Inverse kinematics solution and motion simulation of seven-degree-of-freedom ascending platform based on neural network

Jingyuan Wu, Xiangyi Ren
Leeds Joint School, Southwest Jiaotong University, Chengdu, Sichuan, 611756

Abstract. A pseudoinverse-based inverse kinetic solving scheme for 7-degree-of-freedom redundant aerial platform robotic arm, the scheme combines Minimum Velocity Norm scheme and damped least-square method, a close-loop damped minimum velocity norm scheme is set up which allows to avoid the joints’ limitation and to ensure the solution’s stability near the singularity position. Further, a genetic algorithm optimized BP neural network is built. Genetic algorithm can improve the training efficiency and avoid trapping in local optimal solution. The application of neural network can ensure the real-time of solution.

1. Introduction
The manipulators of the ultra-high meter ascending platform are mostly 2RP4R seven-degree-of-freedom redundant manipulators [1]. Redundant degrees of freedom can ensure the flexibility of the manipulator in space, but this kind of manipulator cannot use the traditional inverse kinematics method to determine each Joint trajectory [2]. At the same time, for the inverse kinematics solution of this type of robotic arm, three requirements should be considered: (1) The solution of the trajectory needs to pay attention to the stability of the solution near the singular pose [3]. The joint speed will also be extremely high at a small terminal speed, and excessive joint speed will destruct the robotic arm [4]. At the same time, large joint acceleration will cause the shaking of the terminal manned platform and threaten the safety of rescuers [5]. (2) The joint motion trajectory must meet the motion limit requirements of each joint, otherwise it will cause damage to the mechanical structure [6]. (3) The solution error is small and converges.

In recent years, domestic and foreign scholars have proposed some algorithms for solving the inverse kinematics of redundant manipulators, such as iterative algorithm, geometric parameter method, etc [7]. This paper will take a brand ascending platform rescue vehicle as an example (Figure 1), a generalized inverse closed-loop solution algorithm that combines the weighted minimum norm method and the error-damped Jacobian matrix is used to solve the motion trajectory of each joint when the robot arm reaches the specified position [8].

Further, the neural network algorithm optimized by genetic algorithm is used to learn the joint initial position, end position, and joint speed [9]. The constructed neural network can improve the real-time calculation speed [10].
Figure 1. A brand ascending platform rescue vehicle

2. Establish the positive kinematics equation of the boom

2.1. Establishment of Homogeneous Transformation Matrix
The positive kinematics equation of the boom is obtained by the DH parameter method. First, define the DH coordinate system on each joint to describe the current spatial pose of each joint, and use the homogeneous transformation matrix to obtain the posture relationship of the adjacent boom, and finally get the spatial pose transformation of each joint relative to the base coordinate system.

Refer to the mechanical arm architecture designed in Figure 1 to simplify the components according to the connection method. The arm label from the origin to the end of the manned platform is 0, 1, 2 ... 7, the simplified model is shown in Figure 2, and the DH parameters are shown in Table 1.

![Figure 2. The sketch map of the robotic arm and the established joint coordinate system](image)

Table 1. DH parameters

| θ   | d   | a   | α   |
|-----|-----|-----|-----|
| θ₁  | 0   | 0   | 90  |
| θ₂  | 0   | 0   | 90  |
| 0   | d₃  | 0   | 90  |
| θ₄  | 0   | l₄  | 0   |
| θ₅  | 0   | l₅  | 0   |
| θ₆  | 0   | 0   | 90  |
| θ₇  | 0   | l₇  | 0   |

The DH method is used to model the kinematics of the robotic arm. The homogeneous transformation matrix of the coordinate of the i-th link to the i-1th link is shown in equation (1)
\[
A_i = \begin{bmatrix}
\cos(\theta_i) & \sin(\theta_i) & a_{i-1} \cos(\theta_i) \\
\sin(\theta_i) & \cos(\theta_i) & a_{i-1} \sin(\theta_i) \\
0 & \sin(a_{i-1}) & \cos(a_{i-1}) \\
0 & 0 & 1
\end{bmatrix}
\] (1)

