Inverse Kinematics of a 5-Axis Hybrid Robot with Non-singular Tool Path Generation

Qi Liu[a] and Tian Huang[a,b,*]

[a] Key laboratory of Modern Mechanisms and Equipment Design of The State Ministry of Education
Tianjin University Tianjin 300072, China

[b] School of Engineering, The University of Warwick, Coventry CV4 7AL, UK

Abstract This paper deals with non-singular tool path generation of a 5-axis hybrid robot named TriMule, which is designed for large part machining in situ. It is observed that at a singularity pose sudden changes occur in rotation of the C-axis and lengths of three telescopic legs. It is found that when the tool axis rotates about the axis normal to the plane expanded by the tool axis and the singular axis, the singular axis itself is forced to rotate simultaneously about the same axis in the opposite direction. This exploration enables the minimum rotation angle of the tool axis to be determined accurately for avoiding singularity and reducing machined surface errors caused by tool axis modification, leading to the development of an algorithm for non-singular tool path generation by modifying a partial set of the control points of B-splines. Both simulation and experiment on a prototype machine are carried out to verify the effectiveness of this approach.

Keywords: Hybrid robot, Tool path planning, Singularity avoidance

1. Introduction

Parallel kinematic machines composed of a 1T2R (T---Translation, R---Rotation) parallel mechanism plus a A/C type wrist serially connected to the platform exhibit desirable performance in terms of rigidity, accuracy, work envelop and reconfigurability. They are therefore suitable to be built as robotized modules for large part manufacturing in situ, drilling, riveting and high-speed milling for example [1]. This statement can be exemplified by very successful applications of the well-known Tricept robots [2]. The similar modules with hybrid architecture are the Exechon [3], TriMule [4], George V [5] and its variant [6]. Similar to the conventional A/C type 5-axis machines with serial architectures, the above mentioned parallel kinematic machines (or hybrid robots) suffer from singularity problem which occurs when the tool axis is coincident with a special axis known as singular axis, leading to sudden changes in rotary and translational drivers, thus resulting in errors on machined surface, and even damage to machines themselves.

In the last two decades, intensive investigations have been carried out towards non-singular tool path generation of the conventional A/C type 5-axis machines where the singularity axis is fixed in space. Various concepts have been proposed for visually and quantitatively defining and detecting the singular domain, for instance, singular cone [7], singular ring [8, 9], admissible orientation domain [10] and acceptable texture orientation region [11] in 2/3D representations. From a kinematic viewpoint, the methods in dealing with singularity problem can be roughly classified into two categories that can be performed in either the Cartesian space or the joint space. For the methods belonging to the first category, the refined subsequence of tool axis vectors falling into a specified singular cone are forced to rotate about a special axis while keeping full sequence of tool tip vectors unchanged. In realistic implementation, several CAD/CAM based algorithms have been developed to generate the refined sequence of tool axis vectors using single or double third order B-splines [8,9]. This allows the parameterized tool axis vectors to be modified by rotating the corresponding control point vectors. Here, the rotation can be represented by either the Roderigues formula or quaternion. The approach to determining the adequate tool orientations is also studied with the goal to minimize the machining error in the neighborhood of singularity [12]. The non-singular tool path can also be generated in post processing using the methods falling into the second category. In this regard, the refined sequence of C-axis commands is modified first by means of linear interpolation using the previous and current cutter locations in a recursive manner while keeping those associated

*The corresponding author
Email: tianhuang@tju.edu.cn

with the other rotary axis (i.e. the A or B axis) and tool tip vectors unchanged. This allows in turn the tool axis vector, thus the commands of translational drives, to be modified accordingly *via* inverse kinematics [13,14]. In addition, some interesting work has been conducted for singularity avoidance by adjusting the workpiece pose with respect to that of the worktable [15,16].

In recent years, tremendous work has been carried out on the design and development of 5-axis hybrid robots having A/C type wrist, the Tricept, the Exechon and the TriMule for example. The relevant work primarily focuses on type synthesis [4,17,18], kinetostatic analysis [19-20], kinematic optimization [21-27], and CNC controller development [28-30], etc. Unfortunately, little attention has been paid to non-singular tool path planning though singularity problem can be avoided, to some extent, by placing such a robot a tilt angle with respect to the workpiece frame at the cost of reducing the useful workspace envelop[23,31]. It should be pointed out that unlike a conventional A/C type 5-axis machine tool where the singular axis is fixed in space, the singular axis of a 5-axis hybrid robot having A/C type wrist varies with system configurations, leading to the change of the singular axis when the tool axis is modified for singularity avoidance. This feature brings two important issues to be investigated in non-singular tool path generation: (1) how to determine the axis about which the tool axis needs to rotate an angle for singularity avoidance, and (2) how to determine the exact value of the angle that allows the machined surface errors caused by tool axis modification to be minimized.

Motivated by the practical needs arising from the abovementioned two issues and by taking the TriMule robot [4] as an example, this paper investigates non-singular tool path generation of the hybrid robots with AC type wrist. The remainder of the paper is organized as followers. Having reviewed the methods for non-singular tool path generation of the conventional A/C type 5-axis machine tools and addressed two issues to be investigated for the 5-axis hybrid robots having A/C type wrist in Section 1, inverse displacement analysis of the robot is carried out in Section 2 with the mission to investigate behaviours of the joint variables in the neighbourhood of singularity configurations. In Section 3, the relationship between rotations of the singular axis and tool axis is revealed, leading to the development of an algorithm for non-singular tool path generation by modifying a partial set of the control points of B-splines. In Section 4, both simulation and experiment on a prototype machine are carried out to verify the effectiveness of the proposed approach before conclusions are drawn in Section 5.

2. Inverse Kinematics and Singularity Analysis

In this section, inverse displacement analysis will be carried out after a brief introduction to the structure of the TriMule robot. This is followed by the investigation into behaviours of the actuated joint variables in the neighbourhood of singular configurations.

2.1. System description

Fig. 1 shows a 3D view of the TriMule robot, which essentially consists of a 1T2R spatial parallel mechanism for positioning and a A/C type wrist attached to the platform *via* a trust bearing. The spatial parallel mechanism features a 6-DOF U<u>P</u>S limb plus a 2-DOF planar parallel mechanism comprising two actuated R<u>P</u>S limbs and a passive RP limb in between with its one extremity being rigidly fixed to the platform. The base link of the planar parallel mechanism is connected by a pair of R joints with the machine frame. Here, R, P, U, and S denote revolute, prismatic, universal, and spherical joints, and the underlined <u>P</u> denotes an actuated prismatic joint.

Fig. 2 shows the schematic diagram of the robot. For convenience, we treat universal/spherical joint as two/three revolute joints with the joint axes intersecting at a common point, and we number three actuated limbs as limb 1, 2 and 3, and the passive limb plus the wrist as limb 4. Let $B_i\left(i=2,3,4\right)$ be the intersections of the rear R joint axes of limb i with the R joint axis connecting the base link to the machine frame, and B_1 be the center of U joint of limb 1; let $A_i\left(i=1,2,3\right)$ be the center of the S joint of limb i, and A_4 be the intersection of the axial axis of the RP limb with its normal plane in which all A_i are placed; and let P be the intersection of two orthogonal axes of the wrist and C be the tool tip. For convenience of inverse displacement analysis, establish the reference frame $\left\{R_0\right\}$ with the x_0 axis being the R joint axis connecting the base link to the machine frame, and the z_0 axis being normal to the plane in which all B_i are placed. Meanwhile, establish body fixed frames $\left\{R_{j,4}\right\}\left(j=0,1,\cdots,4\right)$ with the $z_{j,4}$ axis being the $j+1$ joint axis, and the tool frame $\left\{R_{5,4}\right\}$ with $z_{5,4}$ axis being the spindle axis as shown in Fig 2. Then, the orientation

