Robotics design challenge (MATLAB/SIMULINK)

Jiaqi, Yao. Ruixin, Zhan. Xuzhao, Zhang. Xiaoyu, Zhang

March 31, 2023

Abstract

1 Introduction

Our product is a type of mechanical equipment used for automated welding. Our robotic arm can be widely used in various welding operations in the manufacturing industry, including automotive manufacturing, aerospace, construction, and manufacturing.

Our product has many adavantages. Firstly, it can improve production efficiency and quality by reducing the negative impact of human factors on production through automated welding operations. Secondly, it can reduce the danger of the work environment.

In summary, the four-arm welding robot is an efficient and accurate welding device with many advantages. It will become an important part of automated production in the manufacturing industry, providing a reliable solution for various production operations.

2 Task 1

2.1 Originality and uniqueness of the robot

Our robot's uniqueness lies in the fact that it is a four-arm robot. Although two-arm and three-arm robots can perform the same tasks, they are relatively unstable and more difficult to control. In addition, a four-arm robot can provide a larger workspace and a greater potential for improvement.

2.2 Property of all the arms and the joints

We have designed 4 joints for the robot, 2 of which are revolute type and 2 are prismatic type. At the same time, there are four corresponding robotic arms, whose parameters are shown in the Table 2.1 and Table 2.2.

Furthermore, in order to make the robot operate more stably, we assumed the relevant data of the robot arms, and after simulation, the data we obtained is reasonable.

Name	Body Mass (kg)	Center of mass	Inertia (I_{xx} I_{yy} I_{zz}) ($kg \cdot m^2$)
R1	10	(0 0 0)	(0.27 0.27 0.8)
R2	10	(0 0 0)	(0.27 0.27 0.8)
P1	1.5	(0 0 0)	(0.07 0.07 0.07)
P2	1.5	(0 0 0)	(0.07 0.07 0.07)
Tool	1.2	(0 0 0)	(0.002 0.002 0.004)

Table 2.1: Robot arm parameters

Joint	Type	Position Limit (rad & m)	Joint Axis
1	revolute	$[-5\frac{\pi}{180}, 5\frac{\pi}{180}]$	[0 0 1]
2	revolute	$[-30\frac{\pi}{180}, 30\frac{\pi}{180}]$	[0 1 0]
3	prismatic	$[-0.5, 0.5]$	[1 0 0]
4	prismatic	$[-1, 1]$	[0 1 0]
Fixed	revolute	N/A	N/A

Table 2.2: Joint parameters

Figure 2.1 is a schematic diagram of the robot in MATLAB.

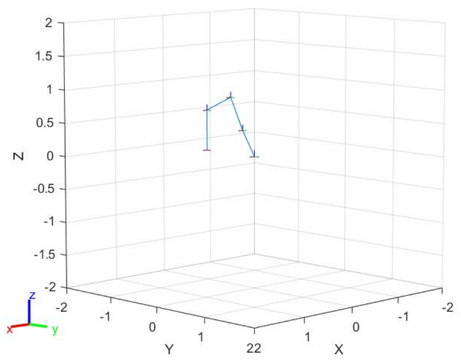

Figure 2.1: Robot schematic

2.3 Collision discussion

Our designed robot is free from collisions. In most cases, collisions occur on the two arms of the rotating joints whose rotation angle is greater than ±$90°$. For our designed robot, the total range of motion for the two movable joints is ±$35°$, so there is no collision. Additionally, the animation of the robot's movement trajectory can be viewed in the dynamic image in Figure 8 (After running the matlab code).

3 Task 2: The Workspace

Figure 3.1 shows the robot's workspace in Task 1. The robot can work with precision within this workspace and also carry out certain tasks outside the range.

