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

Yakun Jiang 1 · Jihong Chen 1 · Huicheng Zhou 1 · Jianzhong Yang 1 · Pengcheng Hu 1 · Junxiang Wang 1

Received: 7 April 2021 / Accepted: 19 August 2021 / Published online: 7 September 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

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 60–85% 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 the machining precision 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 [2–4] of the tool path. To improve the machining accuracy, the contour error should be effectively predicted and compensated [1, 5].

There have been extensive works on contour error prediction and compensation. In general, they are classified into two types: online and offline approaches [6, 7].

As an online method for the contour error estimation and compensation, the concept of cross-coupling control (CCC) was introduced by Koren in [8], 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 [8], an improved concept of CCC is proposed, for which a more effective control law is utilized to build the model of contour error [9]. In the following work of [8], Koren and Lo [10] proposed a variable gain CCC method to estimate the contour error more effectively. For the contour of arbitrary shape, Yeh and Hsu [11] 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 [12], cross-coupled fuzzy logic sliding mode...
control [13], cross-coupled dynamic friction control [14], and the neural-network CCC [15].

Different from CCC algorithms, Huo et al. [16] 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. [17] 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 [18] 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. [19] 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. [20] 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 [21]. Huo and Poo [22] 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. [23] 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. [24] 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 [24]. Also, the predicted value of the model cannot truly reflect the actual contour error, especially when the error generated by the mechanical transmission elements [25, 26] 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). The proposed approach is mainly designed for finishing cutting of CNC machining, for which it has small material removal volume and some physical issues such as the cutting force and tool deflection could not be considered. There are two major contributions in this paper:

1. Based on the internal data of the machine tool, a deep learning network called NAR-LSTM is designed that can precisely predict the tracking error, based on which the contour error can be calculated. For each feeding axis, the nonlinear factor of the tracking error is estimated from the NAR-LSTM network; combined with the linear part as calculated from the steady state of the feeding axis, the entire tracking error of the feeding axis can be precisely predicted. After that, the contour error is calculated based on the predicted tracking error of two feeding axes.

2. To address the lag characteristic as well as find the best compensation strategy, a reinforcement learning method called TS-DQN is proposed that determines the optimal parameters of the compensation strategy, from which the contour error can be largely reduced and the surface machining quality can be improved significantly.
Given the proposed approaches, the contour error for machining a part, regardless if it is a new part or an already machined one, can be precisely predicted and compensated without needing any time-consuming iterative calculation for the compensation value and parameters.

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

### 2 The NAR-LSTM network–based contour error calculation

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

#### 2.1 Tracking error of a feeding axis

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

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 [27]; 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

$$E = u - w$$

(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 [28],

$$v = K_p \cdot E$$

(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.

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

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

(3)

For $E$, 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, and it should be precisely estimated before the contour error can be compensated.
For the X-axis of the machine tool (Fig. 11), an example of the LTE and NLTE is shown in Fig. 3. Considering both the LTE and the NLTE, the overall tracking error (OTE) of a feeding system is the combination of them,

\[ E_a = E + E_{nl} \]  

(4)

where \( E_a \) is the OTE and \( E_{nl} \) is the NLTE.

Based on Eq. (4), building the prediction model for the OTE resorts to the modeling of the LTE and NLTE, as shown in Fig. 4. Since the LTE can be calculated from Eq. (3), the major challenge for modeling the OTE lies in the accurate modeling of the NLTE. In this paper, a big data-based model \( H(U) \) is utilized to predict the NLTE, in which the input \( U \) is the historical data of the CNC machining process. Details regarding the \( U \) and \( H(U) \) will be explained in Section 2.2.

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

### 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 LSTM has direct advantages for modeling of time-series response of the feeding axis. In our early work [29], 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.

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 $\tilde{E}_{nl}^{(t)}$ of $t$th instant, while for the closed-loop architecture the previously predicted result $\tilde{E}_{nl}^{(r-1)}$ is taken as the input for predicting $\tilde{E}_{nl}^{(r-1)}$. In this work, the mathematical expression of these two architectures is given by Eq. (5) and Eq. (6), respectively.

$$
\tilde{E}_{nl}^{(t)} = H\left(E_{nl}^{(t-1)}, E_{nl}^{(t-2)}, \ldots, E_{nl}^{(t-g+1)}; U^{(t)}, U^{(t-1)}, \ldots, U^{(t-g)}\right)
$$