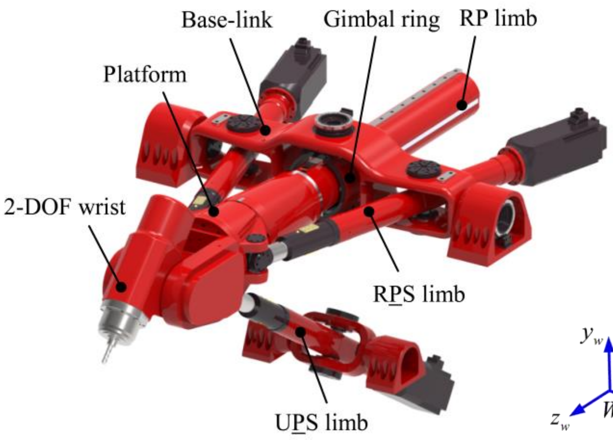


Fig. 1. 3D view of the TriMule robot

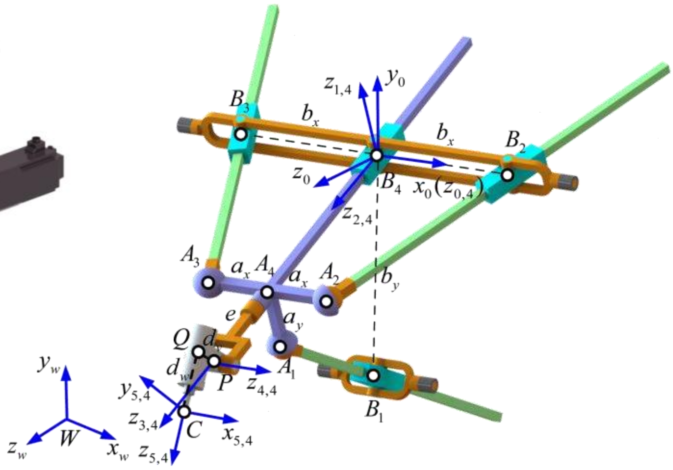


Fig. 2. Schematic diagram of the TriMule robot

matrix of $\{R_{5,4}\}$ with respect to $\{R_0\}$ can be expressed by

$$ {}^{0}\boldsymbol{R}_{5,4} = {}^{0}\boldsymbol{R}_{3,4}\,{}^{3,4}\boldsymbol{R}_{5,4} = \begin{bmatrix} \boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w} \end{bmatrix} \tag{1}$$

where ${}^{0}\boldsymbol{R}_{3,4}$ and ${}^{3,4}\boldsymbol{R}_{5,4}$ are the orientation matrices of $\{R_{3,4}\}$ with respect to $\{R_0\}$, and that of $\{R_{5,4}\}$ with respect to $\{R_{3,4}\}$, respectively.

$$ {}^{0}\boldsymbol{R}_{3,4} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ \sin\theta_1\sin\theta_2 & \cos\theta_1 & -\sin\theta_1\cos\theta_2 \\ -\cos\theta_1\sin\theta_2 & \sin\theta_1 & \cos\theta_1\cos\theta_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{2,4}\times\boldsymbol{s}_{3,4} & \boldsymbol{s}_{2,4} & \boldsymbol{s}_{3,4} \end{bmatrix} \tag{2}$$

$$ {}^{3,4}\boldsymbol{R}_{5,4} = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4\cos\theta_5 & \sin\theta_4\sin\theta_5 \\ \sin\theta_4 & \cos\theta_4\cos\theta_5 & -\cos\theta_4\sin\theta_5 \\ 0 & \sin\theta_5 & \cos\theta_5 \end{bmatrix} \tag{3}$$

where $\boldsymbol{s}_{2,4}\times\boldsymbol{s}_{3,4}$, $\boldsymbol{s}_{2,4}$ and $\boldsymbol{s}_{3,4}$ denote the unit vectors of three axes of $\{R_{3,4}\}$; $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ denote those of $\{R_{5,4}\}$; and $\theta_j\,(j=1,2,4,5)$ is the rotation angle about the $z_{j-1,4}$ axis of $\{R_{j-1,4}\}$, respectively.

2.2. Inverse displacement analysis

Inverse displacement analysis is concerned with the determination of the actuated joint variables for a given pose represented by the tool tip vector $\boldsymbol{r}_C$ and the tool axis vector $\boldsymbol{w}$. Then, the position vector of P can be expressed as

$$\boldsymbol{r}_P = \boldsymbol{r}_C - d_w\boldsymbol{w} - d_v\boldsymbol{v} \tag{4}$$

where $d_w = \left\|\overline{QC}\right\|$ and $d_v = \left\|\overline{PQ}\right\|$. Note that $\boldsymbol{v}$ can be determined by letting $\boldsymbol{n} = \boldsymbol{r}_Q / \left\|\boldsymbol{r}_Q\right\|$ with $\boldsymbol{r}_Q = \boldsymbol{r}_C - d_w\boldsymbol{w}$ such that

$$\boldsymbol{v} = \boldsymbol{w}\times\boldsymbol{u} \quad \text{with} \quad \boldsymbol{u} = \pm\frac{\boldsymbol{n}\times\boldsymbol{w}}{\left\|\boldsymbol{n}\times\boldsymbol{w}\right\|} \quad \text{if} \quad \left\|\boldsymbol{n}\times\boldsymbol{w}\right\| \neq 0 \tag{5}$$

It is easy to see from Eq.(5) that $\boldsymbol{v}$ has two possible solutions, and the one associated with the smaller amplitude of $\boldsymbol{r}_P$ should be considered as the adequate solution for achieving better kinematic performance and avoiding mechanical interference, i.e.

$$\boldsymbol{v} := \left|\min\left(\left\|\boldsymbol{r}_C - d_w\boldsymbol{w} - d_v\boldsymbol{v}\right\|, \left\|\boldsymbol{r}_C - d_w\boldsymbol{w} + d_v\boldsymbol{v}\right\|\right) \quad \text{if} \quad \left\|\boldsymbol{r}_C - d_w\boldsymbol{w} - d_v\boldsymbol{v}\right\| \neq \left\|\boldsymbol{r}_C - d_w\boldsymbol{w} + d_v\boldsymbol{v}\right\| \tag{6}$$

Thus, $\boldsymbol{r}_P$ can uniquely be determined by Eq.(4). Note that $\boldsymbol{r}_P$ can also be expressed as

$$\boldsymbol{r}_P = \left(q_{3,4} + e\right)\boldsymbol{s}_{3,4} \tag{7}$$

Hence, taking norm on both sides of Eq.(7), leads to

$$q_{3,4}=\|\boldsymbol{r}_P\|-e\ ,\quad \boldsymbol{s}_{3,4}=\frac{\boldsymbol{r}_P}{\|\boldsymbol{r}_P\|}=\begin{pmatrix} l \\ m \\ n \end{pmatrix} \tag{8}$$

This enables two rotation angles, θ_1 and θ_2, of the R(RP) limb to be found by

$$\theta_1=\arctan\left(\frac{-m}{n}\right),\quad \theta_2=\arcsin(l) \tag{9}$$

and thus ${}^0\boldsymbol{R}_{3,4}$ to be generated using Eq.(2). Rewrite Eq.(1) as

$${}^0\boldsymbol{R}_{3,4}^{\mathrm{T}}\,{}^0\boldsymbol{R}_{5,4}={}^{3,4}\boldsymbol{R}_{5,4} \tag{10}$$

or

$$\begin{bmatrix} l_1 & l_2 & l_3 \\ n_1 & n_2 & n_3 \\ m_1 & m_2 & m_3 \end{bmatrix}=\begin{bmatrix} \cos\theta_4 & -\sin\theta_4\cos\theta_5 & \sin\theta_4\sin\theta_5 \\ \sin\theta_4 & \cos\theta_4\cos\theta_5 & -\cos\theta_4\sin\theta_5 \\ 0 & \sin\theta_5 & \cos\theta_5 \end{bmatrix} \tag{11}$$

Two rotation angles of the wrist, θ_4 and θ_5, can then be obtained by

$$\theta_4=\arctan\left(\frac{n_1}{l_1}\right),\quad \theta_5=\arctan\left(\frac{m_2}{m_3}\right) \tag{12}$$

