Singularity Avoidance Algorithm Based on Bounded Deviation Path Correction Method

Nei Wang\textsuperscript{1a*}, Bin Li\textsuperscript{1b}, Qiwei Lai\textsuperscript{1c}, Chao Xiang\textsuperscript{1d}, Di Wu\textsuperscript{1e}

\textsuperscript{1} Wuhan Second Ship Design and Research Institute, Wuhan, Hubei, China

\textsuperscript{a}email: neilwong2011@163.com, \textsuperscript{b}email: libinzju@sina.com, \textsuperscript{c}email: hxscabbard@qq.com, \textsuperscript{d}email: 296052049@qq.com, \textsuperscript{e}email: woody9050@126.com

*Corresponding author: \textsuperscript{a}email: neilwong2011@163.com

Abstract: Focused on the abnormal problems of path planning when a custom 6-DOF robot goes passed singular points, this paper makes descriptions based on kinematic analysis and dynamic analysis. It is pointed out that linear interpolation should not be simply abandoned but choose joint motion when singularity problems encounters. Instead, this paper suggests a singularity algorithm to modify the interpolated point sequence in the singular interval after a preliminary path planning of the interpolated point sequences, of which the modification method is called "Bounded Deviation Path Correction Method" (BDPC) in this paper. This algorithm can ensure both the accuracy of the end position and the attitude, and finally realizes the effective and reliable motion planning of the manipulator through the singular interval.

1. Introduction

Traditional 6 axis industrial robot often encounters problems of singularity, which will lead directly to that any linear or circular interpolation near singularity points is nearly impossible but only use joint motion planning. For example: Ye Bosheng\textsuperscript{[1]} mentioned that, you can first determine whether the interpolation trajectory singular points exist, if yes only joint motion could be used near singularity points. This opinion will appear too limited if cases requiring linear interpolation. Liu Haitao\textsuperscript{[2]} and Xu Wenfu\textsuperscript{[3]} respectively mentioned in their paper that by using “Singular Separation + Exponential Damping Derivative" method to deal with linear transition singular interval, which in fact only suppresses the velocity distortion of the singular range through dynamic viewpoint. J.-P. Merlet\textsuperscript{[4]} and P.S. Donelan\textsuperscript{[5]} et al. mentioned by using Jacobi matrix singular interval can be determined, but the trajectory planning method was also not provided. At present, there is no public literature on how to define singular intervals and use linear interpolation method to accomplish the path planning around the singular interval.

In fact, the research of this paper focuses on solving the singularity problem of robot motion and also provides an effective idea and algorithm to deal with path planning problems around the singular interval. The algorithm mentioned in this paper can predict singular points and define singular intervals. After that, corresponding linear interpolation method will be applied to different cases of singularity problems. The algorithm is convenient, effective and can be easily applied to the trajectory planning of a universal 6-DOF serial industrial robot.

2. Dynamic Analysis of Wrist Singularity

As an inherent nature of a 6-DOF robot, the singularity can easily cause vibration and impact of the
robot arm. Therefore, it is very important to analyze and research the singular problems and also the singular avoidance methods of robot to realize high-speed stable motion of robot. Wrist singularity is the most frequently encountered singular problem when do path planning in the working range of universal 6-DOF manipulator. Wrist singularity usually refers to a state that the angle of the 5th joint is 0 which will cause the 4th axis to coincide with the 6th joint, as show in Fig.1.

Fig.1 Common Wrist Singularity of Industrial Robot

Because the robot end and wrist are relatively fixed, and for an effective and feasible singularity analysis and singularity avoidance algorithm, here we simplify the dynamics problem of the robot’s end point to the analysis of the wrist point. The Jacobi matrix using the wrist point as reference is $J_w$, which can be written as:

$$J_w = \begin{bmatrix} z_1(p_w - p_1) & \cdots & 0_{3x1} & 0_{3x1} & 0_{3x1} \\ & & & & \\ & & & & \end{bmatrix}$$

(1)