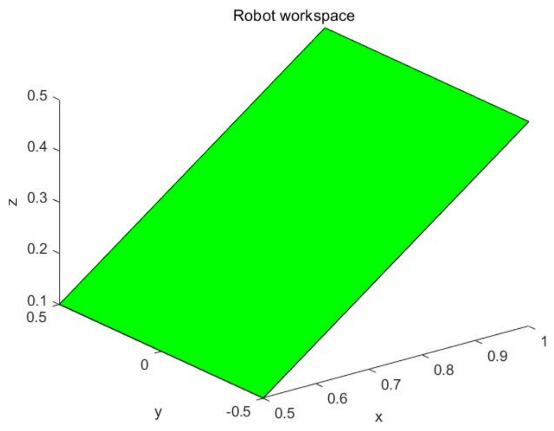

Figure 3.1: Robot's Workspace

4 Task 3: The robot's kinematic diagram

As we can see in the Figure 4.1, we can find that all the z-axis are the revolution or direction of every joints, and use the right-hand rule, we can check if all the x-axis are perpendicular both to its own z axis and the z axis of the frame before it. Apparently, the directions of all the axis are strictly following the Devenit-Hartenburg Framerules.

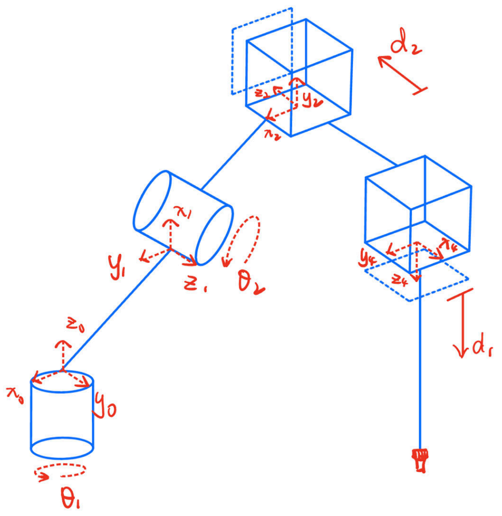

Figure 4.1: The robot's kinematic diagram

5 Task 4: Simulink and Simulation

5.1 Simulink

Figure 5.1 shows a schematic of Simulink.

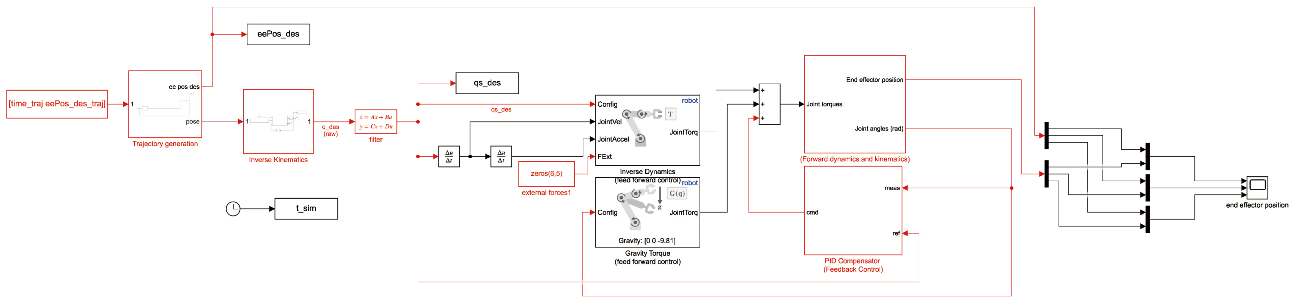

Figure 5.1: The Simulink

Tidy of the model

To ensure aesthetic appeal, a modular design approach was adopted, where different modules represent different functionalities, thereby making the overall model's operation flow appear clear and concise.

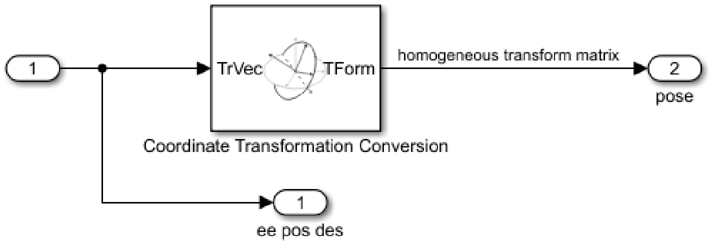
1
TrVec
TForm
homogeneous transform matrix
2
pose
Coordinate Transformation Conversion
1
ee pos des

Figure 5.2: Trajectory generation schematic

Trajectory generation

We first calculate the path coordinate points and time series from the code, and use them as input parameters for trajectory calculation. Through trajectory calculation as shown in Figure 5.2, we can obtain the expected end-effector position and obtain motion trajectory data applicable to each point after coordinate transformation (which records the three-dimensional coordinate points of the path every 0.01 seconds).

