 network-based tracking error modeling approach has very good predicted accuracy.

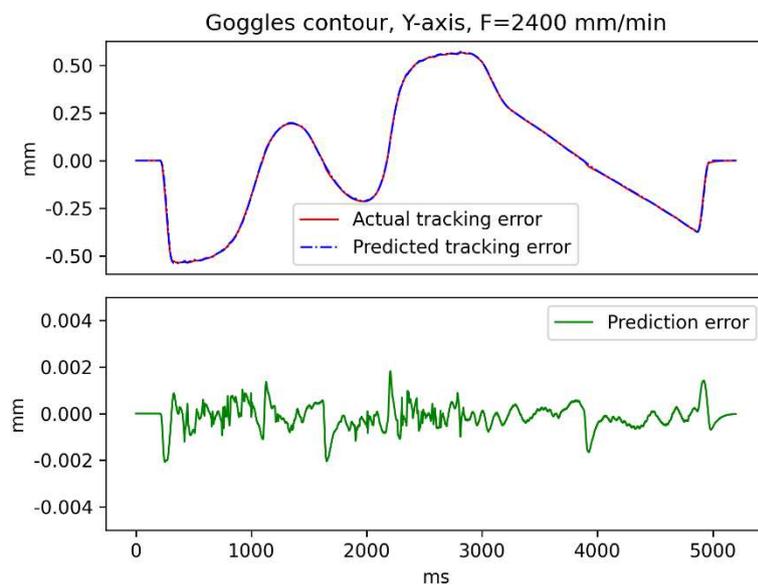


5

6

Fig. 14. Actual (red) and predicted (blue) tracking error, prediction error (green) of the X-axis

7



8

9

Fig. 15. Actual (red) and predicted (blue) tracking error, prediction error (green) of the Y-axis

10

11

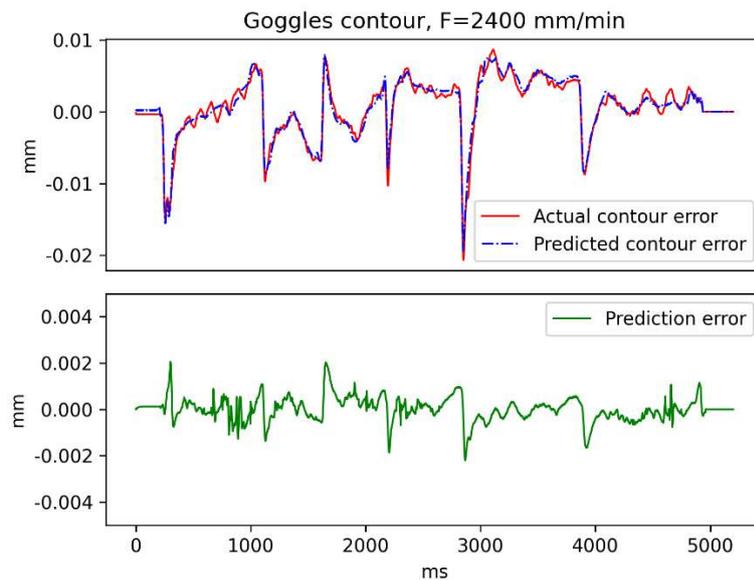
12

13

14

Based on the tracking error as estimated from the NAR-LSTM network, the contour error can be calculated based on the approach as presented in Section 2.3. Fig. 16 shows predicted contour error, actual contour error, and the deviation (prediction error) between them when machining the goggles contour. Results show that, for the location with the maximal actual contour error of 0.0219 mm, the predicted result is 0.0020 mm, with the prediction error being 9.1% of the actual contour error. Results in Fig. 16 validated that, based on the tracking

1 error as modeled via the NAR-LSTM network, the contour error of 2D CNC machining can be precisely
 2 predicted.



3
 4 **Fig. 16.** Actual (red) contour error, predicted (blue) contour error, and prediction error (green) for the
 5 goggles contour

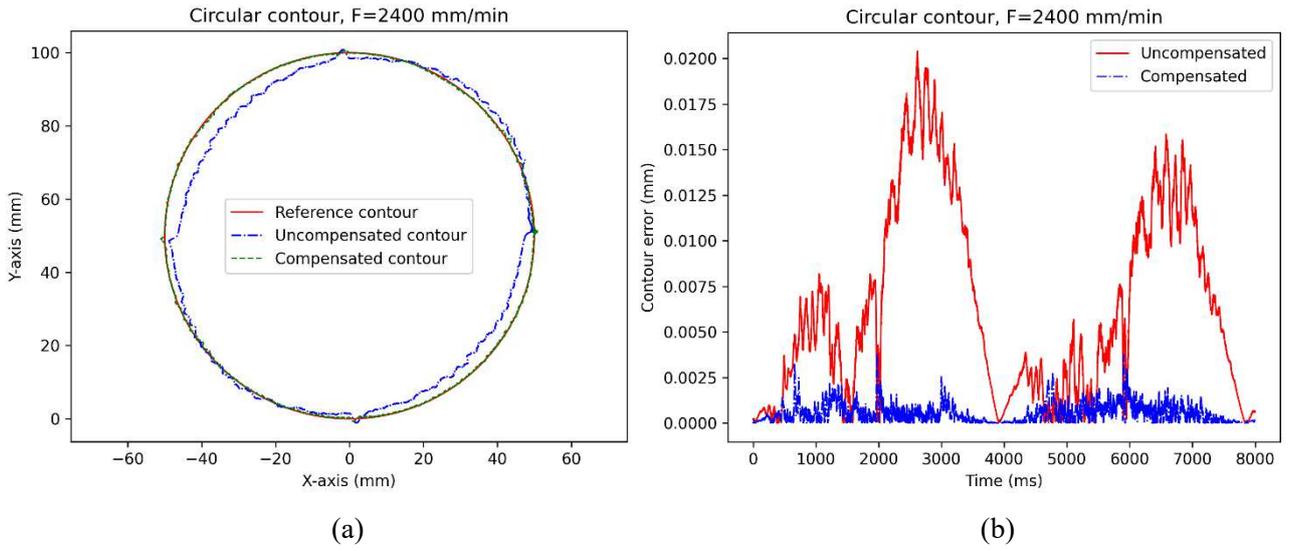
6 4.2.2. Contour error compensation

7 Based on the approach as presented in Section 3, the experiments for contour error compensation are
 8 conducted for circular contour, heart contour and goggles contour, with the results shown in Fig. 17, Fig. 18,
 9 and Fig. 19, respectively.

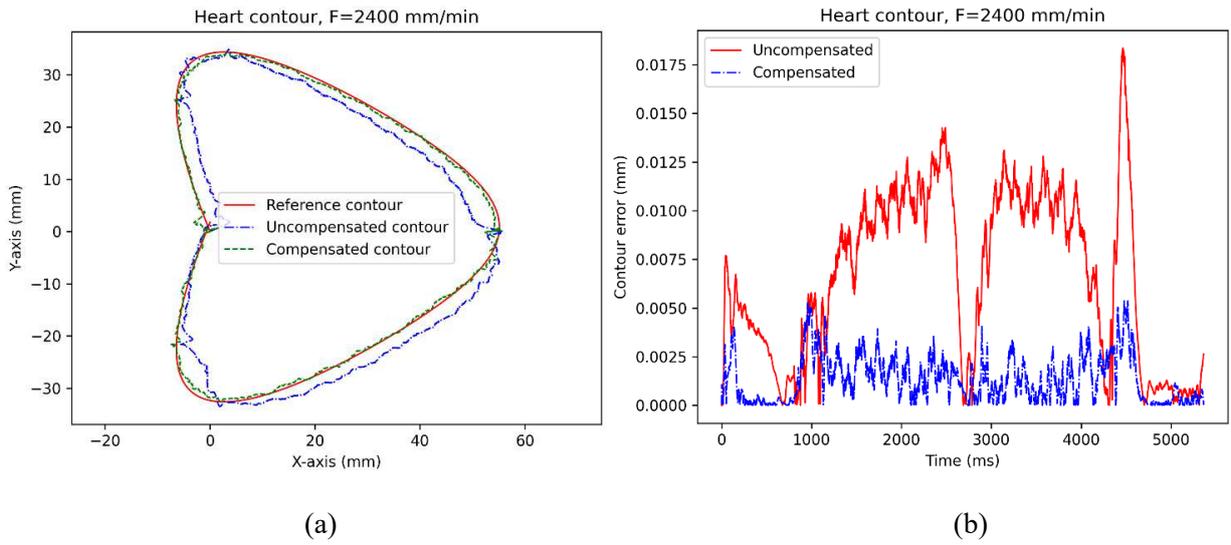
10 For the circular contour, the contour error has some significant reduction after compensation, with the
 11 maximum contour error reduced from 0.0204 mm to 0.0039 mm, an 80.88% reduction; the compensated contour
 12 is very closed to the original reference contour, as shown in Fig. 17. For the heart contour, the contour error is
 13 also largely reduced, with the maximum contour error reduced from 0.0183 mm to 0.0054 mm, as shown in Fig.
 14 18, which has a 70.49% reduction. For the goggles contour shown in Fig. 19, the maximum contour error is
 15 reduced from 0.0262 mm to 0.0048 mm, an 81.68% improvement.

16 Note that, to make the results obvious, the contour error shown in Fig. 17(a), Fig. 18(a) and Fig. 19(a) is
 17 magnified 200 times.

18



1
2
3 **Fig. 17.** Contour error compensation for circular contour. (a) reference, uncompensated, and compensated
4 circle; (b) contour error before and after compensation.



6
7
8 **Fig. 18.** Contour error compensation for heart contour. (a) reference, uncompensated, and compensated heart
9 contour; (b) contour error before and after compensation.

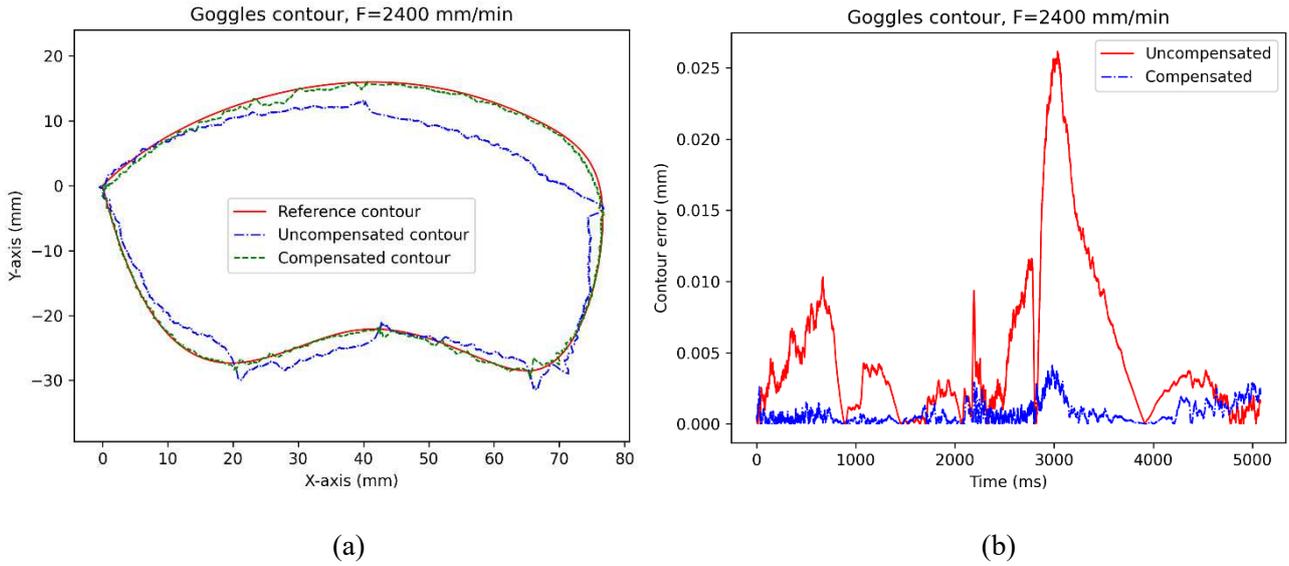


Fig. 19. Contour error compensation for goggles contour. (a) reference, uncompensated, and compensated goggles contour; (b) contour error before and after compensation.

