Kinematics modeling of redundant manipulator based on screw theory and Newton-Raphson method

Dawei Ge
School of Electronic Information, Beijing Information Technology College, Beijing, 100015, China
*Corresponding author’s e-mail: gedw@bitc.edu.cn

Abstract. In this paper, forward kinematics and inverse kinematics algorithms are proposed to solve the problem that the redundant manipulator has more freedom than the traditional manipulator and cannot directly solve the inverse kinematics analytical solution. Firstly, the forward kinematics model is established through the screw theory; secondly, Newton-Raphson method is used to solve the inverse kinematics of the manipulator. Finally, the algorithms of redundant manipulator are verified through an example simulated by Matlab Robotics toolbox. The results show that the kinematic algorithms are correct, which provides a good algorithm basis for subsequent dynamic control.

1. Introduction
Compared with ordinary 6 DOF manipulators, 7 DOF redundant manipulators have become a research hotspot in the industry because of their good characteristics of obstacle avoidance and singularity avoidance. However, its kinematic modeling has always been a difficult problem in the field of robots. The kinematics modeling of the manipulator mainly studies the motion relationship between the manipulator joint and the rigid body of the robot, which is the basis of the trajectory planning and motion control of the robot.

At present, the main methods to solve the forward kinematics of the manipulator are D-H parameter method and screw method. D-H parameter method is more mature and widely used. When D-H parameter method is used to solve the kinematics of the robot, it is necessary to establish a coordinate system on each joint, and the posture (position and orientation) of the coordinate system is different according to the motion mode of each joint, which makes the process of establishing the coordinate system more complicated. The screw method is used to describe the position and orientation of each rotating shaft. Therefore, it is simple to model and the geometric meaning is clear.

The methods for solving the inverse kinematics problem of the manipulator are mainly geometric method, analytical method and numerical method. Geometric method has intuitive advantages, but it is too complex for redundant manipulator. The analytical method is mainly aimed at the manipulator mechanism conforming to the Pieper criterion, and it is also too complex for the manipulator that does not meet the Pieper structure. The numerical method is to establish the iterative equation according to the kinematic relationship of the manipulator and obtain the approximate solution.

In this paper, POE (Product Of Exponentials) is used to build a forward dynamics model of a 7 DOF manipulator based on the screw theory, and Newton-Raphson method, which is a kind of numerical methods, is used to solve the forward dynamics problem of a 7 DOF manipulator.
2. Forward kinematics modeling based on screw theory

2.1. Screw Theory
Assuming that the inertial coordinate system \( \{S\} \) and the tool coordinate system fixed on the rigid body \( \{T\} \), then the rotation matrix of the rigid body relative to the inertial coordinate system is \( R^S_T \), and the posture of the rigid body relative to the inertial coordinate system is \( H^S_T \).

We can obtain:

\[
R^S_T = e^{\theta \tilde{\omega}}, \theta \text{ is the angle of rotation}, \quad \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \text{ is the unit orientation vector of the rotating shaft.}
\]

\( \tilde{\omega} \) is the antisymmetric matrix of \( \omega \),

\[
\tilde{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.
\]

According to the exponential mapping relationship,

\[
e^{\theta \tilde{\omega}} = I + \tilde{\omega} \sin \theta + \tilde{\omega}^2 (1 - \cos \theta)
\] (1)

Let \( r \) be a point along the rotation shaft \( \omega \), then we can obtain:

\[
\xi = \begin{bmatrix} \omega \\ v \end{bmatrix}, \quad \hat{\xi} = \begin{bmatrix} \tilde{\omega} \\ 0 \end{bmatrix}
\] (2)

Where \( v = r \times \omega \), \( \xi \) is the screw coordinate, \( \hat{\xi} \) is the screw.

According to Chasles theorem, any rigid body motion can be realized by spiral motion, i.e. by a compound motion of rotation around an axis and movement along that axis.

According to the POE formula of exponential product, the transformation matrix can be obtained:

\[
e^{\theta \hat{\xi}} = e^{\theta \xi} (I - e^{\theta \xi}) \cdot r + \theta \omega \omega^T v
\] (3)

So the calculation formula of tool coordinate system relative to inertial coordinate system is:

\[
H^S_T(\theta) = e^{\theta \hat{\xi}} H^S_T(0)
\] (4)

\( H^S_T(0) \) is the initial configuration of the robot.