The positive kinematics model of the seven-degree-of-freedom manipulator is obtained as shown in equation (2):

\[
T_7 = A_1(\theta_1)A_2(\theta_2)A_3(d_3)A_4(\theta_4)A_5(\theta_5)A_6(\theta_6)A_7(\theta_7)
\]

\[
= \begin{bmatrix}
n_x & n_y & n_z & p_x \\
n_y & o_y & a_y & p_y \\
0 & a_z & o_z & p_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (2)

2.2. The Establishment Of Jacobian Matrix

In this paper, the differential transformation method is used to calculate the Jacobian matrix of the manipulator.

(1) If joint i is a rotating joint, then

\[
J_i = \begin{bmatrix}
-n_x p_y + n_y p_x \\
-o_x p_y + o_y p_x \\
-a_x p_y + a_y p_x \\
n_z \\
o_z \\
a_z
\end{bmatrix}
\] (3)

(2) If joint i is a sliding joint, then

\[
J_i = \begin{bmatrix}
n_z \\
o_z \\
a_z \\
0 \\
0 \\
0
\end{bmatrix}
\] (4)

The n, o, a, p in equations (3) and (4) represents the transformation matrix \( T_{i-1} \). The four column vectors in, the specific calculation method is as formula (5):

\[
i^{-1}T = A_i A_{i+1} ... A_7
\] (5)

Then the differential motion of the mechanical arm can be expressed as:

\[
\begin{bmatrix}
\frac{dx}{dt} \\
\frac{dy}{dt} \\
\frac{dz}{dt} \\
\frac{\delta x}{dt} \\
\frac{\delta y}{dt} \\
\frac{\delta z}{dt}
\end{bmatrix} = \begin{bmatrix}
J_1 & ... & J_7
\end{bmatrix} \begin{bmatrix}
\frac{d \theta_1}{dt} \\
\frac{d \theta_2}{dt} \\
\frac{d \theta_3}{dt} \\
\frac{d \theta_4}{dt} \\
\frac{d \theta_5}{dt} \\
\frac{d \theta_6}{dt} \\
\frac{d \theta_7}{dt}
\end{bmatrix}, \dot{x} = J \dot{\theta}
\] (6)

In formula (6) \( dx, dy, dz \) respectively, the differential translational motion along the xyz axis, \( \delta x, \delta y, \delta z \). It is a differential rotation motion around the xyz axis.
3. Inverse kinematics solution

3.1. Weighted Least Norm Method

Introduce a weighting matrix \( w \) to avoid the joint angle exceeding the limit. Its solution is as follows:

Restrictions on section variables

\[
\hat{k}_{i_{\text{min}}} < \hat{k}_i < \hat{k}_{i_{\text{max}}}, \quad i = 0, 1, \ldots, 7
\]  

(7)

To achieve the joint avoidance limit, an optimization objective function is added \( O(h) \) and its expression is:

\[
O(h) = \sum_{i=1}^{7} \frac{(\hat{k}_{i_{\text{max}}} - \hat{k}_{i_{\text{min}}})^2}{(\hat{k}_{i_{\text{max}}} - \hat{k}_i)(\hat{k}_i - \hat{k}_{i_{\text{min}}})}
\]  

(8)

Where \( n=7, \hat{k}_{i_{\text{max}}} \) with \( \hat{k}_{i_{\text{min}}} \) is the maximum and minimum value allowed for the \( i \)-th joint variable. Further, will \( O(h) \) correct \( \hat{k}_i \) after finding the partial derivative, the result is as follows:

\[
\frac{\partial O(h)}{\partial \hat{k}_i} = \sum_{i=1}^{7} \frac{(2\hat{k}_{i_{\text{max}}}-\hat{k}_i)(\hat{k}_{i_{\text{max}}} - \hat{k}_{i_{\text{min}}})^2}{(\hat{k}_{i_{\text{max}}} - \hat{k}_i)(\hat{k}_i - \hat{k}_{i_{\text{min}}})^2}
\]  

(9)

As the joint variable approaches the limit, \( O(h) \) is increased to infinity when the joint variable reaches the limit value, when running within the joint limit range, \( \frac{\partial O(h)}{\partial \hat{k}_i} \) almost 0, when the joint variable is close to the limit value, its value is infinite.