The Fig. 17 - Fig. 19 are experimental results for a feedrate of 2400 mm/min; to comprehensively testify the feasibility and advantage of the proposed method, the contour error compensation is also conducted on other three feedrates, i.e., 1200 mm/min, 3600 mm/min and 4800 mm/min. To effectively evaluate the compensation results, the Maximal Absolute Error (MAE) of the contour error, which is the critical condition affecting the machining quality, is listed in the Table-1 for the total four feedrates, and

$$MAE = \max(|\varepsilon(k)|), \quad k = 1, 2, \dots, N \quad (17)$$

where, $\varepsilon(k)$ is the contour error at the k -th instant, and N is the total number of instants.

The experimental results in Table-1 show that, for various contours machined different feedrates, our approach can significantly reduce the contour errors of all cases, with the MAE of the contour error reduced for 70.49% ~ 84.53% after the compensation.

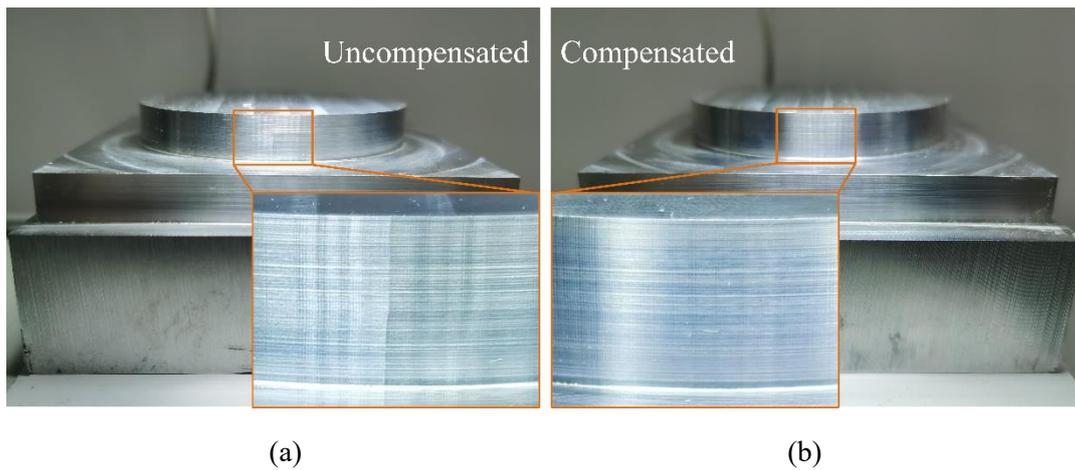
Table-1 Contour error for different contours with various feedrates [MAE (mm)].

Feedrate		1200 mm/min	2400 mm/min	3600 mm/min	4800 mm/min
Circular	Uncompensated	0.0177	0.0204	0.0232	0.0270
	Compensated	0.0028	0.0039	0.0055	0.0064
	Comparison	84.18% ↓	80.88% ↓	76.29% ↓	76.30% ↓
Heart	Uncompensated	0.0135	0.0183	0.0311	0.0472
	Compensated	0.0036	0.0054	0.0062	0.0073
	Comparison	73.33% ↓	70.49% ↓	80.06% ↓	84.53% ↓

Goggles	Uncompensated	0.0151	0.0262	0.0329	0.0379
	Compensated	0.0029	0.0048	0.0055	0.0068
	Comparison	80.79% ↓	81.68% ↓	83.28% ↓	82.06% ↓

1

2 To further testify the effectiveness of the proposed contour error compensation approach on improving the
3 machining quality, physical cutting experiments are conducted on the milling center BM8-H for machining the
4 circular contour of a radius of 35 mm and feedrate of 3000 mm/min. The machined parts before and after
5 compensation are shown in Fig. 20(a) and (b), respectively. Fig. 20 shows that, after contour error compensation,
6 the surface roughness can be improved drastically, especially for the machining region where the X or Y-axis of
7 the machine tool reverses the moving direction and the surface defects such as bump or groove could easily
8 occur. In addition to the surface roughness, the machining precision of the two circles is inspected by the
9 WENZEL LH108 coordinate measuring machine (Fig. 21(a)). Inspection results shown in Fig. 21(b) convince
10 that, based on the proposed compensation technology, the machining error can be reduced significantly with the
11 maximum machining error reduced from 15.1 μm to 7.6 μm .



12

13

14

Fig. 20. Machined parts of circular contour. (a) before compensation; (b) after compensation

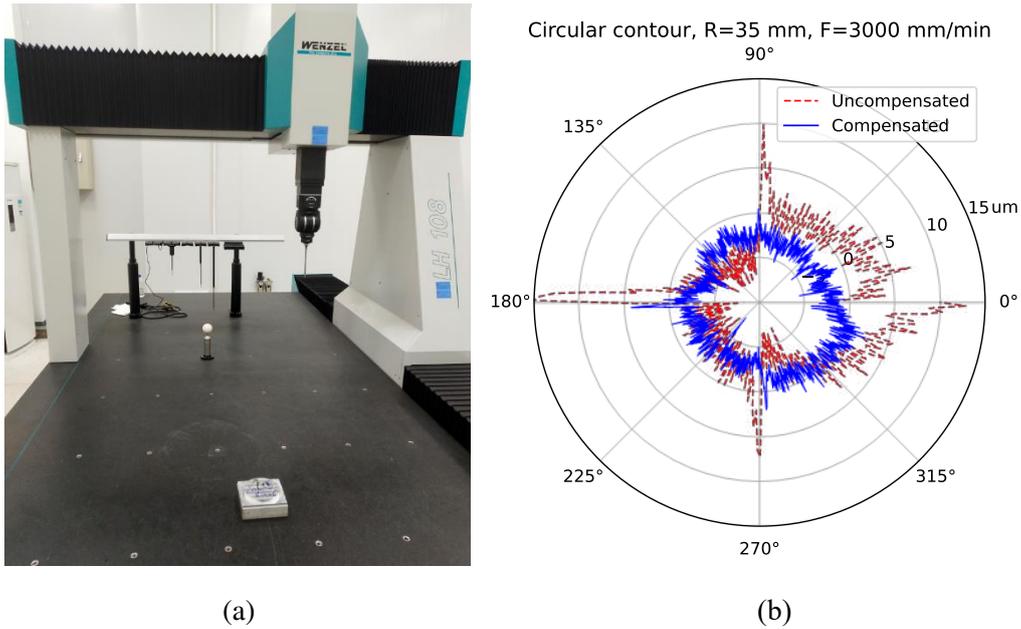


Fig. 21. (a) Inspection scenario of the machined parts; (b) machining error before and after compensation

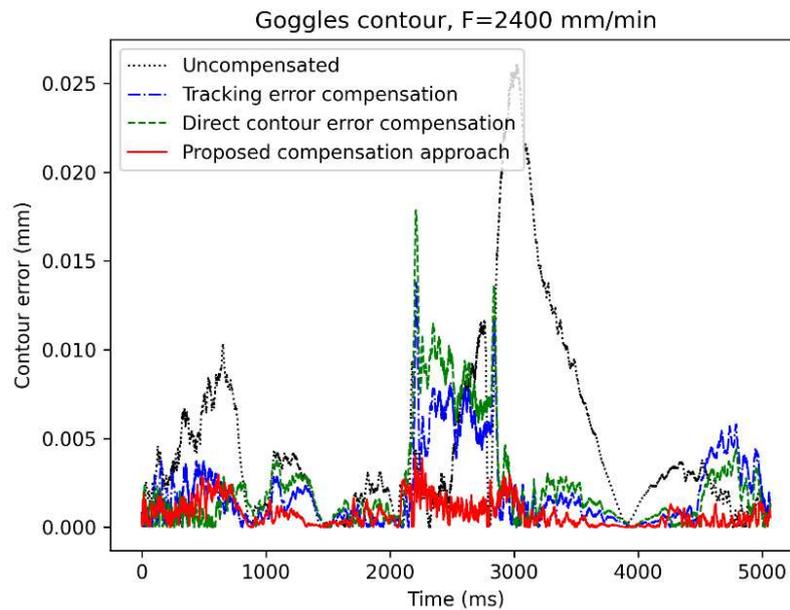
In our work, the compensation parameters of Section 3.1 are identified with the proposed TS-DQN. Compared with the traditional DQN [31], the proposed TS-DQN has time-series replay memory, which enables the rewards in historical steps reupdated based on the time-series state of the feeding system; the TS-DQN designed in this way can improve the parameter identification efficiency. The computational time for recognizing the parameters based on the proposed TS-DQN and traditional DQN is listed in Table-2. Both algorithms are run on the same laptop with Intel Core i7-2.60 GHz CPU and 16GB RAM. The DQN takes 25.51s when identifying the compensation parameters while the proposed TS-DQN only takes 14.83s, i.e., the proposed TS-DQN has a 41.81% reduction of computational time.

Table 2 Computational time of the proposed TS-DQN and the traditional DQN.

Algorithm	Computational time (s)
DQN	25.51
TS-DQN	14.83

In addition to our contour error compensation approach as proposed in Section 3, two more strategies (benchmarks) addressing the contour error are implemented to verify the advantage of the proposed approaches in terms of reducing the contour error. The first strategy is called the tracking error compensation method which directly compensates the tracking error of the feeding axis according to the prediction results of the NAR-LSTM network. The other strategy is called the direct contour error compensation approach that compensates the contour error without considering the two parameters of the forward compensation cycle and the compensation rate (i.e., by setting $T=0$ and $k=1$ in the Eq. (10)). Experimental results on the contour error compensation of our approach and the benchmarks on the goggles contour are presented in Fig. 22. Results show that our approach has much better compensation results with compared to the two benchmarks and the original uncompensated strategy: the maximal contour error of our approach is $4.82 \mu\text{m}$, which is significantly smaller than the two

1 benchmarks (i.e., $13.77\ \mu\text{m}$ and $17.85\ \mu\text{m}$) and the uncompensated strategy (i.e., $26.24\ \mu\text{m}$). Therefore, the
 2 proposed method can achieve better performance on the contour error compensation as compared to the other
 3 strategies.



4

5 **Fig. 22.** Comparison between the proposed approach and the benchmarks on the contour error compensation
 6 of goggles contour

7 In this section, some comprehensive experiments for testifying the proposed NAR-LSTM network-based
 8 contour error prediction and the TS-DQN-based contour error compensation approaches are designed and
 9 conducted. Experimental results show that, the proposed approach can have good prediction accuracy on the
 10 contour error (prediction error less than 9.1%); and for the further contour error compensation strategy as
 11 presented in this paper, it can significantly counterbalance the contour error of the machining process (the MAE
 12 of the contour error is reduced for 70.49% ~ 84.53%) with the machining quality improved significantly after
 13 the compensation (maximal machining error reduced 49.67% for machining circular contour). Besides, the
 14 proposed TS-DQN-based contour error compensation approach has much better performance in terms of both
 15 the computational efficiency compared to the traditional approach (a 41.81% reduction of computational time)
 16 and the compensated results as compared to the benchmarks.