$z_i$ represents the unit vector of the rotation motion around the joint $i$. $p_i$ represents the position vector of the base coordinates relative to coordinate origin. Also the $J_w$ as mentioned above can be simplified as:

$$J_w = \begin{bmatrix} J_{11} \ 0_{3x3} \\ J_{21} \ 0_{3x3} \end{bmatrix}$$

(2)

If defines $\dot{q} = \begin{bmatrix} \dot{q}_u \\ \dot{q}_l \end{bmatrix}$ as the joint velocity vector of a six axis robot, among which $\dot{q}_u = [\dot{q}_4, \dot{q}_5, \dot{q}_6]$. If $p_w$ is defined as a 6 $\times$ 1 linear velocity vector, then:

$$p_w = \begin{bmatrix} v_w \\ w_w \end{bmatrix} = \begin{bmatrix} J_{11} & 0_{3x3} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_u \\ \dot{q}_l \end{bmatrix}$$

(3)

Thus, the inverse kinematics problem is decomposed into 2 problems:

$$\begin{cases} \dot{q}_u = J_{11}^{-1} v_w \\ \dot{q}_l = J_{22}^{-1} (w_w - J_{21} \dot{q}_u) \end{cases}$$

(4)

From above we can find that, $\det(J_{11}) = 0$ determines the condition of the forearm singularity, which is known as position singularity. While $\det(J_{22}) = 0$ determines the wrist singularity condition, called singular attitude. Since we mainly discuss wrist singularity, we get:

$$\det(J_{22}) = - \sin(\theta_5)$$

(5)

$\theta_5 = 0$ or $\pm \pi$ will lead to wrist singularity. We define $k = \sin(\theta_5)$ as singular factor of wrist. When this factor is close to 0, the corresponding point can be called a near-singularity point, the velocity of any of the three joints increases sharply to a critical value $v_{gap}$ (set manually), causing the wrist to be singular. The point triggering this condition is a near singular point:

$$\forall (|\dot{\theta}_4|, |\dot{\theta}_5|, |\dot{\theta}_6|) \leq |v_{gap}|$$

(6)
3. Linear Interpolation Based On UDAA Method

The motion planning algorithm of the six-axis robot mentioned in this paper adopts the attitude angle planning algorithm based on the general rotation transform algorithm. Define \( k = k_i + k_j + k_k \) as a unit vector passing origin point, and the rotation matrix about rotating angle \( \theta \) around axis \( k \) can be written as:

\[
R(k, \theta) = \begin{bmatrix}
k, k \text{Vers}\theta + c\theta & k, k \text{Vers}\theta - k, s\theta & k, k \text{Vers}\theta + k, s\theta \\
k, k \text{Vers}\theta + k, s\theta & k, k \text{Vers}\theta + s\theta & k, k \text{Vers}\theta - k, s\theta \\
k, k \text{Vers}\theta + c\theta & k, k \text{Vers}\theta + k, s\theta & k, k \text{Vers}\theta + c\theta
\end{bmatrix}
\] (7)

Among above: \( s\theta = \sin \theta \), \( c\theta = \cos \theta \), \( \text{Vers}\theta = 1 - \cos \theta \).

The above formula is to establish rotation transformation formula, of which the opposite problem is to solve the equivalent rotation axis \( k \) and the equivalent rotation angle \( \theta \) by using the rotation transformation matrix. For any given rotation transformation matrix:

\[
R = R_i^0 R_2 = \begin{bmatrix}
n_i & a_i & a_i \\
n_j & a_j & a_j \\
n_k & a_k & a_k
\end{bmatrix}
\]

If \( R = R(k, \theta) \), it is easy to calculate that:

\[
\begin{align*}
k_i &= \frac{a_i - a_e}{2 \sin \theta} \\
k_j &= \frac{a_j - a_e}{2 \sin \theta} \\
k_k &= \frac{n_k - a_k}{2 \sin \theta}
\end{align*}
\]

