Modeling and Motion Simulation of a Stamping Manipulator

Zhang Weiwei*
College of Mechanical Engineering Shandong Huayu Institute of Technology, Dezhou, Shandong, 253000, China
Corresponding author’s e-mail: zhangweiwei@huayu.edu.cn

Abstract. With the development of modern industry, manipulator gradually replaces manual operation. The manipulator needs high accuracy in the process of grasping the workpiece, moving and placing the workpiece. After designing the manipulator according to the working environment and related parameters, it is necessary to model and simulate the manipulator, which has determined whether the motion trajectory is accurate and whether the motion is stable and reliable. As a result, the MATLAB software is used to simulate the motion of the manipulator between two points. It can be seen from the curve of displacement, velocity and acceleration that the manipulator runs smoothly and reliably.

1. Introduction
This paper is a simulation study based on the following work. According to the special working environment of the manipulator, the coordinate system type and the number of degrees of freedom are selected for the manipulator; the main structure of the manipulator is designed, and the three-dimensional modeling of each part and component of the manipulator is created by using the Solidworks software; the key parts such as servo motor, synchronous belt and ball screw are analyzed and calculated respectively; and the mechanical part of the main structure of the manipulator is designed.

2. Design of Virtual Model of Manipulator
2.1. 3D Modeling
The whole manipulator includes base, waist, main arm, accessory arm, wrist, claw part. Three dimensional models are created for each part of the manipulator, and they are assembled together according to the position relationship between the parts, thus forming a complete manipulator.

2.2. Virtual Assembly
In virtual assembly, the assembly sequence of each part should be determined first, and then the degree of freedom of each part and the motion relationship between the parts should be determined, so as to determine the constraint form to be used. Based on the bottom-up assembly sequence, the manipulator is assembled virtual, and the collision inspection and interference inspection are carried out [1].

This design uses the order from the base to the claw to install all the parts of the manipulator together, and then verify that the manipulator will produce collision and interference inspection.

1) Addition of constraints
The degree of freedom of each part of manipulator and the motion relation between parts are analyzed, and the form of motion pair constraint is determined. The final completed manipulator
loading ligand is shown in figure 1.

![Figure 1 Manipulator assembly](image)

2) Collision and interference detection
Choose Free Drag, Collision Check, Collision Stop Options under Mobile Parts. By moving and rotating the position of the parts to check whether there is a collision between the parts and other parts, and then determine whether the constraint degree of freedom is set correctly.

Among the more complex manipulator ligands, it is very difficult to check whether there is interference between parts with the naked eye. In the SolidWorks, it is possible to automatically detect interference. If the mechanism design is not reasonable, then the parts can not move to the specified position, and it is difficult to complete a series of preset actions.

3. Kinematics Simulation
There is a very good toolbox in the MATLAB called MATLAB Robotics Toolbox, which contains a number of functional about the manipulator [2]. We can use the correlation function to analyze whether the kinematics of the manipulator we designed is qualified. We can also analyze whether the dynamics can reach the standard, or design the movement route of the manipulator. More conveniently, we can make use of the very convenient drawing function [3], make a simulation for the above analysis process, and get important parameter from it [4]. By using MATLAB Robotics Toolbox to create the 3D modeling and driving of the manipulator, the motion between the two positions at the end of the manipulator is simulated.

3.1. Establishment of Manipulator Simulation Model
We can use the Link function included in the MATLAB Robotics Toolbox to model the manipulator according to this design. The Link function is usually written in the following format [5,6]:

\[
\text{\texttt{L = Link([alpha A theta D sigma], CONVENTION)}}
\]

The number of Link required depends on how many joints the designed manipulator contains. The structure and related parameters of the manipulator are analyzed, and the improved D-H parameters are determined. The simulation is carried out according to the following steps:

With reference to the manipulator body parameters, the upper and lower direction of the motion axis simulation modeling, the program as follows:

\[
\text{\texttt{L\{1\} = link([0 0 0 180 1], 'mod');}}
\]

\[
\text{\texttt{L\{2\} = link([0 110 0 850 0], 'mod');}}
\]

\[
\text{\texttt{L\{3\} = link([-pi/2 0 0 0 1], 'mod');}}
\]

\[
\text{\texttt{L\{4\} = link([pi/2 300 0 20 0],'mod');}}
\]

Establishing four joint models of manipulator.
\[ r = \text{robot}(L, \text{four axis}'); \] Creating an entire manipulator model.

\[ \text{plot}(r); \] % Modeling Stereo Shape by Drawing Function.

\[ \text{drivebot}(r); \] % Let the model move.