17

18 5. Conclusion

19 In this paper, a set of approaches is presented for the contour error compensation of the CNC machining.
 20 To achieve this goal, a deep learning network NAR-LSTM is firstly designed that can precisely predict the
 21 tracking error of the machine's feed axis from the internal data of the CNC system, based on which the contour
 22 error can be calculated. Then, a contour error compensation strategy is proposed for which the compensation
 23 parameters are identified based on the TS-DQN-based reinforcement learning approach. The validities of the
 24 contour error prediction and compensation methods were confirmed by extensive experiments on a three-axis
 25 CNC machine tool. The experimental results show that the proposed method can effectively improve the
 26 accuracy of the contour error prediction, both for the elementary shape like a circle and the freeform curves like

1 the heart contour and goggles contour. Also, experimental results validate the effectiveness and the advantage
 2 of the proposed contour error compensation approach in terms of improving the machining quality and enhance
 3 the computational performance.

4 Regarding the future work of this paper, in addition to the 2D contour, the proposed deep learning-based
 5 tracking error modeling and reinforcement learning-based contour error compensation approaches could be
 6 extended for improving the machining quality of the 3D surface machining. Besides, we will focus on how to
 7 reuse the generated compensation values for different feedrate via the technology of transfer learning. At last,
 8 based on the control error prediction result, some feedforward control methods will be developed to further
 9 increase the contouring accuracy of CNC machining.

11 Declarations

12 **Funding:** This research was supported by the National Science and Technology Major Project [grant
 13 number 2018ZX04035002-002] and the National Natural Science Foundation of China [grant numbers
 14 51575210 and 51675204].

15 **Conflicts of interest:** None.

16 **Authors' contributions:** Yakun Jiang implemented all the algorithms and drafted the paper; Jihong Chen,
 17 Huicheng Zhou and Jianzhong Yang help design the algorithm and experiments of this paper; Pengcheng Hu
 18 proposed the original idea and revised the paper; Junxiang Wang help implement the experiments.

20 Appendix A. Geometric parameters of the two NURBS curves

21 Heart contour: Order: $k = 4$, knot vector: $\{0, 0, 0, 0.15, 0.5, 0.5, 0.85, 1, 1, 1\}$; control points (x, y) : $\{(0.0,$
 22 $0.0), (-20.0, 50.0), (40.0, 20.0), (75.0, 0.0), (40.0, -20.0), (-20.0, -50.0), (0.0, 0.0)\}$; weights: $\{1.0, 1.0, 1.0, 1.0,$
 23 $1.0, 1.0, 1.0\}$.

24 Goggles contour: Order: $k = 4$, knot vector: $\{0, 0, 0, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9, 1, 1, 1\}$; control points
 25 (x, y) : $\{(0.0, 0.0), (10.0, -40.0), (40.0, -10.0), (70.0, -40.0), (80.0, 0.0), (70.0, 10.0), (40.0, 20.0), (10.0, 10.0),$
 26 $(0.0, 0.0)\}$; weights: $\{1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0\}$.

27 References

- 28 [1] D. N. Song, Y. G. Zhong, and J. W. Ma, "Third-order contour-error estimation for arbitrary free-form paths in
 29 contour-following tasks," *Precis. Eng.*, vol. 60, no. April, pp. 85–92, 2019, doi:
 30 10.1016/j.precisioneng.2019.07.009.
- 31 [2] Q. Hu, Y. Chen, and J. Yang, "On-line contour error estimation and control for corner smoothed five-axis tool
 32 paths," *Int. J. Mech. Sci.*, vol. 171, no. August 2019, p. 105377, 2020, doi: 10.1016/j.ijmesci.2019.105377.
- 33 [3] T. jun Li, J. hui Yuan, Y. min Zhang, and C. yu Zhao, "Time-varying reliability prediction modeling of positioning
 34 accuracy influenced by frictional heat of ball-screw systems for CNC machine tools," *Precis. Eng.*, vol. 64, no.
 35 April, pp. 147–156, 2020, doi: 10.1016/j.precisioneng.2020.04.002.
- 36 [4] L. Yang, L. Zhang, and D. Li, "Five-axis CNC machine tool contour error control method," *ACM Int. Conf.*
 37 *Proceeding Ser.*, pp. 12–16, 2019, doi: 10.1145/3366194.3366197.

- 1 [5] S. Yang, A. H. Ghasemi, X. Lu, and C. E. Okwudire, "Pre-compensation of servo contour errors using a model
2 predictive control framework," *Int. J. Mach. Tools Manuf.*, vol. 98, pp. 50–60, 2015.
- 3 [6] D. N. Song, Y. G. Zhong, and J. W. Ma, "Third-order contour-error estimation for arbitrary free-form paths in
4 contour-following tasks," *Precis. Eng.*, vol. 60, no. August, pp. 85–92, 2019, doi:
5 10.1016/j.precisioneng.2019.07.009.
- 6 [7] J. Chin and T. Lin, "Cross-coupled precompensation method for the contouring accuracy of computer numerically
7 controlled machine tools," *Int. J. Mach. Tools Manuf.*, vol. 37, no. 7, pp. 947–967, 1997.
- 8 [8] M. Yang, J. Yang, L. Zhu, and X. Yu, "A novel curvature circle iterative algorithm for contour error control of
9 multi-axis CNC machine tools," *Precis. Eng.*, vol. 65, no. April, pp. 23–31, 2020, doi:
10 10.1016/j.precisioneng.2020.05.005.
- 11 [9] Y. Koren, "Cross-Coupled Biaxial Computer Control for Manufacturing Systems," *ASME J. Dyn. Syst. Meas.
12 Control*, vol. 102, no. December, pp. 265–272, 1980, doi: 10.1115/1.3149612.
- 13 [10] K. Srinivasan and P. K. Kulkarni, "Cross-coupled control of biaxial feed drive servomechanisms," *J. Dyn. Syst.
14 Meas. Control. Trans. ASME*, vol. 112, no. 2, pp. 225–232, 1990, doi: 10.1115/1.2896129.
- 15 [11] Y. Koren and C. C. Lo, "Variable-Gain Cross-Coupling Controller for Contouring," *CIRP Ann. - Manuf. Technol.*,
16 vol. 40, no. 1, pp. 371–374, 1991, doi: 10.1016/S0007-8506(07)62009-5.
- 17 [12] S. S. Yeh and P. Lo Hsu, "Estimation of the contouring error vector for the cross-coupled control design,"
18 *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 1, pp. 44–51, 2002, doi: 10.1109/3516.990886.
- 19 [13] D. Zhang, J. Yang, Y. Chen, and Y. Chen, "A two-layered cross coupling control scheme for a three-dimensional
20 motion control system," *Int. J. Mach. Tools Manuf.*, vol. 98, pp. 12–20, 2015, doi:
21 10.1016/j.ijmachtools.2015.08.001.
- 22 [14] H. Lu *et al.*, "Cross-coupled fuzzy logic sliding mode control of dual-driving feed system," *Adv. Mech. Eng.*, vol.
23 10, no. 2, pp. 1–17, 2018, doi: 10.1177/1687814018755518.
- 24 [15] W. Liu, F. Ren, Y. Sun, and S. Jiang, "Contour error pre-compensation for three-axis machine tools by using cross-
25 coupled dynamic friction control," *Int. J. Adv. Manuf. Technol.*, vol. 98, no. 1–4, pp. 551–563, 2018, doi:
26 10.1007/s00170-018-2189-1.
- 27 [16] F. Du, X. Feng, P. Li, J. Wang, Z. Wang, and C. Yu, "Cross-coupled intelligent control for a novel two-axis
28 differential micro-feed system," *Adv. Mech. Eng.*, vol. 10, no. 6, pp. 1–17, 2018, doi: 10.1177/1687814018774628.
- 29 [17] F. Huo, X. C. Xi, and A. N. Poo, "Generalized Taylor series expansion for free-form two-dimensional contour error
30 compensation," *Int. J. Mach. Tools Manuf.*, vol. 53, no. 1, pp. 91–99, 2012, doi: 10.1016/j.ijmachtools.2011.10.001.
- 31 [18] L. Zhu, H. Zhao, and H. Ding, "Real-time contouring error estimation for multi-axis motion systems using the
32 second-order approximation," *Int. J. Mach. Tools Manuf.*, vol. 68, pp. 75–80, 2013, doi:
33 10.1016/j.ijmachtools.2013.01.008.
- 34 [19] E. K and A. Y, "High speed contouring control algorithm for CNC machine tools," *Proc. ASME Dyn. Syst. Control
35 Div.*, vol. november, pp. 463–469, 1998.
- 36 [20] X. Li, H. Zhao, X. Zhao, and H. Ding, "Contouring compensation control based on high accuracy contour error
37 estimation for multi-axis motion systems," *Int. J. Adv. Manuf. Technol.*, pp. 2263–2273, 2017, doi: 10.1007/s00170-
38 017-0689-z.
- 39 [21] C. Hu, Z. Wang, Y. Zhu, and M. Zhang, "Accurate three-dimensional contouring error estimation and
40 compensationscheme with zero-phase filter," *Int. J. Mach. Tools Manuf.*, vol. 128, pp. 33–40, 2018, doi:

- 1 10.1016/j.ijmachtools.2018.01.001.
- 2 [22] F. Huo and A. N. Poo, "Precision contouring control of machine tools," *Int. J. Adv. Manuf. Technol.*, vol. 64, no.
3 1–4, pp. 319–333, 2013, doi: 10.1007/s00170-012-4015-5.
- 4 [23] F. Huo and A. N. Poo, "Nonlinear autoregressive network with exogenous inputs based contour error reduction in
5 CNC machines," *Int. J. Mach. Tools Manuf.*, vol. 67, pp. 45–52, 2013, doi: 10.1016/j.ijmachtools.2012.12.007.
- 6 [24] K. Erwinski, M. Paprocki, A. Wawrzak, and L. M. Grzesiak, "Neural network contour error predictor in CNC
7 control systems," *2016 21st Int. Conf. Methods Model. Autom. Robot. MMAR 2016*, pp. 537–542, 2016, doi:
8 10.1109/MMAR.2016.7575193.
- 9 [25] D. Zhang, Y. Chen, and Y. Chen, "Iterative pre-compensation scheme of tracking error for contouring error
10 reduction," *Int. J. Adv. Manuf. Technol.*, vol. 87, no. 9–12, pp. 3279–3288, 2016, doi: 10.1007/s00170-016-8735-
11 9.
- 12 [26] M. Yang, J. Yang, and H. Ding, "A two-stage friction model and its application in tracking error pre-compensation
13 of CNC machine tools," *Precis. Eng.*, vol. 51, no. May 2017, pp. 426–436, 2018, doi:
14 10.1016/j.precisioneng.2017.09.014.
- 15 [27] D. Kono, T. Inagaki, A. Matsubara, and I. Yamaji, "Stiffness model of machine tool supports using contact stiffness,"
16 *Precis. Eng.*, vol. 37, no. 3, pp. 650–657, 2013, doi: 10.1016/j.precisioneng.2013.01.010.
- 17 [28] L. Lu, H. Zhao, and P. Zhang, "The Position Control of Machine Tool Based on Fuzzy Adaptive PID Control,"
18 *2019 IEEE 8th Int. Conf. Fluid Power Mechatronics*, 2019, doi: DOI: 10.1109/FPM45753.2019.9035742.
- 19 [29] S. Shi, J. Lin, X. Wang, and X. Xu, "Analysis of the transient backlash error in CNC machine tools with closed
20 loops," *Int. J. Mach. Tools Manuf.*, vol. 93, pp. 49–60, 2015, doi: 10.1016/j.ijmachtools.2015.03.009.
- 21 [30] Y. Jiang, J. Chen, H. Zhou, J. Yang, and G. Xu, "Nonlinear time-series modeling of feed drive system based on
22 motion states classification," *J. Intell. Manuf.*, 2020, doi: 10.1007/s10845-020-01546-5.
- 23 [31] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–
24 533, 2015, doi: 10.1038/nature14236.
- 25 [32] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and Autonomous Control Using
26 Reinforcement Learning: A Survey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062,
27 2018, doi: 10.1109/TNNLS.2017.2773458.
- 28 [33] H. Zhu, M. Wang, and B. Wang, "Research on interpolation of NURBS curve based on fractional power," *J. Phys.
29 Conf. Ser.*, vol. 1633, no. 1, 2020, doi: 10.1088/1742-6596/1633/1/012005.
- 30 [34] R. R. Hiemstra, K. M. Shepherd, M. J. Johnson, L. Quan, and T. J. R. Hughes, "Towards untrimmed NURBS: CAD
31 embedded reparameterization of trimmed B-rep geometry using frame-field guided global parameterization,"
32 *Comput. Methods Appl. Mech. Eng.*, vol. 369, p. 113227, 2020, doi: 10.1016/j.cma.2020.113227.