Additionally, loop closure $B_4-B_i-A_i-A_4-P-B_4$ of the parallel mechanism can be formulated as

$$\boldsymbol{r}_P-\boldsymbol{b}_i-e\boldsymbol{s}_{3,4}+\boldsymbol{a}_i=q_{3,i}\boldsymbol{s}_{3,i}\ ,\quad i=1,2,3 \tag{13}$$

where $q_{3,i}$ and $\boldsymbol{s}_{3,i}$ represent the length and unit vector of the ith actuated limb, and $\boldsymbol{a}_i={}^0\boldsymbol{R}_{3,4}\boldsymbol{a}_{i0}$ with

$$\begin{matrix} \boldsymbol{a}_{20}=a_x\hat{\boldsymbol{x}},\ \boldsymbol{a}_{30}=-a_x\hat{\boldsymbol{x}},\ \boldsymbol{a}_{10}=-a_y\hat{\boldsymbol{y}} \\ \boldsymbol{b}_2=b_x\hat{\boldsymbol{x}},\ \boldsymbol{b}_3=-b_x\hat{\boldsymbol{x}},\ \boldsymbol{b}_1=-b_y\hat{\boldsymbol{y}} \end{matrix}\ ,\quad \begin{matrix} a_x=\|\overrightarrow{A_4A_2}\|=\|\overrightarrow{A_4A_3}\| \\ a_y=\|\overrightarrow{A_4A_1}\| \end{matrix}\ ,\quad \begin{matrix} b_x=\|\overrightarrow{B_4B_2}\|=\|\overrightarrow{B_4B_3}\| \\ b_y=\|\overrightarrow{B_4B_1}\| \end{matrix}\ ,\quad e=\|\overrightarrow{A_4P}\| \tag{14}$$

$$\hat{\boldsymbol{x}}=\begin{pmatrix}1 & 0 & 0\end{pmatrix}^{\mathrm{T}}\ ,\ \hat{\boldsymbol{y}}=\begin{pmatrix}0 & 1 & 0\end{pmatrix}^{\mathrm{T}}$$

Equipped with $\boldsymbol{r}_P$ and ${}^0\boldsymbol{R}_{3,4}$ at hand, $q_{3,i}$ and $\boldsymbol{s}_{3,i}$ can finally be determined by

$$q_{3,i}=\|\boldsymbol{r}_P-\boldsymbol{b}_i-e\boldsymbol{s}_{3,4}+\boldsymbol{a}_i\|,\quad \boldsymbol{s}_{3,i}=\left(\boldsymbol{r}_P-\boldsymbol{b}_i-e\boldsymbol{s}_{3,4}+\boldsymbol{a}_i\right)/\,q_{3,i}\ ,\quad i=1,2,3 \tag{15}$$

2.3. Singularity analysis

It is important to note that the critical condition for achieving unique and continuous solution of the actuated joint variables is that $\|\boldsymbol{n}\times\boldsymbol{w}\|\neq 0$. Otherwise, singularity occurs where C, Q and B_4 are collinear. Therefore, we define $\boldsymbol{n}$ as the singular axis vector. It is clear that $\boldsymbol{n}$ varies with system configurations, depending upon not only orientation of the wrist but also that of the R(RP) limb. This feature makes the singularity problem of the robot different from that of conventional five-axis machine tools where the singular axis is fixed in space.

In order to investigate behaviors of the actuated joint variables in the neighborhood of singular configurations, establish a floating frame $\{R_n\}$ shown in Fig.3 with $\boldsymbol{n}$

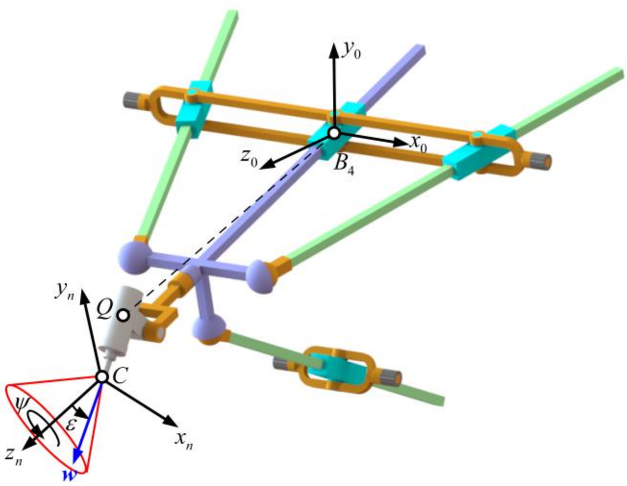


Fig. 3. A floating frame for describing singularity

being unit vector of the z_n axis and $(\boldsymbol{n}\times\hat{z})/\|\boldsymbol{n}\times\hat{z}\|$ being the unit vector of the x_n axis as long as $\boldsymbol{n}$ is not collinear with the z_0 axis. Otherwise, let the x_0 and y_0 axes be the x_n and y_n axes, respectively. Thus, $\boldsymbol{w}$ can be represented by

$$\boldsymbol{w} = {}^0\boldsymbol{R}_n \boldsymbol{w}_n \tag{16}$$

with

$${}^0\boldsymbol{R}_n = \begin{bmatrix} \dfrac{\boldsymbol{n}\times\hat{z}}{|\boldsymbol{n}\times\hat{z}|} & \dfrac{\boldsymbol{n}\times(\boldsymbol{n}\times\hat{z})}{|\boldsymbol{n}\times\hat{z}|} & \boldsymbol{n} \end{bmatrix},\quad \boldsymbol{w}_n = \begin{pmatrix} \sin\psi\sin\varepsilon \\ -\cos\psi\sin\varepsilon \\ \cos\varepsilon \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}$$

where ψ and ε are the procession and nutation angles. Given $\boldsymbol{n}=(0.15 \quad -0.15 \quad 0.98)^{\mathrm{T}}$ and dimensional parameters of the robot shown in Table 1, Fig.4 shows variations of the actuated joint variable over $\psi = 0 \sim 180^\circ$ and $\varepsilon = -5 \sim 5^\circ$. It is easy to see that sudden changes occur in $q_{3,i}\,(i=1,2,3)$ and θ_4 when the tool axis vector is coincident with the singular axis vector, which thus inevitably cause errors on the machined surfaces and even bring damage to the robot itself. Therefore, a method for non-singular tool path generation in the neighbourhood of singular configurations will be investigated in what follows.

Table 1

Dimensional parameters of the TriMule robot (mm)

$a_x(a_y)$	b_x	b_y	e	d_v	d_w
135	320	570	345	120	350

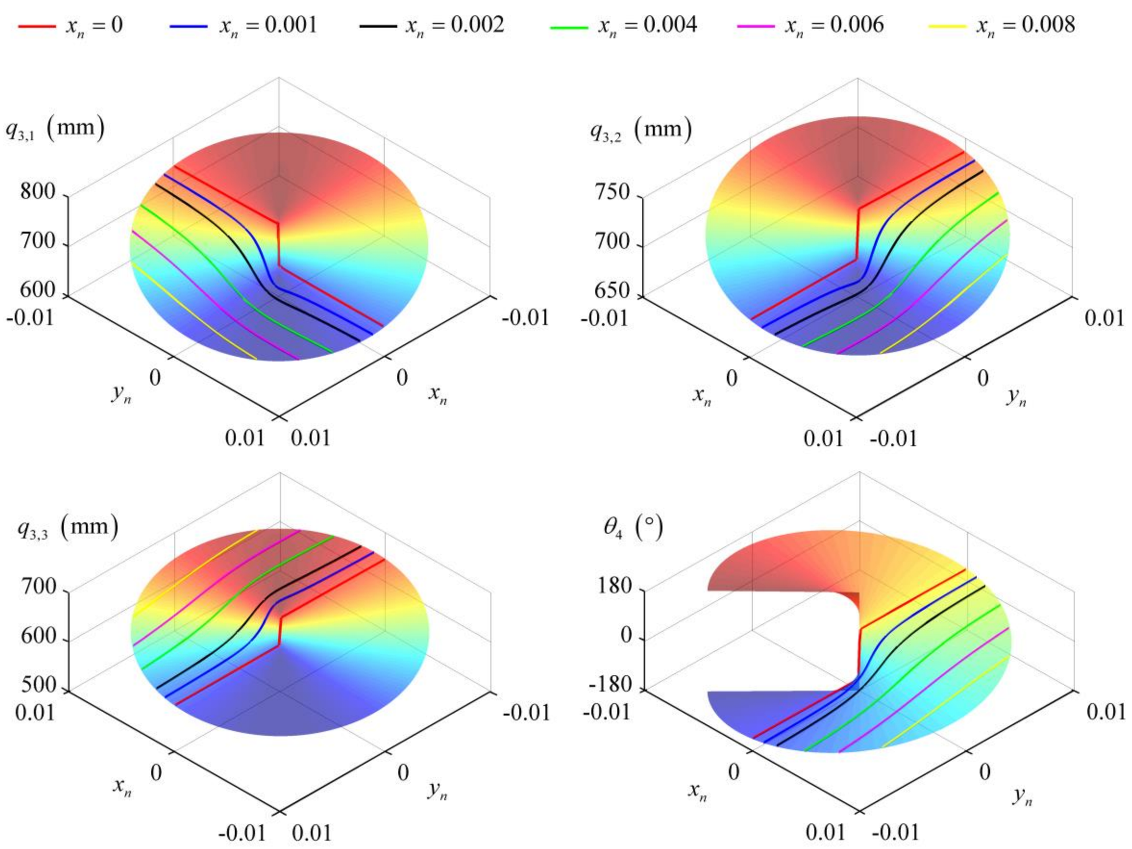


Fig. 4. Variations of $q_{3,i}\,(i=1,2,3)$ and θ_4 in the neighborhood of singular configuration

3. Non-singular Tool Path Generation

Inspired by the method proposed in [8], an approach is proposed in this section for generating non-singular tool path in the neighborhood of singularity. The method involves (1) parameterization of the tool axis vector and the singular axis vector using B-splines; (2) modification of a subsequence of the control point vectors dominating the orientation of the tool axis vector in the neighborhood of a singular configuration.

