论文基本信息表

论文题目	Third-order contour-error estimation for arbitrary free-form paths in contour-following tasks
单位/学校	哈尔滨工程大学
院系、专业及研究方向	机电工程学院
论文指导教师信息	宋得宁 讲师 数字化制造、数控系统技术
论文发表在何期刊	Precision Engineering
期刊索引情况	SCI
影响因子	2.685
论文内容简介	在多轴联动运动控制系统中，由于伺服迟滞、外部扰动等因素，会导致实际运动轨迹与理想轨迹之间产生轮廓误差，因此控制该误差对运动系统轮廓跟踪精度至关重要。轮廓误差的实时估计是对其进行有效控制的前提，估计精度直接影响控制效果。本文针对这一问题，提出一种三阶轮廓误差实时估计方法，实现了对任意自由曲线路径轮廓跟踪过程中轮廓误差的实时高精度估计。
论文创新点	提出了一种用于多轴联动轮廓跟踪控制的三阶轮廓误差实时估计方法，不仅能够在实时性条件下准确估计轮廓误差，还适用于任意自由曲线轨迹。
论文工程应用价值	多轴联动轮廓跟踪任务在工程中常见于数控机床、机械臂等。本文研究的轮廓误差估计方法有利于提高多轴联动轮廓跟踪精度，从而可以在工程上应用于提高数控机床加工精度以及机械臂的操作精度等领域。
灵思创奇设备发挥的科研价值	灵思创奇 RT-CUBE 控制器具有监控各采样周期算法实际执行时间功能，利用这一功能研究了算法计算复杂度，验证了本文提出算法的实时性，极大节约了实验耗时。

Precision Engineering 60 (2019) 85–92

Contents lists available at ScienceDirect

Precision Engineering

journal homepage: www.elsevier.com/locate/precision

Third-order contour-error estimation for arbitrary free-form paths in contour-following tasks

De-Ning Song[a,*], Yu-Guang Zhong[a], Jian-Wei Ma[b]

[a] College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin, 150001, China
[b] Key Laboratory for Precision and Non-traditional Machining Technology of the Ministry of Education, School of Mechanical Engineering, Dalian University of Technology, Dalian, 116024, China

ARTICLE INFO

Keywords:
Contour error
Third-order estimation
Free-form path
Multi-axis motion system

ABSTRACT

The contour error in multi-axis free-form path-following tasks is inevitable due to the existence of factors such as servo lag and external disturbances. Therefore, the control of the contour error is of great significance for improving the precision of multi-axis motion systems. The estimation of the contour error is a premise for its control, and the estimation accuracy should be ensured as high as possible. Existing contour-error estimation methods can be mainly classified into four categories in terms of the first-order method, the second-order method, the iterative method, and their combination methods. Different from them, this paper proposes a third-order contour-error estimation algorithm, so as to improve the estimation accuracy without iterative computation. First, the desired contour is approximated as a third-order arc-length parameterized curve using the Taylor's expansion. Then, the shortest distance from the actual motion position to the approximated contour is solved analytically, thus obtaining the estimated contour error. The proposed estimation algorithm is suitable for arbitrary free-form paths because the analytical equation of the desired contour are not required, but merely kinematic parameters of multi-axis motion systems and the feedback position are utilized. Verification tests illustrate that the proposed method can distinctly improve the estimation accuracy of the three-dimensional contour error, when comparing with typical second-order methods.

1. Introduction

The following of arbitrary free-form path is one of the most important tasks for multi-axis motion systems such as CNC (Computer-Numerical-Control) machine tools and multi degree-of-freedom manipulators. It is of great importance for these multi-axis motion systems to keep well path-following accuracy, because the path-following accuracy affects the motion precision of the end effector directly [1]. However, the contour error during path-following motion is inevitable due to the existence of factors such as the single-axis servo lag, the multi-axis dynamics mismatch, and the external disturbances [2]. Therefore, the control of the contour error in path-following tasks becomes significant.

The contour error is defined as the shortest distance between the actual motion position and the desired path [3]. Doubtlessly, the value of the contour error should be obtained so that the error can be controlled. Although the precise contour errors of linear or circular paths can be easily calculated, it is extremely difficult and time consuming, if possible, to calculate the precise contour errors of arbitrary free-form paths, because the solving of nonlinear equations is required. As a tradeoff, the approximated value of the contour error is commonly adopted in real-time contour control. Therefore, the contour-error estimation plays a key role in contour controllers, and the estimation precision directly affects the control performance [4].

A number of contour-error estimation algorithms have been presented in recent decades. Yeh and Hsu [3] proposed a tangential approximation method for free-form path contour-error estimation. In their method, the contour error was calculated as the perpendicular distance from the actual motion position to the tangential line of the desired contour at the reference point. Cheng and Lee [5] estimated the contour error as the distance from the actual motion position to the secant line passing through the reference point and a tangential-error backstepping point on the desired contour. Yao et al. [6] presented an orthogonal global task coordinate frame (GTCF) for contour error calculation using the analytical equation of the desired path, and it was proofed that their estimation result was exact to the first-order approximation of the actual contour error. Therefore, all of the above methods can be categorized as the first-order method.

* Corresponding author.
E-mail address: songdening@hrbeu.edu.cn (D.-N. Song).

https://doi.org/10.1016/j.precisioneng.2019.07.009
Received 30 April 2019; Received in revised form 17 July 2019; Accepted 30 July 2019
Available online 07 August 2019
0141-6359/

Furthermore, Yang et al. [7] and Chen et al. [8] approximated the desired contour by its osculating circle at the reference position, and the contour error was estimated using the distance from the actual position to the approximated osculating circle. Zhu et al. [9] defined a point-to-curve distance function, and estimated the contour error by applying second-order Taylor expansion to the defined distance function. Song et al. [10] estimated the contour error as the distance from actual motion position to a fitted spatial circle passing through the normal projection point on the desired contour of the actual position. Therefore, these methods can be categorized as the second-order method.

Besides above first-order and second-order contour-error estimation methods, there are also iterative algorithms proposed for further improvement of the estimation accuracy. Yang et al. [11] approximated the desired path by line segments which were formed according to the interpolation points from the interpolator, and the shortest distance from the actual motion position to the line segments, i.e. the estimated contour error, was computed iteratively. Ghaffari and Ulsoy [12] calculated the contour error using a Newton iterative algorithm, and their method can converge to the actual contour error except special cases such as the sharp corner positions. Similarly, Wang et al. [13] presented a Newton algorithm based iterative contour-error estimation method, and it was proofed that their method can achieve second-order convergence.

Note that although there exists other contour-error methods except above three categories in terms of first-order, second-order, and iterative methods, most of other methods can be seen as the generalization or combination of above three categories of methods. For example, Chen et al. [14] proposed several parameter based contour error estimation algorithms by applying two-time first-order or second-order approximations of the desired path; Uchiyama et al. [15] estimated the contour error by using the osculating circle approximation iteratively; Li et al. [16] found the nearest interpolation point from the actual position to the desired contour iteratively, and then computed the contour error by using second-order approximation of the desired path at the found nearest point; Yang et al. [17] approximated the desired path by an arc passing through three points which are searched by iteration; Chen and Sun [18] presented a shifted Frenet frame based contour-error estimation method by two-time first-order approximation.

