Workspace analysis and kinematics simulation of 6-DOF upper limb exoskeleton robot

HuaBin Liu¹, ZongMiao Dai¹, XueFeng Peng¹

¹ Seventh thirteen Institute of China Shipbuilding Industry Corporation, Zhengzhou, Henan, 450015, China

*Corresponding author’s e-mail: huabinsky@yeah.net

Abstract: In this paper, the outer non-humanoid robot 6-DOF upper limb bone as the research object, the establishment of a forward kinematics model of the robot be modeled, using the DH parameter method in Creo software to get the robot based on Monte Carlo Method Point cloud image of workspace. Based on ADAMS, the dynamic of the joint driving torque is solved, and the torque change curve of each joint during the vertical lifting of the arm is obtained. The simulation results show that the working space at the end of the six-degree-of-freedom upper-limb exoskeleton robot model can meet the working requirements, and the joint torque is relatively small, which lays the foundation for the subsequent trajectory planning and drive device design of the upper-limb exoskeleton robot.

1. Introduction

The upper limb exoskeleton robot can enhance the human behavior of moving and lifting operations. It can be widely used in warehouses, logistics and rescue and relief materials to improve work efficiency; The non-human-like 6-DOF upper limb exoskeleton robot is different from the human arm in structure and movement mode. It realizes the end posture change through a specific spatial configuration. Compared with the traditional upper limb exoskeleton, it has the advantages of simple operation and compact structure.

Workspace is the set of all positions end of the robot can reach a point of reference, it is an important measure of the ability of the robot to work [1]. The methods of analyzing the working space of the robotic arm mainly include geometric, analytical and numerical methods. Monte Carlo method is a typical numerical method, its solution process is simple, it is widely used in the analysis of robot workspace. [2] Shengliang Xie et al. used the D-H method to solve the positive kinematics equation of the dual-arm robot, and used the angle division method and extreme value theory to obtain the boundary of the working space. [3] Zhizhong Liu et al. [4] proposed an algorithm to regenerate random points in the small neighborhood of the generated boundary points in response to the problems existing in the Monte Carlo calculation of the space boundary. Zhou Lu et al. [5] used ADAMS to solve the joint torque of the dual-arm robot under specific working conditions, and analyzed the selection of driving joints.

In this paper, the D-H parameter method is used to establish the positive kinematics model of the robot, based on the Monte Carlo method to solve the upper limb exoskeleton robot's workspace point cloud, and then, using the ADAMS software, the joint torque curve of the upper limb exoskeleton robot under specific working conditions is obtained. Finally, a brief analysis is made for the calculation results.
2. D-H parameter method and Kinematics Modeling of Upper Limb Exoskeleton Robot

2.1. D-H parameter Method

In order to describe the pose of a rigid body in space, a rigid body coordinate system is established in this article: the reference coordinate system \{S\} and the object coordinate system \{T\} fixed on the rigid body. The pose of the rigid body relative to \{S\} can be determined by \{T\}. Represents the relative pose of \{S\}[6]. All corresponding pose transformation matrices are expressed as:

\[
\begin{bmatrix}
     c_1 & -s_1 & 0 & 0 \\
     s_1 & c_1 & 0 & 0 \\
     0 & 0 & 1 & 0 \\
     0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
     0 & 0 & 1 & 0 \\
     -c_2 & s_2 & 0 & 0 \\
     -s_2 & -c_2 & 0 & 0 \\
     0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

(1)

\[
\begin{bmatrix}
     c_3 & -s_3 & 0 & L_1 \\
     s_3 & c_3 & 0 & 0 \\
     0 & 0 & 1 & 0 \\
     0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
     1 & 0 & 0 & L_2 \\
     0 & 1 & 0 & 0 \\
     0 & 0 & 1 & 0 \\
     0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(2)

\[
\begin{bmatrix}
     n_x & o_x & a_x & p_x \\
     n_y & o_y & a_y & p_y \\
     n_z & o_z & a_z & p_z \\
     0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(3)