The weighting matrix \( w \) is a diagonal matrix, and its diagonal elements are the weighting coefficients. The expression is as follows:

\[
w_i = 1 + \left| \frac{\partial O(h)}{\partial \hat{k}_i} \right|, \quad i = 1, 2, \ldots, 7
\]  

(10)

When the joint variable changes to the limit value, the weight coefficient gradually increases, and the joint motion speed decreases. When the joint variable approaches the limit value, the weight coefficient tends to infinity, and the joint motion almost stops.

The formula for calculating the inverse kinematics solution using the weighted minimum norm method is:

\[
\dot{\theta} = W^{-1}J^T(JW^{-1}J^T)^{-1}\dot{x}
\]  

(11)

Define \( J^+_W = W^{-1}J^T(JW^{-1}J^T)^{-1} \) For weighted pseudo-inverse, there are:

\[
\dot{\theta} = J^+_W \dot{x}
\]  

(12)

3.2. Damped Least Squares

Therefore, the damping term is introduced at the singular point, and its mathematical expression is:

\[
\min(\|\dot{x} - J\dot{h}\|^2 + \lambda^2\|\dot{h}\|^2)
\]  

(13)

This paper uses a damping coefficient that combines the condition number and the smallest singular value.
There:
\( \sigma \) is the singular value matrix of Jacobian matrix, \( \sigma_m \) is the minimum singular value of Jacobian matrix, \( \sigma_0 \) is the threshold value of singular value. \( k \) is the conditional number, and its expression is as follows:

\[
 k = \frac{\sigma_1}{\sigma_m} \tag{15} 
\]

\( \sigma_1 \) introducing the velocity damping term into the weighted minimum norm represented by equation (7), the solution formula for the weighted minimum norm with singular robustness is as follows:

\[
 \dot{\theta} = W^{-1}J^T(W^{-1}J^T + \lambda^2 I)^{-1}x \tag{16} 
\]

There, \( J^* = W^{-1}J^T(W^{-1}J^T + \lambda^2 I)^{-1} \)

3.3. Weighted Least Norm Method for Closed Loop

In order to improve the accuracy of inverse kinematics solution, the weighted minimum norm method is closed-loop. In the solution process, the error between the end pose of the robot arm and the target pose is calculated simultaneously, and the error is eliminated through multiple iterations until it is lower than expected threshold.

4. Establishment of bp neural network

The BP neural network takes the current joint angles and the expected end displacements in three directions in space as a total of 10 inputs, and the current joint running speed of a total of 7 quantities as the output. Construct a 2-layer BP neural network. Its network structure is as shown below Shown:

![Neural network topology](image)

**Figure 3.** Neural network topology
Using a multi-layer network can improve the calculation accuracy, but it will also increase the solution time accordingly. Therefore, a 2-layer network is used here. The fitness function is the sum of the squared error of the joint velocity. The activation function of the input layer and hidden layer using Sigmoid, the output layer function uses linear Purlin.

5. Solve the simulation

This section uses MATLAB simulation to analyze the joint angle, angular velocity, and end solution error obtained by using the closed-loop damping weighted minimum speed norm algorithm described above under the given trajectory and attitude limitations of the manned platform.

It is known that the length of each boom is \( l_4 = 5.8m, l_5 = 4.85m, l_7 = 4.25m \). Initial value of each joint variable \( \hat{\theta}_0(5^\circ, -80^\circ, 8.5m, -80^\circ, 50^\circ, 50^\circ, 5^\circ) \), the limit range of joint variables of the boom manipulator is as follows: \( \hat{\theta}_{\text{min}}(0^\circ, -90^\circ, 8.5m, -90^\circ, 60^\circ, 60^\circ, -90^\circ) \), \( \hat{\theta}_{\text{max}}(180^\circ, 90^\circ, 22.5m, 90^\circ, 90^\circ, 90^\circ, 90^\circ) \).