(5)

$$
\tilde{E}_{nl}^{(t)} = H\left(\tilde{E}_{nl}^{(t-1)}, \tilde{E}_{nl}^{(t-2)}, \ldots, \tilde{E}_{nl}^{(t-g+1)}; U^{(t)}, U^{(t-1)}, \ldots, U^{(t-g)}\right)
$$

(6)

where in Eq. (5) and Eq. (6), $H$ is the nonlinear function for modeling the NLTE; $U^{(t)}$ is the input vector of the $t$th instant, and $v^{(t)}$, $a^{(t)}$, and $J^{(t)}$ are respectively the reference velocity, acceleration, and jerk as retrieved from the CNC system; $g$ is as a hyperparameter of the network, meaning that $\tilde{E}_{nl}^{(t)}$ is calculated based on the $g$-period of the historical data, and $g$ is set by trial and error method according to experience.

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

$$
E_{nl}^{(t)} = E_{a}^{(t)} - E^{(t)}
$$

(7)

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

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

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

### 2.3 Contour error calculation

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

\[
\hat{w}^{(t)} = u^{(t)} + \hat{E}_a^{(t)}
\]

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 corresponding predicted actual position of the \( u^{(t)} \).

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

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
- \( \hat{E}_{a,x}, \hat{E}_{a,y} \) — the predicted OTE of the feeding axis
- \( \hat{w}_x^{(t)}, \hat{w}_y^{(t)} \) — the predicted position of the feeding axis
- \( \hat{E}_c \) — the predicted contour errors of the feeding system with considering both \( X \)- and \( Y \)-axes

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 \( Y \)-axes at the \( t \)th instant. Taking the \( X \)-axis as the example: make \( \tau g \) (\( g \) is the specific period of the historical data for modeling the NAR-LSTM); given the data of the first \( g \) instances, estimated the NLTE \( \hat{E}_{nl,x}^{(t)} \) with the model of Eq. (6); calculate the OTE \( \hat{E}_{n,x}^{(t)} \) 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 \( P^{(t)} = \left( \hat{w}_x^{(t)}, \hat{w}_y^{(t)} \right) \) 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 \( \{ \ldots, P^{(t)}, P^{(t+1)}, \ldots \} \).

**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 all the line segments of \( \{u^{(t-g)} - u^{(t-g+1)} \), \( u^{(t-g)} - g + 1)u^{(t-g+2)} \), \( \ldots, u^{(t-1)}u^{(t)} \} \), where the \( u^{(t)}(i \in [1, g]) \) is the point on the reference path corresponding to the \( t \)th instant; the contour error is then selected as the minimal value among these distances, e.g., the distance between \( p_x^{(t)} \) and \( P^{(t)} \) as shown in Fig. 7. The contour error of each predicted point is calculated which altogether forms the predicted contour error \( \hat{E}_c \) of the reference path, as shown in Fig. 6. The detailed process for calculating the contour error could refer to the work [16].

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

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 nth sampling instant:

\[ u_x^{(t)}, u_y^{(t)} \text{— the reference position from the given part program} \]
\[ \tilde{e}_c \text{— the contour error calculated from the PMCE of Section 2.3} \]
\[ u_x^{(t)}, u_y^{(t)} \text{— the compensated position of the corresponding reference path} \]
\[ w_x^{(t)}, w_y^{(t)} \text{— the actual position of the feeding axis} \]
\[ c_x^{(t)}, c_y^{(t)} \text{— the compensation values of } X- \text{ and } Y- \text{axes} \]

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

To achieve good compensation performance, we designed two parameters in the proposed compensation model, i.e., the forward compensation cycle \( T \) and the compensation rate \( k \), which are respectively defined to avoid the lag characteristic of the control system and mechanical system of the feeding axis and ensure good compensation performance. Given a reference position \( u^{(t)} \) at the nth instance, the corresponding compensated position \( u^{(t)} \) as calculated from the compensation policy can be regarded as a function of the parameters \( T \) and \( k \), and

\[ u^{(t)} = \Phi(u^{(t)}, k, \tilde{e}_c^{(t+T)}) \] \hspace{1cm} (9)

In Eq. (9), \( u^{(t)} \) at the nth instance is compensated based on the future contour error \( \tilde{e}_c^{(t+T)} \) at the \((t+T)\)th constant; in this way, the lag characteristic is counterbalanced by the parameter \( T \) at the nth instance, as shown in Fig. 9.