\[
\begin{align*}
  n_x &= c_2 \times c_3 \times s_1 - s_1 \times s_2 \times s_3 \\
  n_y &= c_1 \times s_2 \times s_3 - c_1 \times c_2 \times c_3 \\
  n_z &= -c_2 \times s_3 - c_1 \times s_2 \\
  o_x &= -c_2 \times s_1 \times s_3 - c_3 \times s_1 \times s_2 \\
  o_y &= c_1 \times c_2 \times s_3 + c_1 \times c_3 \times s_2 \\
  o_z &= s_2 \times s_3 - c_2 \times c_3 \\
  a_x &= c_1 \\
  a_y &= s_1 \\
  a_z &= 0 \\
  p_x &= L_2 \times (c_2 \times c_3 \times s_1 - s_1 \times s_2 \times s_3) + L_4 \times c_2 \times s_1 \\
  p_y &= -L_2 \times (c_1 \times c_2 \times s_3 - c_1 \times s_2 \times s_3) - L_4 \times c_1 \times c_2 \\
  p_z &= -L_4 \times s_2 - L_2 \times (c_2 \times s_3 + c_3 \times s_2) \\
\end{align*}
\]

(4) - (9)

Where \(s_i\) is the \(\sin(\theta_i)\), and \(c_i\) is the \(\cos(\theta_i)\). The pose transformation matrix of the exoskeleton's end effector relative to the base coordinate system can be expressed as:

Where \(p_x, p_y, p_z\) is the position of the end coordinate system relative to the reference coordinate system.

2.2. Spatial configuration and kinematics model of exoskeleton robot

When designing the human-machine compatibility of the upper limb exoskeleton, the following principles should be followed to ensure the comfort, adaptability and safety between the upper limb exoskeleton and the human body: 1) The mechanical structure of the upper extremity exoskeleton must be similar to the human bone structure and cannot interfere with the normal movement of the human upper limb. 2) Taking into account the height and body shape of the exoskeleton wearer, the designed exoskeleton is required to be adaptable and compatible. 3) The range of freedom of each
joint of the upper extremity exoskeleton must match the body’s normal movements to meet the normal behavior of the body.

From the macro point of view of the human-machine compatibility design of the outer bone of the upper limb, it is almost impossible to completely imitate the various postures of the human body due to the various forms of joint motion. Every movement of the human body generally requires coordinated movement of multiple joints. If every tiny movement is realized on the exoskeleton, it will make the research work extremely complicated, especially the complete simulation of the movement of soft tissues such as muscles and ligaments through mechanical models [7]. Therefore, it is necessary to simplify some joint motion forms, put forward a reliable idea, try to meet the basic movement posture of the upper limbs of the human body, and integrate the assist mechanism on this basis. The human-like structure is adopted for the shoulder joint, and the key degrees of freedom that have a significant impact on the end operation space are retained, namely flexion/extension and adduction/abduction. A non-humanoid structure is adopted for the elbow joint. The elbow joint is rotated 90° along the axis of the big arm. The new elbow joint formed is different from the flexion/extension of the human elbow joint. This degree of freedom is called external flexion/inward extension. This structure can increase the operating space of the adapter at the end while not interfering with the movement of the human body. The exoskeleton structure is shown in Figure 1.

![Figure 1. Upper limb exoskeleton structure diagram](image1)

![Figure 2. Non-human-like 3-DOF configuration and its rod coordinate system](image2)

The human upper limb data of the upper limb exoskeleton robot designed by this research refers to the human body size in the GB10000-1988 "Chinese Adult Human Body Size" standard. The main movable joints of the upper limbs of the human body are the shoulder joint and the elbow joint. The arm of the upper extremity exoskeleton has 3 degrees of freedom, including 2 shoulder joints and 1 elbow joint. The range of joint motion angles is shown in Table 1.

| Degree of freedom                          | Rotation angle range |
|-------------------------------------------|---------------------|
| Shoulder flexion/extension ($\theta_1^t$) | -15°~170°           |
| Shoulder abduction/adduction ($\theta_2^t$) | -15°~125°           |
| Elbow flexion/induction ($\theta_3^t$)    | 0°~125°             |

The human upper limb data of the upper limb exoskeleton robot designed by this research refers to the human body size in the GB10000-1988 "Chinese Adult Human Body Size" standard.
The D-H method is used to establish the non-human-like 3-DOF configuration of the rod coordinate system, and the mapping relationship between the end adapter Cartesian space and the joint space is described. Figure 2 shows the non-human-like 3-DOF configuration and its rod coordinate system.