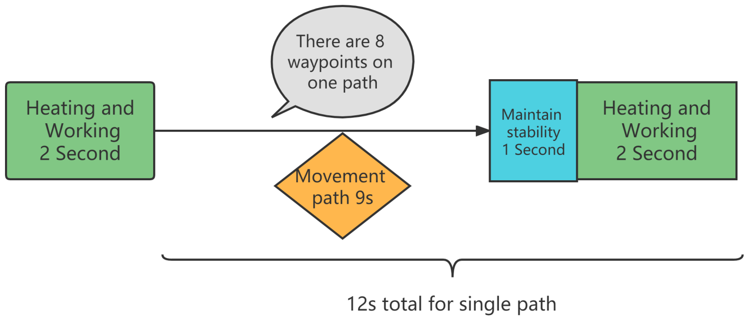
Heating and Working 2 Second
There are 8 waypoints on one path
Movement path 9s
Maintain stability 1 Second
Heating and Working 2 Second
12s total for single path

Figure 5.3: Flowcgart of trajectory generation

Figure 5.3 is a flowchart of the process we use to compute the trajectory of the robot's end effector position. First, we divide the rectangle required for the task into four line segments. For each line segment, we divide it into 9 equally long segments, and expect the robot to take 9 seconds to move along it. After that, the robot needs to spend 1 second for stabilization, followed by 2 seconds of welding work at the work point. Totally, it takes 48 seconds for the robot to complete the welding task. To account for this, we designed a simulation time of 48 seconds.

Inverse kinematics

In the inverse kinematics module, we use the expected end-effector position as input, as shown in Figure 5.4, and calculate the expected joint angles using inverse kinematics. The output is a matrix containing time-series data, which records the rotation angles of different joints of the robotic arm every 0.01 seconds.

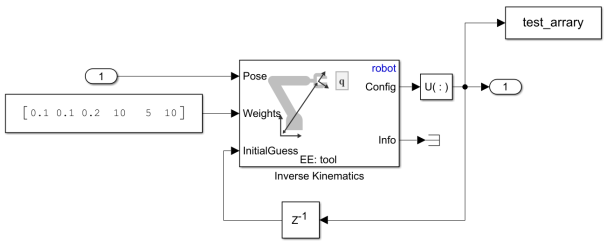

Figure 5.4: Inverse kinematics

Inverse dynamics

Inverse dynamics is to calculate the forces or torques that are required to produce a desired motion of a robotic system. The technique involves using the system's equations of motion to determine the forces or torques required to generate a desired trajectory or motion. As shown in Figure 5.5, the input parameter in this part is the coordinate data of each sampling point calculated by the inverse kinematics. By weighting and differentiating the data, the required angular velocity and angular acceleration can be obtained. Then, the obtained data is imported into the inverse dynamics module for calculation, and the result is the torque required for each joint at each sampling point.

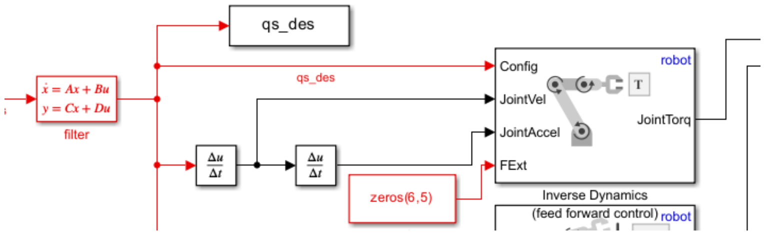

Figure 5.5: Inverse dynamics

Gravity

In this part, the torque generated by gravity is added as a vector to the torque required by each joint calculated by the inverse dynamics module, and the combined torque vector is passed to the next module to simulate the deviation that may occur in the actual robot arm. The process is shown in Figure 5.6.

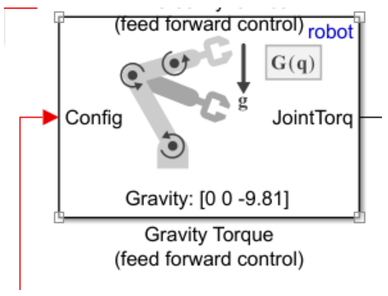

Figure 5.6: Gravity part

Forward dynamics and kinematics

Figure 5.7 shows the forward kinematics and dynamics module, in which the actual position and orientation of the end-effector can be calculated based on the actual joint angles obtained by considering the different torques inputted to each joint.