3.1. Tool path parameterization using B-splines

In this section, we will briefly recall the formulation of B-splines [32] that will be used to generate a refined tool path represented the parameterized tool tip vector $\boldsymbol{r}_C$ and tool axis vector $\boldsymbol{w}$ along the path. Parameterization can be made by a B-spline of degree $p=3$ to achieve G^2 -continuity such that $\boldsymbol{r}_C$ and $\boldsymbol{w}$ can be represented as linear combinations of polynomials $N_{i,p}(u)$, called base or blending functions, weighted by $\boldsymbol{Q}_{C,i}$ and $\boldsymbol{Q}_{w,i}$, named control point vectors, respectively.

$$\boldsymbol{r}_C(u)=\sum_{i=0}^{n}N_{i,p}(u)\boldsymbol{Q}_{C,i}\ ,\quad \boldsymbol{w}(u)=\sum_{i=0}^{n}N_{i,p}(u)\boldsymbol{Q}_{w,i}\quad u\in[0\quad 1] \tag{17}$$

with $N_{i,p}(u)$ satisfying the De Boor formula

$$\begin{cases} N_{i,0}(u)=\begin{cases}1, & u_i\le u<u_{i+1}\\ 0, & \text{otherwise}\end{cases} \\ N_{i,p}(u)=\dfrac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\dfrac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u)\end{cases} \tag{18}$$

where u_i ($i=0,1,\cdots,n+p+1$) form a sequence of knots necessary for building $\boldsymbol{r}_C(u)$ and $\boldsymbol{w}(u)$ by interpolation and can be obtained using the centripetal method [32] as long as a sequence of prescribed tool tip vectors $\boldsymbol{r}_C(\bar{u}_k)$ at $\bar{u}_k$ ($k=0,1,\cdots,n$) are made available by CAM software.

$$\begin{cases} u_0=\cdots=u_p=0,\quad u_{n+1}=\cdots=u_{n+p+1}=1 \\ u_i=\dfrac{1}{p}\sum\limits_{k=i-p}^{i-1}\bar{u}_k,\ i=p+1,\ \cdots,n\end{cases} \tag{19}$$

where

$$\begin{cases} \bar{u}_0=0,\quad \bar{u}_n=1 \\ \bar{u}_k=\bar{u}_{k-1}+\dfrac{\sqrt{\left|\boldsymbol{r}_C(\bar{u}_k)-\boldsymbol{r}_C(\bar{u}_{k-1})\right|}}{\sum\limits_{j=1}^{n}\sqrt{\left|\boldsymbol{r}_C(\bar{u}_j)-\boldsymbol{r}_C(\bar{u}_{j-1})\right|}},\ k=1,2,\cdots,n-1\end{cases} \tag{20}$$

In this way, full sequences of the control point vectors can be determined by solving the following linear algebraic equations.

$$\boldsymbol{r}_C(\bar{u}_k)=\sum_{i=0}^{n}N_{i,p}(\bar{u}_k)\boldsymbol{Q}_{C,i},\quad \boldsymbol{w}(\bar{u}_k)=\sum_{i=0}^{n}N_{i,p}(\bar{u}_k)\boldsymbol{Q}_{w,i}\ ,\quad k=0,1,\cdots,n \tag{21}$$

and thus the parameterized singular axis vector, denoted by $\boldsymbol{n}(u)$, can be obtained by

$$\boldsymbol{n}(u)=\frac{\boldsymbol{r}_C(u)-d_w\boldsymbol{w}(u)}{\left\|\boldsymbol{r}_C(u)-d_w\boldsymbol{w}(u)\right\|} \tag{22}$$

3.2. Tool path modification

Equipped with the parameterized tool tip vector $\boldsymbol{r}_C(u)$ and tool axis vector $\boldsymbol{w}(u)$ to hand, a method to generate the non-singular tool path will be proposed. Here, we use a conical surface with a specified half angle $[\varepsilon]$ to quantitatively represent the boundary of a singular pose. So, for a given pose, it is said that singularity occurs once $\varepsilon = \arccos\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{n}\right) \le [\varepsilon]$. For convenience, the conical surface can visually be represented by a circular ring of radius $\sin[\varepsilon]$ lying on a unit sphere shown in Fig.5.

By keeping $\boldsymbol{r}_C$ unchanged, a natural and straightforward remedy for avoiding singularity is to force $\boldsymbol{w}$ to rotate an angle $\varepsilon_w > 0$ about the axis normal to the plane expanded by $\boldsymbol{w}$ and $\boldsymbol{n}$ such that the modified ε becomes slightly larger than $[\varepsilon]$. Note that ε_w is assumed to be sufficiently small such that $\sin\varepsilon_w \approx \varepsilon_w$ and $\cos\theta_w \approx 1$, the modified $\boldsymbol{w}$, denoted by $\boldsymbol{w}_{new}$, can be expressed as

$$\boldsymbol{w}_{new} = \boldsymbol{w} + \varepsilon_w \boldsymbol{s} \times \boldsymbol{w} \tag{23}$$

where $\boldsymbol{s} = (\boldsymbol{n} \times \boldsymbol{w}) / \|\boldsymbol{n} \times \boldsymbol{w}\|$ denotes the unit vector of the rotation axis, leading to $\boldsymbol{w} \times \boldsymbol{w}_{new} = \varepsilon_w \boldsymbol{s}$.

On the other hand, keeping mind that $\boldsymbol{n}$ is a function of both $\boldsymbol{w}$ and $\boldsymbol{r}_C$ even though $\boldsymbol{r}_C$ remains unmodified. Hence, the modified $\boldsymbol{n}$, denoted by $\boldsymbol{n}_{new}$, can be expressed as

$$\boldsymbol{n}_{new} = \frac{\boldsymbol{r}_C - d_w \boldsymbol{w}_{new}}{\|\boldsymbol{r}_C - d_w \boldsymbol{w}_{new}\|} = \frac{\boldsymbol{r}_C - d_w(\boldsymbol{w} + \varepsilon_w \boldsymbol{s} \times \boldsymbol{w})}{\|\boldsymbol{r}_C - d_w(\boldsymbol{w} + \varepsilon_w \boldsymbol{s} \times \boldsymbol{w})\|} \tag{24}$$

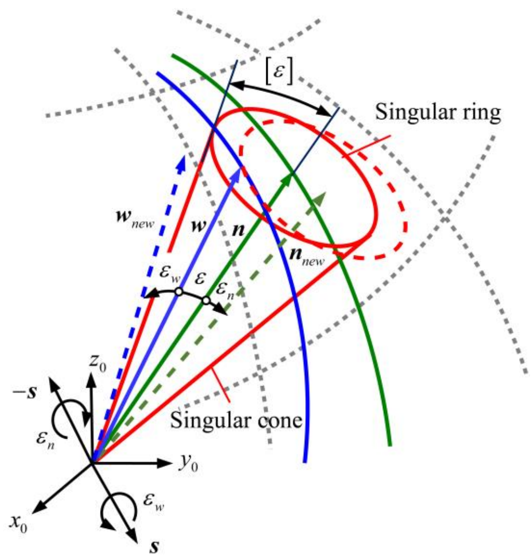


Fig. 5. Simultaneous rotations of $\boldsymbol{w}$ and $\boldsymbol{n}$ in opposite directions

It is easy to prove that

$$\boldsymbol{n} \times \boldsymbol{n}_{new} = -\mu\varepsilon_w \boldsymbol{s} = -\varepsilon_n \boldsymbol{s} \tag{25}$$

where

$$\mu \approx \frac{d_w}{\|\boldsymbol{r}_C - d_w \boldsymbol{w}\|} = \frac{d_w}{\|\boldsymbol{r}_Q\|}$$