In the proposed compensation strategy, the compensation values for the \( X- \) and \( Y- \) axes are denoted as \( c_x^{(t)} \) and \( c_y^{(t)} \), respectively, and

\[ \begin{align*}
  c_x^{(t)} &= k \tilde{e}_c^{(t+T)} \sin(\alpha) \\
  c_y^{(t)} &= k \tilde{e}_c^{(t+T)} \cos(\alpha)
\end{align*} \] \hspace{1cm} (10)

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

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

\[ \begin{align*}
  u_x^{(t)} &= u'_x + c_x^{(t)} \\
  u_y^{(t)} &= u'_y + c_y^{(t)}
\end{align*} \] \hspace{1cm} (11)

The compensated position as calculated from Eq. (11) is then sent to the CNC system to generate the modified contour, which is formed by the compensated positions and close to the original reference contour.

For the proposed compensation strategy, one issue remains how the two compensation parameters (i.e., the \( T \) and \( k \) of Eq. (10)) can be determined so that a good compensation result can be achieved. In Section 3.2, a reinforcement learning-based approach is applied to determine the two parameters.

3.2 TS-DQN-based identification of compensation parameters

As a popular and dominant reinforcement learning method, the Deep Q-network (DQN) learns by interacting with the environment to obtain the optimal policy under a set reward function, and it has been widely used in the domain of optimal control [30, 31]. Considering the time-series feature of the feeding system, a modified DQN, named Time-Series Deep Q-network (TS-DQN), is presented for identifying the compensation parameters. The general framework of the TS-DQN is shown in Fig. 10, in which a time-series replay memory is used to replace the traditional replay memory of the classical DQN [30]. For the replay memory in the proposed TS-DQN, the rewards in historical steps can be re-updated according to the time-series state of the feeding system, so as to achieve higher parameter identification efficiency.

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

The three core components of the TS-DQN algorithm are the prediction performance states’ (superscript “‘t’” represents the data at the nth instance), decision action of compensation parameter \( (T', k') \), and the reward for corresponding decision action \( r' \) 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.

By selecting and executing a decision action according to the $\varepsilon$-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 a specific state.

$$Q(s', T', k') \leftarrow Q(s', T', k') + \alpha [r' + \gamma \max_{a'} Q(s'+1, T'+1, k'+1) - Q(s', T', k')]$$  \hspace{1cm} (12)

where $\alpha$ is the learning rate and $\gamma$ is the discount factor.

The data set $(s', T', k', r')$ 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 $r^t$ in the last step will be re-updated in the time-series replay memory by Eq. (13),

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

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

$$r^{t-n} = r^{t-n} + p \cdot (r^{t-n+1} + \ldots + r^{t-1} + r^t), n = 1, 2, \ldots , m$$  \hspace{1cm} (14)
where in Eq. (13) and Eq. (14), \( p \) is the time-series correlation factor, and \( m \) is the number of repeated decision actions that are continuingly the same.

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

\[
L(\theta) = \mathbb{E}[(\text{Target}Q - Q(s', T', k'; \theta))^2]
\]

and

\[
\text{Target}Q = r^t + \gamma \max_{s^{t+1}, k^{t+1}} Q(s^{t+1}, T^{t+1}, k^{t+1}; \theta)
\]

where \( \theta \) and \( \theta^- \) are the parameters of the main Q-network and the target Q-network, respectively.

The reward value function corresponding to the specific state and action is given in Eq. (16). By estimating the compensation performance through the prediction model of contour error, the reward \( r' \) at the \( t \)th instant can be given as

\[
r' = c \cdot (\tau_E - \varepsilon_E^t)^3 + 1
\]

and

\[
\varepsilon_E = E_{\text{MSE}}^t \cdot 0.3 + \text{abs}(E')_{\text{max}} \cdot 0.7
\]

In Eq. (16), \( E_{\text{MSE}}^t \) and \( \text{abs}(E')_{\text{max}} \) are the mean square error (MSE) and maximal contour error for the \( t \)th instant, respectively; \( \varepsilon_E^t \) is the weighted sum of \( E_{\text{MSE}}^t \) and \( \text{abs}(E')_{\text{max}} \), where the \( E_{\text{MSE}}^t \) accounts for a small weigh of 0.3 and the \( \text{abs}(E')_{\text{max}} \) has a dominant weight of 0.7 because the \( \text{abs}(E')_{\text{max}} \) normally has more effect on the final machining result; \( \tau_E = 2 \mu m \) is the criteria of termination of the current iteration because the best-predicted value is 2 \( \mu m \) for our contour error prediction model toward our machine tool; \( c \) is a constant of 1000.

