Modeling and Control of A Parallel Five-Bar SCARA Robot

Cunxi Dai, Haoran Wang,Peijun Lin, Xinke Mao, Ximan Zhang, Shenglong Liang

Abstract— Singularities is the configuration where task-space motion fails to reasonably map to joint-space motion. Therefore, to over come the in-continuity caused by singularities, inverse-kinematic based control is not sufficient. This paper discusses the developing, modeling and control of a direct-drive planar five-bar robot, and ways to cross singularities without motor failing. We proposed three potential methods for singularity crossing and validated the performance of the three methods robot's via experiments.

I. Introduction

Five-bar linkage mechanism has 2 degrees of freedom(DoFs), and are widely used in manipulation robotic arms and leg structures of legged robots. [1] describe a lightweight arm mechanism with invariant and decoupled inertia characteristics. [2] report the robot for horizontal gap open MRI which controls needle orientation effectively in the vertically limited gantry space. The introduced controller was designed based on the governing ideal Euler-Lagrange equations on the robot but assessed using the on-line dynamic simulation of the mechanism for different target configurations which guarantees the high performance and effectiveness of the designed controller [3]. The detail analysis has been done through the mathematical and geometrical modelling of five-bar spherical metamorphic linkage robotic palm. This mathematical and geometrical model can be used to inspect the feasibility of the mechanism design for a multi fingered robot palm [4].

Direct-drive robot with no gearbox features high-speed motion and force transparency, at cost of low output torque. These features aligns with general request of many robots, especially in manipulation and locomotion where force-sensing is critical. Early literature describe the design concept of a new robot based on the direct-drive method using rare-earth DC torque motors. [1] describe a lightweight arm mechanism with invariant and decoupled inertia characteristics. [5] discuss the control issues related to the performance of a direct-drive robot, specifically, a direct-drive mechanical arm capable of carrying up to 10 kilograms, at 10 meters per second, accelerating at 5 G (a unit of acceleration equal to the acceleration of gravity).

In this paper, we developed a direct-drive five-bar linkage SCARA robot for swift pick-and-place task, and to experiment various ways to operate in different configurations wiout falling due to singularities. The hardware, control frame work, cross-configuration trajectory planning are demonstrated in the following passage.

II. Mechatronics

A. Mechanical Design

1) Five-Bar linkage Design: The distance between the two bases is set as 180mm for our SCARA robot, and the length of each rod is 135mm. In this way, the operable domain can be reached to the maximum, and the space below the two bases can be covered while working. Moreover, it is helpful to solve the problem of crossing the singularity, that is, we do not need to consider the trajectory planning of circumventing the non-operable region during the control.

Before designing the linkage, we discussed the options of the end effector and the functions of the SCARA robot. We found that even for the most stressed part of the whole linkage, the torque under full load is not more than $0.3\mathrm{N}\cdot\mathrm{m}$ and the shear force is not more than 1.5N, which is tolerable for the PLA 3D-printing rod with high filling rate. What needs to be considered is how to minimize the weight and volume while maintaining superior mechanical properties.

Therefore, after designing the concise five-bar structure, we consider using the mechanical simulation plug-in installed in Solidworks to optimize the structure, which can minimize the weight of the bar and improve its appearance. In topology optimization, a tension of 5N and a torque of $1\mathrm{N}\cdot\mathrm{m}$ are applied to the bar. Considering from the perspective of material mechanics, such conditions can ensure that the load at the maximum deformation of the bar can meet the safety factor above 3.0, and the structure will not be damaged under the condition of high-speed movement.

Fig. 1. topological optimization result

After designing the topology form of the five-bar linkage, we use the simulation of static analysis to verify the performance of the individual linkage under force and moment.

Then we found the thickness of linkage is about 30mm, the result of topology shows that our linkage can support the load only 15mm thickness, thus we change the thickness of our bar and desert the hollow structure which generated by topological optimization.