It is interesting to see from Eq.(25) that when $\boldsymbol{w}$ rotates about $\boldsymbol{s}$ a small angle ε_w to reach $\boldsymbol{w}_{new}$, $\boldsymbol{n}$ is forced to rotate simultaneously about $-\boldsymbol{s}$ a small angle ε_n to reach $\boldsymbol{n}_{new}$, and ε_n is proportional to ε_w, i.e. $\varepsilon_n = \mu\varepsilon_w$. Consequently, the angle between $\boldsymbol{n}_{new}$ and $\boldsymbol{w}_{new}$ becomes $\varepsilon_{new} = \varepsilon + \varepsilon_n + \varepsilon_w$ after both rotations. Therefore, singularity at the given pose can be avoided as long as the following relationship hold even if $\varepsilon = 0$.

$$\varepsilon_w \ge \frac{[\varepsilon]}{1+\mu} \tag{26}$$

In the light of Eq.(23), the modified tool axis vector spline can be expressed by

$$\boldsymbol{w}_{new}(u) = \sum_{i=0}^{n} N_{i,p}(u)\left[\boldsymbol{E}_3 + \frac{[\varepsilon]}{1+\mu}[\boldsymbol{s}\times]\right]\boldsymbol{Q}_{w,i} = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{Q}_{new,w,i} \tag{27}$$

where $\boldsymbol{E}_3$ denotes a unit matrix of order 3 and $[\boldsymbol{s}\times]$ denotes the skew matrix of $\boldsymbol{s}$.

According to the foregoing analysis, an algorithm is developed for the realistic implementation:

(1) Generate the sequences of the parameterized tool path vectors and singular axis vector using Eq.(17) and (22) by dividing the path parameter $u \in [0,1]$ into m (e.g. $m = 10^4$) evenly spaced segments with sufficient small increment.

$$\left\{\boldsymbol{r}_C(u_0^*) \ \cdots \ \boldsymbol{r}_C(u_m^*)\right\},\ \left\{\boldsymbol{w}(u_0^*) \ \cdots \ \boldsymbol{w}(u_m^*)\right\},\ \left\{\boldsymbol{n}(u_0^*) \ \cdots \ \boldsymbol{n}(u_m^*)\right\},\ u_k^* = \frac{k}{m},\ k = 0,1,\cdots,m \tag{28}$$

(2) Detect along the path the domain in which singularity may occur. If such a domain $\left[u_f^* \quad u_g^*\right] \subset [0 \quad 1]$ does exist, i.e.

$$\varepsilon_k = \arccos\left(\boldsymbol{w}^{\mathrm{T}}(u_k^*)\boldsymbol{n}(u_k^*)\right) \le [\varepsilon],\ \forall k \in [f \quad g] \tag{29}$$

record the minimum subsequence of knots $\{u_a \quad \cdots \quad u_b\}$ that includes the sequence $\{u_f^* \quad \cdots \quad u_g^*\}$. Then, let u_d^* be the one corresponding to

$$\arccos\left(\boldsymbol{w}^{\mathrm{T}}\left(u_d^*\right)\boldsymbol{n}\left(u_d^*\right)\right)=\min\left\{\varepsilon_k,\forall k\in\left[f \quad g\right]\right\} \tag{30}$$

and go to Step (3); Otherwise remain $\left\{\boldsymbol{w}\left(u_0^*\right) \quad \cdots \quad \boldsymbol{w}\left(u_m^*\right)\right\}$ unchanged.

(3) Modify $\left\{\boldsymbol{w}\left(u_0^*\right) \quad \cdots \quad \boldsymbol{w}\left(u_m^*\right)\right\}$ by only changing the control point vectors associated with $\{u_a \quad \cdots \quad u_b\}$.

$$\boldsymbol{Q}_{new,w,i}=\begin{cases}\left[\boldsymbol{E}_3+\dfrac{[\varepsilon]}{1+\mu}[\boldsymbol{s}\times]\right]\boldsymbol{Q}_{w,i} & a-p\le i\le b\\ \boldsymbol{Q}_{w,i} & \text{Otherwise}\end{cases} \tag{31}$$

where

$$\mu=\frac{d_w}{\left\|\boldsymbol{r}_Q\left(u_d^*\right)\right\|},\quad \boldsymbol{s}=\frac{\boldsymbol{n}\left(u_d^*\right)\times\boldsymbol{w}\left(u_d^*\right)}{\left\|\boldsymbol{n}\left(u_d^*\right)\times\boldsymbol{w}\left(u_d^*\right)\right\|} \tag{32}$$

Note that if $\arccos\left(\boldsymbol{w}^{\mathrm{T}}\left(u_d^*\right)\boldsymbol{n}\left(u_d^*\right)\right)\le[\varepsilon]\times10^{-2}$, $\boldsymbol{s}$ should be determined by

$$\boldsymbol{s}=\frac{\boldsymbol{n}\left(u_d^*\right)\times\left(\boldsymbol{w}\left(u_{d+1}^*\right)\times\boldsymbol{w}\left(u_{d-1}^*\right)\right)}{\left\|\boldsymbol{n}\left(u_d^*\right)\times\left(\boldsymbol{w}\left(u_{d+1}^*\right)\times\boldsymbol{w}\left(u_{d-1}^*\right)\right)\right\|} \tag{33}$$

for ensuring the computational accuracy. Finally, note that machining error could be caused by the modification of $\boldsymbol{w}(u)$, vigilance should be excised in choosing an adequate threshold $[\varepsilon]$.

4. Verification

In this section, the algorithm for non-singular tool path generation will be verified by both simulation and experiment on the TriMule robot having dimensional parameters shown in Table 1 and a cylindrical task workspace shown in Fig. 6 and Table 2. In the simulation, a circular arc of radius $R_c = 1000$ mm and central angle $\theta_0 = 10^\circ$ is used as the tool path with the feed direction downwards as shown. A workpiece frame $\{R_w\}$ is placed where the origin of $\{R_w\}$, the arc centre O and the origin B_4 of $\{R_0\}$ are collinear, the y_w axis is parallel to the y_0 axis, and the z_w axis is coincident with $\left|\overrightarrow{OW}\right|$. Given the tool tip vector ${}^{w}\boldsymbol{r}_C=\left(0 \quad R_c\sin\theta \quad R_c\left(\cos\theta-1\right)\right)^{\mathrm{T}}$ and tool axis vector ${}^{w}\boldsymbol{w}=\left(0 \quad \sin\theta \quad \cos\theta\right)^{\mathrm{T}}$ evaluated in $\{R_w\}$ with $\theta\in\left[-\theta_0/2 \quad \theta_0/2\right]$, those evaluated in $\{R_0\}$ can be obtained by

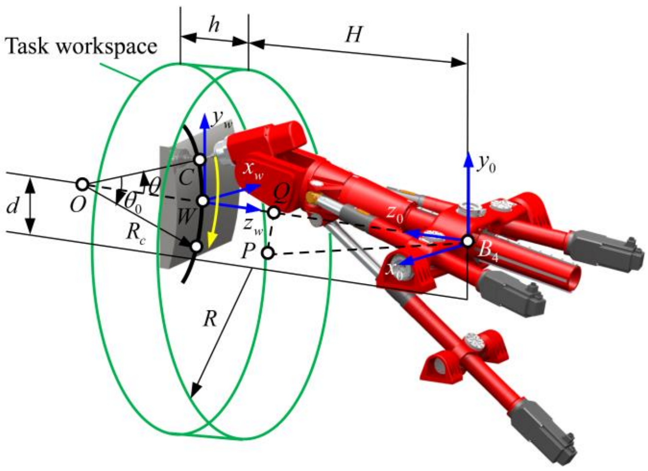


Fig. 6. A circular arc tool path that passes singular pose at $\theta = 0^\circ$

Table 2 Dimensions of task workspace (mm)