According to equations (7) and (8), it is easy to obtain the equivalent rotation axis \( k \) and equivalent rotation angle of the planned path by using the D-H coordinates of the first and last points. Thus, according to equation (9), it is easy to determine the pose matrix of the interpolation point sequence in the process of interpolation, which can be expressed in D-H coordinates as:

\[
\{T_0, T_1, \ldots, T_i, \ldots, T_N\}
\]

where, \( T_i = \left[\begin{array}{ccc}R_{i-1} \cdot R(k, \theta_i/T_N) & P_i + \frac{\Delta P}{N} \\
0 & 0 & 0 & 1\end{array}\right]\) (9)

In the formula, \( \Delta P \) is the position change vector from the start to the end, and \( N \) is the number of interpolation steps calculated according to the step size of the motion planning. The linear interpolation realized by equation (9) is based on the uniformly-divided attitude angles by the equivalent axis. Thus, in this paper, this method is defined as UDAA(Uniformly Divided Attitude Angle Method) interpolation method. The sequence points obtained by this method will cause violent run and other anomalies near the wrist singularity point, so this paper intends to optimize this common linear interpolation method, so as to realize a new and effective singularity avoidance algorithm.

4. Singularity avoidance algorithm

4.1 Overview and flowchart of the algorithm

Based on the D-H coordinate values of the starting point and end point and by using the UDAA method described in Section 2, the joint angle sequence of linear interpolation can be calculated as bellows:

\[
\bigcup_{i=1}^{N} \{\theta_i(i), \theta_2(i), \ldots, \theta_6(i)\} = \bigcup_{i=1}^{N} \text{invKinem}(T_i)
\]

Wherein, \( \text{invKinem}(T_i) \) represents the inverse kinematic solution function for D-H coordinate
\[ \lambda_s = \frac{(j_1 \cdot j_N)}{OR(j_1 \cdot j_N)} = 1 \] denotes that the interpolation process do not pass singular points, but if 0, it is necessary to consider using the singularity avoidance algorithm to achieve smoothly passing the singularity points.

The main calculation idea of the singularity avoidance algorithm described in this paper is as follows. Firstly, the exact and the fore-and-aft singular interval are predicted and defined among the interpolation region, and then different calculation methods are applied respectively. Finally, the interpolation path is merged and updated. The flowchart of the calculation is as follows.

4.2 Path planning based on singular interval prediction

In order to accurately identify the singular interval, different from the current method of manually setting a small value interval of joint 5 as the singular interval, the wrist singular factor is used to define the near singular point, so as to qualitatively define the singular interval.

Step1. The interpolation sequence is calculated by UDAA algorithm based on the joint angle or homogeneous coordinate values of the starting point and the end point. The sequence can be denoted as \[ \{\theta_1(i), \theta_2(i), \ldots, \theta_{i+N}(i)\} \].

Step2. By formula (6), the point of determining the trigger wrist singular factor in the above sequence could be called a near singular point, the sequence number of the dynamic mutation point is \( i_0 \), the degree of joint angle 5 is \( \theta_5(i_0) \).

Step3. The \( i_0 \) homogeneous coordinate obtained by the kinematics solution is \( T_{i_0} \), with this point as the starting point and the original end point as the end point, the position quantity \( \{P_3, P_y, P_2\} \) is still interpolated according to the pre-planned interpolation path, while the attitude quantity \( \{R_x, R_y, R_z\} \) is interpolated in the joint space according to UDA (Uniformly Divided Angle Method), so as to obtain the new posterior segment sequence \[ \cup_{i=i_0}^{i+N} \{\theta'(i), \theta'_2(i), \ldots, \theta'_{i+N}(i)\} \].

Step4. Detect the sequence number of the interpolation point in the new posterior segment sequence of \( |\theta'(i) - \theta_5(i_0)| \leq \epsilon |\theta_5(i_0)| \) (where \( \epsilon \) is a small value, which can be set referred to
interpolation precision, \(1 \times 10^{-3}\), for example). If no such point, the interpolation can be completed according to the new sequence in Step 3. If there is, the serial number of this point will be denoted as \(i_1\). In general, the singular interval can be defined as a sequence \(\{\theta_1(i), \theta_2(i) ... \theta_6(i)\}\).