For the proposed TS-DQN, the learning rate \( \alpha \), the discount factor \( \gamma \) of Eq. (12), the time-series correlation factor \( p \) of Eq. (13), and the constant \( c \) of Eq. (16) are all hyperparameters set by trial and error method according to experience. For the machine learning–based method, the manual setting of hyperparameters is essentially a drawback, and the criterion for selecting these parameters could refer to [30, 32]. Regarding the compensation set \((T, k)\), they are the parameters of the proposed TS-DQN, which can be obtained through the training of the TS-DQN.

For the proposed TS-DQN as shown in Fig. 10, an algorithm called TS-DQN with Experience Replay is given below for a clear explanation. As already alluded, a time-series replay memory of the proposed TS-DQN is used to replace the traditional replay memory of DQN, and the re-updated method of rewards is added in the time-series replay memory by setting \( r^{t-1} = r^{t-1} + p \cdot (r' - r^{t-1}) \) when \( d^{t-1} = d' \) and \( s^{t-1} > s' \).
Algorithm TS-DQN with Experience Replay

Initialize time-series replay memory \( D \)

Initialize action-value function \( Q \) with random parameters \( \theta \)

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

for episode = 1, \( M \) do

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

for \( t = 1, T \) do

With probability \( \varepsilon \) select a random compensation parameter decision action \( (T^i, k^i) \)

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

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

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

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

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

Set \( y^j = \begin{cases} 
   r^j & \text{if episode terminates at step } j+1 \\
   r^j + \max_{T,k} \hat{Q}(\phi^{j+1}, T^j, k^j; \theta^-) & \text{otherwise}
\end{cases} \)

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

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

Every \( N \) steps reset \( \hat{Q} = Q \)

end for

end for
TS-DQN with Experience Replay, the two compensation parameters (i.e., the $T$ and $k$ of Eq. (10)) at each instance of the machining process are identified which minimizing the contour error. Results on the identification of the two parameters and the follow-up compensation will be presented in Section 4.

4 Experiments

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

4.1 Experimental setup

The experimental verifications of the proposed contour error prediction and compensation approaches were conducted toward the BM8-H three-axis CNC milling center equipped with a Huazhong HNC-818B controller, as shown in Fig. 11. In addition, the structure of the three-axis CNC milling center is given in Fig. 12. For the machine tool, the 2D contour error resulting from tracking error (both linear and nonlinear)
of $X$- and $Y$-axes is calculated and compensated, where the reference positions $u_x$ and $u_y$ are generated and retrieved from the CNC system; and the actual positions of the feeding axis $w_x$ and $w_y$ are detected by the linear encoder mounting on the working table (shown in Fig. 13); both data are collected via the data acquisition function of the CNC system with the sampling frequency of 1 kHz. Furthermore, the HSV-180UD-075 servo controller is utilized and set as the position control mode. To show the ability of our model for predicting the results of the unmatched axial, the proportional gains for the two axes were set to be different with $K_{px} = 160$ Hz and $K_{py} = 120$ Hz.

To testify the proposed approaches in terms of contour error prediction and compensation, two elementary contours (i.e., circle shown in Fig. 14a and rhombus shown in Fig. 14d) and three more freeform contours (i.e., heart contour, goggles contour, and star contour shown in Fig. 14b, c, e, respectively) are chosen as test examples. For each contour, experiments are conducted on four feedrates of 1200 mm/min, 2400 mm/min, 3600 mm/min, and 4800 mm/min. Among the five contours, the circle and heart contours are taken as the training and validation samples for the proposed NAR-LSTM network and TS-DQN, while the goggles contour serves as the test sample. Details about the five contours are listed below.
• Circular contour: a circle centered at the origin with a radius of 50 mm.
• Heart contour: a NURBS curve with order=4 and 7 control points [33], details on its geometric parameters are listed in Appendix.
• Goggles contour: a NURBS curve with order=4 and 9 control points [34], details on its geometric parameters are also listed in Appendix.
• Rhombus contour: a polygon with 4 points listed in Appendix.
• Star contour: a NURBS curve with order=4 and 11 control points, details on its geometric parameters are listed in Appendix.

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 cells and one fully connected layer were used. The number of nodes in each cell and the parameter $g$ are hyperparameters that were found by trial and error approach [35, 36], 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-axes from the approach of Section 2.1 and Section 2.2 are shown in Fig. 15 and Fig. 16, 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. 15), the prediction error, which is the deviation between the actual position and the predicted position, has the maximum value of 0.0019 mm, which is only 0.34% of the corresponding actual tracking error. For the Y-axis which has the largest actual tracking error of 0.5661 mm, the maximum prediction error is 0.0021 mm, which is only 0.37% of the actual tracking error at this location. Results shown in Fig. 15 and Fig. 16 validated that the proposed NAR-LSTM network–based tracking error modeling approach has very good predicted accuracy.