H	R	h	d
1000	600	300	190

$$\boldsymbol{r}_C={}^{w}\boldsymbol{R}_0^{\mathrm{T}}\left({}^{w}\boldsymbol{r}_C-{}^{w}\boldsymbol{d}\right),\quad \boldsymbol{w}={}^{w}\boldsymbol{R}_0^{\mathrm{T}}\,{}^{w}\boldsymbol{w},\quad {}^{w}\boldsymbol{R}_0=\begin{bmatrix}-1&0&0\\0&1&0\\0&0&-1\end{bmatrix} \tag{34}$$

where ${}^{w}\boldsymbol{d}=\left(0 \quad 0 \quad 1350\text{mm}\right)^{\mathrm{T}}$ denotes the position vector pointing from W to B_4, ${}^{w}\boldsymbol{R}_0$ denotes the orientation matrix of $\{R_0\}$ with respect to $\{R_w\}$. It is easy to see that singularity occurs in the neighbourhood of $\theta=0^{\circ}$ for this arrangement. Then, the refined sequences $\left\{\boldsymbol{r}_C\left(u_k^*\right)\right\}$, $\left\{\boldsymbol{w}\left(u_k^*\right)\right\}$ and $\left\{\boldsymbol{n}\left(u_k^*\right)\right\}$ （$k=0,1,\cdots,m$） with $m=10^4$ are generated using Eqs.(17) and (22) by parameterizing the sequence $\left\{\boldsymbol{r}_C\left(\bar{u}_k\right)\right\}$ （$k=0,1,\cdots,n$） with $n=10^2$. By taking $[\varepsilon]=10^{-3}$ rad as recommended by [7], the singular domain is detected by.

$$\theta\in\left[-0.042^{\circ} \quad 0.042^{\circ}\right] \text{ or } u_k^*\in\left[u_f^* \quad u_g^*\right]\subset\left[u_a \quad u_b\right] \tag{35}$$

where

$$u_f^*=0.4978,\ u_g^*=0.5022,\ u_a=0.49,\ u_b=0.51$$
$$f=4978,\ g=5022,\ a=51,\ b=53$$

Then, the rotation angle ε_w and unit vector $\boldsymbol{s}$ can be determined by Eq.(26) and Eq.(32) . This allows the sequence of the control point vector to be partially modified by Eq.(31) for generating a non-singular tool path. Fig.7 shows variation of ε_{new} over u in the neighbourhood of singularity with and without considering μ. According to [8] and Eq.(26), the machined surface error arising from tool axis modification can also be formulated as

$$e=r\sin\varepsilon_w\approx r\varepsilon_w=r[\varepsilon]/(1+\mu) \tag{36}$$

where r denotes radius of an end mill. It is easy to see from Eq.(36) that taking $\mu>0$ into account is helpful to decrease ε_w and thus to reduce the machined surface error in comparison with the case without this consideration, i.e. $\mu=0$.

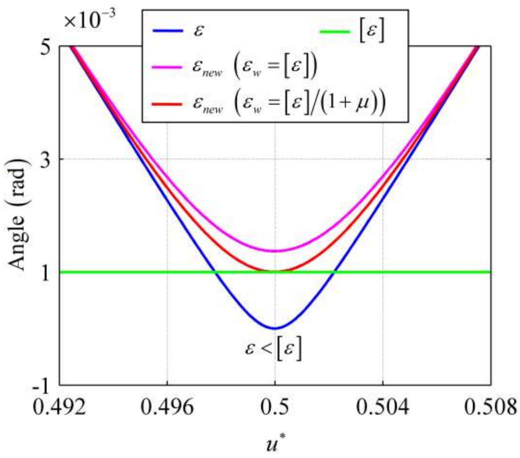


Fig. 7. ε and ε_{new} vs. u^*

In order to examine bearings of the modified tool path on motions of the actuated joint variables，the parameterized tool path is interpolated on a periodic increment of 10 ms using the trapezoid acceleration motion profile with feed rate of 20 mm/s and maximum acceleration of 200 mm/s^2. Fig.8 shows the joint variables $q_{3,i}\left(i=1,2,3\right)$ and θ_4 vs. time before and after modification. It can be seen that sudden changes occur in the neighborhood of singularity before the modification, resulting in $\Delta q_{3,1}=108$ mm , $\Delta q_{3,2}=\Delta q_{3,3}=31$ mm , and $\Delta\theta_4=\pi$ in a very short period of time. Nevertheless, motions of these joints become continuous and smooth in the same time period after the modification.

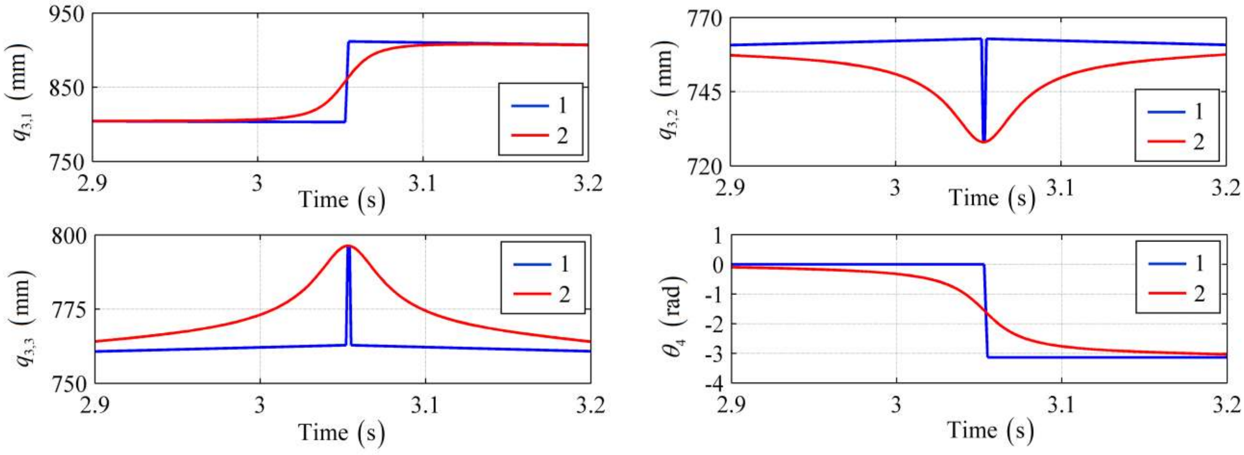
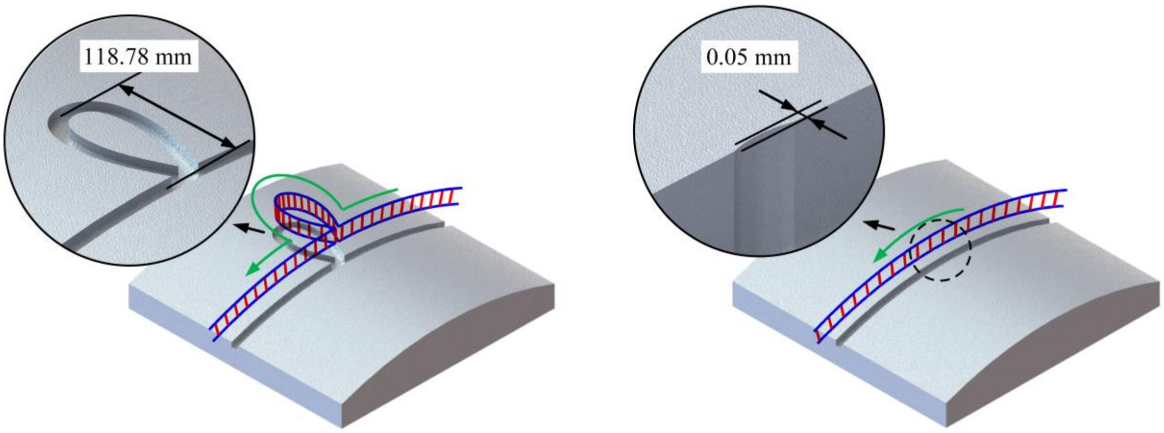


Fig. 8. $q_{3,i}\left(i=1,2,3\right)$ and θ_4 vs. time, 1: before modification, 2: after modification

Simulation is carried out to evaluate the machining effects in the neighbourhood of singularity before and after tool path modification. This can be done by interpolating the joint variables shown in Fig.8 with 1.0 ms increment and utilizing the data to generate the tool path *via* forward kinematics. Fig.9 shows the machining effects simulated by the path simulator provided in UG NX 8.0 before and after tool path modification. It can be seen from Fig.9(a) that sudden changes in the original joint variables produce a ring like tool path with a maximum deviation up to 118.78 mm, which may cause serious machining error and even damages the tool. However, the modified joint variables enable the tool to pass through the singular configuration very smoothly, leaving only a maximum machining error of 0.05 mm as depicted in Fig.9(b), thus demonstrating the effectiveness of the proposed approach.



(a) The original machining effect (b) The modified machining effect

Fig. 9. Simulation of the original and modified machining effects

Also, note that $\mu = d_w / \|\boldsymbol{r}_Q\| = d_w / \sqrt{\|\boldsymbol{r}_P\| - d_v^2}$ varies with $\|\boldsymbol{r}_P\|$ across the cylindrical task space, reaching the maximum value when $\|\boldsymbol{r}_P\|$ takes the minimum value and vice versus as two extreme cases shown in Fig.10. Both will produce less machined surface error arising from tool axis modification compared with the case where $\mu = 0$.