For the manipulator with \( n \) rotating shafts,

\[
H^S_T(\theta) = e^{\xi_1 \theta_1} e^{\xi_2 \theta_2} e^{\xi_3 \theta_3} \cdots e^{\xi_n \theta_n} H^S_T(0)
\] (5)

The Jacobi matrix \( J_\xi(\theta) \) consists of \( \omega \) and \( v \). The \( i \)th column of the Jacobian matrix \( J_\xi(\theta) \) is \( [\omega_i, v_i] \).

2.2. Forward kinematics modeling
This paper takes ABB IRB14000 left arm as an example for analysis, and its structural diagram is shown in Figure 1.
Fig. 1 ABB IRB14000 left arm structural diagram

\[ l_1 = 166, \ l_2 = 30, \ l_3 = 251.5, \ l_4 = 40.5, \ l_5 = 40.5, \ l_6 = 265, \ l_7 = 27, \ l_8 = 36. \]

In Figure 1, we can see two coordinate systems, the inertial coordinate system \( \{ S \} \) and the tool coordinate system \( \{ T \} \). \( \omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7 \) are the orientation unit vectors of the rotation shafts of seven joints relative to the inertial coordinate system \( \{ S \} \). \( r_1, r_2, r_3, r_4, r_5, r_6, r_7 \) are the position vectors of the rotation shafts of seven joints relative to the inertial coordinate system \( \{ S \} \).

From Figure 1, we can obtain:

\[
\begin{align*}
\omega_1 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ 
\omega_2 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ 
\omega_3 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ 
\omega_4 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ 
\omega_5 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \ 
\omega_6 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ 
\omega_7 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\
\end{align*}
\]

\[
\begin{align*}
r_1 &= \begin{bmatrix} 0 \\ -30 \\ 0 \end{bmatrix}, \\
r_2 &= \begin{bmatrix} 0 \\ 0 \\ 166 \end{bmatrix}, \\
r_3 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \\
r_4 &= \begin{bmatrix} 40.5 \\ 0 \\ 147.5 \end{bmatrix}, \\
r_5 &= \begin{bmatrix} 0 \\ 0 \\ 458 \end{bmatrix}, \\
r_6 &= \begin{bmatrix} 0 \\ 0 \\ 431 \end{bmatrix}, \\
r_7 &= \begin{bmatrix} 0 \\ 0 \\ 458 \end{bmatrix} \\
\end{align*}
\]

We can get the posture of tool coordinate system relative to inertial coordinate system:

\[
H_T^S(\theta) = e^{\xi_1 \theta_1}e^{\xi_2 \theta_2}e^{\xi_3 \theta_3}e^{\xi_4 \theta_4}e^{\xi_5 \theta_5}e^{\xi_6 \theta_6}e^{\xi_7 \theta_7}H_T^S(0) \\
(6)
\]

The initial configuration of the robot \( H_T^S(0) \) is:

\[
\begin{bmatrix} 0 & 0 & 1 & 341.5 \\ 0 & 0 & 1 & 458 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

3. Solving inverse kinematics problem by Newton-Raphson method

Inverse kinematics is to solve the angles of joints \( \theta \in \mathbb{R}^7 \) under the condition of known robot posture \( H_T^S \).

Newton-Raphson iteration method has the advantages of fast convergence and self-correction, and is an important method for solving nonlinear equations.

Newton-Raphson iterative method is a good method to solve inverse kinematics problem.

Let \( p^e \) be the expected ending position relative to \( \{ S \} \), and the actual ending position relative to \( \{ S \} \) is \( p(\theta^d) \) when the joint angle vector of the manipulator is \( \theta^d \), then the error function is \( g(\theta_d) = p_e - p(\theta^d) \), the objective is to find the joint coordinate \( e \) and ensure that \( g(\theta_d) < \text{setting least error} \).

The specific calculation steps are as follows:
1. Initialization: According to the current angle vector $\theta^0$, the forward kinematics is used to obtain the current pose $H^S_0$, so as to obtain the actual ending position $p(\theta^0)$ and the current coordinate system Jacobi matrix $J_S(\theta^0)$.