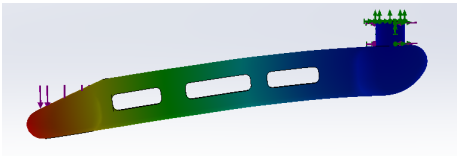

Fig. 2. static analysis (red part has the most deformation)

In the first designing scheme, we follow the type in DexCAR's User manual, which place the first two linkage are place in the same height. But while simulating the SCARA robot we found, that the first two bars will crash under some configurations. So that, we place the four linkages in different heights in our second prototype.

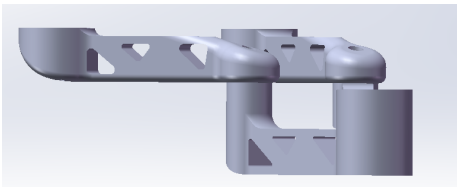

Fig. 3. the first designing scheme

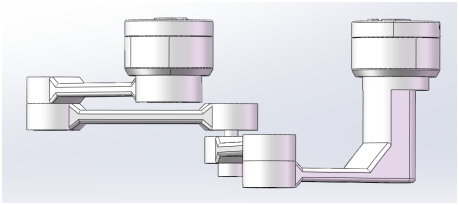

Fig. 4. the second designing scheme

2) Final controlling action: Before determining which scheme to choose for the end effector, we need to make it clear that the parallel five-bar linkage itself has no degree of freedom in the direction of Z axis, so it cannot move vertical. If the end is expected to move in the Z axis, it needs to add rack and pinion or guide rail to achieve smooth movement. However, because the joint diameter of the end is only 200mm, we need to reduce the volume of the moving structure as much as possible. At the same time, because the end is far away from the fixed linkage position, the end should not load too much, so the rack and pinion structure made by steel is not suitable. Then, through research, we consider using the steering gear to connect the capstan to drive the rope. Pull the end to compress the spring then achieve a Z axis motion.

We set the actuator rely on suction cups and air pumps to absorb objects with smooth surfaces, suck the objects off the ground and move them to other positions. Because the distance from the base to end effector is always changing while terminal is moving, and the steering gear has the effect of self-locking if power on, it cannot be placed on the platform, but can only be fixed on the end effector.

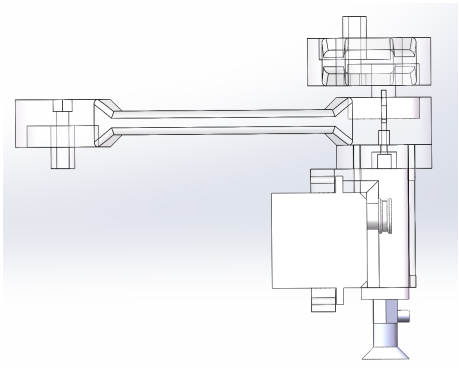

Fig. 5. the perspective drawing of end effector

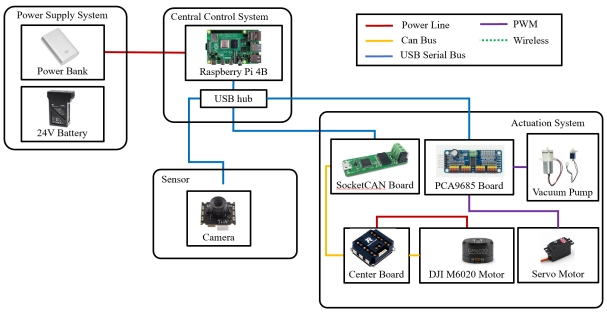

Fig. 6. Electronics system of the five-bar linkage robot.

B. Electronics Systems

The main computation power on this robot is the Raspberry Pi 4B embedded computer. Two DJI M6020 brushless motors are used as the power unit. The controllers accept the voltage command and provide current, speed, and position feedback. A socketcan board that converts the CAN signal to the USB signal is used to connect the FOCs and the embedded computer.