Figures

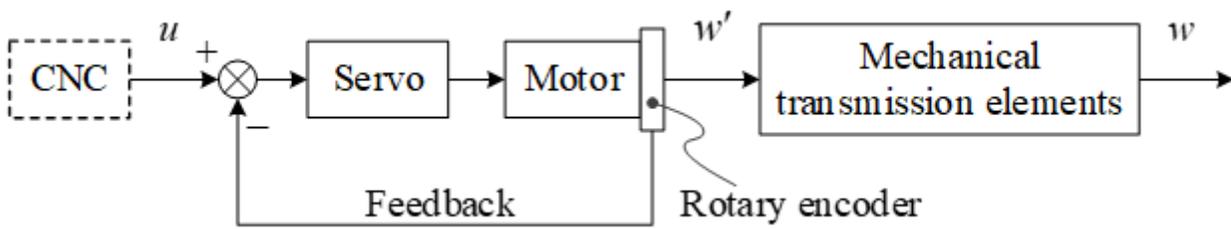


Figure 1

The general structure of the semi-closed loop control of a feeding axis

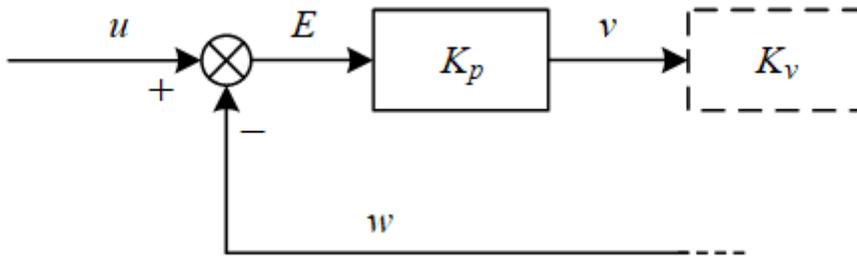


Figure 2

The position control loop of the feeding system

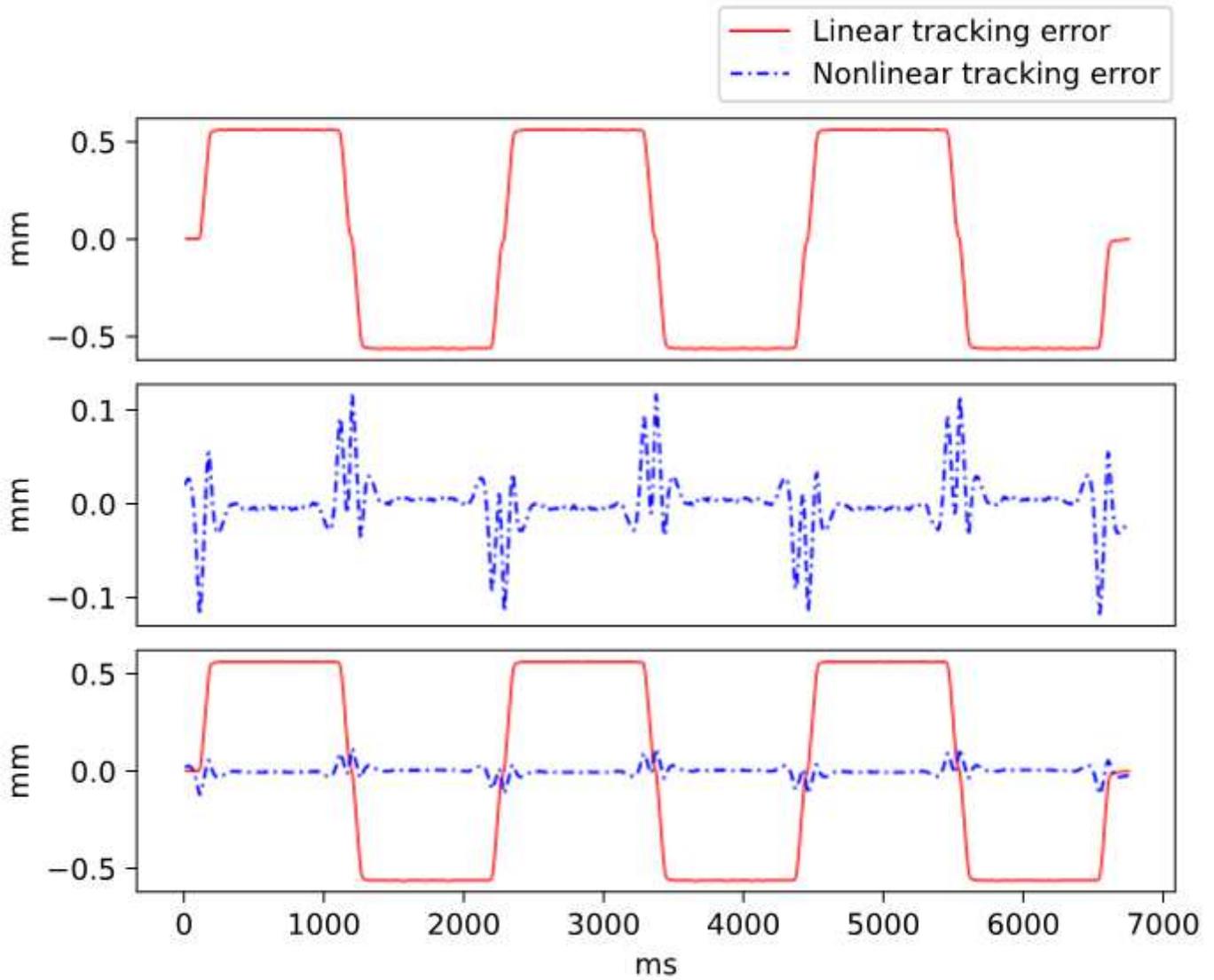


Figure 3

LTE and NLTE of the X-axis of the machine tool of Fig. 13

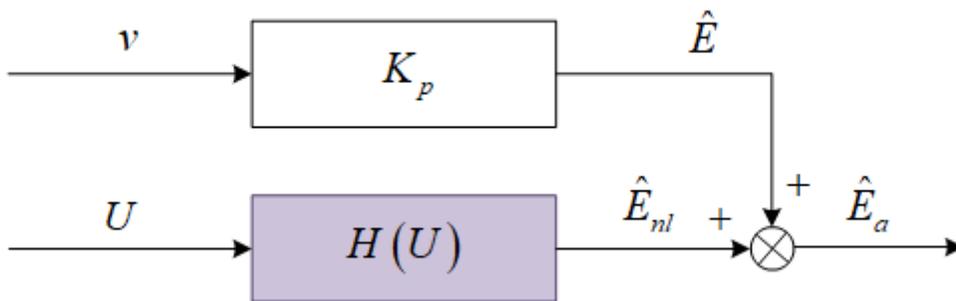


Figure 4

Prediction model of the OTE

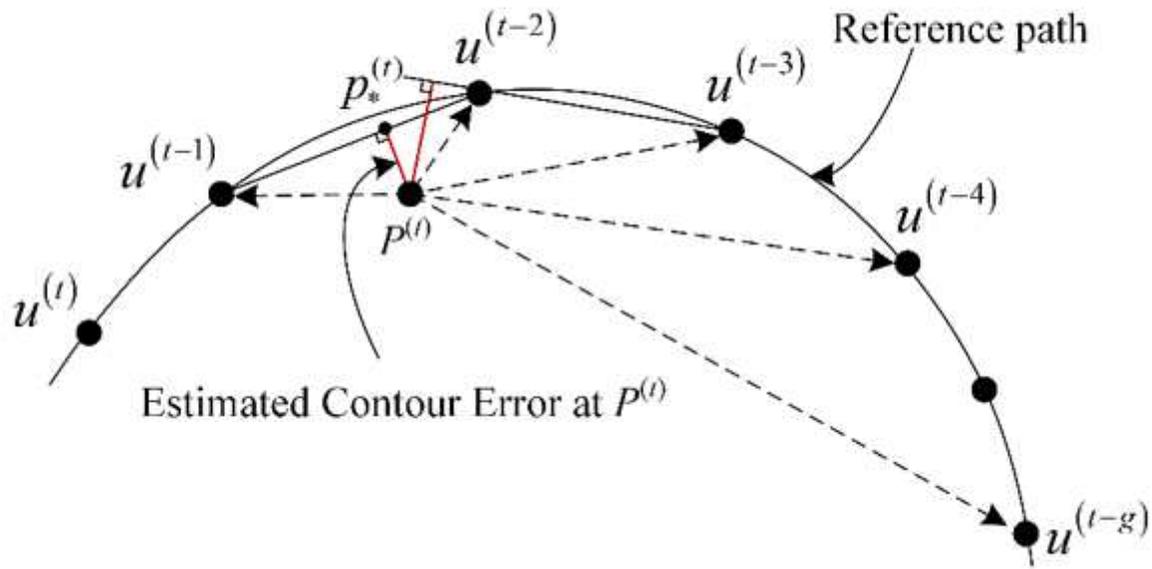


Figure 7