Figures 2 and 3 represent three-dimensional models and control windows. Four degrees of freedom variables are zero. A black, used in horizontal view when a manipulator moves Create a coordinate at the terminal of the terminal pickup. And represented by different colors, red represent the movement of the x direction, green represent the movement of the y direction, bluer represent the movement of the z direction. Then, Get the model moving with \text{drivebot}, The same as the real control end picker.

3.2. Manipulator trajectory planning

This design uses the \[ jtraj(q^A, q^B, t) \] command in the MATLAB Robotics Toolbox to simulate the motion of each joint[47,48]. The program is as follows:

\[
q^A = \begin{bmatrix}
q_1^A, q_2^A, q_3^A, q_4^A, q_5^A, \ldots, q_n^A
\end{bmatrix};
\]

\[
q^B = \begin{bmatrix}
q_1^B, q_2^B, q_3^B, q_4^B, q_5^B, \ldots, q_n^B
\end{bmatrix};
\]

\[ t = 0:0.01:T; \]

\[ [q, \dot{q}, \ddot{q}] = jtraj(q^A, q^B, t) \]

The movement route of the end of the manipulator's claw is shown in figure 4.
According to the above simulation, we understand how the position of the end of the manipulator claw changes gradually from the beginning of grasping the material to the position waiting for processing, and realize the movement route of the end of the claw between the two positions. From this, we get the trajectory between the given two points. By MATLAB, we can get the motion trajectory between two points, and we can further calculate the displacement, speed and acceleration of the four moving joints in the hands of the feeding and discharging machinery. The designed feeding and discharging manipulator has four degrees of freedom, corresponding to the four corresponding parameters of $d_1$, $\theta_2$, $d_3$ and $\theta_4$ in turn.

Through figures 5 and 7, we can see how the displacement, velocity, and acceleration of the manipulator lift shaft and the telescopic shaft change with time. Also in the graph representing the change of acceleration, we can see that at the beginning of motion and before the end of motion, there is a peak and a valley value, respectively. The peak and valley values are the maximum values of acceleration and deceleration. From Fig.6 and Fig.8, we can see that the angular displacement, angular velocity and angular acceleration of the axis of rotation in the hand of the machine change with time. From the diagram, we can see that the rotation axis is similar to the two axes of up and down movement and back movement.
4. Conclusion
By using MATLAB software, the manipulator motion model is established, and the running route of the manipulator between two positions is simulated. The change trend of velocity, displacement and acceleration of the four joints is obtained. It can be seen from the curve that the curve changes smoothly. We can draw a conclusion that the manipulator runs stably and reliably.

Acknowledgment
This achievement is supported by the project of Shandong Huayu University of Technology School-level scientific research platform "Intelligent Manufacturing Equipment Design Engineering Technology Research and Development Center"(NO. 201901), thanks for the financial support from the school.

References
[1] Wang Q (2011). Hu Xiaojian. Kinematics Analysis and Simulation of Matlab Welding Manipulator[J]. Modern Welding .(4):26-29
[2] Xie B (2010). Cai Zixing. Teaching of MATLAB Robotics Toolbox-based Robotics Simulation Experiment [J]. Computer Education.19:140-143
[3] John J. Craig. Introduction to robotics mechanics and control [M].US:Prentice Hall, 2004
[4] Zang Q.K. (2010). Li Chungui, Yan Xianglei. Motion simulation and trajectory planning of PUMA560 robot based on MATLAB [J]. Journal of Guangxi Academy of Sciences .26(4):397-400
[5] Fu Y.Y. (2013). Ge Aping. Research on Kinematics Simulation of Industrial Robot Based on MATLAB [J]; and Mechanical Engineering and Automation .3:41-45
[6] Wang Z.X. (2012). Fan Wenxin, Zhang Baocheng, Shi Yuanyuan. Kinematics Analysis and Simulation of Industrial Robot Based on Matlab [J] Electrical and Mechanical Engineering .29(1):33-37