NURBS interpolator with pre-compensation based on discrete inverse transfer function for CNC high-precision machining

Taiyong Wang1,2 · Libo Cao1 · Yongbin Zhang1 · Jingchuan Dong1 · Songhui Jia1 · Chong Tian1 · Qingjian Liu3

Received: 30 July 2021 / Accepted: 18 May 2022 / Published online: 28 May 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
In the field of computerized numerical control (CNC) machining, high-speed and high-precision machining has been regarded as the key research by many scholars. In conventional methods, high-speed machining and high-precision machining are contradictory. It is inevitable to reduce the feedrate to improve the processing accuracy. In the paper, a pre-compensation based on discrete inverse transfer function (PDIT) theory is proposed. PDIT is able to improve machining contour accuracy without decreasing feedrate. The proposed PDIT theory is divided into three parts: NURBS interpolator, feedrate scheduling, and interpolator with pre-compensation. The NURBS interpolator has great advantage of directly interpolating the parametric curve. Therefore, the paper adopts the NURBS interpolator to accomplish interpolation. In the feedrate scheduling, S-type flexible acceleration and deceleration are used for path planning, and the maximum starting feedrate is obtained under the feedrate constraint. In the interpolator with pre-compensation, the NURBS interpolator is pre-compensated by PDIT. For the input, the response of the transfer function reaches a steady-state response within periods of time. Before steady-state response, the unsteady-state response exists in the transfer function. The unsteady-state response usually sustains dozens of interpolation periods, which inevitably lead to contour error in machining. Hence, the PDIT theory is employed to compensate the contour error caused by the unsteady-state response of transfer function to NURBS interpolator. The drive system is a transfer function, so the unsteady-state response of drive system can also cause machining errors before the steady-state response. In the paper, the NURBS interpolator is pre-compensated by PDIT theory before drive system to reduce contour errors and improve machining accuracy. Finally, the performance of the proposed PDIT is evaluated by simulations and experiments. The experimental results illustrate that PDIT theory obviously improve the machining accuracy.

Keywords Discrete · Transfer function · Interpolator with pre-compensation · Feedrate scheduling · Contour error

1 Introduction
In the modern manufacturing industry, the demand for high-speed and high-precision machining is increasing, and high-speed and high-precision machining has always been the topic research by many scholars. Many scholars reduce the vibration of the machine tool by reducing the fluctuation of the feedrate to achieve the purpose of improving the machining accuracy. Dong and Wang proposed the target feedrate filter method to achieve smooth feedrate [1, 2]. Liu Huan and other scholars proposed a linear programming mathematical model for the optimal feedrate in sensitive areas to obtain the optimal feedrate [3]. However, the disadvantage of this method is that the real-time performance is low. It needs the help of MATLAB linprog. Guo transformed the axial and tangential feedrate constraints into a convex optimization problem to find the optimal feedrate [4]. There are also many scholars to improve the machining accuracy by optimizing the interpolation algorithm. Scholars such as Wolfgang BOHM proposed parameter interpolation algorithms based on Lagrange [5], and Sang et al. proposed interpolation algorithms based on Taylor expansion [6–9]. Scholars Qin Hu
proposed the optimized Taylor parameter interpolation algorithm to compensate the truncation error [10–15]. The above scholars mainly focus on the CNC system, and some scholars also focus on the drive system. Scholars who pay attention to drive system mainly study contour errors caused by drive system. Hu proposed an analytical contour compensation algorithm to improve machining accuracy [16, 17]. Scholars Li proposed a method based on monocular vision to obtain and compensate contour errors [17]. Whether the feedrate smoothing algorithm and parameter interpolation algorithm in the CNC system, or the contour error compensation algorithm in the drive system, they are all subjectively divided into two parts. However, very little research literature combines the two. Therefore, the paper proposes the PDIT theory and makes full use of the relationship between the two to improve machining accuracy without losing feedrate. The PDIT theory studies the influence of transfer function of drive system on contour error, and directly compensates the contour error to the CNC commanded position.

In order to improve the performance of the NURBS interpolator, PDIT theory is proposed in the paper. PDIT is consist by NURBS interpolator, feedrate scheduling, and interpolator with pre-compensation. NURBS curve has great advantage of expressing free curve and is adopted in the paper. In order to obtain the maximum smooth feedrate and improve the efficiency of planning path, the feedrate scheduling based on S-type flexible acceleration and deceleration algorithm is adopted. This is introduced in Sect. 2. In Sect. 3, the principle of PDIT is introduced in detail. PDIT has ability to resolve the contour error caused by unsteady-state response. In the Sect. 4, simulations and experiments are carried out to evaluate the proposed NURBS interpolator. And conclusions are in Sect. 5.

2 Implementation of feedrate scheduling for NURBS interpolator

Non-uniform rational B-spline (NURBS) is defined by the node vector U, a set of control points P, and the weight of each control point. Assuming that the tool path is described by a NURBS curve of order p, the curve can be expressed as [19–21]:

\[
C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) \omega_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) \omega_i}
\]  

(1)

where \(P_i\) is the control point, \(\omega_i\) is the corresponding weights of \(P_i\), \((n + 1)\) is the number of control points, and \(p\) is the order of a NURBS curve. \(N_{i,p}(u)\) is the \(p\)th-degree of B-spline basis functions defined on the non-uniform knot vector \(u = \{u_0, u_1, ..., u_{n+p+1}\} = \{a, ..., a, u_{p+1}, ..., u_{m-p-1}, b, ..., b\}\), and \(m + 1\) is the number of knots. The de Boor-Cox algorithm can be used to calculate the \(p\)-degree B-spline basic function \(N_{i,p}(u)\) is [18]:

\[
\left\{
\begin{array}{ll}
N_{i,0} = \begin{cases} 1 & \text{if } i \in [u_i, u_{i+1}] \\
0 & \text{otherwise}
\end{cases} \\
N_{i,p}(u) = \frac{u - u_{i+p} - u_{i+p+1}}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p}(u)
\end{array}
\right.
\]  

(2)

NURBS interpolator has ability to directly interpolate parameter curve. Compared with traditional micro-segmented interpolators, NURBS interpolators are outstanding in improving machining accuracy and efficiency. It is indispensable for high-speed and high-precision machining. Therefore, the paper adopts the NURBS interpolator to accomplish the interpolation, and the relationship between interpolation position \(u_{i+1}\) and interpolation position \(u_i\) is [20–25]:

\[
u_{i+1} = u_i + \frac{V(u_i) \cdot T_s}{\left. \left( \frac{dC(u)}{du} \right) \right|_{u=u_i}} - \frac{V(u_i)^2 \cdot T_s^2}{2 \left. \left( \frac{d^2C(u)}{du^2} \right)^2 \right|_{u=u_i}}
\]  

(3)

It can be observed that the parameter \(u_{i+1}\) of the next interpolation period can be obtained from Formula (3); the feedrate \(V(u_i)\) is indispensable. Therefore, the feedrate scheduling is the highlight in the section. The first part is the S-type flexible acceleration and deceleration algorithm, which is the basis of feedrate scheduling. The second part is the feedrate scheduling based on motion model. The third part is the double-direction look-ahead algorithm. The double-direction look-ahead algorithm is divided into backward planning and forward planning. The purpose of backward planning is to obtain the maximum starting speed under the constraint of sensitive speed area. And the purpose of forward planning is using the maximum starting speed to plan the duration time \((\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7)\) of each segment in the 7-segment S-type acceleration and deceleration for a path with a fixed length of \(L\).

2.1 S-type acceleration and deceleration algorithm

Compared with trapezoidal acceleration and deceleration, the S-type acceleration and deceleration algorithm is more stable and has less impact on the drive system. It is suitable for high-speed and high-precision machining.