Contour error calculation of for a free-form reference path

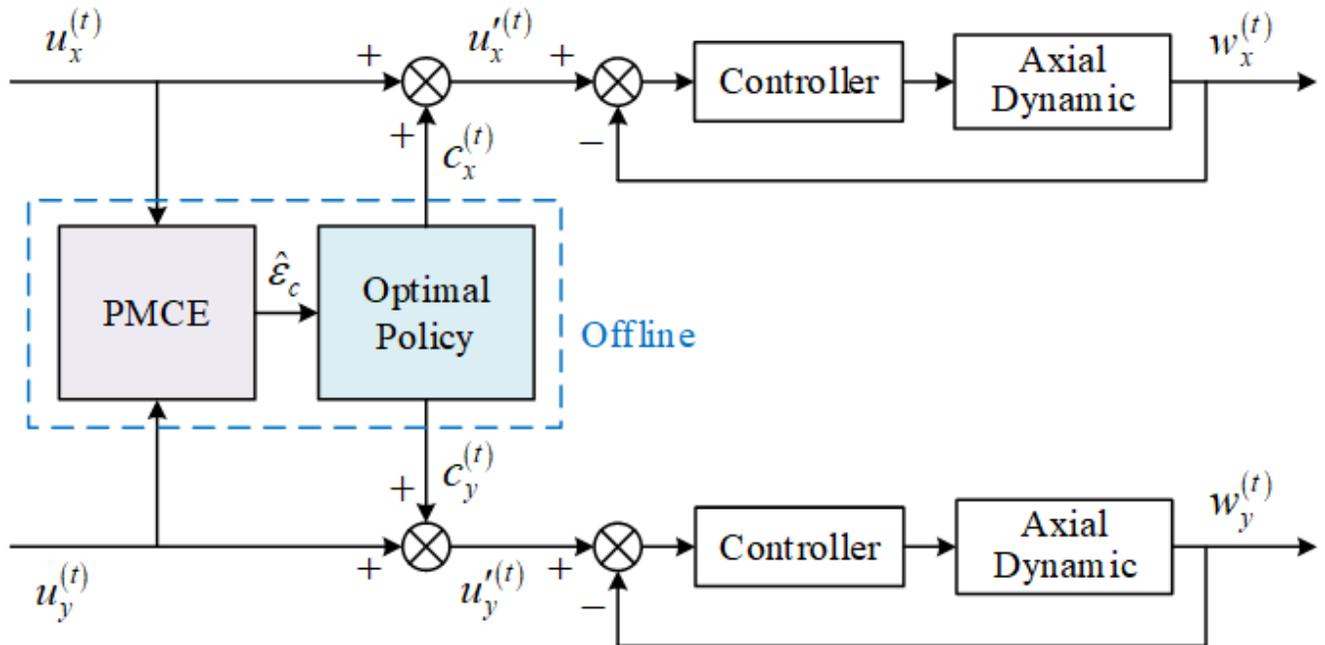


Figure 8

The general idea of the contour error compensation method

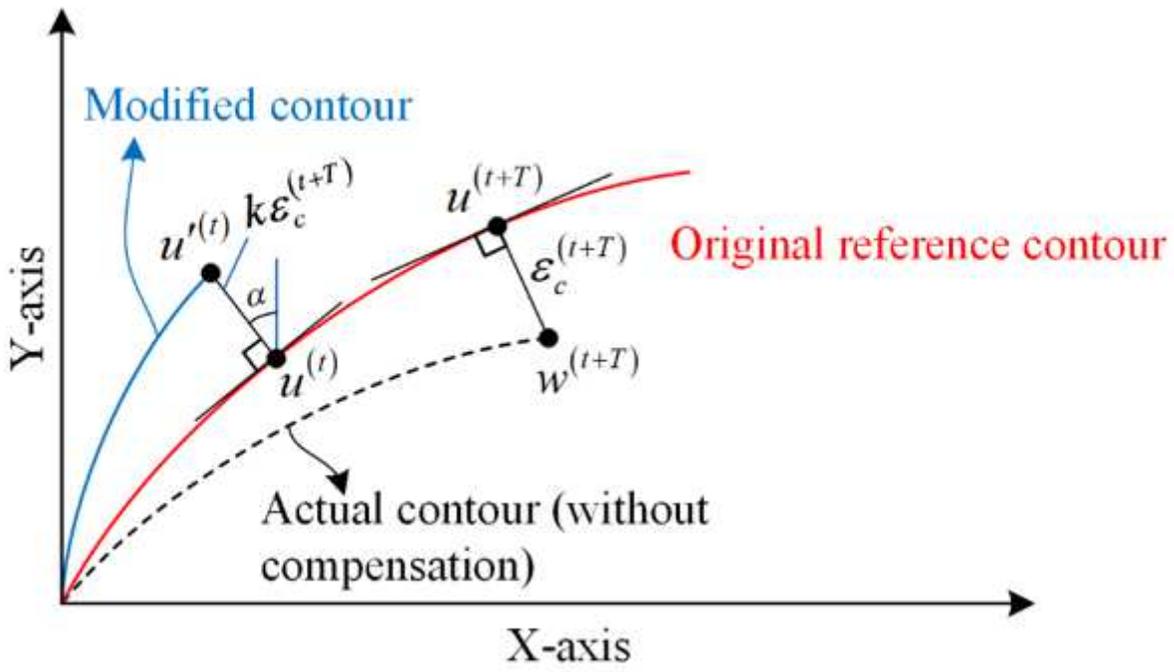


Figure 9

Contour error compensation of the reference contour

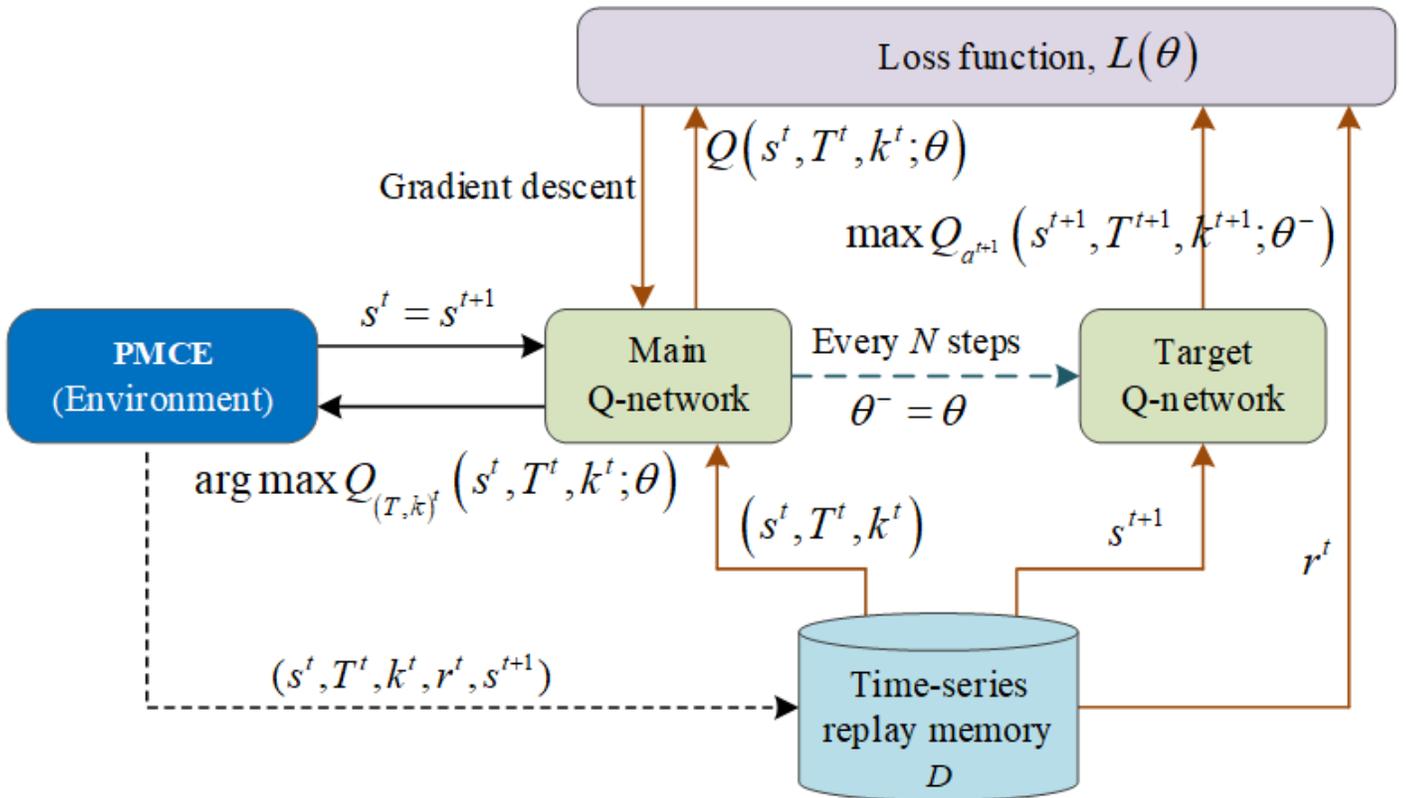


Figure 10

The general structure of the proposed TS-DQN

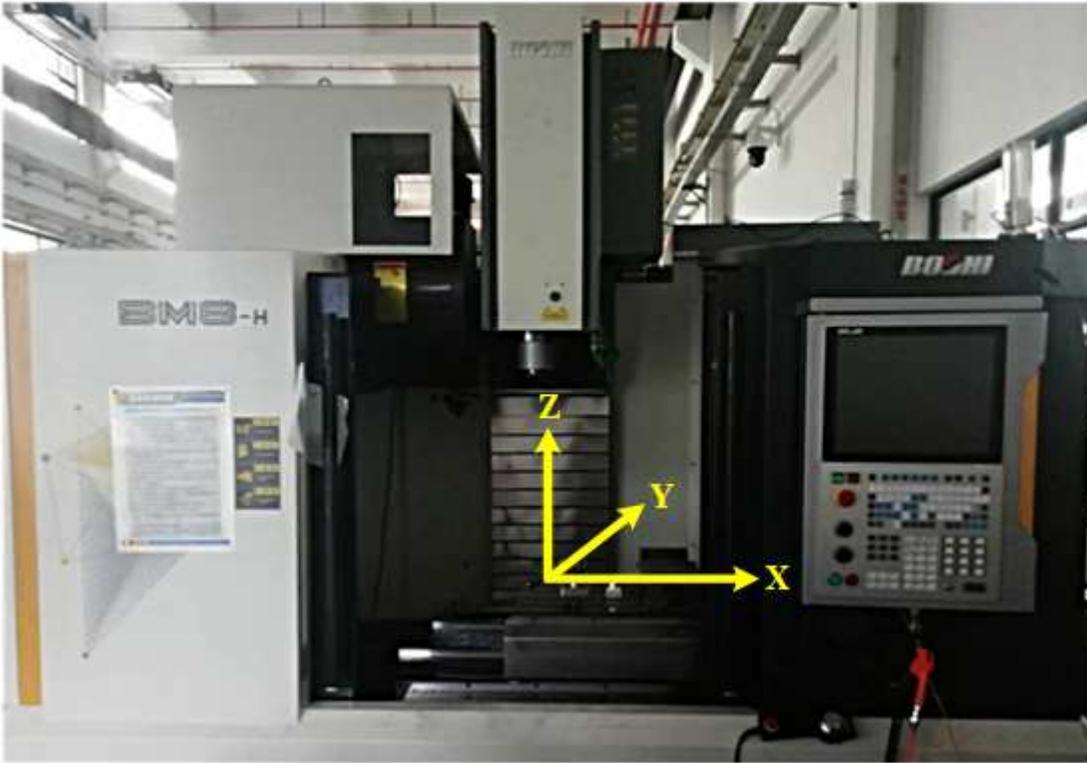


Figure 11

The BM8-H three-axis CNC milling center

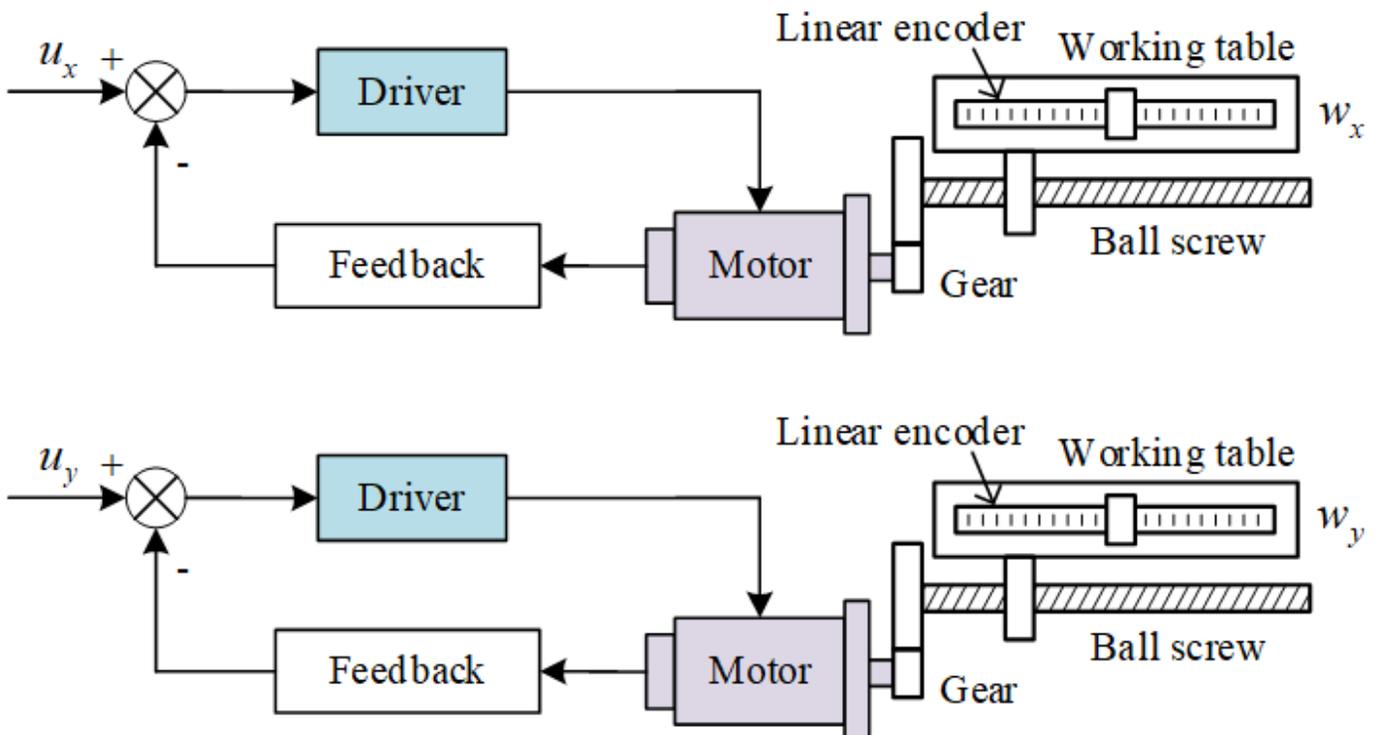