From above analysis, it is concluded that most of existing contour-error estimation methods can be categorized as following four categories in terms of the first-order, the second-order, the iterative methods and their combination methods. Different from them, this paper presents a third-order contour-error estimation algorithm, in order to improve the estimation accuracy without iterative computation, so that it is suitable for real time contour control due to the high computational efficiency. Additionally, the geometric information such as the curvature of the desired path is not required during the proposed third-order estimation procedure, which indicates that the presented method is not only convenient to implementation, but also universal for arbitrary free-form paths.

Following sections are organized as follows. The third-order approximation based contour-error estimation model is established in Section 2. The solving process of the established estimation model is presented in Section 3. After that, the performance of the presented method is evaluated by comparison with existing methods in Section 4, and finally, conclusions are summarized in Section 5.

2. Contour-error estimation model based on third-order approximation

Most existing direct contour-error estimation methods use first-order and second-order approximation. In this paper, a third-order estimation method is presented to improve the estimation accuracy without iterative computation. The desired arbitrary free-form path to be followed is approximated as a cubic parametric curve using third-order Taylor's expansion, and then, the cost function for calculation of the distance from the actual motion position to the approximated cubic contour is established.

Without loss of generality, the desired path to be followed of the multi-axis motion systems is expressed as $\boldsymbol{r}(s)$, where s stands for the arc length parameter. The arc length parameter corresponding to the reference position is denoted as s_0, thus, the desired path neighboring the reference position can be expanded at s_0 as

$$\boldsymbol{r}(s) = \boldsymbol{r}(s_0) + \boldsymbol{r}'(s_0)\cdot\delta_s + \frac{1}{2}\boldsymbol{r}''(s_0)\cdot\delta_s^2 + \frac{1}{6}\boldsymbol{r}'''(s_0)\cdot\delta_s^3 + o(\delta_s^3) \tag{1}$$

where δ_s is defined as $\delta_s = s\text{-}s_0$, and $o(\delta_s^3)$ represents the higher order infinitesimal.

By ignoring $o(\delta_s^3)$, the desired path neighboring the reference position $\boldsymbol{r}(s_0)$ can thus be approximated as a cubic curve with respect to δ_s and s_0, and the approximated contour is denoted as $\boldsymbol{r}_{ap}(s_0, \delta_s)$

$$\boldsymbol{r}_{ap}(s_0, \delta_s) = \boldsymbol{r}(s_0) + \boldsymbol{r}'(s_0)\cdot\delta_s + \frac{1}{2}\boldsymbol{r}''(s_0)\cdot\delta_s^2 + \frac{1}{6}\boldsymbol{r}'''(s_0)\cdot\delta_s^3 \tag{2}$$

The cost function f_c is defined as

$$f_c(s_0, \delta_s, \boldsymbol{p}) = \boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p} \tag{3}$$

Above, $\boldsymbol{p}$ is the actual motion position corresponding to the reference motion position $\boldsymbol{r}(s_0)$, and $\cdot$ stands for the Euclidean norm. As can be seen from Eq. (3), the physical meaning of the cost function f_c is the distance from the actual position to the approximated contour, and the smallest value of f_c is hence the third-order estimated contour error. For a given reference position and an actual position, the δ_s corresponding to the smallest f_c should be found to obtain the contour error, which is equivalent to solve the following equation

$$\frac{\partial f_c(s_0, \delta_s, \boldsymbol{p})}{\partial \delta_s} = \frac{\frac{\partial \boldsymbol{r}_{ap}(s_0, \delta_s)}{\partial \delta_s}, \boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p}}{\boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p}} = 0 \tag{4}$$

Above, $\cdot,\cdot$ means the operator of inner product of two vectors. When $\boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p} \neq 0$, Eq. (4) can be rewritten as

$$\frac{\partial \boldsymbol{r}_{ap}(s_0, \delta_s)}{\partial \delta_s}, \boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p} = 0 \tag{5}$$

Note that if $\boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p} = 0$, the actual position $\boldsymbol{p}$ is on the approximated desired contour $\boldsymbol{r}_{ap}$, which means that the contour error equals to zero. Therefore, although Eq. (4) cannot be rewritten as Eq. (5) when $\boldsymbol{r}_{ap}(s_0, \delta_s) - \boldsymbol{p} = 0$, the estimated contour error can be obtained directly as zero in this case.

The δ_s corresponding to the nearest point on the approximated contour from the actual position can be obtained by solving Eq. (5). Differentiating Eq. (2) with respect to δ_s, one has

$$\frac{\partial \boldsymbol{r}_{ap}(s_0, \delta_s)}{\partial \delta_s} = \boldsymbol{r}'(s_0) + \boldsymbol{r}''(s_0)\cdot\delta_s + \frac{1}{2}\boldsymbol{r}'''(s_0)\cdot\delta_s^2 \tag{6}$$

Substitute Eqs. (2) and (6) to Eq. (5), one obtains

$$\begin{aligned}(\tfrac{2}{3}\boldsymbol{r}'(s_0)\boldsymbol{r}'''(s_0) + \tfrac{1}{2}\boldsymbol{r}''(s_0)^2)\delta_s^3 + (\tfrac{3}{2}\boldsymbol{r}'(s_0), \boldsymbol{r}''(s_0) + \tfrac{1}{2}\boldsymbol{r}'''(s_0), \boldsymbol{r}(s_0) - \boldsymbol{p})\delta_s^2 \\ + (\boldsymbol{r}'(s_0)^2 + \boldsymbol{r}''(s_0), \boldsymbol{r}(s_0) - \boldsymbol{p})\delta_s + \boldsymbol{r}'(s_0), \boldsymbol{r}(s_0) - \boldsymbol{p} = 0\end{aligned} \tag{7}$$

Obviously, Eq. (7) is a cubic equation with respect to δ_s. Once Eq. (7) is solved, the estimated third-order contour error $\hat{\varepsilon}$ can thus be obtained as

$$\hat{\varepsilon} = \boldsymbol{r}_{ap}(s_0, \delta_{s,f}) - \boldsymbol{p} \tag{8}$$

where $\delta_{s,f}$ is the desired solution of Eq. (7).

The coefficients of Eq. (7) must be obtained in order to solve it. However, the analytical values of $\boldsymbol{r}'(s_0)$, $\boldsymbol{r}''(s_0)$, and $\boldsymbol{r}'''(s_0)$ are difficult to get due to the fact that the arc length parameterization of a free-form curve is not an easy task. In the following section, the solving methods of the coefficients and Eq. (7) without using analytical expressions of

the desired path $\boldsymbol{r}(s)$ are provided, in order to make the presented contour-error estimation method be suitable for arbitrary free-form paths.

3. Solving of the third-order contour-error estimation model

This section solves the exact solution of the presented third-order contour-error estimation model in Section 2 without using analytical equation of the desired path. First, the values of $\boldsymbol{r}'(s_0)$, $\boldsymbol{r}''(s_0)$, and $\boldsymbol{r}'''(s_0)$ in the coefficients of Eq. (7) are computed according to the kinematic parameters of the multi-axis motion system. Then, the third-order equation is solved analytically to obtain the estimated contour error.