Step5. The \(i_1\)-th homogeneous coordinate value is \(T_{i_1}\), with this point as the starting point and the original end point as the end point, the new interpolation sequence is calculated according to the UDAA algorithm described in Section 3 of this paper, so as to update the point sequence after the singular interval is to \(\{\theta_1(i), \theta_2(i) ... \theta_6(i)\}\).

Update the above interpolation point sequences, and reorganize to form a new interpolation point sequence as the final linear interpolation trajectory.

In conclusion, according to the above singularity avoidance algorithm, the positional relationship between the starting point and the end point will cause linear interpolation to encounter the following three situations:

1) Criterion \(\lambda_s = (j_1 \cdot j_N) OR (!j_1 \cdot j_N) = 1\) represents there is no singular point, the interpolation point sequence according to the normal programming mode could be calculated as follows:

\[
\bigcup_{i=N} \{\theta_1(i), \theta_2(i) ... \theta_6(i)\} = \bigcup_{i=N} \text{invKinem}(T_i)
\]

2) Criterion \(\lambda_s = 0\) means that the interpolation process passes through singular points. Solve \(i_0\), detect the interpolation point number of \(|\theta_1'(i) - \theta_3(i_0)| \leq \epsilon |\theta_3(i_0)|, \epsilon = 10^{-3}\) in the new posterior sequence. If no point detected, use the BDPC method (detailed in section 4.2) to re-interpolate the attitude:

\[
\begin{align*}
\bigcup_{i=N} \{\theta_1(i), \theta_2(i) ... \theta_6(i)\} &= \bigcup_{i=N} \text{invKinem}(T_i) \\
\bigcup_{i=N} \{\theta_1(i), \theta_2(i) ... \theta_6(i)\} &= \bigcup_{i=N} \{\theta_1(i), \theta_2(i), \theta_3(i)\} + \bigcup_{i=N} \{\theta_4(i), \theta_5(i), \theta_6(i)\} + \\
(i-i_0) \{\theta_1(N), \theta_2(N), \theta_3(N)\} - \{\theta_4(i), \theta_5(i), \theta_6(i)\} &+ \{\delta \theta_1(i), \delta \theta_2(i), \delta \theta_3(i)\}
\end{align*}
\]

Among them, \(\{\delta \theta_1(i), \delta \theta_2(i), \delta \theta_3(i)\}\) is the posture angle correction to be obtained, and the second part denotes the singular interval.

3) Criterion \(\lambda_s = 0\) means that the interpolation process passes through singular points. Solve \(i_0\), detect the interpolation point number of \(|\theta_1'(i) - \theta_3(i_0)| \leq \epsilon |\theta_3(i_0)|, \epsilon = 10^{-3}\) in the new posterior sequence. If exit point (the serial number is \(i_1\)) detected, use the BDPC method (detailed in section 4.2) to re-interpolate the attitude:

\[
\begin{align*}
\bigcup_{i=N} \{\theta_1(i), \theta_2(i) ... \theta_6(i)\} &= \bigcup_{i=N} \text{invKinem}(T_i) \\
\bigcup_{i=N} \{\theta_1(i), \theta_2(i) ... \theta_6(i)\} &= \bigcup_{i=N} \{\theta_1(i), \theta_2(i), \theta_3(i)\} + \bigcup_{i=N} \{\theta_4(i), \theta_5(i), \theta_6(i)\} + \\
(i-i_0) \{\theta_1(N), \theta_2(N), \theta_3(N)\} - \{\theta_4(i), \theta_5(i), \theta_6(i)\} &+ \{\delta \theta_1(i), \delta \theta_2(i), \delta \theta_3(i)\}
\end{align*}
\]

Among them, \(\{\delta \theta_1(i), \delta \theta_2(i), \delta \theta_3(i)\}\) is the posture angle correction to be obtained, and the second part denotes the singular interval.