The end effector consists of a servomotor and a vacuum pump. An electromagnetic valve is connected to the vacuum pump to control its opening and closing. The three components are all controlled by pulse width modulation. But it's not easy for Raspberry Pi 4B to generate three PWM signals simultaneously, so a PCA9685 servomotor controller is connected between Raspberry Pi 4B and the three components. The servo motor controller is able to generate most 16 PWM signals. For the use of just controlling three components, it is sufficiently powered directly by Raspberry Pi 4B. To use the PCA9685 PWM servo controller with the Raspberry Pi 4B, we referred to the source code on GitHub and wrote the end effector control python code.

III. Kinematics Analysis

A. Direct Kinematics

In the direct kinematics we know the active-joint angles θ_1 and θ_2 and we look for the position of C. Essentially, the problem consists in finding the intersection points between circle CA_1 , centred at A_1 and of radius l, and circle CA_2 , centred at A_2 and of radius l.

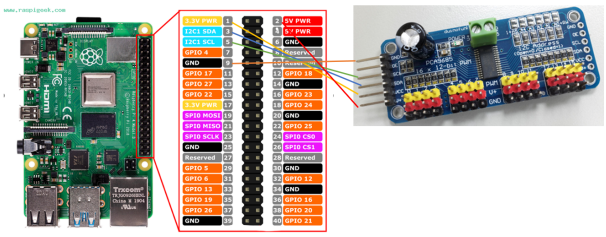

Fig. 7. GPIO pins connection

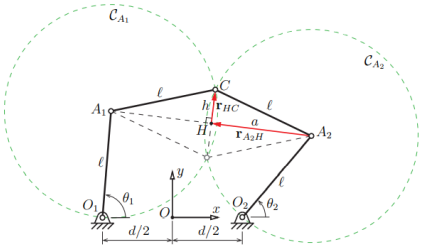

Fig. 8. Schematic illustrating of the direct kinematics

$$r_{OC} = r_{OA} + r_{A_2H} + r_{HC} \tag{1}$$

$$r_{A_2H} = 1/2(r_{0A_1} - r_{OA_2}) \tag{2}$$

$$r_{0A_1} = \begin{bmatrix} -d/2 + lcos\theta_1 \\ lsin\theta_1 \end{bmatrix} \tag{3}$$

$$r_{0A_2} = \begin{bmatrix} d/2 + lcos\theta_2 \\ lsin\theta_2 \end{bmatrix} \tag{4}$$

In general, there are two possible positions for point C. In other words, vector r_{HC} can be obtained by rotating vector r_{A_2H} 90° clockwise or 90° counterclockwise, normalizing it, and then multiplying the resulting vector by the length of segment HC

$$r_{HC} = -\gamma \frac{h}{a} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} r_{A_2H} \tag{5}$$

$$h = \sqrt{l^2 - a^2} \tag{6}$$

$$x = \frac{1}{2}(-d + lcos\theta_1 + lcos\theta_2) + \frac{1}{2}\frac{\sqrt{l^2 - a^2}}{a}(lsin\theta_1 - lsin\theta_2) \tag{7}$$

$$y = \frac{1}{2}(lsin\theta_1 + lsin\theta_2) + \frac{1}{2}\frac{\sqrt{l^2 - a^2}}{a}(d - lcos\theta_1 - lcos\theta_2) \tag{8}$$

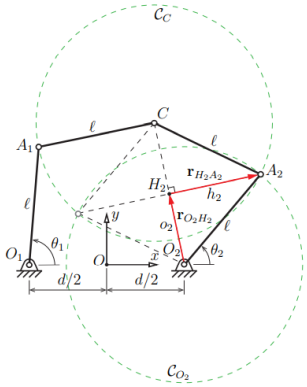

Fig. 9. Schematic illustrating of the inverse kinematics

B. Inverse Kinematics

Let us denote by H_2 the midpoint between C and O_2, and let O_2 be the length of segments O_2H_2 and H_2C, and H_2 be the length of segment H_2A_2. Obviously, the line segment H_2A_2 is normal to the line segment O_2H_2.