A complete S-type acceleration and deceleration has a total of 7 segments [26–29], named acc-acceleration (AA) segment, uniform acceleration (UA) segment, dec-acceleration (DA) segment, uniform speed (US) segment, dec-deceleration (DD)
segment, uniform deceleration (UD) segment, and acc-deceler-ation (AD) segment. Figure 1 is a complete S-type acceleration and deceleration curve. The details of each segment are described in detail in Appendix 1, and the meaning of $t_s$, $T_p$, $r_p$, $a(t)$, $a_p$, $v_p(t)$, $v_s(t)$, $s_p$, $s_{total}$ is also introduced in Appendix 1.

Jerk, acceleration, velocity, and displacement have common characteristics. The unified form algorithm.

$$J(t) = \begin{cases} J_1(t) = J_m, & t_0 < t < t_1 \ 
J_2(t) = 0, & t_1 < t < t_2 
J_3(t) = -J_m, & t_2 < t < t_5 \ 
J_4(t) = 0, & t_3 < t < t_4 
J_5(t) = -J_m, & t_4 < t < t_5 
J_6(t) = 0, & t_5 < t < t_6 
J_7(t) = J_m, & t_6 < t < t_7 
\end{cases} \quad (4)$$

$$v(t) = \begin{cases} v_1(t) = v_0 + \frac{1}{2} J_m (t - t_0)^2, & t_0 < t < t_1 
 v_2(t) = v_1 + A_m (t - t_1), & t_1 < t < t_2 
v_3(t) = v_2 + A_m (t - t_2) - \frac{1}{2} J_m (t - t_2)^2, & t_2 < t < t_3 
v_4(t) = v_3, & t_3 < t < t_4 
v_5(t) = v_4 - \frac{1}{2} J_m (t - t_4)^2, & t_4 < t < t_5 
v_6(t) = v_5 - A_m (t - t_5), & t_5 < t < t_6 
v_7(t) = v_6 - A_m (t - t_6) + \frac{1}{2} J_m (t - t_6)^2, & t_6 < t < t_7 \end{cases} \quad (6)$$

$$s(t) = \begin{cases} s_1(t) = v_2 (t - t_0) + \frac{1}{2} J_m (t - t_0)^3, & t_0 < t < t_1 
s_2(t) = s_1 + v_1 (t - t_1) + \frac{1}{6} A_m (t - t_1)^2, & t_1 < t < t_2 
s_3(t) = s_2 + v_2 (t - t_2) + \frac{1}{2} A_m (t - t_2)^2 - \frac{1}{6} J_m (t - t_2)^3, & t_2 < t < t_3 
s_4(t) = s_3 + v_3 (t - t_3), & t_3 < t < t_4 
s_5(t) = s_4 + v_4 (t - t_4) - \frac{1}{2} J_m (t - t_4)^3, & t_4 < t < t_5 
s_6(t) = s_5 + v_5 (t - t_5) - \frac{1}{6} A_m (t - t_5)^2, & t_5 < t < t_6 
s_7(t) = s_6 + v_6 (t - t_6) - \frac{1}{2} A_m (t - t_6)^2 + \frac{1}{6} J_m (t - t_6)^3, & t_6 < t < t_7 \end{cases} \quad (7)$$

To sum up, in the 7 segments of S-type acceleration and deceleration, the formulas of jerk, acceleration, velocity, and displacement have common characteristics. The unified form of the formula is:

$$J(t) = C \quad (8)$$

Therefore, the paper proposes 4 typical motion models to reduce the complexity of combination. For segments 4–7, they can all be mapped to these 4 types. For 3 segments and below, only the acceleration or the deceleration exist, which is easy to determine. The application of the four proposed motion models effectively reduces the complexity of

2.2 Feedrate scheduling

The process of path planning will not always appear all the 7 segments. Only when the path is long enough, all the 7 segments will exist in the drive system, and it is a complete acceleration and deceleration speed curve. If the path is not long enough, not all the 7 segments but parts of them will appear in the drive system. Therefore, it is important to determine which part appears. The detailed analysis is given below.

In the permutation and combination rules of probability theory, the combination of 7 segments are $2^7$ different combinations in total. Some permutation and combination results do not exist in practice, even so, there are still many combinations. It is too complex to analyze for each situation.
2.2.1 Establish motion models

For a certain length of $L$ path, 4 special models are analyzed.

Model a 7-segment S-type acceleration and deceleration

In the case, the given path $L$ is long enough, and the speed curve is a complete 7-segment acceleration and deceleration curve. The maximum feedrate can reach $V_m$, and the maximum acceleration can reach $A_m$. The time of each segment satisfies the relation $T_1 = T_3 = T_5 = T_7$, $T_2 = T_6$. And $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$, $T_7$ can be obtained by using Table 1 and Formulas (4), (5), (6) and (7). Formula $J_m * \tau_1 * \tau_2 + J_m * \tau_1 * \tau_4 = V_m$ is introduced in Appendix 2.

Model b 4-segment S-type acceleration and deceleration

In the case, the given path $L$ is not long enough, and the speed curve is a 4-segment acceleration and deceleration curve. This speed curve only includes AA segment ($t_0$-$t_1$), DA segment ($t_2$-$t_3$), DD segment ($t_4$-$t_5$), AD segment ($t_6$-$t_7$). The maximum feedrate can reach $V_m'$ and not reach $V_m$, and the maximum acceleration can reach $a_m$ or just reach $A_m$. The time of each segment satisfies the relation $T_1 = T_3 = T_5 = T_7$, $T_2 = T_6$. And $T_1$, $T_3$, $T_5$, $T_7$ can be obtained using Table 2 and Formulas (4), (5), (6).
Table 2  The duration of each segment of 4-segment S-type acceleration and deceleration

| 4-seg | τ1 | τ2 | τ3 | τ4 | τ5 | τ6 | τ7 |
|-------|----|----|----|----|----|----|----|
| \(a_m(A_m)\) | \(a_m/J_m\) | r1 | r1 | r1 | r1 | r1 |

and (7). Figure 2b is the 4-segment S-type acceleration and deceleration.

**Model c** 6-segment S-type acceleration and deceleration

In the case, the given path \(L\) is not long enough, and the speed curve is a 6-segment acceleration and deceleration curve. This speed curve only includes AA segment \((t_0-t_1)\), UA segment \((t_1-t_2)\), DA segment \((t_2-t_3)\), DD segment \((t_3-t_4)\), UD segment \((t_4-t_5)\), AD segment \((t_5-t_6)\). The maximum feedrate reach \(V_m/V_m'\), and the maximum acceleration can reach \(A_m\). The time of each segment satisfies the relational expression \(T_1 = T_2 = T_3 = T_4 = T_5 = T_6 = 0\). And \(T_1, T_2, T_3, T_4, T_5, T_6, T_c\) can be obtained by using Table 3 and Formulas (4), (5), (6) and (7). Figure 2c is the 6-segment S-type acceleration and deceleration.

**Model d** 5-segment S-type acceleration and deceleration

In the case, the given path \(L\) is not long enough, and the speed curve is a 5-segment acceleration and deceleration curve. This speed curve only includes AA segment \((t_0-t_1)\), UA segment \((t_1-t_2)\), DA segment \((t_2-t_3)\), DD segment \((t_3-t_4)\), UD segment \((t_4-t_5)\), AD segment \((t_5-t_6)\). Taking the UA segment as an example, the maximum feedrate can reach \(V_m\) which is smaller than \(V_m'\) and the maximum acceleration can reach \(A_m\). The time of each segment satisfies the relational expression \(T_1 = T_2 = T_3 = T_4 = T_5 = 0\), \(T_6 = \tau_1\). And \(T_1, T_2, T_3, T_4, T_5, T_6\) can be obtained using Table 4 and Formulas (4), (5), (6) and (7). Figure 2d is the 5-segment S-type acceleration and deceleration.

2.2.2 Feedrate scheduling

As shown in Fig. 3, the feedrate scheduling is introduced in the section based on the four proposed motion model in detail. 