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. Figure 17 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. 17 validated that, based on the tracking error as modeled via the NAR-LSTM network, the

| 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% ↓    |

Fig. 21 Machined parts of circular contour. a Before compensation; b after compensation
contour error of 2D CNC machining can be precisely predicted.

4.2.2 Contour error compensation

Based on the approach as presented in Section 3, the experiments for contour error compensation are firstly conducted for circular contour, heart contour, and goggles contour, with the results shown in Fig. 18, Fig. 19, and Fig. 20, respectively.

The tracking error for machining the contours of Fig. 18, Fig. 19, and Fig. 20 is first predicted with the proposed NAR-LSTM of Section 2.2, and the contour error is then calculated with the model as presented in Section 2.3. After that, the compensation parameters \((T,k)\) are identified by the proposed TS-DQN. Given the predicted contour error and the optimal compensation parameters, the compensated positions of the \(X\)-axis and \(Y\)-axis are calculated from Eq. (11) and sent to the CNC system to execute the compensation. In this way, the three contours can be machined with the compensated results.

For the circular contour, the contour error has some significant reduction after compensation, with the maximum contour error reduced from 0.0204 to 0.0039 mm, an 80.88% reduction; the compensated contour is very close to the original reference contour, as shown in Fig. 18. For the heart contour, the contour error is also largely reduced, with the maximum contour error reduced from 0.0183 to 0.0054 mm, as shown in Fig. 19, which has a 70.49% reduction. For the goggles contour shown in Fig. 20, the maximum contour error is reduced from 0.0262 to 0.0048 mm, an 81.68% improvement.

Note that, to make the results obvious, the contour error shown in Fig. 18 a, Fig. 19 a, and Fig. 20 a is magnified 200 times.

Figures 18–20 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 Table 2 for the four feedrates, and

\[
MAE = \max(\epsilon(k)), \quad k = 1, 2, \ldots, N
\]

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

The experimental results in Table 2 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.

To further testify the effectiveness of the proposed contour error compensation approach on improving the machining quality, physical cutting experiments are conducted on the milling center BM8-H for machining the circular contour of a radius of 35 mm and feedrate of 3000 mm/min. The machined parts before and after compensation are shown in Fig. 22.

Table 3 Computational time of the proposed TS-DQN and the traditional DQN

| Algorithm | Computational time (s) |
|-----------|------------------------|
| DQN       | 25.51                  |
| TS-DQN    | 14.83                  |
21 a and b, respectively. Figure 21 shows that, after contour error compensation, the surface roughness can be improved drastically, especially for the machining region where the $X$- or $Y$-axes of the machine tool reverses the moving direction and the surface defects such as bump or groove could easily occur. In addition to the surface roughness, the machining precision of the two circles is inspected by the WENZEL LH108 coordinate measuring machine (Fig. 22a). Inspection results shown in Fig. 22 b convince that, based on the proposed compensation technology, the machining error can be reduced significantly with the maximum machining error reduced from 15.1 to 7.6 μm, and the roundness of the circular part is improved from 12.7 to 5.3 μm.

In our work, the compensation parameters of Section 3.1 are identified with the proposed TS-DQN. Compared with the traditional DQN [30], 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 3. 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.

To further verify the contour error compensation performance at the location of high curvatures, more experiments are conducted toward some complex trajectory, i.e., the rhombus and star curves as respectively shown in Fig. 14 d and e. For the rhombus contour, as shown in Fig. 25 a, the contour error has some significant reduction after compensation, with the maximum contour error reduced from 0.0138 to 0.0054 mm, a 60.87% reduction; the compensated contour is very close to the original reference contour, as shown in Fig. 23. For the star contour, the contour error is also largely reduced, with the maximum contour error reduced from 0.0247 to 0.0061 mm, as shown in Fig. 25 b, which has a 75.30% reduction; the compensated contour is also close to the original reference contour, as shown in Fig. 24. Experimental results toward the rhombus and star contours validate the robustness of the proposed contour error compensation method: even when the contours have some drastic changes of curvature (i.e., the shape corner), the proposed approach can still achieve good compensation results (Fig. 25).
In addition to our contour error compensation approach as proposed in Section 3, two more strategies (benchmarks) 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 of our approach and the two benchmarks on the goggles contour are presented in Fig. 26. Results show that our approach based on the optimal selection of the compensation parameters has much better compensation results compared to the two benchmarks and the original uncompensated strategy: the maximal contour error of our approach is 4.82 μm, which is significantly smaller than the two benchmarks (i.e., 13.77 μm and 17.85 μm) and the uncompensated strategy (i.e., 26.24 μm).