The coordinate system {0} is the base coordinate system, used to fix the position of the shoulder joint and the entire upper limb exoskeleton, and is connected to the shoulder joint flexion/extension (θ1) joint. The coordinate system {1} is fixed to the proximal end of the shoulder joint flexion/extension (θ1) joint, and the origin of the coordinate system is at the intersection of the two straight lines Z1 and Z2. The coordinate system {2} is fixed to the proximal end of the shoulder joint adduction/abduction (θ2) joint, and the origin of the coordinate system is at the intersection of the two straight lines Z2, Z3, and Z2. The coordinate system {3} is fixed to the proximal end of the elbow joint flexion/adduction (θ3) joint, and the origin of the coordinate system is at the center of the joint. The coordinate system {h} is fixed with the end fitting end, and it is obtained by translating L2 along the X axis on the basis of the coordinate system {3}. The D-H parameters of each link in the non-human-like 3-DOF configuration are shown in Table 2.

Table 2 The range of freedom of each joint

| link | α_i | a_i | θ_i | d_i |
|------|-----|-----|-----|-----|
| 1    | -90°| 0   | θ_1 | 0   |
| 2    | -90°| 0   | θ_2 | 0   |
| 3    | 0   | L_1 | θ_3 | 0   |
| 4    | 0   | L_2 | 0   | 0   |

3. Solution and analysis of robot workspace

Monte Carlo [8] method is a numerical method that uses random sampling to solve mathematical problems. This method has a fast calculation speed and is suitable for working space analysis of various articulated manipulators. The specific steps of using Monte Carlo method to obtain the working space of the robot arm in Matlab are as follows:

1. Use the positive kinematics equation to solve the position vector \((p_x, p_y, p_z)\) of the end reference point in the reference coordinate system as shown in formulas (7) (8) (9).

2. Use random functions to generate random values for each joint variable, traverse each joint angle interval separately, and calculate the position of the end reference point relative to the reference coordinate system each time.

3. The workspace of the robot can be obtained by plotting the obtained reference point position coordinates.

The length of the upper arm and forearm of the non-humanoid 3-DOF knot refers to the human body size, which are 330mm and 370mm. Use the robot toolbox in MATLAB software to solve the operating space of the non-humanoid 3-DOF structure. Based on the Monte Carlo method, 5000 points are set in the operating space, and the non-human-like 3-DOF structure operating space in the Cartesian coordinate system can be solved (as shown in Figure 3).
Figure 3. Non-human-like 3-DOF structure workspace

Figure 4. Robot structure dynamic parameter model

Figure 5. ADAMS calculation model of robot structure

The XOY plane in Figure 3 is the sagittal plane across the shoulder joint. By the analysis of the working space, it can be seen that the non-human-like 3-DOF structure of a single arm can reach the operating space on the front side of the body, and the two arms can meet the needs of the upper extremity exoskeleton robot moving and lifting operations.

4. Joint dynamic parameter analysis

When the upper extremity exoskeleton robot is carrying out the lifting operation, when a certain external load is loaded, the dynamic parameters (output torque, speed) of each joint will change due to the change of the upper limb posture. Geared motors designed according to the maximum load torque and speed indicators will be very bulky. This paper is based on ADAMS software to simulate the dynamic parameters of joints under specific working conditions and provide data support for the design of driving joints.

The dynamic parameter model of the robot structure is shown in Figure 4. In the figure, $\theta_1$, $\theta_2$ and $\theta_3$ are the rotation angles of shoulder joint flexion/extension, shoulder joint adduction/abduction, and elbow joint flexion/adduction, respectively. $m_1$, $m_2$ represent the weight of the upper arm and the weight of the forearm respectively. $L_1$ and $L_2$ represent the arm length and forearm length, respectively. $p_1$, $p_2$ represent the center of mass of the big arm and the position of the forearm respectively. $g$ represents the acceleration of gravity.

In ADAMS, based on the fully parameterized geometric model of the mechanical system, the Lagrangian dynamic equations in multi-rigid body dynamics are used to establish the dynamic equations of the system, and then the geometric model system is analyzed for statics, kinematics and dynamics. Finally, the joint torque characteristics of the robot are obtained.