$$r_{O_2A_2} = r_{O_2H_2} + r_{H_2A_2} \tag{9}$$

$$r_{O_2H_2} = \frac{1}{2}\begin{bmatrix} x - d/2 \\ y \end{bmatrix} \tag{10}$$

In general, there are two possible positions for point A_2. In other words, vector $r_{H_2C_2}$ can be obtained by rotating vector $r_{O_2H_2}$ 90° clockwise or 90° counterclockwise, normalizing it, and then multiplying the resulting vector by the length of segment H_2C_2

$$r_{H_2A_2} = -\delta_2 \frac{h_2}{o_2} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} r_{O_2H_2} \tag{11}$$

$$h = \sqrt{l^2 - o_2^2} \tag{12}$$

$$\theta_1 = atan2(y + \delta_2 \frac{\sqrt{l^2 - o_2^2}}{o_2}(d/2 - x), x - d/2 + \delta_2 \frac{\sqrt{l^2 - o_2^2}}{o_2} y) \tag{13}$$

$$\theta_2 = atan2(y + \delta_1 \frac{\sqrt{l^2 - o_1^2}}{o_1}(d/2 + x), x + d/2 + \delta_1 \frac{\sqrt{l^2 - o_1^2}}{o_1} y) \tag{14}$$

C. Workspace and Singularity Analysis

The workspace is defined as the reachable region of the end-effector. Within the workspace of manipulator there are some positions where the manipulator loses its controllability, which is known as singular position and the phenomenon is known as singularity.

The workspace of the planar five-bat robot is related to the length of links. The detailed figure is as follow, where the tow arcs is centered at the position of motors separately, with R equal to the sum of two movable links. Because of the identity of links, there is no

inner unreachable space among the workspace in the shape of leaves. And for the special arrangement of the links, no mechanical interference will occur with any configuration.

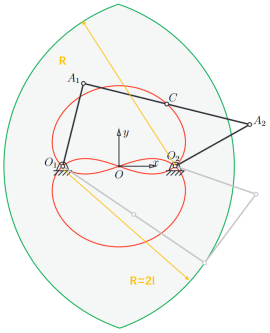

Fig. 10. workspace and singularity

However, the singular configurations still deserve attention. If we consider the singularity of five-bar mechanism individually, there are many configurations causing reduction of freedom. But focusing on the purpose of analysing singularity which is control, then, only one kind of singular configuration matters. The configuration is shown in Fig. 9, the upper one in black. And the red curve is the singularity loci in workspace. In this singularity, two horrible situations will interface the control of the end effector. First, if the two motors rotates in opposite directions, the robot is jammed. Second, this position is on the edge of two different configurations and it is impossible to control the moving direction of distal links. This means that when motors move, the robot will go into a random configuration and it will influence the precise analysis of the effector position.

To avoid the singularity, we calculate the singular loci in jointspace simultaneously, so that we can avoid it when plan trajectory.

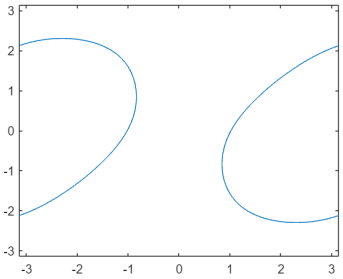

Fig. 11. jointspace and singularity

IV. Control and Simulation

A. Control Hierarchy and Computer Vision

Based on the computer version, the goal is detecting the position of the target in the workspace so that the end effector can move to that position to pick up the target. We choose a USB drive-free camera to get video stream. The camera hangs over the workspace independent from the robot to reduce vibration disturbance.

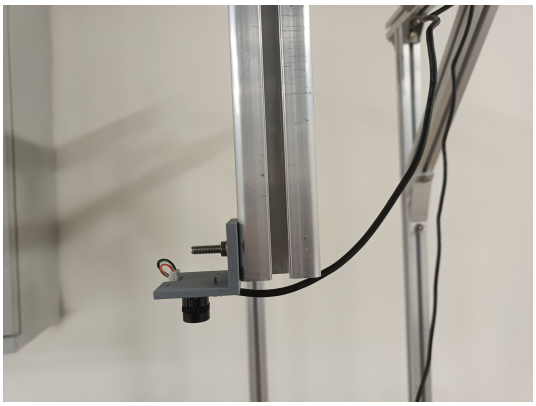

Fig. 12. camera

Based on OpenCV, we take some ways to process the video stream to get the centroid coordinate of the object in the screen. First, we define the tracking color range in RGB value. Second, detect the object in HSV color space and construct a mask of the object. Third, find contours in the mask and calculate the centroid coordinate. To ensure the accuracy, the process is repeated 50 times for one centroid coordinate. Then we can transform the coordinate in the screen into the coordinate in the workspace.