Theoretically, the two benchmarks are supposed to have good performance for improving the contouring accuracy. However, in the real cutting practice, the strategies of direct tracking error compensation and direct contour error compensation are difficult to work because the actual responses lag behind the reference inputs due to the lag characteristic of the control system and mechanical system of the feeding system. Therefore, the compensation values cannot work immediately through direct compensation. To achieve good compensation performance, we design two compensation parameters (i.e., the forward compensation cycle $T$ and the compensation rate $k$) and utilize the deep learning approach to determine their optimal values. In this way, the lag-characteristic feeding system can be counterbalanced and good compensation performance can be achieved. Therefore, the proposed method can achieve better performance on the contour error compensation as compared to the two benchmarks.

In this section, some comprehensive experiments for testing the proposed NAR-LSTM network–based contour error prediction and the TS-DQN-based contour error compensation approaches are designed and conducted. Experimental results show that the proposed approach can have good prediction accuracy on the contour error (prediction error less than 9.1%); and for the further contour error compensation strategy, as presented in this paper, it can significantly counterbalance the contour error of the machining process (the contour error is reduced for 60.87–84.53%) with the machining quality improved significantly after the compensation (maximal machining error reduced 49.67% for machining circular contour). Besides, the proposed TS-DQN-based contour error compensation approach has much

![Fig. 25](image1.png)  
**Fig. 25** a Contour error compensation for rhombus contour; b contour error compensation for star contour

![Fig. 26](image2.png)  
**Fig. 26** Comparison between the proposed approach and the benchmarks on the contour error compensation of goggles contour
better performance in terms of both the computational efficiency compared to the traditional approach (a 41.81% reduction of computational time) and the compensated results as compared to the benchmarks.

5 Conclusion

In this paper, a set of approaches is presented for the contour error compensation of the CNC machining. To achieve this goal, a deep learning network NAR-LSTM is designed that can precisely predict the tracking error of the machine’s feed axis from the internal data of the CNC system, based on which the contour error can be calculated. Then, a contour error compensation strategy is proposed for which the compensation parameters are identified based on the TS-DQN-based reinforcement learning approach. The validities of the contour error prediction and compensation methods were confirmed by extensive experiments on a three-axis CNC machine tool. Experimental results show that the proposed method can effectively improve the accuracy of the contour error prediction, both for the elementary shape like a circle and the freeform curves like the heart contour and goggles contour. Also, experimental results validate the effectiveness and the advantage of the proposed contour error compensation approach in terms of improving the machining quality and enhance the computational performance.

Regarding the method proposed in this paper, it has two major limitations. One is that our current method can only be used in a thermal steady state of the machine tool because the effective training samples can only acquire in that state. The other one is that there are many hyperparameters (e.g., learning rate α, discount factor γ, time-series correlation factor ρ) in the proposed networks that should be determined in a trial and error way, which is complicated and time-consuming. Regarding the future work of this paper, in addition to the 2D contour, the proposed deep learning–based tracking error modeling and reinforcement learning–based contour error compensation approaches could be extended for improving the machining quality of the 3D surface machining. Besides, we will focus on how to reuse the generated compensation values for different feedrate via the technology of transfer learning. At last, based on the control error prediction result, some feedforward control methods will be developed to further increase the contouring accuracy of CNC machining.

Appendix

Geometric parameters of four contours

Heart contour: NURBS curve with order \( k = 4 \), knot vector: \( \{0, 0, 0, 0.15, 0.3, 0.45, 0.6, 0.75, 0.9, 1, 1\} \); control points \((x, y)\): \{(0.0, 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, 1.0, 1.0, 1.0, 1.0\}.

Goggles contour: NURBS curve with order \( k = 4 \), knot vector: \( \{0, 0, 0, 0.111, 0.222, 0.333, 0.444, 0.555, 0.666, 0.777, 0.888, 1, 1\} \); control points \((x, y)\): \{(0.0, 0.0), (10.0, 60.0), (20.0, 60.0), (30.0, 20.0), (40.0, −20.0), (50.0, −60.0), (0.0, −30.0), (−50.0, −60.0), (−40.0, −20.0), (−80.0, 20.0), (−30.0, 20.0), (0.0, 60.0)\}; weights: \{1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0\}.

Rhombus contour: Polygon with 4 points: \{(350.0, −200.0), (380.0, −200.0), (400.0, −150.0), (370.0, −150.0), (350.0, −200.0)\}.