**4.3 Path correction based on BDPC method**

In Section 3.2, the method of defining the singular interval and the calculation of the interpolation
sequence in the linear interpolation interval are described in detail. The remaining problem is how to revise the interval planned by UDA method within the defined singular interval to meet the established pose error requirements. The correction method is the "Bounded Deviation Path Correction Method", this article is abbreviated as the BDPC method. The method and the calculation process of the attitude angle correction are as follows:

1) Taking a typical example of singularity-point avoiding calculation, i.e., points $i_0$ and $i_1$ in the algorithm described in Section 3.2 do exist as examples, the attitude matrix sequence $R$ could be obtained as $\bigcup_{i \in A_i} \{ R_{i-3,i} \}$ according to the attitude planning algorithm described in Section 3.

2) Equalize the interval between point $i_0$ and in the joint space according to the previous step length, and then the calculate the D-H coordinate matrix of the corresponding Cartesian space, and the pose matrix could be denoted as sequence $S$: $\bigcup_{i \in A_i} \{ S_{i-3,i} \}$.

3) Calculate the deviation error between each point of sequence $R$ and $S$. The deviation error is expressed as the equivalent rotation angle $\{ \phi(i) \}$ of sequence $R$ relative to sequence $S$:

$$V_{3;3}(i) = S(i) \text{Rot}(k, \frac{\tau e^{\phi(i)/\delta_{R}^\max}}{e + e^{\phi(i)/\delta_{R}^\max}} \text{sgn}(\phi(i)) \delta_{R}^\max - \phi(i))$$

4) Check the error limit, if $|\phi(i)| \leq \delta_{R}^\max$, this point does not need to be corrected, otherwise $\phi(i)$ does not tend to 0, and the rotation axis $k$ calculated in step 3) just no longer appears ill-conditioned solution[4], the corresponding pose matrix sequence $V$ can be calculated according to the following formula to meet the deviation requirements:

$$V_{3;3}(i) = S(i) \text{Rot}(k, \frac{\tau e^{\phi(i)/\delta_{R}^\max}}{e + e^{\phi(i)/\delta_{R}^\max}} \text{sgn}(\phi(i)) \delta_{R}^\max - \phi(i))$$

Where, $\text{sgn}()$ is a symbolic function, which will be 1 when $\phi(i)$ is positive and -1 otherwise. $f(\zeta) = \frac{\tau e^\zeta}{e + e^\zeta}$ is the gain function, $\zeta = \frac{\phi(i)}{\delta_{R}^\max}$, $\tau$ is a the gain coefficient. Gain function can ensure that the sequence $R$ approaches the sequence $S$ more smoothly, thus ensuring the attitude accuracy of the robot’s end-effector.

5) Use the pose matrix sequence $V$ and the position quantity $\{ \gamma_i, \gamma_i, \gamma_i \}$, inverse solution can obtain the joint angle sequence of the new singular interval. Thus, all interpolation points of the singular interval mentioned in Section 3.2 can be obtained and afterwards substitute them into the initial calculation sequence points of the singular interval.

5. Simulation and verification

Simulation verification is carried out on a six-axis manipulator, and the joint angles given at the starting point and the end point were respectively $\{5.5, 56.1, 10.3, 20.2, 36.3, 10 \}$ and $\{10.3, 51.2, 21.3, 30.2, -60.2, 22 \}$. First, the general linear interpolation method, namely UDA method, is used to do the trajectory planning. The simulation results of joint Angle and velocity variation obtained are shown in Fig. 3.
Fig. 3 Linear interpolation results using UDAA method

The UDA method described in this paper is used for re-planning, and the results are shown in Fig.4.

Fig. 4 Interpolation results using UDA method

After the UDA method described in this article is used for motion planning, sharp change of the joint angle is significantly improved, and the robotic arm can pass the singularity in the path relatively smoothly. However, analysis shows that when the singular interval is interpolated only by simple UDA method, the attitude error will become larger and larger, as shown in Fig.5.