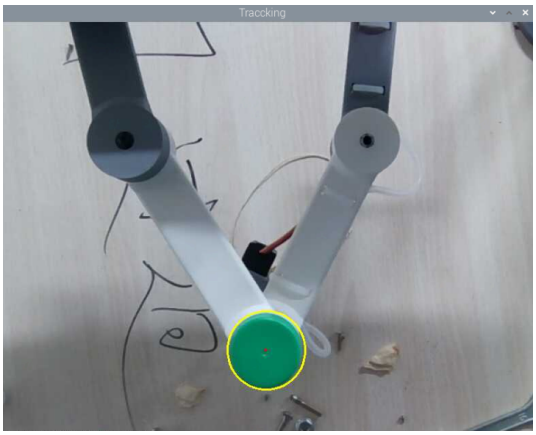

Fig. 13. detecting the object

B. Cross-Configuration Trajectory Planning

1) Algorithm introduction: Trajectory planning is a very important part in robot control. And Dijkstra algorithm is commonly used in the optimization calculation of trajectory planning. Dijkstra algorithm is a typical shortest path algorithm, which is used to calculate the shortest path from one node to other nodes. Its main algorithm starts from the starting point, adopts the greedy algorithm strategy, and traverses the adjacency nodes of the nearest and unvisited vertices to the starting point every time, until it expands to the end point.The

basic idea is to introduce two sets, A and B, whose function is to record the vertices where the shortest path has been found, and B is to record the vertices where the shortest path has not been found.At the beginning, there is only the initial point given by us in A, and the vertex path in B is "the path from the starting point to the vertex", then find the vertex with the shortest path from B and add it to A, and then update the vertex in B and the corresponding path of the vertex, and repeat the operation until all vertices have been traversed, and the last vertex is the end point set by us.

2) Problems and difficulties: In this paper, the trajectory planning uses the Dijkstra algorithm. It should be noted that this algorithm can only solve the shortest path problem between different points. In our parallel manipulator, there are many singular configurations, and the existence of singular points is relatively complex. It is impossible to avoid singular configurations directly by using the Dijkstra algorithm in the workspace, and it may even lead to excessive motor speed and direct error reporting due to singularity.

3) Implementation procedure: Under our joint efforts, the above problems have been successfully solved, and the specific implementation process is as follows.

First, the problem that trajectory planning may pass through singular points is solved. On the basis of Dijkstra algorithm, we supplement the obstacle avoidance function, and finally solve the problem. Firstly, we obtained the end effector 's positions of each singular point in the workspace through the forward and inverse kinematics and singularity analysis of the parallel manipulator mentioned above, and recorded these coordinates. Then, a small diamond shape with the end as the center point was established around the end coordinates of each possible singular point, which was regarded as an obstacle and set as uncrossable. The singularity and its surrounding position are regarded as obstacles to avoid the singularity in the way of avoiding obstacles.Besides the circle, regular polygon is closest to the space around the singularity. While the rhombus can surround the singularity in a small range, its vertices can fit well to satisfy the Dijkstra algorithm.

Second, solve the problem of taking points in the Dijkstra algorithm. Although the manipulator should move in a continuous way in the actual working space, it is actually more convenient and faster to let the manipulator reach the destination only through a few wires.In this project, the boundary points of a certain number of workspaces and the boundary vertices of singular point diamond are taken as point points, and they are connected and the midpoint of each line is taken for the application of Dijkstra algorithm. Taking the midpoint can avoid the position of all singularities very well, and the path is still in the interval of the shortest path. This solves the problem of taking points in Dijkstra algorithm.