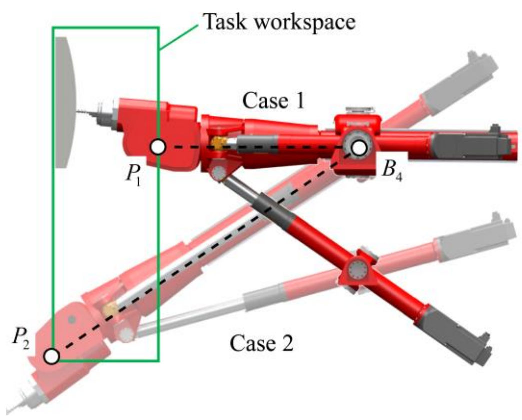


Fig. 10. Two extreme cases for values of μ

Case 1: $\mu_{\max} = 0.3525$ for $\boldsymbol{r}_P = \begin{pmatrix} 0 & 0 & 1000 \text{ mm} \end{pmatrix}^{\mathrm{T}}$

Case 2: $\mu_{\min} = 0.2183$ for $\boldsymbol{r}_P = \begin{pmatrix} 0 & -790 \text{ mm} & 1400 \text{ mm} \end{pmatrix}^{\mathrm{T}}$

On the basis of simulation, experiment is carried out on a prototype machine of the TriMule robot (See Fig.11) having the maximum movement capability of 50 m/min in speed and 1g in acceleration. Hardware of the CNC system of the robot is built upon IPC + Turbo PMAC-PCI motion controller and its software is developed by NI LabVIEW®. The tool path is generated using UG NX8.0, and then detected and modified offline by the proposed algorithm programmed by Matlab®. This allows G codes to be generated and interpolated with 10 ms in the Cartesian space first, and then converted into the joint NC commands *via* inverse kinematics with servo feedback rate of 2 KHz in real implementation.

In the experiment, the following joint limits are set for safety reasons in a neighbourhood of singularity configuration：

$$[\dot{q}_{3,i}] = 250 \ \text{mm/s}, \ [\ddot{q}_{3,i}] = 2000 \ \text{mm/s}^2, \ i = 1,2,3$$

$$[\dot{\theta}_4] = 1.5 \ \text{rad/s}, \ [\ddot{\theta}_4] = 9.5 \ \text{rad/s}^2,$$

$$[\dot{\theta}_5] = 1.3 \ \text{rad/s}, \ [\ddot{\theta}_5] = 7.8 \ \text{rad/s}^2$$

Building upon encoder signals acquired by DSP in the motion controller, Fig.12 and Fig.13 show displacement, velocity, acceleration, and tracking errors of the actuated joints vs. time after the modification. It is easy to see that continuous and smooth joint variables and their first and second derivatives can be achieved in the neighbourhood of singularity, resulting in a tracking error of < 0.015 mm for the telescopic legs and $< 2\times10^{-4}$ rad for the A/C wrist, respectively, thus verifying the effectiveness of the proposed algorithm for singularity free tool path generation.

Fig. 11. A prototype machine of the TriMule robot

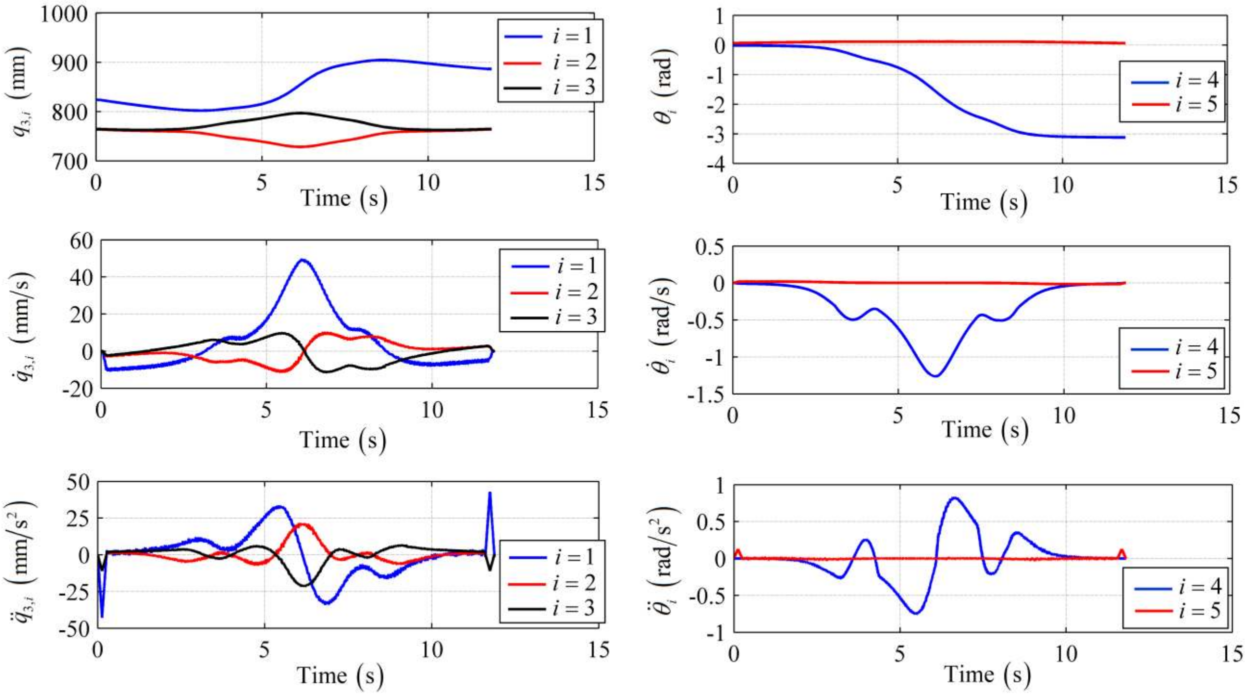


Fig. 12. Displacement, velocity and acceleration of the actuated joints vs. time in the neighborhood of singular pose

5. Conclusions

By taking the TriMule robot as an example, this paper presents a method for non-singular tool path generation of the hybrid robots having A/C type wrist. The conclusions are drawn as follows.

(1) At a singular configuration, sudden changes occur in both rotation of the C-axis and lengths of three telescopic legs if the eccentricity $d_v \neq 0$; otherwise singularity merely causes a sudden change in rotation of the C-axis.

(2) When the tool axis rotates a small angle about the axis normal to the plane expanded by the tool axis and the singular axis, the singular axis is forced to rotate a small angle simultaneously about the same axis in the opposite direction. The magnitude of singular axis rotation is linearly proportional to that of the tool axis if the latter is sufficient small, allowing the minimum rotation angle of the tool axis to be determined accurately.

(3) An algorithm for non-singular tool path generation is developed by modifying a partial set of the control points of B-splines. The results of both simulation and experiment on a prototype machine show that in the neighborhood of singularity, continuous and smooth joint motions and the acceptable tracking errors can be ensured after the modification, they thereby verify the effectiveness of the proposed approach.

(4) The proposed method is general such that it can be employed to generate non-singular tool path of the other hybrid robots having A/C type wrist.