It should be noted that the path interpolation must be performed in the interpolator before the contour control in multi-axis motion systems, so as to generate reference positions of all axes at each sampling interval. Therefore, the kinematic parameters in terms of velocity, acceleration, and jerk of each axis can be easily obtained from the interpolator. Taking the three-axis CNC system as an instance, denote the axial velocity, acceleration, and jerk vectors as $\boldsymbol{v} = [v_x, v_y, v_z]^{\mathrm{T}}$, $\boldsymbol{a} = [a_x, a_y, a_z]^{\mathrm{T}}$, and $\boldsymbol{j} = [j_x, j_y, j_z]^{\mathrm{T}}$, respectively, and they can be expressed as

$$\boldsymbol{v} = \frac{d\boldsymbol{r}(s)}{dt} = \frac{d\boldsymbol{r}(s)}{ds}\cdot\frac{ds}{dt} = \boldsymbol{r}'(s)\cdot v_p \tag{9}$$

$$\boldsymbol{a} = \frac{d^2\boldsymbol{r}(s)}{dt^2} = \frac{d^2\boldsymbol{r}(s)}{ds^2}\cdot\left(\frac{ds}{dt}\right)^2 + \frac{d\boldsymbol{r}(s)}{ds}\cdot\frac{d^2s}{dt^2} = \boldsymbol{r}''(s)\cdot v_p^2 + \boldsymbol{r}'(s)\cdot a_p \tag{10}$$

$$\begin{aligned} j &= \frac{d^3\boldsymbol{r}(s)}{dt^3} = \frac{d^3\boldsymbol{r}(s)}{ds^3}\cdot\left(\frac{ds}{dt}\right)^3 + 3\frac{d^2\boldsymbol{r}(s)}{ds^2}\cdot\frac{d^2s}{dt^2}\cdot\frac{ds}{dt} + \frac{d\boldsymbol{r}(s)}{ds}\cdot\frac{d^3s}{dt^3} \\ &= \boldsymbol{r}'''(s)\cdot v_p^3 + 3\boldsymbol{r}''(s)\cdot a_p\cdot v_p + \boldsymbol{r}'(s)\cdot j_p \end{aligned} \tag{11}$$

Above, v_p, a_p, and j_p are the first-order, second-order, and third-order derivatives of the arc length s with respect to the time t, respectively, i.e. the path velocity, acceleration, and jerk, respectively, and they all can be obtained from the interpolator. According to Eq. (9), the value of $\boldsymbol{r}'(s_0)$ can be calculated using the current axial velocity $\boldsymbol{v}_0$ and the current path velocity $v_{p,0}$ as

$$\boldsymbol{r}'(s_0) = \frac{\boldsymbol{v}_0}{v_{p,0}} \tag{12}$$

Substituting Eq. (12) to Eq. (10), the value of $\boldsymbol{r}''(s_0)$ can thus be obtained according to the current axial velocity $\boldsymbol{v}_0$, acceleration $\boldsymbol{a}_0$, path velocity $v_{p,0}$ and path acceleration $a_{p,0}$ as

$$\boldsymbol{r}''(s_0) = \frac{v_{p,0}\boldsymbol{a}_0 - a_{p,0}\boldsymbol{v}_0}{v_{p,0}^3} \tag{13}$$

Substituting Eqs. (12) and (13) to Eq. (11), the value of $\boldsymbol{r}'''(s_0)$ is computed by the current axial velocity $\boldsymbol{v}_0$, axial acceleration $\boldsymbol{a}_0$, axial jerk $\boldsymbol{j}_0$, path velocity $v_{p,0}$, path acceleration $a_{p,0}$, and path jerk $j_{p,0}$ as

$$\boldsymbol{r}'''(s_0) = \frac{v_{p,0}^2\boldsymbol{j}_0 - 3v_{p,0}a_{p,0}\boldsymbol{a}_0 + 3a_{p,0}^2\boldsymbol{v}_0 - v_{p,0}j_{p,0}\boldsymbol{v}_0}{v_{p,0}^5} \tag{14}$$

Hence, the unknown variables in the coefficients of Eq. (7) are obtained. Eq. (7) can thus be rewritten by substituting Eqs. (12-14) to it as

$$c_1\delta_s^3 + c_2\delta_s^2 + c_3\delta_s + c_4 = 0 \tag{15}$$

where the coefficients are

$$c_1 = \frac{4v_{p,0}^2\langle\boldsymbol{v}_0, \boldsymbol{j}_0\rangle - 18v_{p,0}a_{p,0}\langle\boldsymbol{v}_0, \boldsymbol{a}_0\rangle + 15a_{p,0}^2\boldsymbol{v}_0^2 - 4v_{p,0}j_{p,0}\boldsymbol{v}_0^2 + 3v_{p,0}^2\boldsymbol{a}_0^2}{6v_{p,0}^6} \tag{16}$$

$$c_2 = \frac{\begin{pmatrix} 3v_{p,0}^2\langle\boldsymbol{v}_0, \boldsymbol{a}_0\rangle - 3v_{p,0}a_{p,0}\boldsymbol{v}_0^2 + v_{p,0}^2\boldsymbol{j}_0, \boldsymbol{r}(s_0) - \boldsymbol{p} \\ -3v_{p,0}a_{p,0}\langle\boldsymbol{a}_0, \boldsymbol{r}(s_0) - \boldsymbol{p}\rangle + (3a_{p,0}^2 - v_{p,0}j_{p,0})\langle\boldsymbol{v}_0, \boldsymbol{r}(s_0) - \boldsymbol{p}\rangle \end{pmatrix}}{2v_{p,0}^5} \tag{17}$$

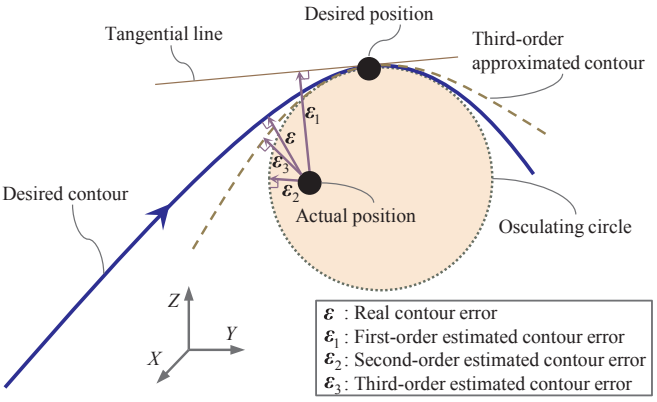

Fig. 1. Comparison of the first-order, second-order, and the proposed third-order contour-error estimation methods.