Figure 12

Data retrieved for the X and Y-axis from the controller

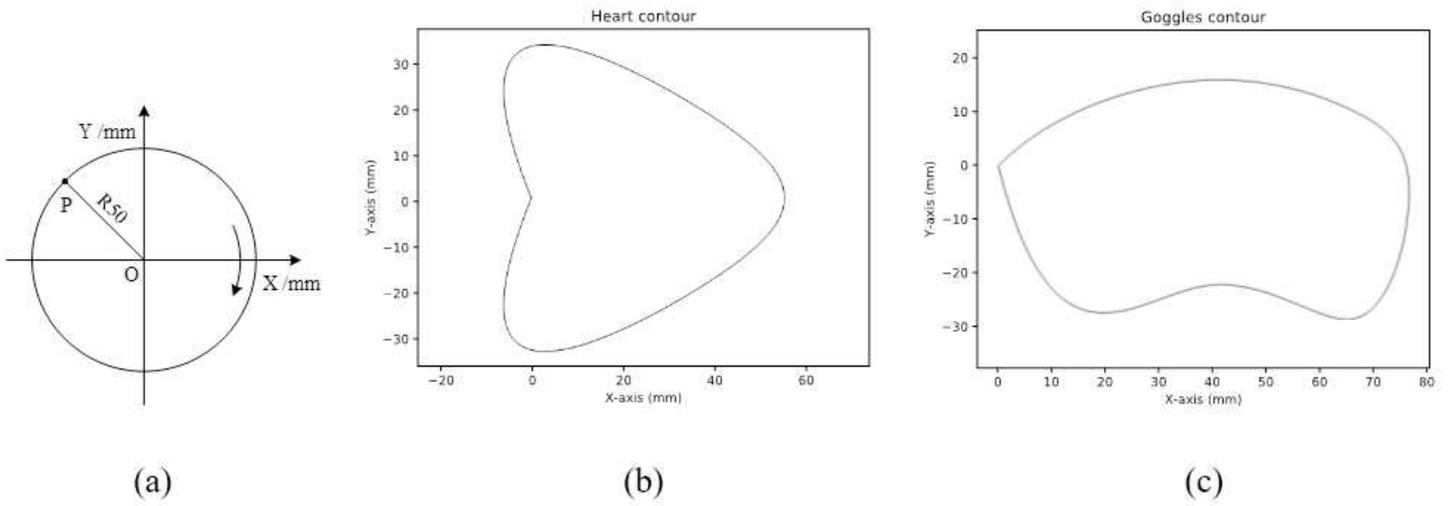


Figure 13

Three test contours. (a) circular contour; (b) heart contour; (c) goggles contour.

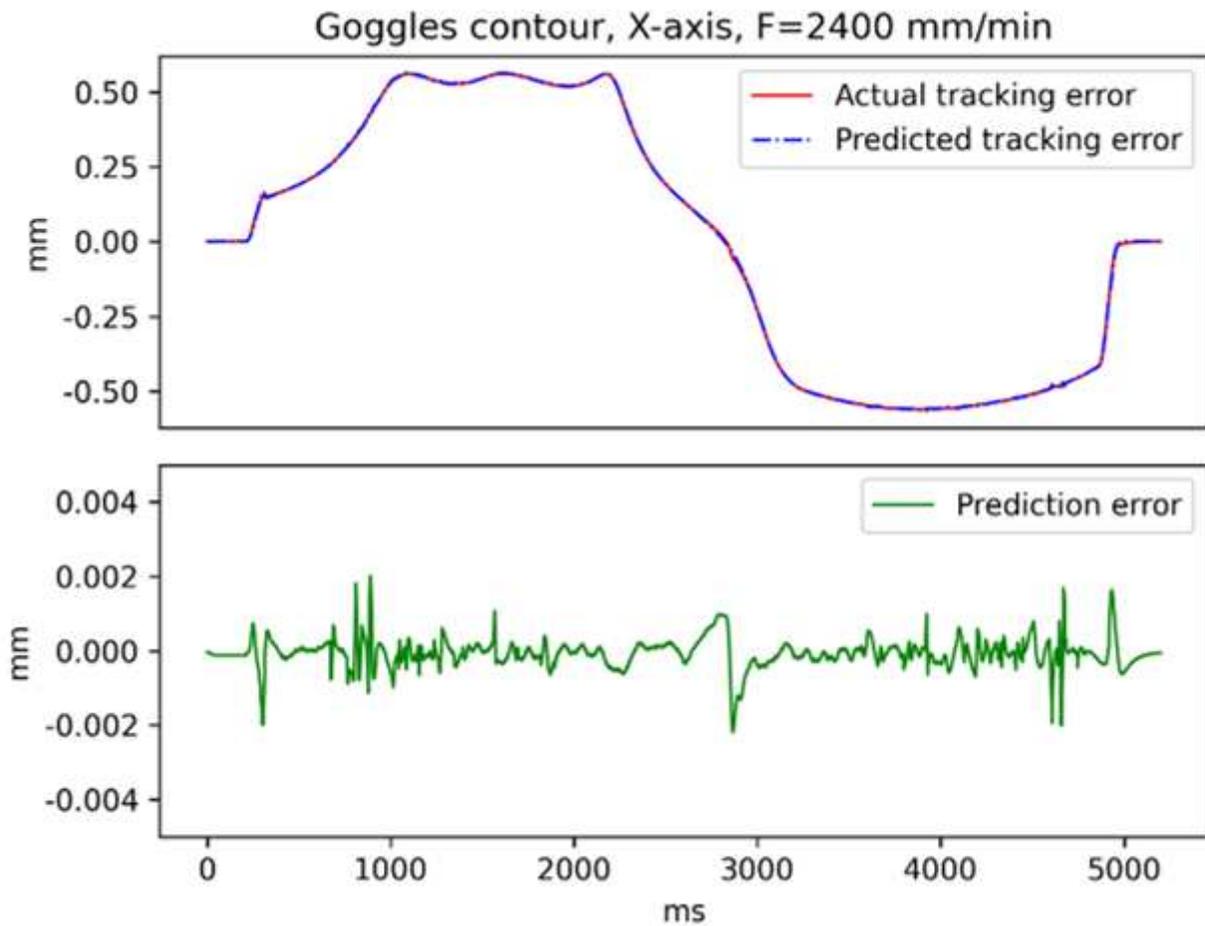


Figure 14

Actual (red) and predicted (blue) tracking error, prediction error (green) of the X-axis

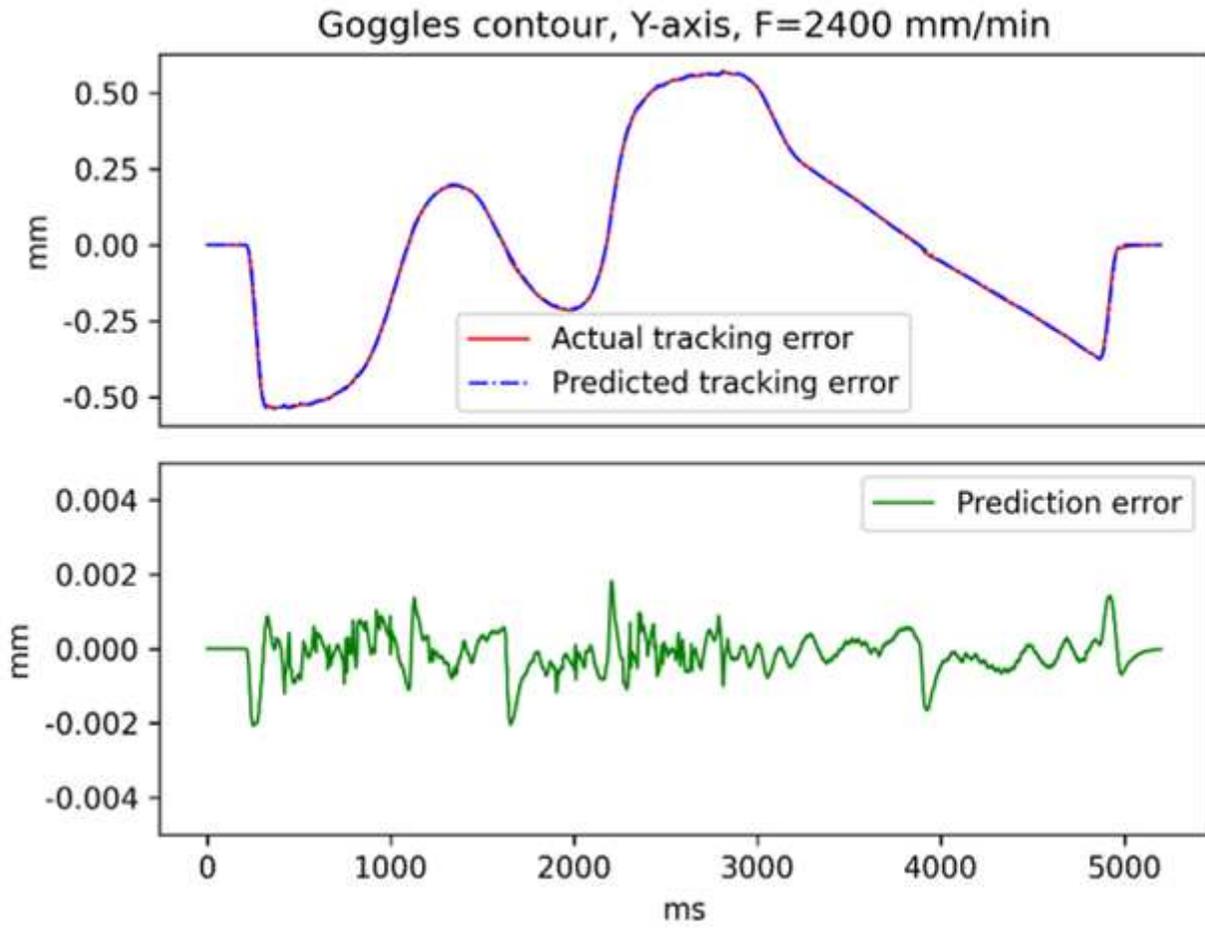


Figure 15

Actual (red) and predicted (blue) tracking error, prediction error (green) of the Y-axis