Star contour: NURBS curve with order \( k = 4 \), knot vector: \( \{0, 0, 0, 0.111, 0.222, 0.333, 0.444, 0.555, 0.666, 0.777, 0.888, 1, 1\} \); control points \((x, y)\): \{(0.0, 0.0), (30.0, 20.0), (80.0, 20.0), (40.0, −20.0), (50.0, −60.0), (0.0, −30.0), (−50.0, −60.0), (−40.0, −20.0), (−80.0, 20.0), (−30.0, 20.0), (0.0, 60.0)\}; weights: \{1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0\}.

Author contribution Yakun Jiang implemented all the algorithms and drafted the paper; Jihong Chen, Huicheng Zhou, and Jianzhong Yang helped design the algorithm and experiments of this paper; Pengcheng Hu proposed the original idea and revised the paper; Junxiang Wang helped implement the experiments.

Funding This work is supported by the Major Science and Technology Project of Hubei Province (Grant No. 2020AGA018-01).

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Song DN, Zhong YG, Ma JW (2019) Third-order contour-error estimation for arbitrary free-form paths in contour-following tasks. Precis Eng 60:85–92. https://doi.org/10.1016/j.precisioneng.2019.07.009
2. Hu Q, Chen Y, Yang J (2020) On-line contour error estimation and control for corner smoothed five-axis tool paths. Int J Mech Sci 171:105377. https://doi.org/10.1016/j.ijmecsci.2019.105377
3. Jun LT, Hui YJ, Min ZY, Yu ZC (2020) Time-varying reliability prediction modeling of positioning accuracy influenced by frictional heat of ball-screw systems for CNC machine tools. Precis Eng 64:147–156. https://doi.org/10.1016/j.precisioneng.2020.04.002
4. Yang L, Zhang L, Li D (2019) Five-axis CNC machine tool contour error control method. ACM Int Conf Proceeding Ser 12–16. https://doi.org/10.1145/3366194.3366197
5. Yang S, Ghasemi AH, Lu X, Okwudire CE (2015) Precompensation of servo contour errors using a model predictive control framework. Int J Mach Tools Manuf 98:50–60
6. Chin J, Lin T (1997) Cross-coupled precancellation method for the contouring accuracy of computer numerically controlled machine tools. Int J Mach Tools Manuf 37:947–967
7. Yang M, Yang J, Zhu L, Yu X (2020) A novel curvature circle iterative algorithm for contour error control of multi-axis CNC machine tools. Precis Eng 65:23–31. https://doi.org/10.1016/j.precisioneng.2020.05.005
8. Koren Y (1980) Cross-coupled biaxial computer control for manufacturing systems. ASME J Dyn Syst Meas Control 102: 265–272. https://doi.org/10.1115/1.3149612

9. Srinivasan K, Kulkarni PK (1990) Cross-coupled control of biaxial feed drive servomechanisms. J Dyn Syst Meas Control Trans ASME 112:225–232. https://doi.org/10.1115/1.2896129

10. Koren Y, Lo CC (1991) Variable-gain cross-coupling controller for contouring. CIRP Ann Manuf Technol 40:371–374. https://doi.org/10.1016/S0007-8506(79)60095-5

11. Yeh SS, Lo HP (2002) Estimation of the contouring error vector for the cross-coupled control design. IEEE/ASME Trans Mechatronics 7:44–51. https://doi.org/10.1109/3516.990886

12. Zhang D, Yang J, Chen Y, Chen Y (2015) A two-layered cross coupling control scheme for a three-dimensional motion control system. Int J Mach Tools Manuf 98:12–20. https://doi.org/10.1016/j.ijmachtools.2015.08.001

13. Lu H, Fan W, Zhang Y, Ling H, Wang S, Alghannam E, Duan M (2018) Cross-coupled fuzzy logic sliding mode control of dual-driving feed system. Adv Mech Eng 10:1–17. https://doi.org/10.1177/1687814018755518

14. Liu W, Ren F, Sun Y, Jiang S (2018) Contour error pre-compensation for three-axis machines by using cross-coupled dynamic friction control. Int J Mach Tools Manuf 98:551–563. https://doi.org/10.1016/j.ijmachtools.2018.01.001

15. Du F, Feng X, Li P et al (2018) Cross-coupled intelligent control for a novel two-axis differential micro-feed system. Adv Mech Eng 10:1–17. https://doi.org/10.1177/1687814018774628