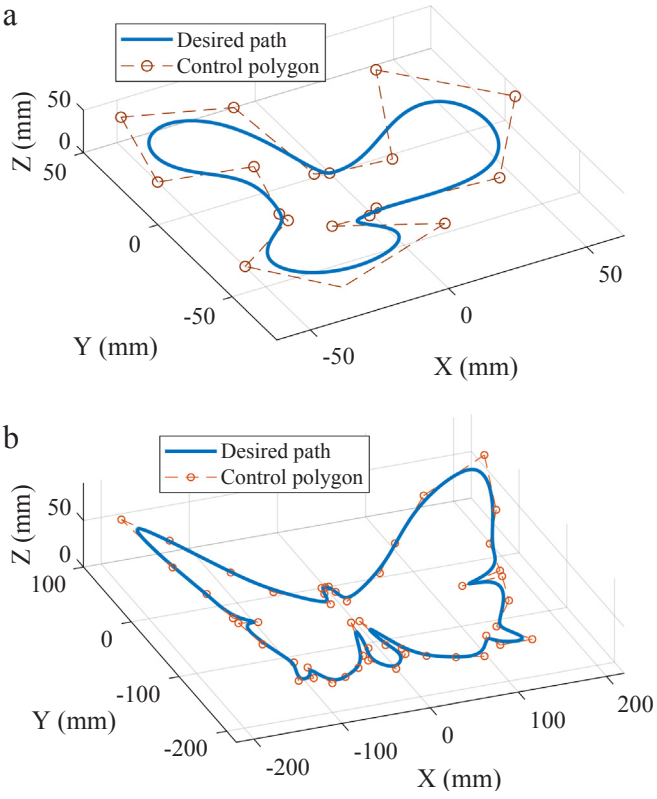

Fig. 2. Desired contour geometries. (a) Impeller contour. (b) Butterfly contour.

$$c_3 = \frac{v_{p,0}\boldsymbol{v}_0^2 + v_{p,0}\langle\boldsymbol{a}_0, \boldsymbol{r}(s_0) - \boldsymbol{p}\rangle - a_{p,0}\langle\boldsymbol{v}_0, \boldsymbol{r}(s_0) - \boldsymbol{p}\rangle}{v_{p,0}^3} \tag{18}$$

$$c_4 = \frac{\langle\boldsymbol{v}_0, \boldsymbol{r}(s_0) - \boldsymbol{p}\rangle}{v_{p,0}} \tag{19}$$

It can be seen from Eqs. (16-19) that all of the coefficients of Eq. (15) are known values. Therefore, the cubic Eq. (15) can be solved. Note that although the coefficients are calculated according to the kinematics parameters in terms of velocity, acceleration, and jerk, the results do not change with the variation of the specific kinematics parameters generated by different interpolator. This is because the coefficients are related to the fixed ratio of the kinematics parameters instead of the specific values of them, which is proofed below.

Denote the motion time of a piece of path s_p under the path velocity $v_{p,1}$ as $t_{s,1}$. Once the path velocity $v_{p,1}$ is generated as $v_{p,2}$, the motion time of s_p becomes $t_{s,2}$

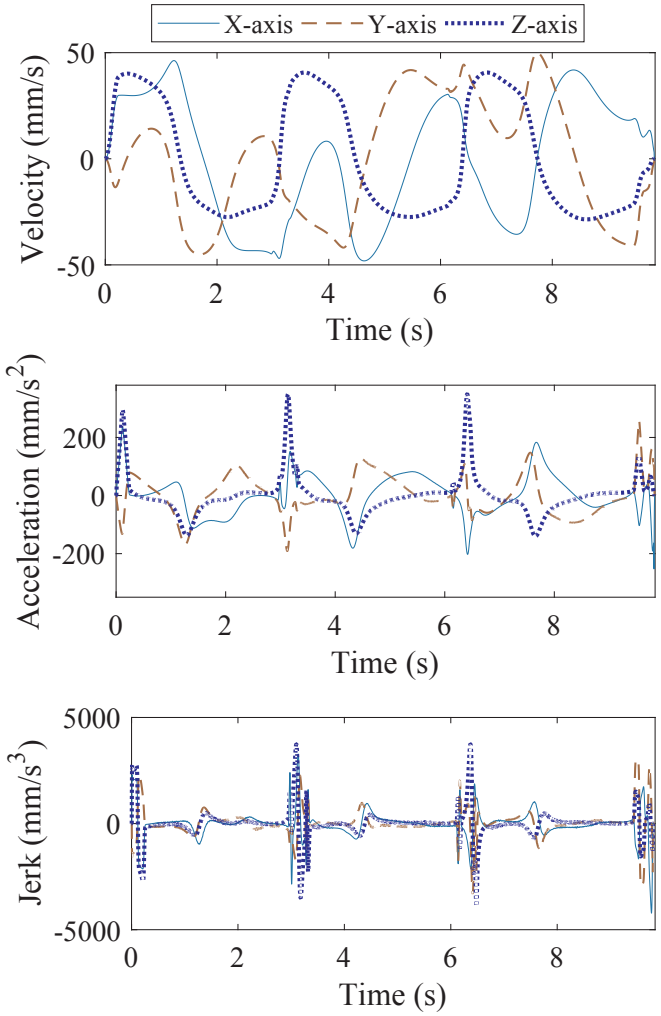

Fig. 3. Kinematic parameters of the impeller contour path.

$$t_{s,2} = \frac{s_p}{v_{s,2}} = \left(\frac{v_{s,1}}{v_{s,2}}\right)\cdot\frac{s_p}{v_{s,1}} \tag{20}$$

Hence, the path acceleration $a_{p,1} = \frac{v_{p,1}}{t_{s,1}}$ will be changed to

$$a_{p,2} = \frac{v_{p,2}}{t_{s,2}} = \frac{v_{p,2}}{v_{p,1}}\cdot\frac{v_{p,1}}{t_{s,2}} = \left(\frac{v_{p,2}}{v_{p,1}}\right)^2\cdot\frac{v_{p,1}}{t_{s,1}} = \left(\frac{v_{p,2}}{v_{p,1}}\right)^2\cdot a_{p,1} \tag{21}$$

Additionally, the path jerk $j_{p,1} = \frac{a_{p,1}}{t_{s,1}}$ will be changed to

$$j_{p,2} = \frac{a_{p,2}}{t_{s,2}} = \frac{a_{p,2}}{a_{p,1}}\cdot\frac{a_{p,1}}{t_{s,2}} = \left(\frac{v_{p,2}}{v_{p,1}}\right)^2\cdot\frac{a_{p,1}}{t_{s,2}} = \left(\frac{v_{p,2}}{v_{p,1}}\right)^3\cdot\frac{a_{p,1}}{t_{s,1}} = \left(\frac{v_{p,2}}{v_{p,1}}\right)^3\cdot j_{p,1} \tag{22}$$

Denote the axial velocity, acceleration, and jerk corresponding to $v_{p,1}$, $a_{p,1}$, and $j_{p,1}$ as $\boldsymbol{v}_1$, $\boldsymbol{a}_1$, and $\boldsymbol{j}_1$, respectively. According to Eqs. (9-11), once $v_{p,1}$, $a_{p,1}$, and $j_{p,1}$ are changed to $v_{p,2}$, $a_{p,2}$, and $j_{p,2}$, respectively, $\boldsymbol{v}_1$, $\boldsymbol{a}_1$, and $\boldsymbol{j}_1$ will change as $\boldsymbol{v}_2$, $\boldsymbol{a}_2$, and $\boldsymbol{j}_2$, respectively, and they are computed as

$$\begin{cases} \boldsymbol{v}_2 = \left(\frac{v_{p,2}}{v_{p,1}}\right)\cdot v_{p,1} \\ \boldsymbol{a}_2 = \left(\frac{v_{p,2}}{v_{p,1}}\right)^2\cdot a_{p,1} \\ \boldsymbol{j}_2 = \left(\frac{v_{p,2}}{v_{p,1}}\right)^3\cdot j_{p,1} \end{cases} \tag{23}$$

By substituting Eqs. (21-23) to Eqs. (16-19), it is easy to find that the coefficients c_1, c_2, c_3, and c_4 do not change when kinematics parameters $v_{p,1}$, $a_{p,1}$, $j_{p,1}$, $\boldsymbol{v}_1$, $\boldsymbol{a}_1$, and $\boldsymbol{j}_1$ are altered to $v_{p,2}$, $a_{p,2}$, $j_{p,2}$, $\boldsymbol{v}_2$, $\boldsymbol{a}_2$, and $\boldsymbol{j}_2$. This indicates that the coefficients can reflect the invariable geometry of the toolpath, although the kinematics may be different when using different interpolator.