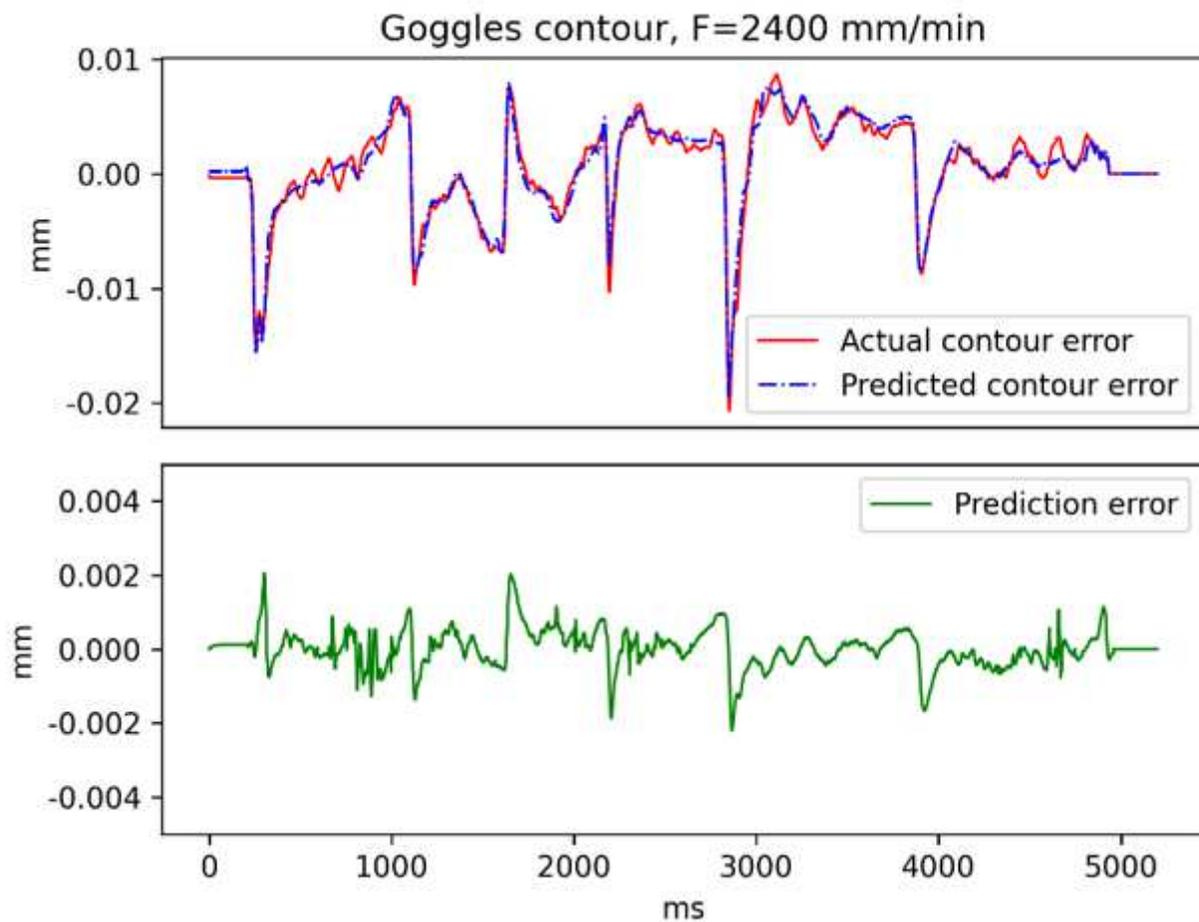


Figure 16

Actual (red) contour error, predicted (blue) contour error, and prediction error (green) for the goggles contour

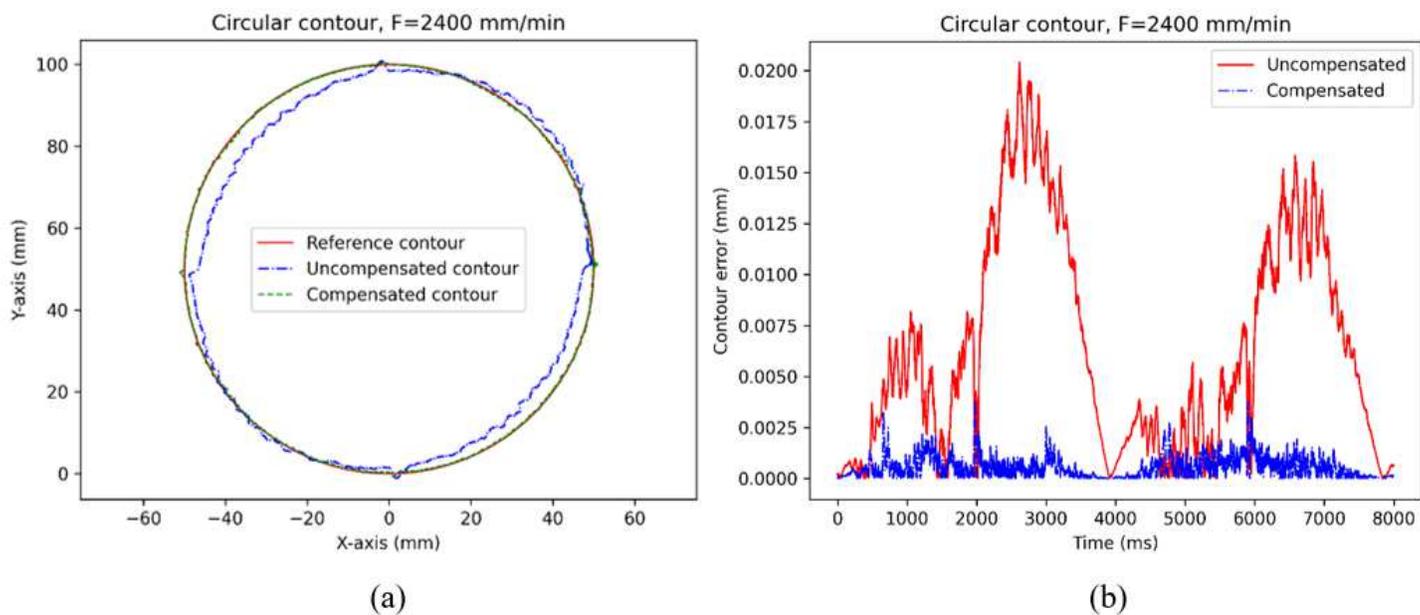


Figure 17

Contour error compensation for circular contour. (a) reference, uncompensated, and compensated circle; (b) contour error before and after compensation.

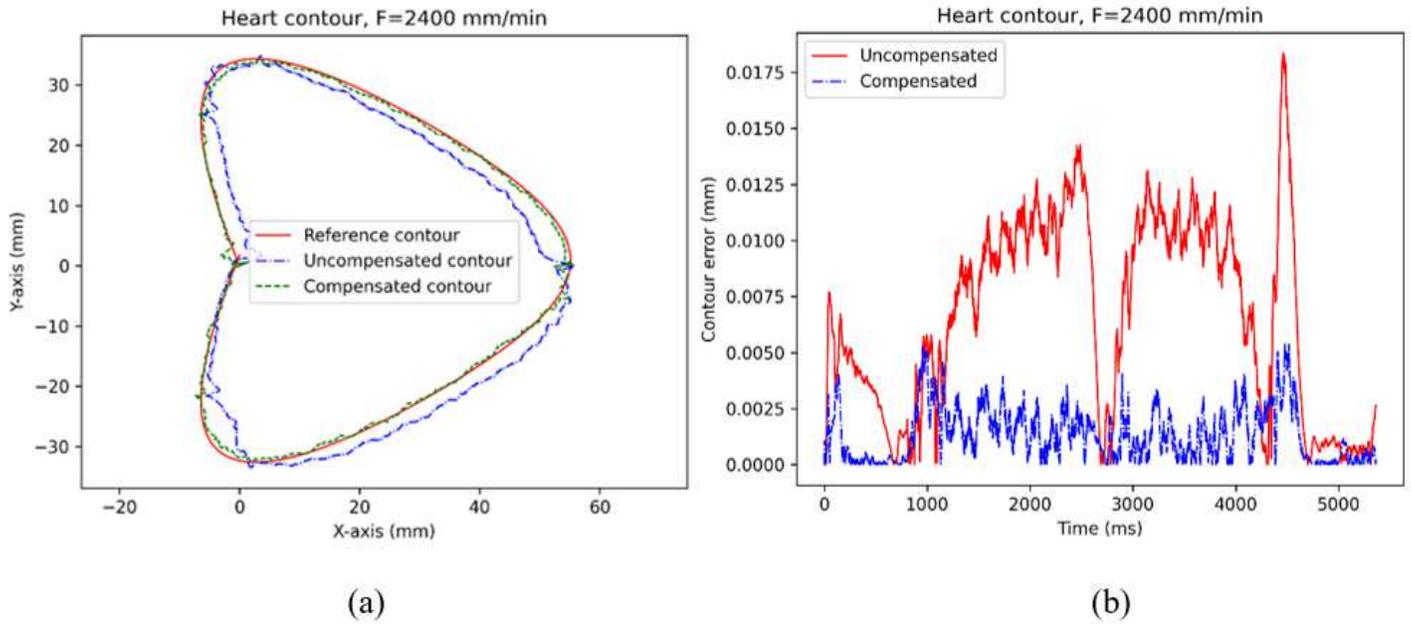


Figure 18

Contour error compensation for heart contour. (a) reference, uncompensated, and compensated heart contour; (b) contour error before and after compensation.

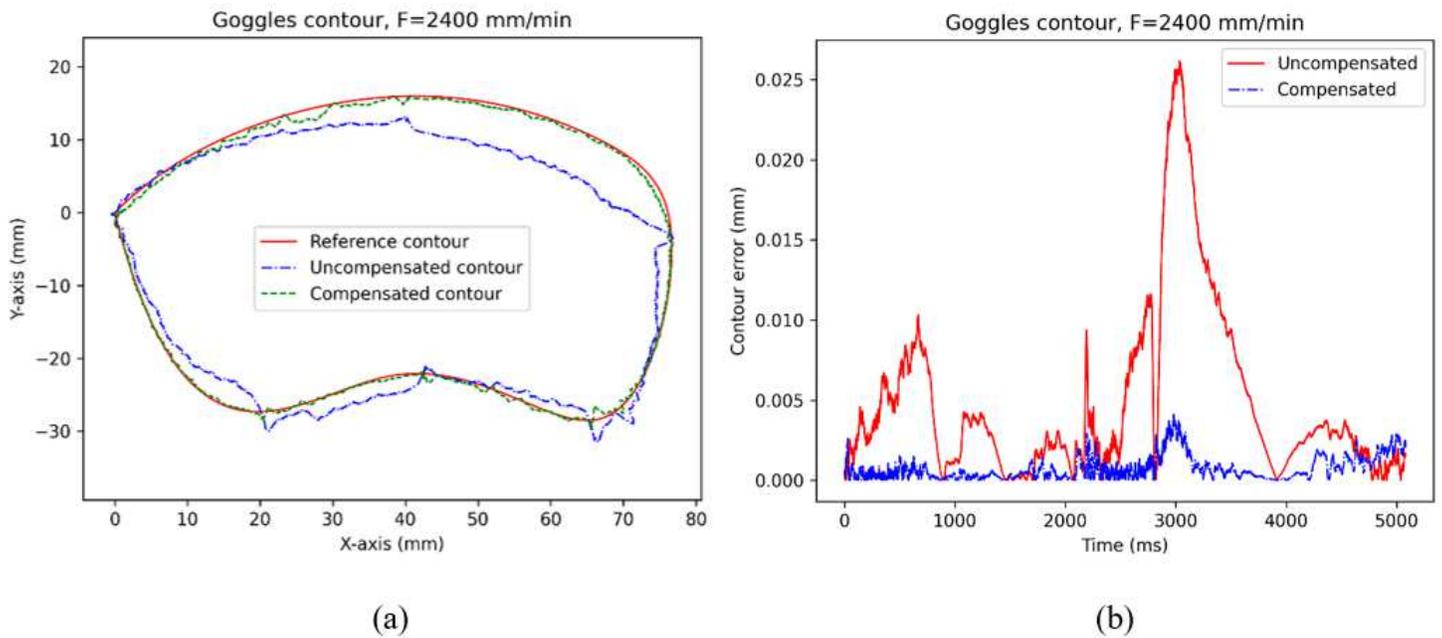


Figure 19

Contour error compensation for goggles contour. (a) reference, uncompensated, and compensated goggles contour; (b) contour error before and after compensation.