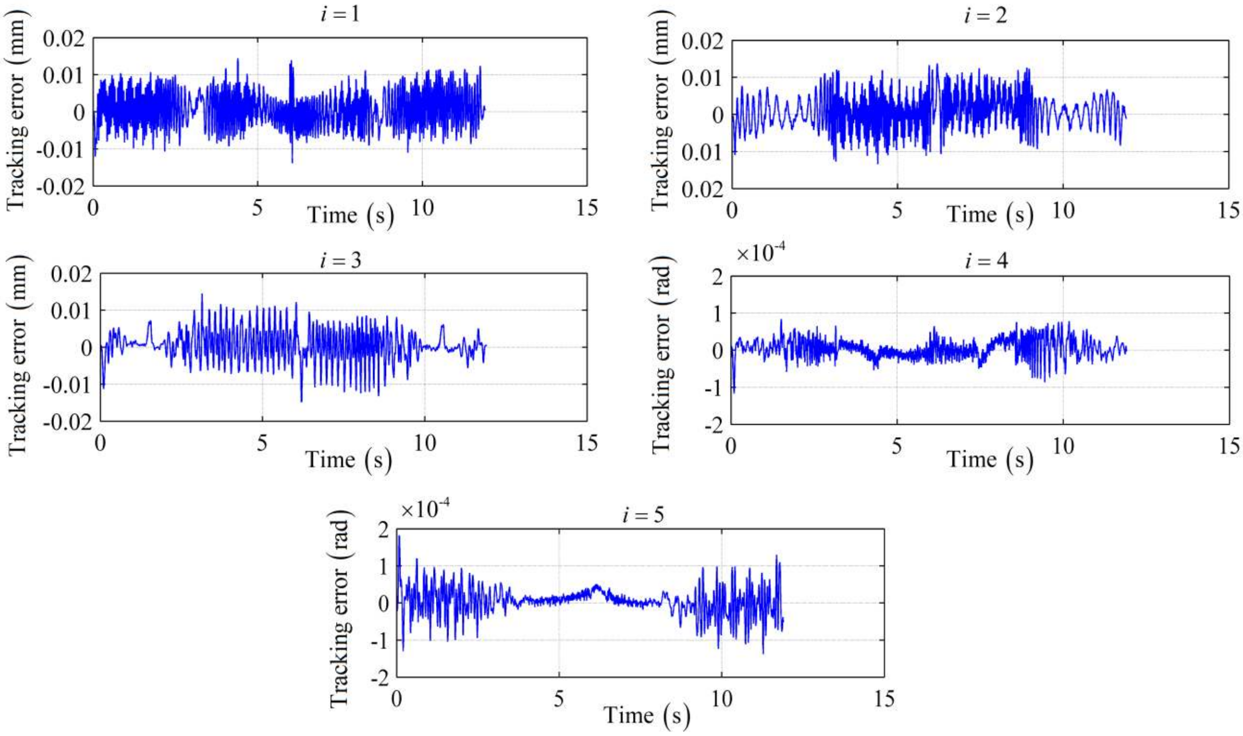


Fig. 13. Tracking errors of the actuated joints vs. time in the neighbourhood of singular pose

Acknowledgement

The research is partially supported by National Natural Science Foundation of China (Grant 51420105007) , National key research and development program（Grant 2017YFE0111300）and EU H2020-RISE-ECSASDP (Grant 734272).

References

[1] L. Uriarte, M. Zatarain, D. Axinte, et al., Machine tools for large parts, CIRP Annals-Manufacturing Technology 62 (2) (2013) 731-750.

[2] K. E. Neumann, Tricept application, in: Proceedings-3rd Chemnitz Parallel Kinematics Seminar, 2002.

[3] K. E. Neumann, The key to aerospace automation, in: SAE Aerospace Manufacturing and Automated Fastening Conference and Exhibition, 2006.

[4] C. Dong, H. Liu, Q. Liu, T. Sun, T. Huang, D. G. Chetwynd, An approach for type synthesis of overconstrained 1T2R parallel mechanisms, in: Computational Kinematics, 2018.

[5] H. K. Tönshoff, H. Grendel, R. Kaak, Structure and characteristics of the hybrid manipulator Georg V, in: Parallel Kinematic Machines, C.R. Boer, L. Molinari-Tosatti, K.S. Smith (Eds.), Springer-Verlag, London, 1999, pp. 365–376.

[6] Z. Gao, D. Zhang, Performance analysis, mapping, and multiobjective optimization of a hybrid robotic machine tool, Industrial Electronics IEEE Transactions on 62 (1) (2015) 423-433.

[7] A. Affouard, E. Duc, C. Lartigue, J.-M. Langeron, P. Bourdet, Avoiding 5-axis singularities using tool path deformation, International Journal of Machine Tools and Manufacture 44 (4) (2004) 415-425

[8] J. Yang, Y. Altintas, Generalized kinematics of five-axis serial machines with nonsingular tool path generation, International Journal of Machine Tools & Manufacture 75 (12) (2013) 119-132.

[9] Z. Lin, J. Fu, H. Shen, W. Gan, Non-singular tool path planning by translating tool orientations in C-space, International Journal of Advanced Manufacturing Technology 71 (9-12) (2014) 1835-1848.

[10] C. Castagnetti, E. Duc, P. Ray, The domain of admissible orientation concept: a new method for five-axis tool path optimization, Computer-Aided Design 40 (9) (2008) 938-950.

[11] Z. Lin, J. Fu, H. Shen, Improving machined surface texture in avoiding five-axis singularity with the acceptable-texture orientation region concept, International Journal of Machine Tools & Manufacture 108 (2016) 1-12.

[12] M. Munlin, S.S. Makhanov, E. Bohez, Optimization of rotations of a five-axis milling machine near stationary points, Computer-Aided Design 36 (12) (2004) 1117-1128.

[13] K. Sørby, Inverse kinematics of five-axis machines near singular configurations, International Journal of Machine Tools and

Manufacture 47 (2) (2007) 299-306.
[14] C.H. She, Z.T. Huang, Postprocessor development of a five-axis machine tool with nutating head and table configuration, International Journal of Advanced Manufacturing Technology 38 (7-8) (2008) 728-740.
[15] Z. Lin, J. Fu, H. Shen, W. Gan, On the workpiece setup optimization for five-axis machining with RTCP function, International Journal of Advanced Manufacturing Technology 74 (1-4) (2014) 187-197.
[16] R.J. Cripps, B. Cross, M. Hunt, G. Mullineux, Singularities in five-axis machining: cause, effect and avoidance, International Journal of Machine Tools and Manufacture 116 (2017) 40-51.
[17] F. Xie, X. Liu, Z. You, et al., Type synthesis of 2T1R-type parallel kinematic mechanisms and the application in manufacturing, Robotics & Computer Integrated Manufacturing 30 (1) (2014) 1-10.
[18] Q. Li, J. M. Hervé, Type synthesis of 3-DOF RPR-equivalent parallel mechanisms, IEEE Transactions on Robotics 30 (6) (2017) 1333-1343.
[19] D. Zhang, F. Xi, C. M. Mechefske, et al. Analysis of parallel kinematic machine with kinetostatic modelling method, Robotics & Computer Integrated Manufacturing 20 (2) (2004) 151-165.
[20] J. Zhang, Y. Zhao, Y. Jin. Kinetostatic-model-based stiffness analysis of Exechon PKM, Robotics & Computer Integrated Manufacturing 37 (2016) 208-220.
[21] Y. Li, Q. Xu. Kinematic analysis of a 3-PRS parallel manipulator, Robotics & Computer Integrated Manufacturing 23 (4) (2007) 395-408.
[22] S. Son, T. Kim, S. E. Sarma, et al., A hybrid 5-axis CNC milling machine, Precision Engineering 33 (4) (2009) 430-446.
[23] M. Milutinović, N. Slavković, D. Milutinović, Kinematic modelling of hybrid parallel-serial five-axis machine tool, FME Transactions 41 (1) (2013) 1-10.
[24] Z. M. Bi, Y. Jin, R. Gibson and P. McTotal, Kinematics of parallel kinematic machine Exechon, in: Proceedings of the 2009 IEEE International Conference on Information and Automation, 2009.
[25] Z. M. Bi, Y. Jin, Kinematic modeling of Exechon parallel kinematic machine, Robotics & Computer Integrated Manufacturing 27 (1) (2011) 186-193.
[26] M. A. Hosseini, H. R. M. Daniali, Kinematic analysis of Tricept parallel manipulator, IIUM Engineering Journal 12 (5) (2012) 7-16.
[27] P. Xu, C. Cheung, B. Li, et al. Kinematics analysis of a hybrid manipulator for computer controlled ultra-precision freeform polishing, Robotics & Computer Integrated Manufacturing 44 (2017) 44-56.
[28] F. Caccavale, B. Siciliano, L. Villani, The Tricept robot: dynamics and impedance control, IEEE Transactions on Mechatronics 8 (2) (2003) 263-268.
[29] M. Petko, K. Gac, G. Góra, et al., CNC system of the 5-axis hybrid robot for milling, Mechatronics 37 (2016) 89-99.
[30] M. Petko, G. Karpiel, K. Gac, et al., Trajectory tracking controller of the hybrid robot for milling, Mechatronics 37 (2016) 100-111.
[31] J. L. Olazagoitia, S. Wyatt, New PKM Tricept T9000 and its application to flexible manufacturing at aerospace industry, Sae International (2007) Paper No. 07ATC-94.
[32] L. Piegl, W. Tiller, The NURBS Book, second ed., Springer-Verlag, New York, Inc (1997) 646.