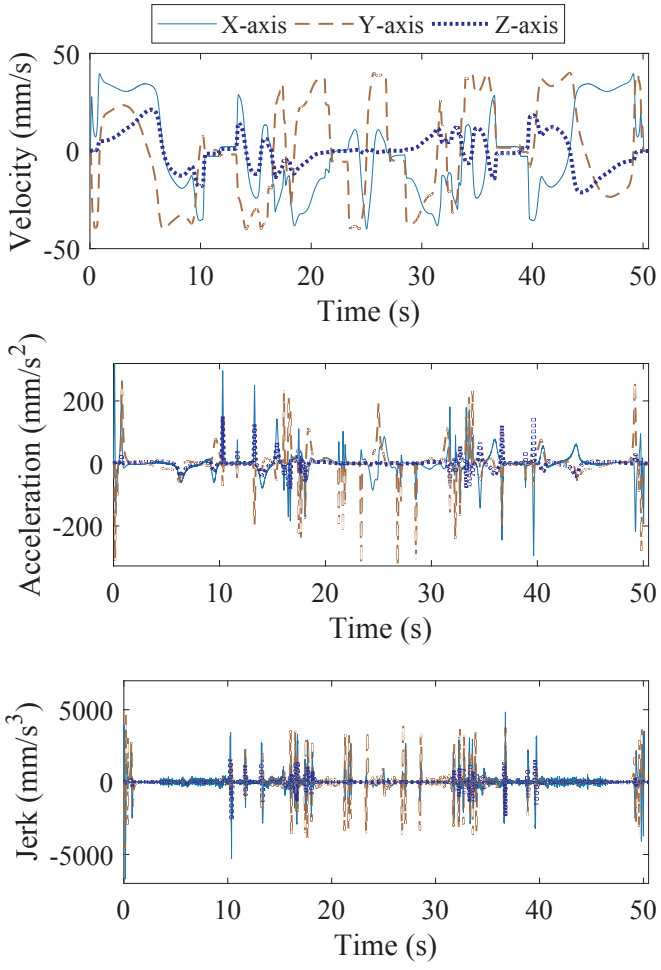

Fig. 4. Kinematic parameters of the butterfly contour path.

In addition, it is seen from Eqs. (16-19) that the path velocity v_p is performed as the denominator during the coefficients calculation, therefore, these equations become singularity once v_p equals to zero which will occur when encountering sharp corners. In this case, the actual contour error will be zero because an instant stop is executed at this kind of positions. Hence, the contour error is directly estimated as zero when $v_p = 0$. Unless $v_p = 0$, there are three roots of an arbitrary cubic equation. The three roots of Eq. (15) are calculated in this paper according to the Shengjin's formula proposed by Fan [19] and utilized by Liu [20], and the detail procedure is introduced below.

First, establish the discriminants

$$\Delta = B^2 - 4AC \tag{24}$$

where

$$\begin{cases} A = c_2^2 - 3c_1c_3 \\ B = c_2c_3 - 9c_1c_4 \\ C = c_3^2 - 3c_2c_4 \end{cases} \tag{25}$$

If $A = B = 0$, Eq. (15) has three equal real roots and they are

$$\delta_{s,1} = \delta_{s,2} = \delta_{s,3} = \frac{-c_2}{3c_1} \tag{26}$$

In this case, the desired solution of Eq. (7) is

$$\delta_{s,f} = \frac{-c_2}{3c_1} \tag{27}$$

Otherwise, the solving method of Eq. (15) can be classified into three cases according to the sign of the discriminants Δ.

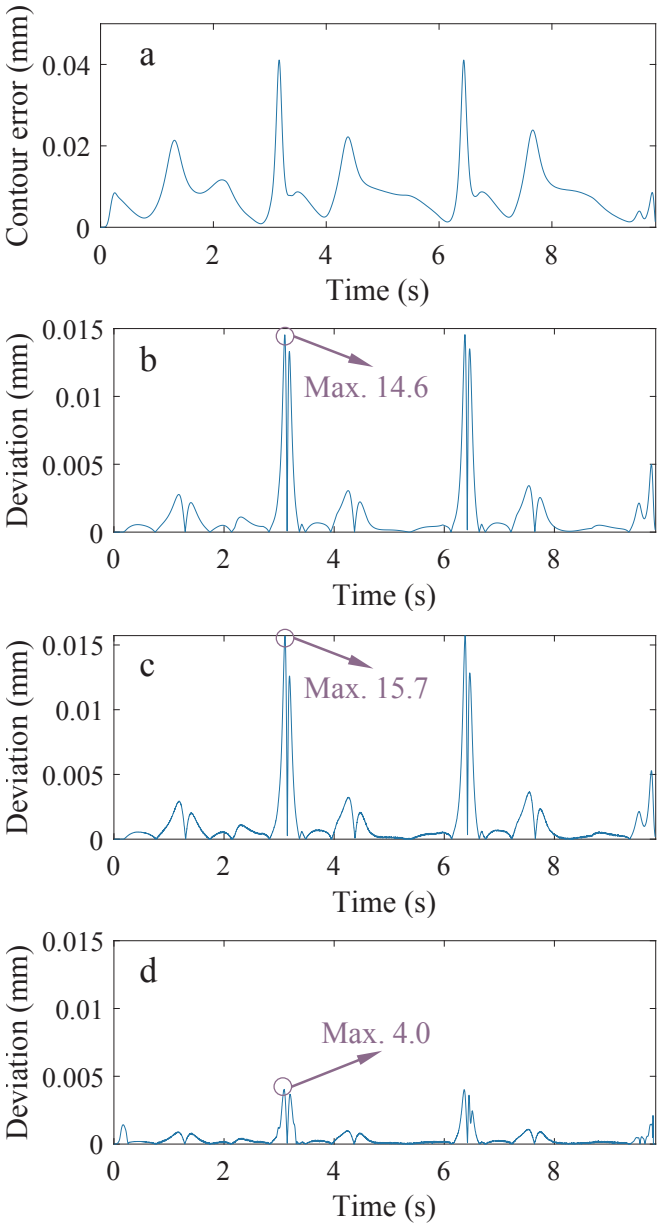

Fig. 5. Real contour error and its estimation deviations of different methods for the impeller contour. (a) Real contour error. (b) Estimation deviation of Chen's osculating method [8]. (c) Estimation deviation of Zhu's second-order method [9]. (d) Estimation deviation of the proposed third-order method.