\(L\) is the given path length. Model c is a 6-segment acceleration and deceleration curve, and \(L6\) is defined as the critical displacement when the velocity just reaches the maximum speed \(V_m\). Model d is a 5-segment acceleration and deceleration curve, and \(L5\) is defined as the critical displacement when the acceleration can reach the maximum acceleration \(A_m\) but the velocity cannot reach \(V_m\). Model b is a 4-segment acceleration and deceleration curve, and \(L4\) is the critical displacement when the acceleration just reaches the maximum acceleration \(A_m\).

**Category 1** When \(L6 < L < L7\) or \(L > L7\), it is a complete acceleration and deceleration curve of 7 segments. \(τ1-τ7\) can be solve using Table 1.

**Category 2** When \(L5 < L < L6\), \(L6\) is the displacement when the velocity just reaches the maximum speed \(V_m\), so the displacement \(L\) only reaches the speed \(V_m'\) which is smaller than \(V_m\). Because the displacement \(L5\) must be able to reach the maximum acceleration \(A_m\), the displacement \(L\) must be able to reach the maximum acceleration \(A_m\). Consequently, \(L\) belonging to the interval \([L5, L6]\) is enough long to reach \(A_m\) but not enough long to reach \(V_m\). Therefore, category 2 belongs to model c and \((τ1, τ2, τ3, τ4, τ5, τ6, τ7)\) is solved by the formula in Table 3. For example, \(τ1\) can be solved by formula \(A_m/J_m\) and \(τ2\) can be solved by formula \(s_3 = f(τ1, τ2) = L/2\).

**Category 3** When \(L4 < L < L5\), because the displacement \(L4\) is just length to reach \(A_m\), the displacement \(L\) must be able to reach the maximum acceleration \(A_m\). The displacement \(L5\) is enough long to reach \(A_m\) but not enough long to reach the maximum speed \(V_m\). Consequently, \(L\) belonging to the interval \((L4, L5)\) interval is enough long to reach \(A_m\) but not enough long to reach \(V_m\). Thereby, category 3 belongs to model d and \((τ1, τ2, τ3, τ4, τ5, τ6, τ7)\) can be solved by the formula in Table 4. \(τ1\) is solved by formula \(A_m/J_m\), \(τ2\) is solved by formula \(J_m^*τ1^*τ2 + J_m^*τ1^*τ1 = V_m^\prime\), and \(τ3\) is solved by formula \(s_3 = f(τ1, τ2, τ3) = L\).

**Category 4** When \(L < L4\), because \(L4\) is the displacement just reaching the maximum acceleration \(A_m\). \(L\) belonging to the \((0, L4)\) interval is only enough to reach \(a_m\) which is smaller than \(A_m\). Therefore, category 4 belongs to model b. \((τ1, τ2, τ3, τ4, τ5, τ6, τ7)\) can be solved by using Formula in Table 2 and \(τ1\) can be solved by using \(A_m/J_m\).

Table 3  The duration of each segment of 6-segment S-type acceleration and deceleration

| 6-seg | τ1 | τ2 | τ3 | τ4 | τ5 | τ6 | τ7 |
|-------|----|----|----|----|----|----|----|
| \(A_m\) | \(A_m/J_m\) | r1 | r1 | r1 | r1 | r1 |
| \(L\) | \(s_3 = f(τ1, τ2) = L/2\) | r2 | r1 | r1 | r1 |
2.3 Backward and forward planning

2.3.1 Backward planning

Path planning is divided into backward planning and forward planning [26]. Backward planning is introduced in the part, and forward planning is introduced in next part.

The purpose of backward planning is to obtain the maximum starting speed with a limit of sensitive speed area. The paths are planned in groups not one by one, and how to get the maximum starting speed with the end speed under the feedrate constraint [30–32]. This is the meaning of backward planning. The backward planning process is as follows:

![Diagram](image)
Step 1: Getting the end speed $V_e$ of last path.
Step 2: For the last path with a given length of $L_n$, $V_e$ is regarded as the starting speed $V_s$. The last path is planned reversely to obtain the end speed $V_{e1}$ with the feedrate scheduling introduced in previous section.
Step 3: $V_{e2}$ is the smallest velocity between the end velocity $V_{e1}$ and the sensitive velocity $V_{sensor}$. For the penultimate path with a given length of $L_{n-1}$, $V_{e2}$ is the starting speed of $L_{n-1}$. $L_{n-1}$ is planned reversely to obtain the end speed $V_{e2}$.
Step 4: Repeat step 3 until the first path with a given length of $L_1$ is planned reversely and the end speed $V_{en}$ is obtained. This end speed $V_{en}$ is the maximum starting speed $V_s$ of the set of paths.

2.3.2 Forward planning

In the previous section, the starting velocity $V_s$ of a set of paths is obtained by backward planning. With $V_s$ planned by the backward planning and the feedrate scheduling based on Tables 1, 2, 3 and 4, the duration $t_1$, $t_2$, $t_3$, $t_4$, $t_5$, $t_6$, $t_7$ of each segment of the 7-segment S-type acceleration and deceleration can be obtained for a given length $L$ [32, 33].

3 Interpolation pre-compensation based on discrete inverse transfer function

In Sect. 2.2, the feedrate scheduling will be generated with the proposed models a–d, and the path will be planned by forward planning and backward planning in Sect. 2.3. After that, CNC system will send the planned path to drive system with Formula (3) in Sect. 2. In Formula (3), $u_i$ is the current NURBS interpolation parameter, $V(u_i)$ is the scheduled feedrate. The next interpolation parameter $u_{i+1}$ will be obtained with $u_i$, current feedrate $V(u_i)$, and interpolation period $T_s$. Each motion axis of CNC system will move to the commanded position with the obtained $u_{i+1}$ and NURBS curve Formula (1), finally. However, it is impossible to precisely move the axis consisting of the drive system to the commanded position. The reason will be introduced in Sect. 3.

Dispersed micro-segment from CAM is input to the CNC system. The CNC system receives micro-segment from CAM and transmits the commanded position $P_r$ to the drive system. The output of the drive system is the actual position $P_a$. As shown in Fig. 4.

The commanded position $P_r$ output by the CNC system can accurately express the contour of the workpiece, but the actual position $P_a$, which is output from the drive system cannot completely follow the commanded position $P_r$. As a result, the actual position $P_a$ and the commanded position $P_r$ are not exactly same, leading to a reduction in machining accuracy [34]. This machining deviation is called contour error [35–39], as shown in Fig. 5.

The difference is caused by the unsteady-state response of the drive system, which is a transient state before steady-state response of drive system. The paper proposes interpolation pre-compensation theory to avoid contour errors caused by the unsteady-state response. In Sect. 3.1, the reason of leading contour error is analyzed. In Sect. 3.2, before proposing the pre-compensation theory, transfer function discretization theory is induced at first, and pre-compensation theory is introduced in Sect. 3.3.

![Fig. 4 Flow diagram of interpolation position](image)

![Fig. 5 Contour error](image)
3.1 Response analysis of drive system

The drive system consists of a closed-loop transfer function \( G(s) \) [40–43]. For a typical drive, system is composed by a position loop and a velocity loop and PMSM mathematical model. The control block diagram is shown in Fig. 6 [44]. The position loop is controlled by the P controller, and the velocity loop is controlled by the PI controller.

Therefore, the closed-loop transfer function \( G(s) \) of drive system can be calculated as

\[
G(s) = \frac{k_{pp}k_{vp}k_{vi}}{Ts^3 + (1 + Tk_{vp}k_{vi})s^2 + k_{vp}k_{vi}s + k_{pp}k_{vp}k_{vi}} \quad (12)
\]

In this situation [45], the drive system can be reasonably approximated as a first-order inertial system (FOIS). The step response of the closed-loop transfer function \( G(s) \) is depicted in Fig. 7. From Fig. 7, it is easy to find that, before reaching steady-state, the low bandwidth zone of FOIS with time constant \( T \) is about \((3 \sim 4)T\). The low bandwidth zone can lead to contour error, especially in the field of high-precision machining. \( T \) is generally greater than interpolation period \( T_s \). Therefore, it is difficult, even impossible, to reach steady-state within \( T_s \). To address this issue, the drive system is simplified as a first-order inertial system in Chen et al. [45, 46], while the simplify model is not the real mathematical model. Simplifying a model inevitably introduces model errors, even if it is a higher-order simplified system. In the paper, a new theory is proposed to eliminate the model error.