2. Calculation
$$\Delta \theta^0 = J_S^{-1}(\theta^0)(p_e - p(\theta^0))$$

3. Obtain new valuation
$$\theta^1 = \theta^0 + \Delta \theta$$

4. Continuing the iteration and generating new $\theta$ values, eventually converging at $\theta^d$.

Note that the Moore-Penrose pseudoinverse form $J_S^+$ can be used to avoid the irreversibility of coordinate Jacobi $J_S$ in the solution process.

4. Simulation and verification
To verify the correctness of the above model, Matlab Robotics toolbox is used for verification. From Fig. 1, we can tabulate D-H parameter table (Table 1).

| $\theta_i$ | $d_i$/mm | $a_i$/mm | $\alpha_i$/° | $\text{offset}$/° | Angle Limit          |
|-----------|-----------|-----------|--------------|-------------------|----------------------|
| $\theta_1$| 166       | -30       | -90          | 0                 | (-168.5,168.5)       |
| $\theta_2$| 0         | 30        | 90           | 0                 | (-143.5,43.5)        |
| $\theta_3$| 251.5     | 40.5      | -90          | 0                 | (-123.5,80)          |
| $\theta_4$| 0         | 40.5      | -90          | -90               | (-290,290)           |
| $\theta_5$| 265       | 27        | -90          | 180               | (-88,138)            |
| $\theta_6$| 0         | -27       | 90           | 0                 | (-229,229)           |
| $\theta_7$| 36        | 0         | 0            | 0                 | (-168.5,168.5)       |

With Matlab Robotics toolbox, the 7 DOF manipulator simulation model is built, shown in Figure 2.

Fig. 2  Simulation model using matlab robotics toolbox

Firstly, 50 groups of angle vectors are randomly given, and the forward kinematics results (postions and orientation) are solved by using the screw method, and the position $p_{screw}$ and angle vector $\theta$ are obtained. Then $\theta$ is brought into the Robotics Toolbox model to obtain the position $p_{model}$. Compared with $p_{screw}$, we can get $\text{error} = p_{screw} - p_{model}$. The error diagram is shown in Fig. 3.
As shown in Fig. 3, the maximum position error is no more than $1 \times 10^{-13} mm$, which meets the accuracy requirements.

![Fig 3. Position error diagram](image1)

Then, we randomly give the initial posture and target posture, wherein 80 times are interpolated. The joints angles of the manipulator moving from the initial posture to the end posture changes with time as shown in Fig. 4.

![Fig 4. 7 joints angles of the manipulator moving from the initial posture to the end posture](image2)

5. Conclusion
In this paper, the kinematics analysis of 7 DOF redundant manipulator is carried out. The forward kinematics and inverse kinematics models are established by using the screw method and Newton-Raphson method, and the simulation implementation of this method is carried out. The experimental results show that this method has high accuracy and good practical value for the subsequent motion control.

References
[1] Kevin, M. L., Frank, C. P. (2017) Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press, Cambridge.
[2] Siciliano, B., Sciavicco, L., Villani, L. (2010) Robotics: modelling, planning and control. Springer Publishing Company, New York.
[3] John, J.C. (2003) Introduction to Robotics: Mechanics and Control. Prentice Hall, Upper Saddle River.

[4] Wang, G.J., Shu, Q.L., Wang, J. (2018). Kinematics Modelling and Analysis for 6R Industrial Robot Based on Screw Theory. Machine tool & Hydraulics, 46:35-42

[5] Liu, G.P., Yang, X.Y., Zhong, F.F., Wang, D.H. (2021) Kinematics Screw Inverse Solution and Simplified Algorithm of 6-DOF Manipulator. Modular Machine Tool & Automatic Manufacturing Technique, 9:11-15

[6] Shi, J.P., Li, P.S., Liu, G.P., Zhu, L. (2021) Inverse Kinematics Solution of Redundant Manipulator Based on Hybrid Learning Fruit Fly Optimization Algorithm. Transactions of the Chinese Society for Agricultural Machinery, 52:410-416