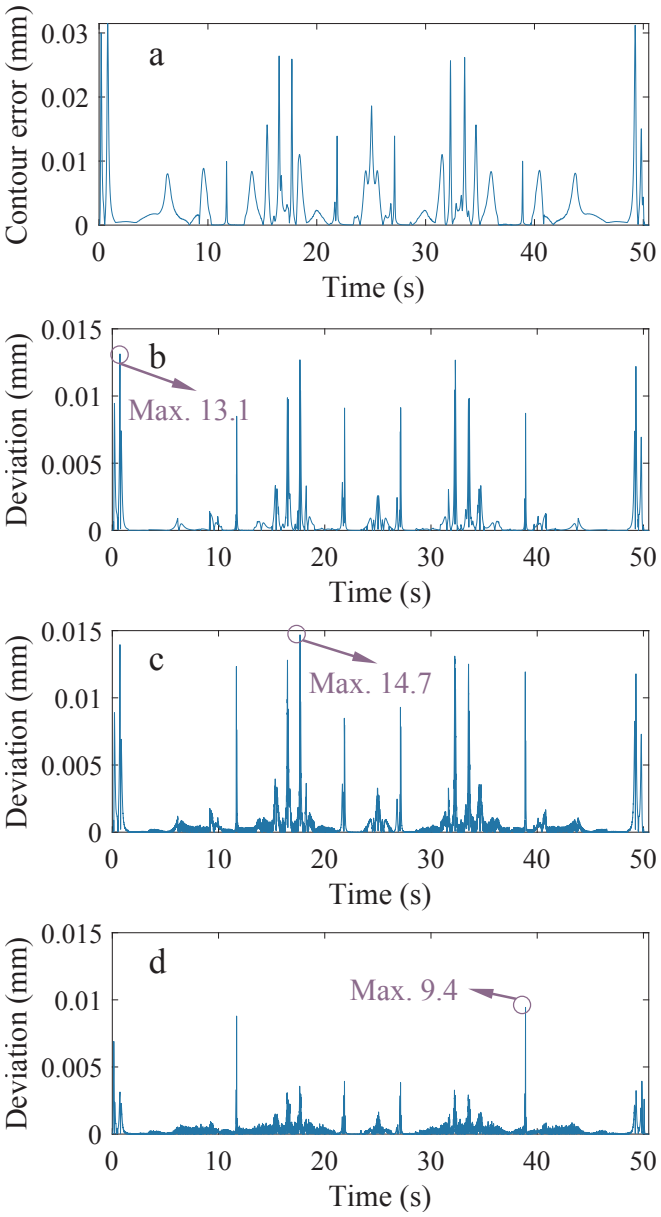

Fig. 6. Real contour error and its estimation deviations of different methods for the butterfly contour. (a) Real contour error. (b) Estimation deviation of Chen's osculating method [8]. (c) Estimation deviation of Zhu's second-order method [9]. (d) Estimation deviation of the proposed third-order method.

1) If $\Delta > 0$, there are two complex conjugate roots and one real root for Eq. (15), thus the single real root must be the desired solution $\delta_{s,f}$, and it is computed as

$$\delta_{s,f} = \frac{-c_2 - \left(\sqrt[3]{Ac_2 + 3c_1\left(\frac{-B+\sqrt{\Delta}}{2}\right)} + \sqrt[3]{Ac_2 + 3c_1\left(\frac{-B-\sqrt{\Delta}}{2}\right)}\right)}{3c_1} \tag{28}$$

2) If $\Delta = 0$, there are two equal real roots and another one different real root for Eq. (15), and they are

$$\begin{aligned} \delta_{s,1} &= \frac{-c_2}{c_1} + \frac{B}{A} \\ \delta_{s,2} &= \delta_{s,3} = \frac{-B}{2A} \end{aligned} \tag{29}$$

In this case, the root closest to zero is selected as the desired solution $\delta_{s,f}$.

Table 1
Performance comparison of contour-error estimation methods for the impeller contour.

Estimation methods	Deviation indexes (μm)		
	MAX	IAE	RMS
Chen's osculating circular method [8]	14.6	5996.8	2.5
Zhu's second-order method [9]	15.7	5998.3	2.6
Proposed third-order method	4.0	1932.5	0.7

$$\delta_{s,f} = \delta_{s,i},\ i = 1\ or\ 2,\ |\delta_{s,i}| = \min\{|\delta_{s,1}|, |\delta_{s,2}|\} \tag{30}$$

3) If $\Delta < 0$, there are three different real roots for Eq. (15) and they are

Table 2
Performance comparison of contour-error estimation methods for the butterfly contour.

Estimation methods	Deviation indexes (μm)		
	MAX	IAE	RMS
Chen's osculating circular method [8]	13.1	14016.1	1.5
Zhu's second-order method [9]	14.7	15186.4	1.5
Proposed third-order method	9.4	6615.3	0.6

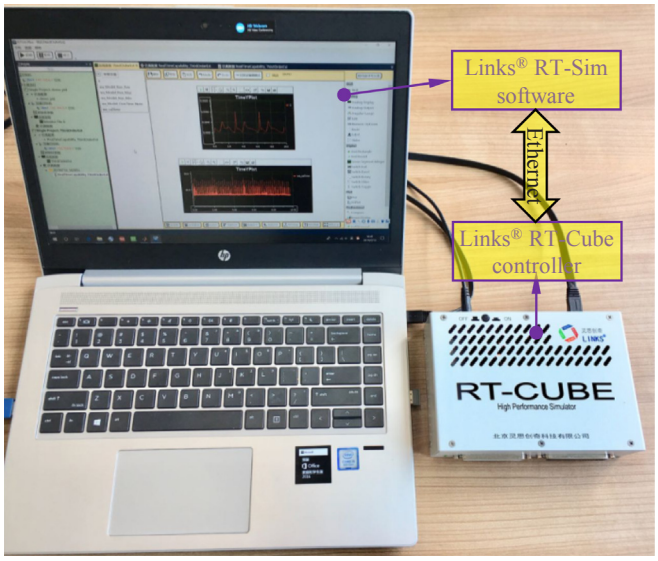

Fig. 7. Experimental setup for real-time capability tests.

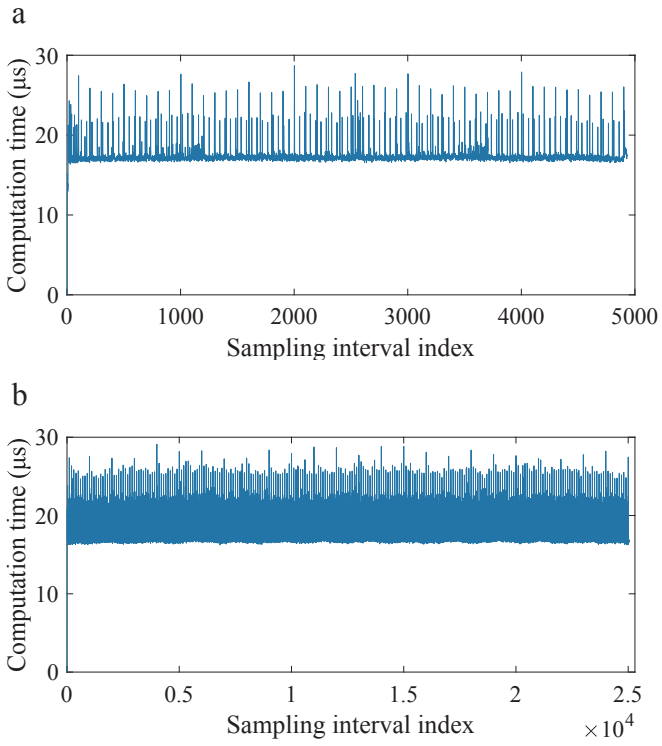

Fig. 8. Computation time of the proposed third-order contour-error estimation method. (a) Computation time for the impeller-path contour-error estimation. (b) Computation time for the butterfly-path contour-error estimation.

$$\delta_{s,1} = \frac{-c_2 - 2\sqrt{A}\cos\frac{\theta}{3}}{3c_1}$$
$$\delta_{s,2,3} = \frac{-c_2 + \sqrt{A}\left(\cos\frac{\theta}{3} \pm \sqrt{3}\sin\frac{\theta}{3}\right)}{3c_1} \tag{31}$$

in which

$$\theta = \arccos\frac{2Ac_2 - 3c_1B}{2A\sqrt{A}} \tag{32}$$

It is worth noting that there always exists $A \geq 0$ in this case, so that the roots $\delta_{s,1}$, $\delta_{s,2}$, and $\delta_{s,3}$ are all real roots. This is proofed by contradiction method as follows.