3.2 Transfer function discretization theory

The response of transfer function from unsteady-state to steady-state causes contour errors. Therefore, the article focuses on the unsteady-state process of transfer function to solve the issue. Before discussing the issue, a brief explanation of transfer function discretization theory is induced at first.

The CNC system is a linear time invariant systems (LTI) [46–48]. The response of LTI system is:
Theory 1 The transfer function is the ratio of the output function to the input function in Z-transform [46], which is $H(z) = \frac{Y(z)}{X(z)}$. If the input function is a unit pulse which is impulse signal, the Z-transform of unit pulse is 1. Thereby, the transfer function is the output in Z-transform, $H(z) = Y(z)$.

Using the Theory 1 and Formula (15), the transfer discrete function $h(n)$ can be obtained.

Theory 2 Delay, multiplication, and addition shown in Fig. 8 are the basic operations of LTI.

The basic operations of LTI system. a The delay of LTI system. b The addition of LTI system. c The multiplication of LTI system.
Because of the last element $\Delta x(0) = 0$ in the last line of matrix $\Delta x(i)$, the last row of matrix $\Delta x(i)$ is all zero in Formula (19). Therefore, $h(m)$ will not work and will not affect the discrete transfer function $H(m)$. According to the principle of one-to-one correspondence between the matrix equation and its augmented matrix, the augmented matrix of the matrix equation (19) is,

\[
\begin{bmatrix}
\Delta x(0) & 0 & 0 & 0 & \cdots & 0 & 0 \\
\Delta x(1) & \Delta x(0) & 0 & 0 & \cdots & 0 & 0 \\
\Delta x(2) & \Delta x(1) & \Delta x(0) & 0 & \cdots & 0 & 0 \\
\Delta x(3) & \Delta x(2) & \Delta x(1) & \Delta x(0) & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\Delta x(m-2) & \Delta x(m-3) & \Delta x(m-4) & \Delta x(m-5) & \cdots & \Delta x(0) & 0 \\
\Delta x(m-1) & \Delta x(m-2) & \Delta x(m-3) & \Delta x(m-4) & \cdots & \Delta x(1) & \Delta x(0) \\
\end{bmatrix}
\begin{bmatrix}
\Delta y(0) \\
\Delta y(1) \\
\Delta y(2) \\
\Delta y(3) \\
\vdots \\
\Delta y(m-2) \\
\Delta y(m-1) \\
\end{bmatrix}
\]  

And correspondingly, the unknown vector $H(m)$ to be solved for is

\[
(h(1), h(2), h(3), h(4), \ldots, h(m-1), h(m))^T
\]  

Using the elementary transformation principle of the matrix and $\Delta x(0) = \Delta y(0) = 0$, augmented matrix (20) is simplified to

\[
\begin{bmatrix}
\Delta x(1) - \Delta x(0) & 0 & 0 & \cdots & 0 & 0 \\
\Delta x(2) - \Delta x(1) & \Delta x(1) - \Delta x(0) & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\Delta x(m-1) - \Delta x(m-2) & \Delta x(m-2) - \Delta x(m-3) & \Delta x(m-3) - \Delta x(m-4) & \cdots & \Delta x(1) - \Delta x(0) & \Delta y(m-1) - \Delta y(m-2) \\
\end{bmatrix}
\]  

However, a set of unit pulses cannot be directly input to the actuator, nor are they regular inputs in CNC machine. Assuming that the regular inputs are $x(n) = \{x(0), \ldots, x(n-1)\}$, which are the commanded position. $x(0)$ is usually regarded as the initial value, named init, and the increments $\Delta x(i)$ which is impulse is $x(i)$ relative to $x(0)$, that is $x(i) - x(0)$, $i = 0, \ldots, n-1$. The outputs $y(n)$ is same to $x(n)$. Obviously, $\Delta x(0) = \Delta y(0) = 0$.

On the basic of Formula (15), $H(m)$ can be obtained by Formula (19) with the known $\Delta x(i)$ and $\Delta y(i)$.

\[
\begin{bmatrix}
\Delta x(0) & 0 & 0 & 0 & \cdots & 0 & 0 \\
\Delta x(1) & \Delta x(0) & 0 & 0 & \cdots & 0 & 0 \\
\Delta x(2) & \Delta x(1) & \Delta x(0) & 0 & \cdots & 0 & 0 \\
\Delta x(3) & \Delta x(2) & \Delta x(1) & \Delta x(0) & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\Delta x(m-2) & \Delta x(m-3) & \Delta x(m-4) & \Delta x(m-5) & \cdots & \Delta x(0) & 0 \\
\Delta x(m-1) & \Delta x(m-2) & \Delta x(m-3) & \Delta x(m-4) & \cdots & \Delta x(1) & \Delta x(0) \\
\end{bmatrix}
\begin{bmatrix}
h(1) \\
h(2) \\
h(3) \\
h(4) \\
\vdots \\
h(m-1) \\
h(m) \\
\end{bmatrix}
= 
\begin{bmatrix}
\Delta y(0) \\
\Delta y(1) \\
\Delta y(2) \\
\Delta y(3) \\
\vdots \\
\Delta y(m-2) \\
\Delta y(m-1) \\
\end{bmatrix}
\]  

$\Delta x(i) - \Delta x(i-1)$ is the impulse pulse in the $i$th interpolation period, and $\Delta y(i) - \Delta y(i-1)$ is represented by the $y_{inc}(i)$. Augmented matrix (22) can be simplified as matrix (23). According to the principle of elementary transformation of matrices, matrix (23) can be further simplified as matrix (24),

\[
\begin{bmatrix}
\delta(1) & 0 & 0 & \cdots & 0 & 0 & y_{inc}(1) \\
\delta(2) & \delta(1) & 0 & \cdots & 0 & 0 & y_{inc}(2) \\
\delta(3) & \delta(2) & \delta(1) & \cdots & 0 & 0 & y_{inc}(3) \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\delta(m-2) & \delta(m-3) & \delta(m-4) & \cdots & \delta(1) & 0 & y_{inc}(m-2) \\
\delta(m-1) & \delta(m-2) & \delta(m-3) & \cdots & \delta(2) & \delta(1) & y_{inc}(m-1) \\
\end{bmatrix}
\begin{bmatrix}
h(1) \\
h(2) \\
h(3) \\
h(4) \\
\vdots \\
h(m-1) \\
h(m) \\
\end{bmatrix}
= 
\begin{bmatrix}
\Delta y(0) \\
\Delta y(1) \\
\Delta y(2) \\
\Delta y(3) \\
\vdots \\
\Delta y(m-2) \\
\Delta y(m-1) \\
\end{bmatrix}
\]
\[
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
y_{inc}(1) \\
y_{inc}(2) - y_{inc}(1) \\
y_{inc}(3) - y_{inc}(2) \\
\vdots \\
y_{inc}(m) - y_{inc}(m-2) \\
y_{inc}(m-1) - y_{inc}(m-2)
\end{bmatrix}
\]

(24)

And obviously, the discrete transfer function \(H(m)\) is

\[
H(m) = \begin{bmatrix}
h(1) \\
h(2) \\
h(3) \\
\vdots \\
h(m-1) \\
h(m)
\end{bmatrix}
= \begin{bmatrix}
y_{inc}(1) \\
y_{inc}(2) - y_{inc}(1) \\
y_{inc}(3) - y_{inc}(2) \\
\vdots \\
y_{inc}(m) - y_{inc}(m-2) \\
y_{inc}(m-1) - y_{inc}(m-2)
\end{bmatrix}
\]

(25)

When the motor is in enabled state, a slight vibration will be kept all the time. Considering the long-term existence of jitter, the average of the discrete transfer function \(H(m)\) is adopted and shown in Formula (26).