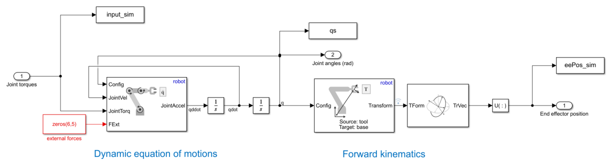

Figure 5.7: Forward dynamics and kinematics

5.2 PID design

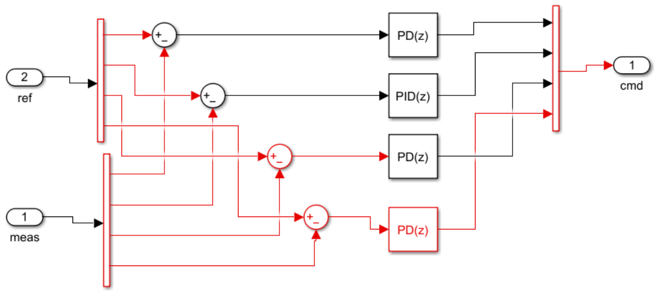

Figure 5.8: PID part

In the PID control module,as shown in Figure 5.8, we subtract the desired data from the actual data to obtain the error value, which is then used for PID calculation. We noticed that joints 1, 3, and 4 only use PD controllers because these three joints require a fast response speed and low overshoot, and have lower requirements for steady-state error. For joint 2, which uses a PID controller, it is sensitive to steady-state error due to its rotation around the y-axis. Although there may be difficulty in tuning, we successfully completed the debugging process.

5.3 Results

Finally, as shown in Figure 5.9set the end effector positions and the desired end effector positoins as the input, draw the plot with their x y z positions followed by the time respectively.

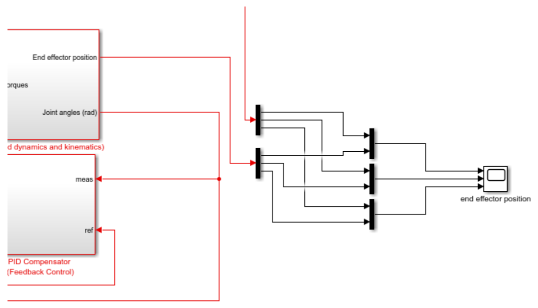

Figure 5.9: Export figure part

Input torque

As shown in Figure 5.10, this is the torque figure of each joint required for the robot during operation. By observing the figure, it can be found that, except for the initial instability during startup, we only need less than $30N \cdot m$ of torque to drive the robot to complete the welding task.