Assume that there exists $A < 0$, i.e. $c_2^2 - 3c_1c_3 < 0$ when $\Delta < 0$. Denote the third-order function as $f_3(\delta_s) = c_1\delta_s^3 + c_2\delta_s^2 + c_3\delta_s + c_4$, and its first-order derivative can be obtained as $f_3'(\delta_s) = 3c_1\delta_s^3 + 2c_2\delta_s + c_3$. When $c_2^2 - 3c_1c_3 < 0$, there exists no real root for the function $f_3'(\delta_s) = 0$. Therefore, $f_3'(\delta_s) > 0$ is satisfied when $c_1 > 0$ and $f_3'(\delta_s) < 0$ is satisfied when $c_1 < 0$. Thus, $f_3(\delta_s)$ monotonically increases on $(-\infty, +\infty)$ when $c_1 > 0$ and monotonically decreases when $c_1 < 0$ on $(-\infty, +\infty)$, which indicates that there exists only one real roots for the third-order function $f_3(\delta_s) = 0$. Hence, it can be deduced that the discriminants $\Delta > 0$ according to the Shengjin's theorem [19], and this is contradict with the premise of the case $\Delta < 0$. Therefore, we have $A \geq 0$ when $\Delta < 0$.

In this case, similarly, the root closest to zero is selected as the desired solution $\delta_{s,f}$.

$$\delta_{s,f} = \delta_{s,i},\ i = 1\ or\ 2\ or\ 3,\ |\delta_{s,i}| = \min\{|\delta_{s,1}|, |\delta_{s,2}|, |\delta_{s,3}|\} \tag{33}$$

The estimated contour error $\hat{\varepsilon}$ can thus be calculated by substituting the solution $\delta_{s,f}$ to Eq. (8). Fig. 1 shows the difference of the proposed third-order contour-error estimation method with existing first-order and second-order methods. It is acceptable that the third-order approximated contour of the desired curved path is more precise than the lower-order approximated ones, therefore, the contour-error estimation accuracy can be improved effectively. When comparing with iterative methods, the estimation accuracy of the proposed method may be lower, but the advantage is that the iterative computation which consumes uncertain calculation time is not required in the proposed third-order contour-error estimation algorithm.

The integral procedure of the proposed third-order contour-error estimation is summarized as follows. First, read the reference position and kinematic parameters in terms of the axial and path velocities, accelerations, and jerks from the interpolator, and read the actual motion position from the feedback sensors of the multi-axis motion system. Second, calculate the coefficients c_1, c_2, c_3, and c_4 according to Eqs. (16)–(19), respectively. Third, solve the solution $\delta_{s,f}$ according to Eqs. 27–33. Fourth, calculate the estimated contour error $\hat{\varepsilon}$ by substituting the solved $\delta_{s,f}$ to Eq. (8). As can be seen from above procedure, the analytical formula of the desired path is not required, which indicates that the presented method can be used for any arbitrary free-form path.

4. Verification tests

This section conducts verification tests so as to evaluate the performance of the presented contour-error estimation method. Note that the contour error belongs to motion error who has no relationship with the dimensional accuracy of the multi-axis motion system, and the contour error estimation is a pure mathematical computation issue once the desired contour and actual position are obtained, therefore, a real motion system is not necessary for verification tests. A three-axis motion system model is established in Matlab/Simulink environment, so as to generate the actual motion positions corresponding to the desired contour. The close-loop transfer functions of the three axes are set as

$$G_x = G_y = G_z = \frac{37}{0.01s^2 + s + 37} \tag{34}$$

Two three-dimensional desired contours in terms of an impeller contour and a butterfly contour, expressed by NURBS (non-uniform rational B-spline), are adopted for the verification tests, and their geometries are shown in Fig. 2. The control points of the two spline contours are shown in Fig. 2, and the rest of NURBS parameters of the impeller contour shown in Fig. 2(a) are:

Order: 5;

Knot vector: [0, 0, 0, 0, 0, 0.216, 0.280, 0.323, 0.359, 0.418, 0.485, 0.549, 0.613, 0.656, 0.692, 0.751, 0.818, 0.883, 0.947, 1, 1, 1, 1, 1];

Weights: [1, …, 1].

The rest of NURBS parameters of the butterfly contour shown in Fig. 2(b) are:

Order: 4;

Knot vector: [0, 0, 0, 0, 0.0083, 0.015, 0.0361, 0.08550, 0.1293, 0.1509, 0.1931, 0.2273, 0.2435, 0.2561, 0.2692, 0.2889, 0.317, 0.3316, 0.3482, 0.3553, 0.3649, 0.3837, 0.4005, 0.4269, 0.451, 0.466, 0.4891, 0.5, 0.5109, 0.534, 0.54890, 0.57310, 0.5994, 0.6163, 0.6351, 0.6447, 0.6518, 0.6683, 0.683, 0.7111, 0.7307, 0.7439, 0.7565, 0.77290, 0.8069, 0.8491, 0.8707, 0.9145, 0.9639, 0.985, 0.9917, 1, 1, 1, 1];

Weights: [1, 1, 1, 1.2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 5, 3, 1, 1.1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1.1, 1, 3, 5, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1.2, 1, 1, 1].

As can be seen, the two testing contours are all curved paths with both large curvature and large curvature variation. A NURBS interpolator [21] is employed to generate the reference interpolation point coordinates of the desired contours at each sampling interval. The generated interpolation point coordinates are utilized as input reference commands of the established three-axis motion system model. By sampling of the outputs of the motion system, actual motion positions corresponding to the desired contours can be obtained. Thus the desired and actual contours are all known. The maximum feedrate of the impeller contour path is set as 50 mm/s, while that for the butterfly contour path is 40 mm/s. The kinematic parameters in terms of the axial velocity, acceleration, and jerk that should be used by the proposed contour-error estimation method are obtained from the interpolator, and they are shown in Fig. 3 and Fig. 4.

Two typical second-order contour-error estimation methods, i.e. Chen's osculating approximation method [8] and Zhu's second-order method [9], are token for comparison with the presented third-order method. First, the real contour errors of the two testing contours are computed according to the mathematical definition of the contour error. Then, the above three kinds of methods are utilized to compute the estimated contour errors respectively. The real contour errors of the impeller contour and their deviations with the estimated contour errors of different methods are illustrated in Fig. 5. Similarly, the real contour errors and estimation deviations of the more complicated butterfly contour are illustrated in Fig. 6.