\[
\bar{h}(i) = \frac{1}{N} \sum_{k=0}^{N-1} h(k), \quad i = 1, 2, \ldots, m
\]

(26)

For the law of conservation of energy, the sum of the output should be a unit pulse if the input is a unit pulse. However, the sum of the output is not a unit pulse due to the data accuracy error, the \(H(m)\) should be normalized.

\[
H(m) = \left(\begin{array}{cccc}
n(1) & n(2) & n(3) & \cdots & n(m-1) & n(m) \\
\sum_{i=1}^{m} h(i) & \sum_{i=1}^{m} h(i) & \sum_{i=1}^{m} h(i) & \cdots & \sum_{i=1}^{m} h(i) & \sum_{i=1}^{m} h(i)
\end{array}\right)
\]

(27)

\(P_r, P_a, \text{ and } P_c\) represent the commanded position, the actual position, and the pre-compensated position. Generally, the commanded position \(P_r\) is transmitted to the drive system which is a transfer function, and the actual position \(P_a\) is output. Based on the analysis of the response of transfer function, \(P_c\) is not in accordance with \(P_r\). If the consistency is achieved between \(P_a\) and \(P_r\) after processed by \(H(m)\), \(P_c\) must be compensated and the \(P_c\) is the pre-compensated position. That is, the commanded position \(P_r\) is replaced by the pre-compensated position \(P_c\), and actual position \(P_a\) is replaced by the commanded position \(P_r\). The pre-compensation principle for the commanded position will be introduced in detail.

If a constant value is continuously input to the transfer function, it is always a steady-state response after a while. Only when the input changes, will the unsteady-state response appear. Therefore, the concept of input increment \(W(n)\) which is named weight is introduced.

\[
W(n) = \begin{bmatrix}
w(1) \\
w(2) \\
w(3) \\
\vdots \\
w(m) \\
w(m+1) \\
w(n)
\end{bmatrix} = \begin{bmatrix}
P_r(1) - n(0) \\
P_r(2) - n(0) \\
P_r(3) - n(0) \\
\vdots \\
P_r(m) - n(0) \\
P_r(m+1) - n(0) \\
P_r(n) - n(0)
\end{bmatrix}
\]

(28)

The input increment \(W(n)\) is the increment of the current commanded position \(P_r(i)\) relative to the first commanded position \(P_r(0)\) which is the initial value. Based on the above analysis of pre-compensation principle, \(W(n)\) is the output of the transfer function. Similarly, \(PW(n)\) is the increment of pre-compensation position and is the input.

\[
PW(n) = \begin{bmatrix}
pw(1) \\
pw(2) \\
pw(3) \\
\vdots \\
pw(m) \\
pw(m+1) \\
pw(n)
\end{bmatrix} = \begin{bmatrix}
P_r(1) - n(0) \\
P_r(2) - n(0) \\
P_r(3) - n(0) \\
\vdots \\
P_r(m) - n(0) \\
P_r(m+1) - n(0) \\
P_r(n) - n(0)
\end{bmatrix}
\]

(29)

Then with the discrete inverse transfer function \(H(m)\), the relationship between input \(PW(n)\) and output \(W(n)\) is

\[
PW(n) = \begin{bmatrix}
pw(1) \\
pw(2) \\
pw(3) \\
\vdots \\
pw(m) \\
pw(m+1) \\
pw(n)
\end{bmatrix} = \begin{bmatrix}
w(1) \\
w(2) \\
w(3) \\
\vdots \\
w(m) \\
w(m+1) \\
w(n)
\end{bmatrix}
\]

(30)

Then the pre-compensation position \(P_c(i)\) which is inputted to the drive system is:
Fig. 9 The mathematical simplified model of drive system
pre-compensation theory based on the discrete inverse transfer function proposed in this paper has a significant improvement effect in terms of contour.

In the second simulation, a spiral curve is used as an example. The spiral curve [49] Formula (32) is as follows. In Fig. 12, the red trajectory is the actual trajectory.
compensated by PDIT theory proposed in this paper, the blue trajectory is the CEEC theory, and the black point is the ideal commanded position. Figure 12 shows that the red trajectory of the PDIT theory is closer to the ideal curve than the blue curve of the CEEC theory. Figure 13 and Table 6 are the contour error of PDIT theory and CEEC theory. The contour error of PDIT is smaller than CEEC. For the contour error of the spiral curve, the contour error of PDIT is $0.000849–0.147475$ mm, and the contour error of CEEC is $0.001189–0.496462$ mm.

$$\begin{align*}
c_x(t) &= \cos(2\pi)t \\
c_y(t) &= 3\sin(2\pi)t
\end{align*}$$

(32)

### Table 5 The contour error analysis of six corner in different methods

| Compensation method | PDIT  | FOIS | CEEC  |
|---------------------|-------|------|-------|
| Contour error       |       |      |       |
| Maximum             | Mean  | Minimum | Mean | Minimum | Mean  | Minimum |
| The 1st corner      | 0.000484 | 0.00047 | 0.000464 | 0.015263 | 0.00020544 | 0.000238 | 0.002873 | 0.0008215 | 0.00008 |
| The 2nd corner      | 0.000433 | 0.00042392 | 0.000412 | 0.002172 | 0.00156605 | 0.000716 | 0.001119 | 0.0009185 | 0.00059 |
| The 3rd corner      | 0.000491 | 0.000485 | 0.000477 | 0.011689 | 0.00288815 | 0.000394 | 0.002595 | 0.0012704 | 0.000561 |
| The 4th corner      | 0.000442 | 0.00043371 | 0.000422 | 0.009988 | 0.00259823 | 0.000373 | 0.002272 | 0.0011701 | 0.000462 |
| The 5th corner      | 0.000431 | 0.00042125 | 0.000407 | 0.00215 | 0.0014043 | 0.000542 | 0.001109 | 0.0008504 | 0.000404 |
| The 6th corner      | 0.000457 | 0.00044999 | 0.000438 | 0.014388 | 0.00195746 | 0.000217 | 0.002691 | 0.0007969 | 0.000104 |

4.2 Experiment

In addition to the simulation studies applied in the previous section, experiments are performed to verify the contour accuracy improvements of the proposed method. As shown in Fig. 14, the experimental platform is a CNC machine tool consisted of two axes with build-in position loops, and the motion parameter, which is $K_p$, $K_i$ and forward parameter of position, velocity, and current loop in servo, are in Table 7. The position loop is a proportional controller, while the velocity loop and the current loop are both proportional and integral controllers. The axes are controlled by a PC-based CNC system with a real-time Linux operating system and EtherCAT network. EtherCAT is the real-time industrial
Ethernet technology owned by the EtherCAT Technology Group (ETG). As shown in Fig. 14, the commanded position is transmitted to axes by EtherCAT networks. Moreover, the EtherCAT network can retrieve the actual position via its object 6064 in each interpolation period.

With NURBS curve in Formula (1) and the Taylor expansion in Formula (3), the next position $C(u_{i+1})$ which is named weight in Formula (31) can be obtained. In Formula (3), $C(u_i)$ is the current position of NURBS curve, and $V(u_i)$ is the feedrate which is achieved by feedrate scheduling with the proposed models a–d in Sect. 2.2. The pre-compensation position $p_{w_1}(i)$ of the next interpolation period can be get by Formula (31) with the obtained $C(u_{i+1})$ in Sect. 2. In Formula (31), $W(n)$ is the output of the transfer function, $PW(n)$ is the increment of pre-compensation position and is also the inputs, and as we can see, the key of getting pre-compensation position $P_{w_1}(i)$ is to obtain the discrete transfer function $H(m)$.