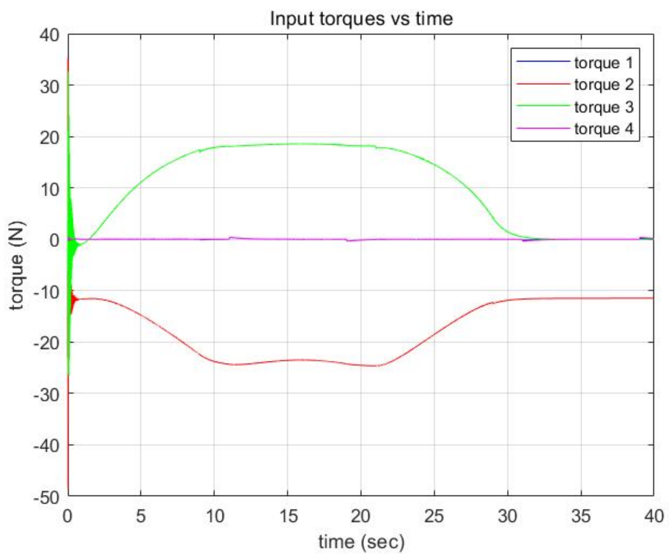

Figure 5.10: Torque vs Time

Joint angle

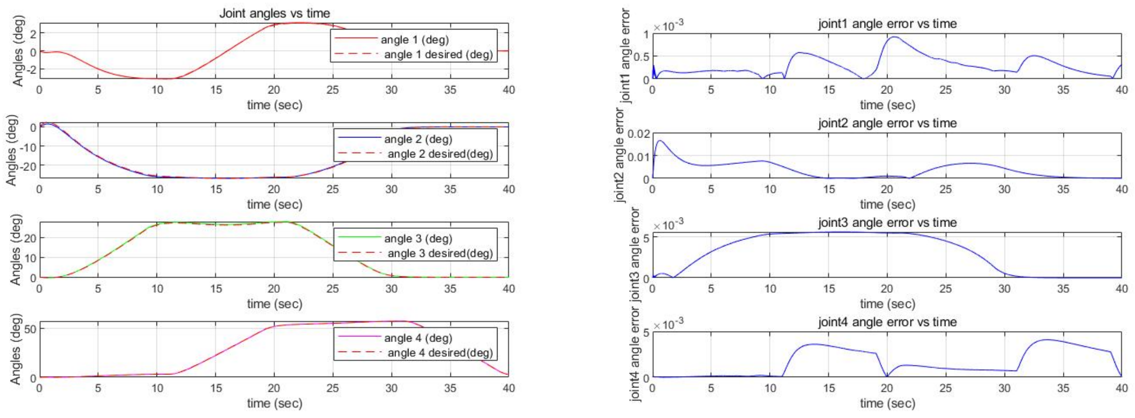

Figure 5.11: Joint angles vs Time

As we can see in Figure 5.11, the error of joint1 angle does not exceed the limit of 10^{-3}, but it fluctuates throughout the process; The error of joint2 angle is the largest, and it nearly reaches 0.02, the error tends to be stable except for large fluctuations at the peak throughout the process; Both the maximum error of joint3 angle and joint4 angle are below $5 * 10^{-3}$, but joint3 angle error are steady increase to the maximum error, and remain stable on the maximum angle error; as for joint4 angle, there is almost no error in the first ten seconds, and there is always a period of stability after the changes of the error. Overall, the errors are small and will not have a significant impact on the operation of the robot.

End Effector Position

As we can see in Figure 5.12, the maximum error in end effector position on the x and z axes is below 0.05, while the maximum error on the y axis is slightly over 0.10. The errors in the x and z axes fluctuate simultaneously, while the error in the y axis gradually increases as the errors in the former axes begin to decrease. There is a small amount of error throughout the entire process, but it is not significant. It has almost no impact in reality.

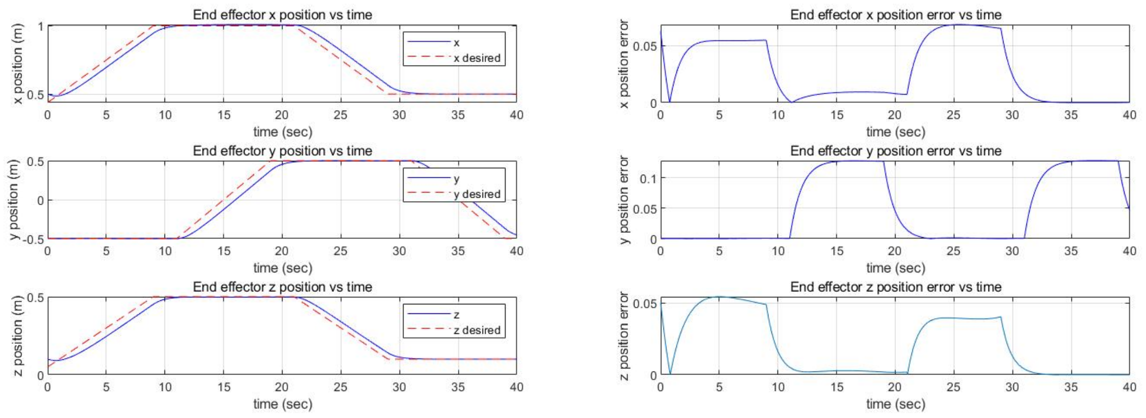

Figure 5.12: End Effector Position vs Time

6 Conclusion

In conclusion, we successfully built the four-arm robot and achieve the function of welding in a specific area. We designed all the relevant data and reduced the errors by simulink. The maximum error in end effector position on the x and z axes is below 0.05, while the maximum error on the y axis is slightly over 0.10, and the largest error of joint angle is joint2 angle, and it just nearly reaches 0.02. These errors are far better than the standards set. Overall, we achieved the desired results. This is certainly a product worth buying and will bring unimaginable benefits to your company

References