The testing results of the impeller and butterfly contours are further compared in Table 1 and Table 2, respectively. In Tables 1 and 2, the index MAX means the maximum estimation deviation; the index IAE means the absolute integral value of the estimation deviations; the index RMS means the root mean square value of the estimation deviations. It can be concluded from Figs. 5 and 6 and Tables 1–2 that the estimation deviations of Chen's and Zhu's methods are nearly the same because they are all second-order methods, and the proposed third-order estimation method can distinctly improve the estimation accuracy. Note that the contour-error estimation accuracy can be further improved if the first-order and second-order methods in existing iterative or multi-time contour-error estimation methods, such as the methods in Ref. [14], are replaced by the proposed third-order estimation method.

In order to further evaluate the real-time capability of the proposed third-order contour-error estimation method, above estimation procedures are executed in a real-time rapid-control-prototype controller RT-Cube shown in Fig. 7. The RT-Cube controller is produced by Beijing Links® Corporation, and it is embedded with a 866 MHz dual-core ARM Cortex-A9 processor. By using the affiliated Links® RT-Sim software, the code of the proposed estimation model is downloaded into the RT-Cube controller, and the computation time of the estimation model in each sampling interval is monitored and saved into the host PC through the RT-Sim software. The computation time for estimation of the contour errors of the impeller and butterfly contours is illustrated in Fig. 8. It is seen from Fig. 8 that the maximum contour-error estimation time of the two tests are all shorter than 30 μs, and this is much less than the commonly used interpolation period and sampling interval of multi-axis motion systems which are always larger than 1 ms. As a result, it can be concluded that the proposed third-order contour-error estimation method processes not only high estimation accuracy, but also high computational efficiency.

5. Conclusion

In this paper, a third-order contour-error estimation method is presented for accurate estimation of the contour error of arbitrary free-form paths in multi-axis contour-following tasks. The desired contour is approximated by an arc length parameterized cubic curve using third-order Taylor's expansion first, and then, the analytical solving method of the distance from the actual motion position to the cubic curve is provided. The whole procedure uses merely the feedback positions and the axial kinematic parameters of the motion systems which can be obtained easily from the interpolator. The analytical expression of the desired contour and the axial amounts and configurations are not required, which indicates that the presented method can be used for arbitrary free-form path and most serial multi-axis motion systems. Verification tests are conducted to evaluate the performance of the presented approach. It is seen from the test results that the estimation deviation of the three-dimensional contour error using the proposed third-order estimation method is less than 10 μm which is much smaller than that using the typical second-order methods, and the computation time of the estimation method is shorter than 30 μs, which indicates a high computational efficiency. It is also noted that the estimation accuracy can be further improved if the proposed third-order contour-error estimation algorithm is combined with iterative computation methods. Contribution of this study is significant for improvement of the contour-error estimation precision in real time, thus enhancing the contour control performance. In future works, the proposed contour-error estimation method will be employed for design of novel contour controllers for multi-axis motion systems.

Acknowledgements

This research is supported by the Fundamental Research Funds for the Central Universities, China (3072019CFJ0701).

References

[1] Huo F, Poo AN. Precision contouring control of machine tools. Int J Adv Manuf Technol 2013;64(1–4):319–33.
[2] Lie T, Robert GL. Multiaxis contour control—the state of the art. IEEE Trans Control Syst Technol 2013;21(6):1997–2010.
[3] Yeh SS, Hsu PL. Estimation of the contouring error vector for the cross-coupled control design. IEEE/ASME Trans Mechatronics 2002;7(1):44–51.
[4] Jia ZY, Song DN, Ma JW, Qin FZ, Wang XL. High-precision estimation and double-loop compensation of contouringerrors in five-axis dual-NURBS toolpath following tasks. Precis Eng 2018;54. 243-53.
[5] Cheng MY, Lee CC. Motion controller design for contour-following tasks based on real-time contour error estimation. IEEE Trans Ind Electron 2007;54(3). 1686-95.
[6] Yao B, Hu C, Wang Q. An orthogonal global task coordinate frame for contouring control of biaxial systems. IEEE/ASME Trans Mechatronics 2012;17(4). 622-34.
[7] Yang J, Li Z. A novel contour error estimation for position loop-based cross-coupled control. IEEE/ASME Trans Mechatronics 2011;16(4). 643-55.
[8] Chen M, Sun Y. A moving knot sequence-based feedrate scheduling method of parametric interpolator for CNC machining with contour error and drive constraints. Int J Adv Manuf Technol 2018;98:487–504.
[9] Zhu L, Zhao H, Ding H. Real-time contouring error estimation for multi-axis motion

systems using the second-order approximation. Int J Mach Tool Manuf 2013;68:75–80.
[10] Song DN, Ma JW, Jia ZY, Qin FZ, Zhao XX. Synergistic real-time compensation of tracking and contouring errors for precise parametric curved contour following systems. Proc IMechE Part C: J Mech Eng Sci 2018;232(19). 3367-83.
[11] Yang J, Ding H, Zhao H, Yan S. A generalized online estimation algorithm of multi-axis contouring errors for CNC machine tools with rotary axes. Int J Adv Manuf Technol 2016;84(5–8):1239–51.
[12] Ghaffari A, Ulsoy AG. Dynamic contour error estimation and feedback modification for high-precision contouring. IEEE/ASME Trans Mechatronics 2016;21(3). 1732-41.
[13] Wang Z, Hu C, Zhu Y, He S, Mu H. Newton-ILC contouring error estimation and coordinated motion control for precision multiaxis systems with comparative experiments. IEEE Trans Ind Electron 2018;65(2). 1470-80.
[14] Chen HR, Cheng MY, Wu CH, Su KH. Real time parameter based contour error estimation algorithms for free form contour following. Int J Mach Tool Manuf 2016;102:1–8.
[15] Khalick MAE, Uchiyama N. Contouring controller design based on iterative contour error estimation for three-dimensional machining. Robot Comput Integr Manuf 2011;27:802–7.
[16] Li X, Zhao H, Zhao X, Ding H. Dual sliding mode contouring control with high accuracy contour error estimation for five-axis CNC machine tools. Int J Mach Tool Manuf 2016;108:74–82.
[17] Yang M, Yang J, Ding H. A high accuracy on-line estimation algorithm of five-axis contouring errors based on three-point arc approximation. Int J Mach Tool Manuf 2018;130–131:73–84.
[18] Chen M, Sun Y. Contour error–bounded parametric interpolator with minimum feedrate fluctuation for five-axis CNC machine tools. Int J Adv Manuf Technol 2019;103:567–84.
[19] Fan S. A new extracting formula and a new distinguishing means on the one variable cubic equation. Nat Sci J Hainan Teach Coll 1989;2(2). 91-8.
[20] Liu H, Liu Q, Zhou S, Li C, Yuan S. A NURBS interpolation method with minimal feedrate fluctuation for CNC machine tools. Int J Adv Manuf Technol 2015;78(5–8):1241–50.
[21] Jia ZY, Song DN, Ma JW, Hu GQ, Su WW. A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. Int J Mach Tool Manuf 2017;116:1–17.