In order to minimize the effects of magnetic and mechanical losses, a total of 50,000 inputs are used to obtain the discrete transfer functions $H(m)$ and named $x(0), x(1), \ldots, x(49,999)$. The 50,000 inputs are transmitted to the actuators by EtherCAT network with object 607A, and the duration is 50 s in total if the interpolation period is 1 ms. Correspondingly, the output of drive are $y(0)$,

| Theory   | Maximum     | Minimum     | Delta     |
|----------|-------------|-------------|-----------|
| PDIT     | 0.147475    | 0.000849    | 0.146626  |
| CEEC     | 0.496462    | 0.001189    | 0.495273  |

Table 6 Datasheet of the contour error of the spiral curve

Fig. 13 Contour error of the entire path of the spiral curve

Fig. 14 The PC-based experimental two-axis platform with EtherCAT communication
$y(1), \ldots, y(49,999)$, which are the actual position of the inputs and can be get by EtherCAT with its object 6064 in every interpolation period. Thereby, $H(m)$ can be obtained directly by Formula (24) with the retrieved actual position $y(i)$ by EtherCAT. In the experiment, only 100 input signals are required to detect the transfer function $H(m)$ once. Therefore, the number of detecting discrete transfer function $H(m)$ is $N = 50,000/100 = 500$. Thereby, the obtained discrete transfer function $H(m)$ can be averaged by Formula (26) with $N$. And finally, the $H(m)$ can be normalized by Formula (27).

The method proposed in the paper focuses on reducing the effect of unsteady-state response caused by the increment between adjacent positions. The position increment changes significantly at the corners, resulting in an overshoot with unsteady-state response. There are many types of corners in the butterfly, so the butterfly is still used for processing experiments. The butterfly is machined by the experimental bi-axial table in Fig. 14. All the selected six corners used to evaluate the performance of the proposed PDIT theory are same to the simulation in Sect. 4.1. The contour of all the six corners of the experiment is shown in Fig. 15, which is comparison among PDIT, FOIS, and CEEC methods. The red line is actual axes trajectory which is compensated by PDIT, the blue line represents the output of commanded position in servo with FOIS, the purple line is the actual trajectory compensated by CEEC, and the black line is the commanded position which is the ideal position without contour error. As we can see, in Fig. 15, the red line is closer to the black line than the blue line and purple line. It is obvious that the proposed PDIT theory can significantly improve the contour accuracy in corners. The contour error is shown in Fig. 16, and the contour error data analysis of the six corner is given in Table 8. The maximum contour error of PDIT theory is 0.038 mm, the maximum contour error of CEEC theory is 0.09 mm, and the maximum contour error is 0.071 mm in FOIS. In terms of mean contour error, the proposed PDIT improves the contour accuracy of the selected six corners by an average of 28.712% compared to FOIS, and 57.35% compared to CEEC. The interpolation pre-compensation

| Servo parameter | Position loop | Velocity loop | Current loop |
|-----------------|---------------|---------------|--------------|
| Proportional gain | 1000          | 210           | 100          |
| Integration time | —             | 50            | 2            |
| Forward parameter | 130          | —             | —            |

Fig. 15 Contour comparison of the selected six corners with PDIT CEEC and FOIS methods
theory based on the discrete inverse transfer function proposed in this paper has a significant improvement effect in terms of contour. It can be seen that the compensation effect of CEEC is slightly worse in experiment than in simulation, because the pre-compensated position $P_c(i)$ in the experiment is approximated. The position in servo is based on encoder and is represented by the number of pulses, so the command position $P_r(i)$ and the actual position $P_a(i)$ are both integers in servo. However, the pre-compensated position $P_c(i)$ obtained by CEEC from the actual position $P_a(i)$ is float. Therefore, in experiment, the float $P_c(i)$ needs to be rounded to the nearest integer and then sent to servo, while

**Fig. 16** Datasheet of the contour error of the six butterfly’s corner

| Compensation method | Contour error | PDIT | FOIS | CEEC |
|---------------------|---------------|------|------|------|
|                      | Maximum       | Mean | Minimum | Maximum | Mean | Minimum | Maximum | Mean | Minimum |
| The 1st corner       | 0.030580917   | 0.011089221 | 0.002843508 | 0.065683096 | 0.016076768 | 0.003670214 | 0.054164995 | 0.016818097 | 0.000306508 |
| The 2nd corner       | 0.020413201   | 0.009752957 | 0.000356231 | 0.024714134 | 0.011495158 | 0.000255175 | 0.061472962 | 0.027952368 | 0.000247829 |
| The 3rd corner       | 0.038586492   | 0.012161042 | 0.001702205 | 0.071204958 | 0.019930073 | 0.003176925 | 0.05425369 | 0.025377633 | 1.21E-05 |
| The 4th corner       | 0.038140352   | 0.013317994 | 0.000651309 | 0.056238117 | 0.015769927 | 0.00311747 | 0.065689408 | 0.030321205 | 0.001308382 |
| The 5th corner       | 0.011807606   | 0.00801809 | 0.002752759 | 0.020612177 | 0.013045497 | 0.006273115 | 0.032937432 | 0.026344607 | 0.009912912 |
| The 6th corner       | 0.037543588   | 0.006472918 | 0.000195523 | 0.068640353 | 0.009664652 | 0.000217465 | 0.091537749 | 0.019739463 | 0.001049655 |
the float $P_c(i)$ is sent directly to the mathematical simulation model of servo in simulation. The same goes for the spiral experiment.

In the second experiment, in order to maintain consistency between the simulation and experiment, the spiral curve is still taken as the example. In Figs. 17 and 18, the red line is about the PDIT theory proposed in this article, the purple line is about the CEEC theory, and the blue line is the FOIS theory. The black point is the ideal commanded position in Fig. 17. Figure 17 shows that the red trajectory of

Fig. 17 The entire path of the spiral curve with PDIT CEEC and FOIS methods

Fig. 18 Contour error after compensation by PDIT CEEC and FOIS methods for the full path of a spiral curve
the PDIT theory is closer to the ideal curve than the purple curve of the CEEC theory and the blue curve of the FOIS theory. Figure 18 is the contour error after compensation by PDIT CEEC and FOIS methods for the full path of a spiral curve. As we can see in Fig. 18, The red line representing the compensation effect of PDIT theory proposed by the paper is minimal. And Table 9 is the contour error data analysis of the entire spiral curve. The contour error of compensation by PDIT is smaller than FOIS and CEEC. For the contour error of the spiral curve, the maximum contour error of PDIT is 0.0739 mm, the CEEC is 0.138 mm, and FOIS is 0.0772 mm. In terms of mean contour error, the proposed PDIT improves the contour accuracy of the entire spiral curve by an average of 47.372% compared to CEEC, and 5.5% compared to FOIS.

### 5 Conclusions

The NURBS interpolator with pre-compensation based on discrete inverse transfer function is important to guarantee high-efficiency and high-accuracy machining. In conventional methods, high-speed machining and high-precision machining are contradictory, and the proposed PDIT breaks this point. Without decreasing feedrate, the proposed PDIT can effectively improve the machining contour accuracy, especially at the corners. With the proposed method, the NURBS interpolator can pre-compensate for contour errors due to the unsteady-state response of the servo system. As compared to the other two algorithms, the proposed method can significantly improve machining accuracy. In simulation, the commanded position with PDIT can completely avoid the influence of unsteady-state response, which is helpful to realize high-speed and high-precision machining in CNC machining field. Even in the experiment, the contour accuracy is improved by 28.163% and 57.35% compared to the other two methods, respectively. If the interpolation period is smaller, the contour accuracy is further improved. Moreover, PDIT only includes basic addition, subtraction, multiplication and division operations, without complex operations such as integral in FOIS. Therefore, PDIT does not tax the CPU and hardly affects the real-time performance of the NURBS interpolator.

### Appendix

#### Appendix 1

In the acc-acceleration segment, the jerk is a positive constant value, and the acceleration increases at a constant value. In the uniform acceleration segment, the value of the is 0, and the acceleration reach a maximum value $A_m$. In the dec-acceleration segment, the jerk is a negative constant value, and the acceleration decreases at a constant value. In the constant speed segment, the values of jerk and acceleration are both 0, and the velocity reach the maximum value $V_m$. In the decCELERATION segment, the jerk is a positive constant value, and the acceleration increases at a constant value until it reaches zero.