Fig.5 Attitude error trend using UDA method

Use the BDPC method to correct the interpolation point sequence of the singular interval, and re-calculate the joint angle of the manipulator to obtain the corresponding D-H coordinate value sequence. After that, extract the X, Y, and Z axes of the Cartesian coordinate system to get the left
graph of Fig.6, it is obvious that the robot end’s linear planning is executed with high precision. Also extract the corrected singular interval and the theoretical value of the attitude angle error in the vicinity, we can get the result of the BDPC algorithm shown in thick line of the Fig.6’s right graph, compared by the error value of Fig.5 shown in thin line.

Thus, the wrist point or end track of the manipulator is still straight without any distortion. Moreover, the attitude angle error can be controlled within the set bounded error range, which indicates that the proposed algorithm can control the mechanical arm wrist point to go smoothly and effectively through the wrist singular interval of joint 5 as 0 in the premise of ensuring the desired attitude error by linear interpolation.

6. Conclusions
Because this algorithm adopts the method of position and attitude separation to calculate and analyze, the terminal path is controlled by the allowable deviation of the attitude angle on the planned path while the accuracy of the terminal position is guaranteed, the smoothness and stability of the motion of the manipulator near the singularity point are guaranteed. It is worth pointing out that the optimization strategy of the singularity-avoidance algorithm in this paper is basically to use D-H coordinates in Cartesian space to work out the linear interpolation of position precision and allowable deviation control of attitude data. However, in the final execution end, the change of robot pose depends on the movement of each joint motor to realize the interpolation process. Therefore, it is necessary to transfer the trajectory of the task space to the joint space, and then to execute the trajectory sequence of the interpolation points in the joint space, thereafter to obtain the joint trajectory to meet the limited physical performance of the robots (such as motor capacity or speed acceleration limit). The interpolation of point sequence in joint angle space can be performed by three-to-seven spline curves\(^{[11]}\), which is already relatively mature and thus not necessary to describe in this paper.

For the six-axis serial robot, the singularity avoidance algorithm of other common singular points such as velocity singularity and position singularity\(^{[1]}\) can also refer to this paper for analysis and calculation.

References
[1] Ye Bosheng, Guo Xianjin, Xiong Shuo et al. Inverse Solution Algorithm of 6-axis Industrial Robot Considering Joint Properties [J]. Journal of Huazhong University of Science and Technology [J]. Journal of Huazhong University of Science and Technology, 2013,41(3): 68-72.
[2] Liu Haitao. Research on High-Speed and High-Precision Control of Industrial Robot [D]. South China University of Technology,2012.
[3] Xu Wenfu, Liang Bin, Liu Yu et al. A Novel Singularity Avoidance Algorithm for PUMA Type Robots [J]. Acta Automatica Sinica 2008,34(06):670-675.
[4] Cai Zixing, Xie Bin et al. Robotics [M]. Beijing: Tsinghua University Press,2015.
[5] Jiang Xinsong. Robotics and Control Problems in Robotics [J]. Robots, 1990, 12(5): 1-12.
[6] Taylor R. Planning and Execution of Straight Line Manipulator Trajectories in Robot Motion [J]. Massachusetts: The MIT Press, 1983.
[7] Tchon K. and R. Muszynski. Singular inverse kinematic problem for robotic manipulators: a normal form approach [J]. Robotics and Automation, IEEE Transactions on, 1998, 14(1): 93-104.
[8] Yin Shibin. Research on Positioning Error Hierarchical Compensation and Precision Maintenance Method of Industrial Robot [D]. Tianjin University, 2015.
[9] Zhang Xiaoping. Calibration method and experimental study of six-degree-of-freedom articulated robot. Huazhong University of Science and Technology, 2013.
[10] Liu Songguo. Research on Motion Optimization and Trajectory Tracking Control of 6-DOF Series Robot [D]. Zhejiang University, 2009.
[11] Liu Chengli, Luan Nan. Continuous Point-to-point Motion Planning Algorithm Based on Splines in Joint Space [J]. Machinery & Electronics, 2014(8): 74-78.