In the simulation process, the trajectory and parameters of the end adapter are set as a vertical straight line 415mm from the center of rotation of the shoulder joint, and the end adapter is lifted up at a constant speed, as shown in Figure 5. Kinetic parameters are shown in Table 3. Choose 30kg as the load weight on the end adapter during the simulation. The completion time of the action significantly affects the dynamic simulation results. Therefore, based on measuring the time for the human body to complete the lifting, the simulation time of the lifting action is 9s after the time is rounded, the lifting height is 0.9m, and the lifting speed is 0.1m/s. According to these parameters, the dynamic simulation analysis of the upper limb exoskeleton is performed. The simulation results are shown in Figure 6-11.

| $L_1$ (mm) | $P_1$(mm) | $m_1$(kg) | $L_2$ (mm) | $P_2$(mm) | $m_2$(kg) |
|------------|-----------|-----------|------------|-----------|-----------|
| 330        | 321.6     | 1.73      | 372        | 183.3     | 0.82      |
Figure 6. Angular velocity curve of shoulder joint flexion/extension

Figure 7. Shoulder flexion/extension moment curve

Figure 8. Shoulder adduction/abduction angular velocity curve

Figure 9. Shoulder joint adduction/abduction moment curve

Figure 10. Elbow joint flexion/induction angular velocity curve

Figure 11. Elbow joint flexion/induction moment curve

It can be seen that since the distance between the end load and the shoulder joint flexion/extension rotation center remains basically the same, the shoulder joint flexion/extension torque curve changes very little, with an average torque of 119.7Nm. The rotation speeds all reach the maximum value of 14.7°/s at 6s, when the end load and the shoulder joint flexion/extension rotation center are on the same level. Because only the weight of the upper arm and forearm affects the shoulder joint adduction/abduction torque, and no torque is generated when the upper arm and forearm are horizontal, Therefore, the torque amplitude first decreases, then increases, and then increases. The curve fluctuates little and passes through the zero point. Since the end adapter moves at a constant speed along the trajectory during the lifting process, there is an obvious sudden change in the rotation speed curve at the starting point, so the maximum rotation speed of shoulder joint adduction/abduction is about 25°/s. The elbow joint flexion/extension moment increases first, then decreases, and then increases; the maximum torque is 47.6Nm, and the maximum speed is 16.8°/s.
5. Conclusions
In this paper, through the kinematics and dynamics modeling of the upper limb exoskeleton robot, the D-H parameter method is used to calculate the end reference point position. In the MATLAB environment, Monte Carlo method was used to obtain the cloud image of the robot's workspace and the workspace was analyzed. Then, for specific working conditions, the dynamic parameters of each joint were calculated by using ADAMS software. The simulation conclusions drawn in this paper have certain reference value for the research and development of upper limb exoskeleton robots.

Acknowledgments
This article is one of the phased achievements of the national key research and development project 《Research on the key technology of weight-bearing and moving lifting assisted exoskeleton robots and verification of typical applications》 (2017YFB1300500).

References
[1] Dong C J, Liu W W, Li X B, et al. (2020), Workspace analysis and regional performance study of six-axis industrial robots[J]. China Test, v.46;No.262(05):157-163.
[2] Li Y J, Fang K L, Zhou H M. (2020), Workspace analysis of six-axis industrial robots[J]. Internal Combustion Engines and Parts, 000(007): P.262-264.
[3] Xie S L, Liu Z S. (2018), Analysis and simulation of workspace of dual-arm robot [J]. Journal of Mechanical Transmission, 42(6):139-143.
[4] Liu Z Z, Liu H Y, LUO Z, et al. (2013), Improvement on Monte Carlo method for robot workspace determination[J]. Transactions of the Chinese Society for Agricultural Machinery, 44(1):230-235.
[5] Zhou L, Zhou Y M, Wang L, et al. (2014), Dual-arm robot workspace analysis based on the Monte Carlo method [J]. Journal of Mechanical Transmission, 38(6):85-87.
[6] Amanpreet, Singh, Ashish, et al. Extension of D-H parameter method to hybrid manipulators used in robot-assisted surgery.[J]. Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine, 2015.
[7] Hamaya M, Matsubara T, Noda T, et al. Learning assistive strategies for exoskeleton robots from user-robot physical interaction[J]. Pattern Recognition Letters, 2017, 99(nov.1):67-76.
[8] Blanes S, Casas F, Sanz-Serna J M. Numerical integrators for the Hybrid Monte Carlo method[J]. SIAM Journal on Scientific Computing, 2014, 36(4).