$t_1$ represents the critical moment in segments, so $t_1$-$t_7$ are the end moments of the first to seventh segments. $T_1$ represents the duration of each segment, so $T_1 = t_1 - t_0$ is the duration of the first segment, and $T_7 = t_7 - t_0$ is the duration of the seventh segment. Others are the same. $T_1$ represents the time difference between the any moment in one segment and the starting moment of the same segment. For example, in the interval $t_6 < t < t_7$, $T_7 = t - t_6$. The acceleration in each segment is represented by $a(t)$, so $a_1(t)$-$a_7(t)$ are the acceleration of the first to seventh segments. The acceleration at the end of each segment is represented by $a_r$, so $a_1$-$a_7$ are the final accelerations of the first to seventh segments. The speed in each segment is represented by $v(t)$, so $v_1(t)$-$v_7(t)$ are the speed of the first to seventh segments. The speed at the end of each segment is represented by $v_r$, so $v_1$-$v_7$ are the final speeds of the first to seventh segments. The displacement in each segment is represented by $s(t)$, so $s_1(t)$-$s_7(t)$ are the displacement in the first to seventh segments. The displacement at the end of each segment is represented by $s_r$. The total displacement of the entire 7 segments is represented by $S_{total}$.

#### Appendix 2

In model a, after the first three segments, which are AA, UA, and DA, the speed increases from 0 to the maximum speed $V_m$. The speed increment in AA segment is $\Delta v_1$. The speed increments in UA and AD segment are $\Delta v_2$ and $\Delta v_3$, individually. And: $t_1 = T_1$, $t_2 = T_1 + T_2$, $t_3 = T_1 + T_2 + T_3$. $T_1 = T_3 = T$.

$$
\Delta v_1 = \int_0^{T_1} v(t) \, dt = \frac{1}{2} J_m T_1^2
$$

(33)

$$
I_1, I_2:
$$

| Spiral error | Theory | Maximum | Mean | Minimum |
|--------------|--------|---------|------|---------|
| PDIT         | 0.073914066 | 0.018473075 | 1.9999E-05 |
| CEEC         | 0.13807344  | 0.035100988 | 1.6865E-05 |
| FOIS         | 0.077225269 | 0.019549595 | 5.44837E-06 |

Table 9 Datasheet of the contour error of the entire spiral curve

$T$ represents the duration of each segment, so $T_1 = t_1 - t_0$ is the duration of the first segment, and $T_7 = t_7 - t_0$ is the duration of the seventh segment. Others are the same. $T_1$ represents the time difference between the any moment in one segment and the starting moment of the same segment. For example, in the interval $t_6 < t < t_7$, $T_7 = t - t_6$. The acceleration in each segment is represented by $a(t)$, so $a_1(t)$-$a_7(t)$ are the acceleration of the first to seventh segments. The acceleration at the end of each segment is represented by $a_r$, so $a_1$-$a_7$ are the final accelerations of the first to seventh segments. The speed in each segment is represented by $v(t)$, so $v_1(t)$-$v_7(t)$ are the speed of the first to seventh segments. The speed at the end of each segment is represented by $v_r$, so $v_1$-$v_7$ are the final speeds of the first to seventh segments. The displacement in each segment is represented by $s(t)$, so $s_1(t)$-$s_7(t)$ are the displacement in the first to seventh segments. The displacement at the end of each segment is represented by $s_r$. The total displacement of the entire 7 segments is represented by $S_{total}$. 

The total displacement of the entire 7 segments is represented by $S_{total}$.
\[ \Delta v_1 = \int_{t_1}^{t_1 + \Delta t} J_m T_1 dt = J_m t_1 + J_m t_2 = J_m T_1 T_2 \] (34)

\[ t_2 - t_1; \]

\[ a = J_m T_1 - J_m (t - t_2) \] (35)

\[ \Delta v_2 = \int_{t_1}^{t_1 + \Delta t} \frac{1}{2} J_m T_1^2 dt = \frac{1}{2} J_m T_1^2 \] (36)

\[ 0-t_3; \]

\[ \Delta v = \Delta v_1 + \Delta v_2 + \Delta v_3 = \frac{1}{2} J_m T_1^2 + J_m T_1 T_3 + \frac{1}{2} J_m T_3^2 = J_m T_1 T_3 + J_m T_1 T_2 \] (37)

Author contribution Libo Cao: conceptualization, software, validation, formal analysis, data curation, writing—original draft, writing—review and editing. Taiyong Wang: conceptualization, funding acquisition, supervision, resources. Yongbin Zhang: methodology, investigation, review, data curation. Jingchuan Dong: conceptualization. Songhui Jia: methodology, investigation, writing—review and editing. Chong Tian: data curation, investigation. Qingjian Liu: funding acquisition.

Funding This work was supported by the National Natural Science Foundation of China (grant number 51975402) and Tianjin Science and Technology Committee (project number 20YDTPJC01730).

Declarations

Ethics approval Approval.

Consent to participate Consent.

Consent for publication Consent.

Competing interests The authors declare no competing interests.

References

1. Dong JC, Wang TY, Li B, Ding YY (2014) Smooth feedrate planning for continuous short line tool path with contour error constraint. Int J Mach Tools Manuf 76:1–12. https://doi.org/10.1016/j.ijmachtools.2013.09.009
2. Liu M, Huang Y, Yin L, Guo JW, Shao XY, Zhang GI (2014) Development and implementation of a NURBS interpolator with smooth feedrate scheduling for CNC machine tools. Int J Mach Tools Manuf 87:1–15. https://doi.org/10.1016/j.ijmachtools.2014.07.002
3. Liu H, Liu Q, Yuan S (2017) Adaptive feedrate planning on parametric tool path with geometric and kinematic constraints for CNC machining. Int J Adv Manuf Technol 90:1889–1896. https://doi.org/10.1007/s00170-016-9483-6
4. Guo JX, Zhang K, Zhang Q, Gao XS (2013) Efficient time-optimal feedrate planning under dynamic constraints for a high-order CNC servo system. CAD Comput Aided Des 45(12):1538–1546. https://doi.org/10.1016/j.cad.2013.07.002
5. Boehm W, Farin G, Kahmann J (2015) A survey of curve and surface methods in CAGD. Comput Aided Geom Des 11(1):1–60. https://doi.org/10.1016/j.cad.2015.01.004
6. Sang Y, Yao C, Lv Y, He G (2020) An improved feedrate scheduling method for NURBS interpolation in five-axis machining. Precis Eng. https://doi.org/10.1016/j.precisioneng.2020.03.012
7. Wu J (2013) Research on NURBS fitting and interpolation algorithm of programming trajectory for surface NC machining. Dissertation, Huazhong Univ Sci Technol
8. YI ZF (2017) Research on high-speed and high-precision NURBS interpolation algorithm and its interpolator implementation. Dissertation, Heifei Univ Technol
9. Zhang F, Gao S, Cheng X, Zhang W (2017) Innovation research on Taylor’s iterative interpolation algorithm for NURBS curve and simulation. Adv Mater Mach Electron. https://doi.org/10.1063/1.4977370
10. Qin H, Chen Y, Jin X, Yang J (2019) A real-time C3 continuous local corner smoothing and interpolation algorithm for CNC machine tools. J Manuf Sci Eng 141(4):1. https://doi.org/10.1115/1.4042606
11. Shen HY (2010) Research on adaptive NURBS interpolation and implementation. Dissertation, Zhejiang Univ
12. Guo WL (2003) Double arc fitting algorithm and realization of curve equation. Dissertation, Harbin Univ Sci Technol
13. Feng Z, Gao S, Cheng X, Zhang W (2017) Research on an innovative modification algorithm of NURBS curve interpolation. AIP Conf Proc 10(1063/1):4981606
14. Duong QK, Hubinsky P, Paszto P, Florek M, Sovcik J, Kajan M (2014) Effectiveness of input shaping and real-time nurb interpolation for reducing feedrate fluctuation. Procedia Eng 96:81–90. https://doi.org/10.1016/j.proeng.2014.12.101
15. Ji SI, Lei LG, Zhao J, Lu XQ, Gao H (2020) An adaptive real-time NURBS curve interpolation for 4-axis polishing machine tool. Robot Comput Integr Manuf. https://doi.org/10.1016/j.rcim.2020.102025
16. Zhang X, Shi T, Wang Z, Xia C (2018) Generalized predictive contour control of the biaxial motion system. IEEE Trans Indus Electron. https://doi.org/10.1109/TIE.2018.2808899
17. Li X, Liu W, Pan Y, Liang B, Zhou M, Bi H (2018) Monocular vision-based contouring error detection and compensation for CNC machine tools. Precis Eng. https://doi.org/10.1016/j.precisioneng.2018.10.015
18. Les P, Wayne T (2010) Non-uniform rational B-spline. Tsinghua University Press, Beijing
19. Jin YQ (2015) Research on trajectory optimization modeling and forward-looking interpolating technology for high-speed machining of small line segments. Dissertation, Shanghai Jiaotong Univ
20. Shi J (2014) Research on the curvature continuous smoothing and its interpolation method for five-axis machining trajectory. Dissertation, Shanghai Jiaotong Univ
21. Lei WT, Sung MP, Lin LY, Huang JJ (2007) Fast real-time NURBS path interpolation for CNC machine tools. Int J Mach Tools Manuf 47(10):1530–1541. https://doi.org/10.1016/j.ijmachtools.2006.11.011
22. Liu Y, Dai L, Liu J, Wang J (2009) Taylor expansion NURBS curve interpolation algorithm. Dissertation, J Northeast Univ (Nat Sci Ed) 30(1):117–120
23. Yau HT, Lin MT, Tsai MS (2006) Real-time NURBS interpolation using FPGA for high speed motion control. Comput Aided Des 38(10):1123–1133. https://doi.org/10.1016/j.cad.2006.06.005
24. Zhang W, Gao S, Cheng X, Zhang W (2017) Research on an innovative modification algorithm of NURBS curve interpolation. American Institute of Physics Conference Series. AIP Publishing LLC doi 10.1063/1.4981606
25. Zhang YB, Wang TY, Dong JC, Liu YF, Ke RJ (2020) A corner smoothing method with feedrate blending for linear segments under geometric and kinematic constraints. Proc Inst Mech Eng Part B J Eng Manuf 234(5):095440542091133. https://doi.org/10.1177/0954405420911336
26. Dong JC, Wang TY, Ding YY, Li B, Liu Z (2013) Real-time adaptive feed rate planning algorithm for free-form surface NC machining. Manuf Auton 000(011):167–172. https://doi.org/10.3969/j.issn.1009-0134.2013.11.50