16. Hao F, Xi XC, Poo AN (2012) Generalized Taylor series expansion for free-form two-dimensional contour error compensation. Int J Mach Tools Manuf 53:91–99. https://doi.org/10.1016/j.ijmachtools.2011.10.001

17. Zhu L, Zhao H, Ding H (2013) Real-time contouring error estimation for multi-axis motion systems using the second-order approximation. Int J Mach Tools Manuf 68:75–80. https://doi.org/10.1016/j.ijmachtools.2013.01.008

18. K E, Y A (1998) High speed contouring control algorithm for CNC machine tools. Proc ASME Dyn Syst Control Div-November:463–469

19. Li X, Zhao H, Zhao X, Ding H (2017) Contouring compensation control based on high accuracy contour error estimation for multi-axis motion systems. Int J Mach Tool Manuf 93:2263–2273. https://doi.org/10.1016/j.ijmachtools.2017.07.009

20. Hu C, Wang Z, Zhu Y, Zhang M (2018) Accurate three-dimensional contouring error estimation and compensation scheme with zero-phase filter. Int Mach Tools Manuf 128:33–40. https://doi.org/10.1016/j.ijmachtools.2018.01.001

21. Hao F, Poo AN (2013) Precision contouring control of machine tools. Int J Adv Manuf Technol 64:319–333. https://doi.org/10.1007/s00170-012-0215-5

22. Hao F, Poo AN (2013) Nonlinear autoregressive network with exogenous inputs based contour error reduction in CNC machines. Int J Mach Tools Manuf 67:45–52. https://doi.org/10.1016/j.ijmachtools.2012.12.007

23. Erwinski K, Paprocki M, Wawrzak A, Grzesiak LM (2016) Neural network contour error predictor in CNC control systems. 2016 21st Int Conf Methods Model Autom Robot MMAR 2016 537–542. https://doi.org/10.1109/MMAR.2016.7575193

24. Zhang D, Chen Y, Chen Y (2016) Iterative pre-compensation scheme of tracking error for contouring error reduction. Int J Adv Manuf Technol 87:3279–3288. https://doi.org/10.1007/s00170-016-8735-9

25. Yang M, Yang J, Ding H (2018) A two-stage friction model and its application in tracking error pre-compensation of CNC machine tools. Precis Eng 51:426–436. https://doi.org/10.1016/j.precisioneng.2017.09.014

26. Kono D, Inagaki T, Matsubara A, Yamaji I (2013) Stiffness model of machine tool supports using contact stiffness. Precis Eng 37:650–657. https://doi.org/10.1016/j.precisioneng.2013.01.010

27. Lu L, Zhao H, Zhang P (2019) The position control of machine tool based on fuzzy adaptive PID control. 2019 IEEE 8th Int Conf Fluid Power Mechatronics. https://doi.org/10.1109/FPM45753.2019.9035742

28. Shi S, Lin J, Wang X, Xu X (2015) Analysis of the transient backlash error in CNC machine tools with closed loops. Int J Mach Tools Manuf 93:49–60. https://doi.org/10.1016/j.ijmachtools.2015.03.009

29. Jiang Y, Chen J, Zhou H, Yang J, Xu G (2020) Nonlinear time-series modeling of feed drive system based on motion states classification. J Intell Manuf 31:1935–1948. https://doi.org/10.1007/s10845-020-01546-5

30. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518:529–533. https://doi.org/10.1038/nature14236

31. Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL (2018) Optimal and autonomous control using reinforcement learning: a survey. IEEE Trans Neural Networks Learn Syst 29:2042–2062. https://doi.org/10.1109/TNNLS.2017.2773458

32. Dong X, Shen J, Wang W et al (2018) Hyperparameter optimization for tracking with continuous deep Q-learning. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit:518–527. https://doi.org/10.1109/CVPR.2018.00061

33. Zhu H, Wang M, Wang B (2020) Research on interpolation of NURBS curve based on fractional power. J Phys Conf Ser 1633:012005. https://doi.org/10.1088/1742-6596/1633/1/012005

34. Hiemstra RR, Shepherd KM, Johnson MJ, Quan L, Hughes TJR (2020) Towards fully 3-D freeform trimming of NURBS: CAD embedded reparameterisation of trimmed B-rep geometry using frame-field guided global parameterization. Comput Methods Appl Mech Eng 369:113227. https://doi.org/10.1016/j.cma.2020.113227

35. Gers FA, Schmidhuber J, Cummins F (2018) Learning to forget: Continual prediction with LSTM. Neural Comput 12:2451–2471. https://doi.org/10.1162/neco.2018.999766030015015

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.