![Figure 4. Space Expected Trajectory](image)

According to the desired end trajectory, the inverse kinematics method is used to obtain the joint variables of the rotary joint and the translation joint.

Using the above data, take 70 groups as the training set and 30 groups as the test set to train the BP neural network. After 50 iterations, the training results meet the accuracy requirements.

![Figure 5. Planned joint displacement values](image)
Figure 6. Planned value of angular velocity of rotating joint

Figure 7. Planned values of translational joint angular velocity

Figure 8. End error

Figure 9. Planned values of angle of rotation joint

Observing the above solution results, given the expected motion trajectory of the platform, the motion trajectory of each joint can be obtained, and the change is relatively stable and does not exceed
the joint limit, which proves that the target optimization function is effective. Observation of the variable speed planning of each joint can be seen, the acceleration is relatively stable, which proves that the set damping coefficient is effective. And the neural network learning result is more accurate. The error becomes a convergence trend.

The traditional inverse kinematics algorithm takes about 0.1s to solve each joint trajectory of the movement, and it takes only 0.027s to solve the joint motion trajectory through the neural network of the training institute.

6. Conclusion
This paper introduces a closed loop inverse kinematics algorithm based on the pseudo-inverse of the Jacobian matrix combined with the weighted least norm method and the least damping square method. The algorithm achieves joint circumvention through the weighted least norm method, and the damped least square method. The multiplication guarantees the stability of the solution at the singular position. The closed-loop structure of the algorithm ensures that the error is small and has convergence. The simulation verification of the operation of the seven-degree-of-freedom ascending rescue platform further proves the effectiveness and feasibility of the method. Moreover, the solution method is universal and can be used to find various redundant degrees of freedom inverse kinematics solutions of the robotic arm. Further training of the neural network greatly improves the speed of inverse kinematics solution, and its role in the field of kinematics solution can be further explored.

References
[1] Liu Shiping, Cao Junfeng, Sun Tao, Hu Jiangbo, Fu Yan, Zhang Shuai, Li Shiqi. Inverse kinematics analysis of redundant manipulator based on BP neural network [J]. China Mechanical Engineering, 2019, 30(24): 2974-2977+2985.
[2] Chen Jinliang. Research and Application of Motion Control Technology of Seven Degrees of Freedom Redundant Manipulator [D]. Huazhong University of Science and Technology, 2019.
[3] Wang Zhihao. Research on the Design and Kinematics Calibration Method of Offset Seven Degrees of Freedom Manipulator System [D]. Huazhong University of Science and Technology, 2019.
[4] Gao Han, Zhang Minglu, Zhang Xiaojun, Gao Baijun, Huang Xiaochen. Kinematics analysis of the 7-degree-of-freedom manipulator [J]. Modern Manufacturing Engineering, 2017, 12(08): 29-33+77.
[5] He Xiangyang. Research on the direct teaching method of the 7-DOF humanoid manipulator [D]. Huazhong University of Science and Technology, 2017.
[6] Liu Shiping, He Xiangyang, Han Liangliang, Li Shiqi, Fu Yan. Self-motion optimization of redundant robotic arms for robot astronauts [J]. Manned Spaceflight, 2017, 23(02): 143-149.
[7] Yu Qiulei. Research on trajectory planning and tracking technology of seven-degree-of-freedom redundant manipulator [D]. Huazhong University of Science and Technology, 2016.
[8] Guan Xiaoqing, Chang Qing, Liang Guanhao, Ge Zhuo. A multi-movement obstacle avoidance algorithm for redundant manipulators [J]. Computer Measurement and Control, 2015, 23(08): 2802-2805.
[9] Yan Wenbin, Sun Lei. An inverse kinematics optimization algorithm for a seven-degree-of-freedom redundant manipulator [J]. System Science and Mathematics, 2015, 35(07): 823-837.
[10] Dong Xiaoxing, Li Ge, Liu Gangfeng, Zhao Jie. Kinematics and Cartesian impedance control methods of redundant space manipulators [J]. China Mechanical Engineering, 2014, 25(01): 36-41.