27. Zhang LB (2016) Research on the control technology of motion stability and contour accuracy of CNC system. Dissertation, Nanjing Univ Aeronaut Astronaut

28. Zhang L, Du J (2018) Acceleration smoothing algorithm based on jounce limited for corner motion in high-speed machining. Int J Adv Manuf Technol 95(1–4):1487–1504. https://doi.org/10.1007/s00170-017-1272-3

29. Jia ZY, Song DN, Ma JW, Hu GQ, Su WW (2017) A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. Int J Mach Tools Manuf Des Res Appl 116:1–17. https://doi.org/10.1016/j.ijmachtools.2016.12.007

30. Dong JC, Wang TY, Wang ZI, Li B, Ding YY, Jiang YX (2013) General-purpose forward-looking speed planning algorithm. Comput Integr Manuf Syst 019(003):529–539. https://doi.org/10.13196/j.cims.2013.03.83.dongch.005

31. Zhang LX (2010) High-speed CNC machining S-type acceleration/deceleration method of continuous small straight line interpolation algorithm. Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences 29:227–237. https://doi.org/10.1016/B978-0-444-53599-3.10005-8

32. Wang TY, Zhang YB, Dong JC, Ke RJ, Ding YY (2020) NURBS interpolator with adaptive smooth feedrate scheduling and minimal feedrate fluctuation. Int J Precis Eng Manuf 21(2):273–290. https://doi.org/10.1007/s12541-019-00288-6

33. Wang HT (2011) Numerical control system speed forward control algorithm and its realization. Dissertation, Nanjing Univ Aeronaut Astronaut

34. Hu K (2018) Servo system over-quadrant protrusion suppression technology. Dissertation, Harbin Inst Technol

35. Altintas Y, Khoshdarregi MR (2012) Contour error control of CNC machine tools with vibration avoidance. CIRP Ann Manuf Technol 61(1):335–338. https://doi.org/10.1016/j.cirp.2012.03.132

36. Liu J, Wang YL, Jiang RH (2019) Contour control of CNC system based on real-time contour error estimation. Manuf Techn Mach Tool 000(011):168–172. https://doi.org/10.19287/j.cnki.1005-2402.2019.11.036

37. Lu H, Guan SQ, Xiao X, Lei M (2020) Research on a real-time contour error estimation method. Mech Sci Technol Aerosp Eng 39(10):5. https://doi.org/10.13433/cnki.1003-8728.20190308

38. Chen Y, Wang TY, Dong JC, Li B, Yu ZQ (2016) Research on contour accuracy control method of CNC system. Dissertation, Tianjin Univ

39. Chen Y, Wang TY, Dong JC, Li B, Yu ZQ (2016) Contour error control based on contour optimal circle approximation method. Comput Eng Sci 6:1281–1285. https://doi.org/10.3969/j.issn.1007-130X.2016.06.033

40. Shi T, Zhang X, Zhou Z, Xia C (2018) Precise contour control of biaxial motion system based on MPC. IEEE J Emerg Sel Top Power Electron 6:1711–1721. https://doi.org/10.1109/JESTPE.2018.2842238

41. Shi XM, Dai Y, Ma JX (2011) Fault simulation of CNC machine tool feed servo system. Manuf Autom 33(5):4

42. Zhang L (2018) Research on gain scheduling control of ball screw feed system based on time-varying dynamic characteristic. Dissertation, Tianjin Univ

43. Wang WP (2013) Servo system adjustment technology of CNC Lathe. Dissertation, Dalian Univ Technol

44. Jia Z, Song D, Ma JW (2017) Pre-compensation for continuous-path running trajectory error in high-speed machining of parts with varied curvature features. Chinese J Mech Eng 30(01):46–54. https://doi.org/10.3901/CJME.2016.0127.015

45. Chen M, Sun Y, Xu J (2020) A new analytical path-reshaping model and solution algorithm for contour error pre-compensation in multi-axis CNC machining. J Manuf Sci Eng 142(6):1–14. https://doi.org/10.1115/1.4046749

46. Zheng JL, Ying QH, Yang WL (2000) Signals and systems, 2nd edn. Higher Education Press, Beijing

47. Zhang Y, Xiong W, Hang DW (2016) Signal and system. China Water Power Press, Beijing

48. Zeng ZZ (2010) Signal and system, revised. Hunan University Press, Hunan

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

50. Bristow DA, Tharayil M, Alleyne AG (2006) A survey of iterative learning control. IEEE Control Syst 26(3):96–114. https://doi.org/10.1109/MCS.2006.1636313

51. Wang DB (2006) Research on AC servo feed system and its mathematical model. Mach Manuf Autom 1:86–88. https://doi.org/10.3969/j.issn.1671-5276.2006.01.027

52. Zhang W (2009) Research on AC servo system based on fuzzy parameter self-tuning PID control. Dissertation, Harbin Univ Sci Technol

53. Zhang T (2012) Research on the fuzzy PID parameter self-tuning vector control system of AC synchronous servo motor. Dissertation, Hefei Univ Technol

54. Yan J, Sun CG, Zhu XL, Xu J (2003) AC servo system based on parameter self-adjusting fuzzy control. J East China Jiaotong Univ 20(001):20–23. https://doi.org/10.3969/j.issn.1005-0523.2003.01.006

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