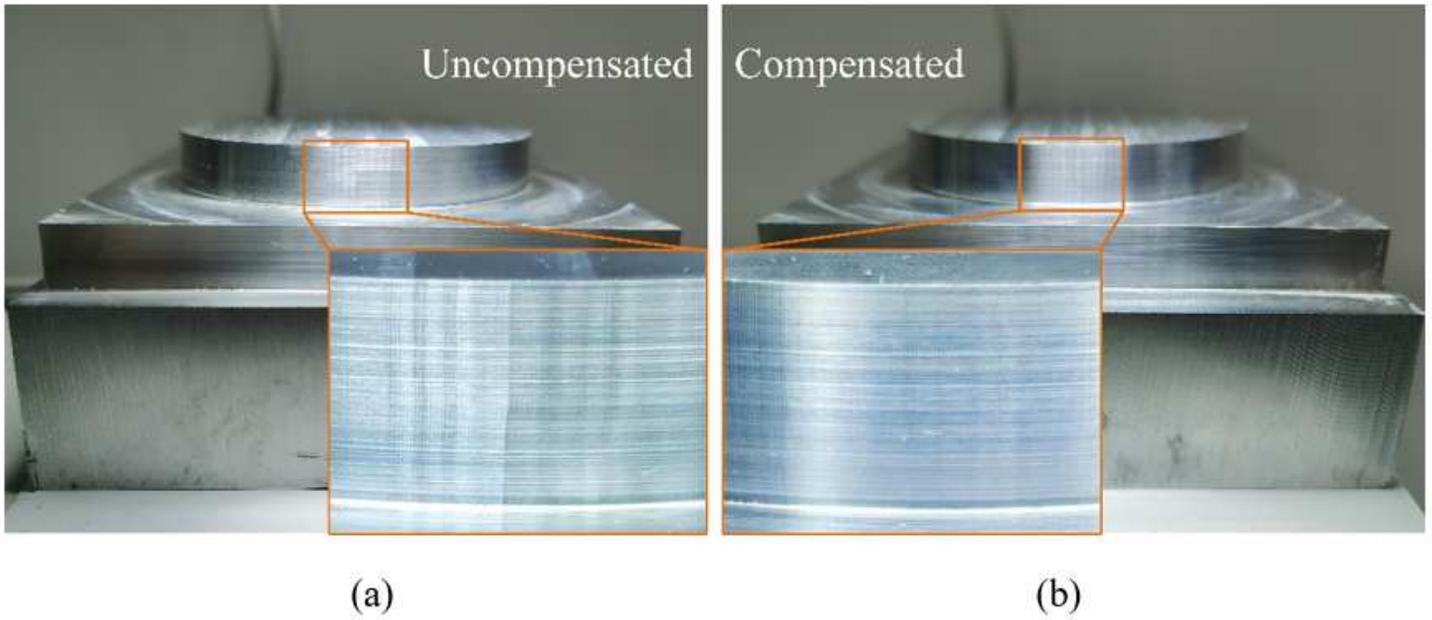
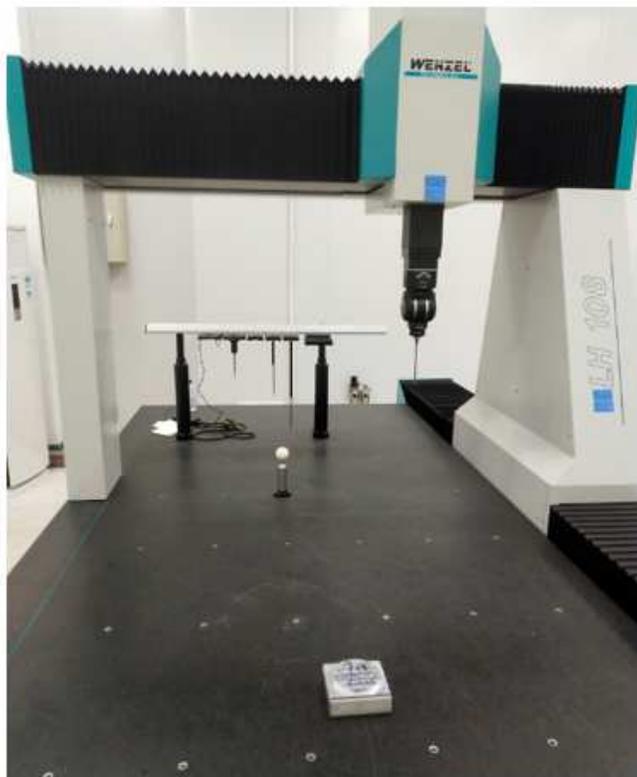
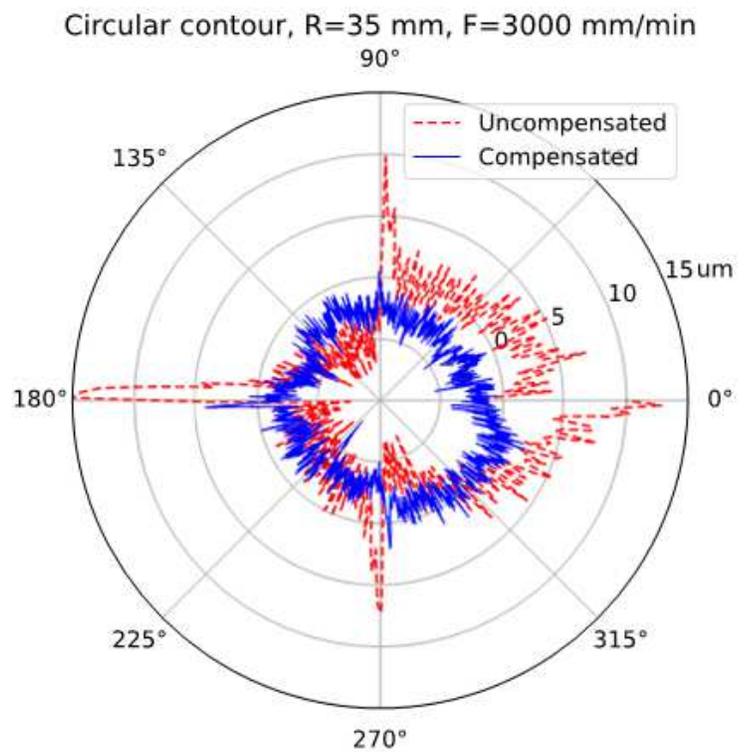


Figure 20

Machined parts of circular contour. (a) before compensation; (b) after compensation



(a)



(b)

Figure 21

(a) Inspection scenario of the machined parts; (b) machining error before and after compensation

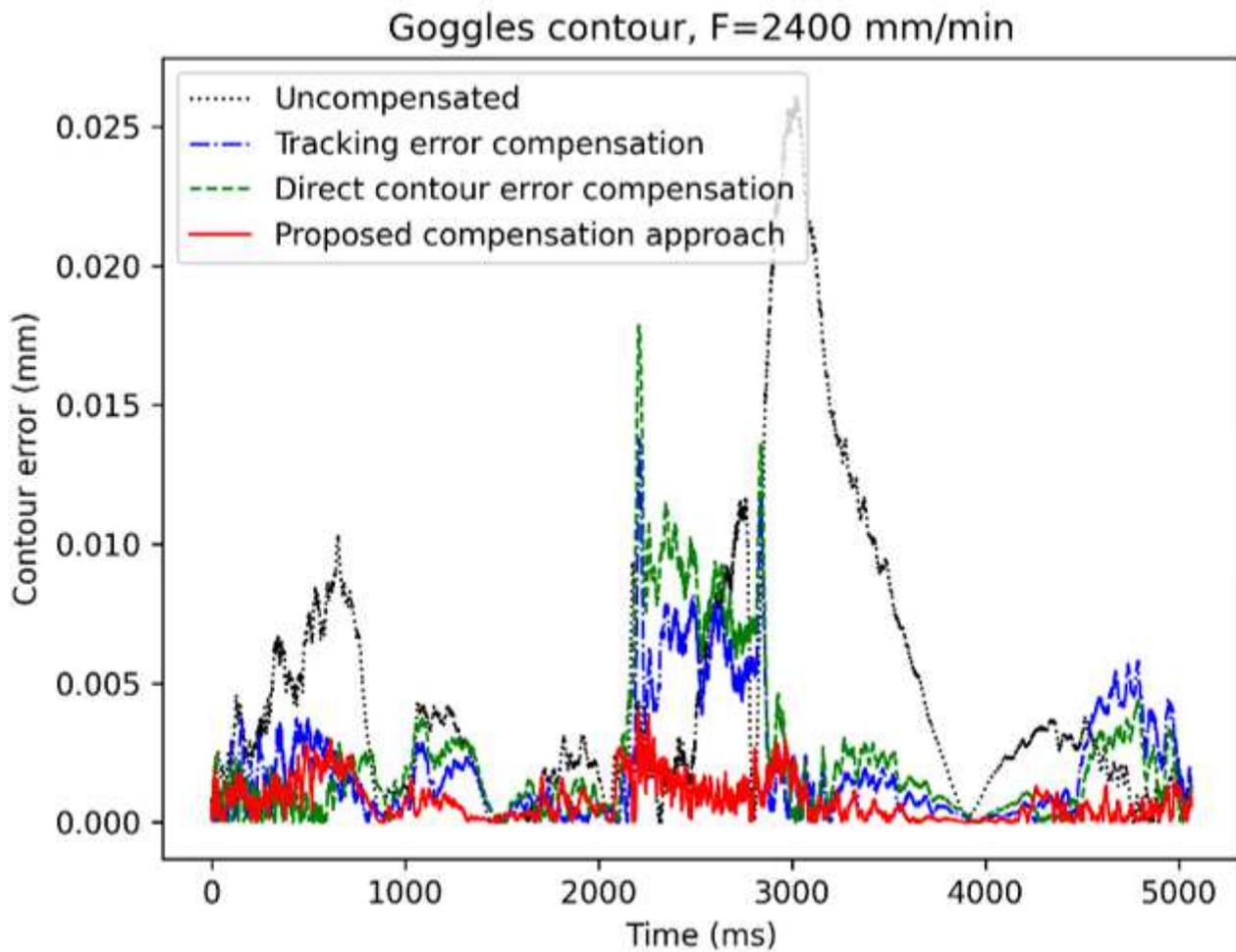


Figure 22

Comparison between the proposed approach and the benchmarks on the contour error compensation of goggles contour