Now that the preparation is complete, it's time to

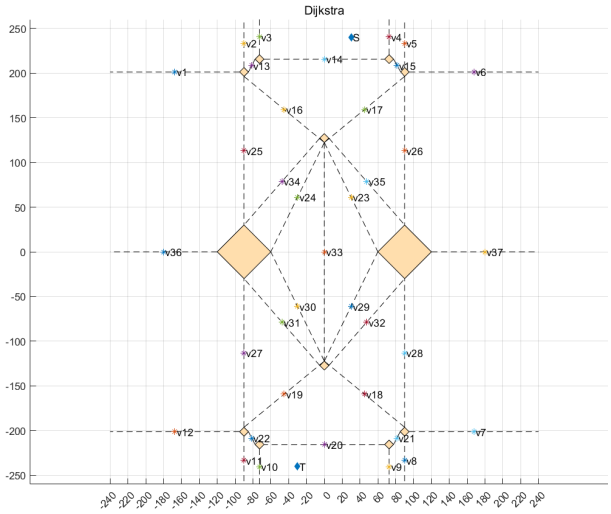

Fig. 14. Workspace

apply the Dijkstra algorithm. In the previous paper, we took the midpoint of each line segment as the optional point of the Dijkstra path, but did not specify the connection relation between points. Therefore, the matrix of the reachable point position of each point was first written, including the starting point and the final point. Finally, the reachable path diagram between points is obtained, as shown in Figure 12.

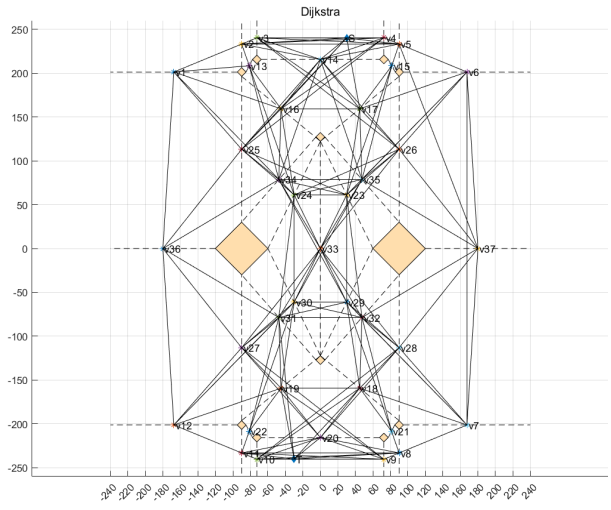

Fig. 15. Accessible path

On this basis, the length of each path can be calculated, as shown in Figure 14.

Then, we officially started to use the Dijkstra algorithm, starting from the starting point S and finally reaching the final point T. The planned path is shown in figure 15, which meets our requirement of avoiding singularity, and the process is relatively simple.

Next, the equation of the path is solved, the linear equation is written and discredited by code, and finally the discretized path array is obtained.Finally, this path

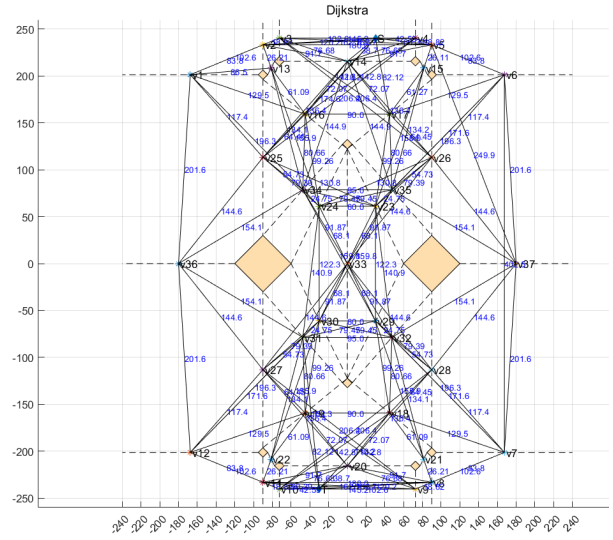

Fig. 16. Accessible path distance

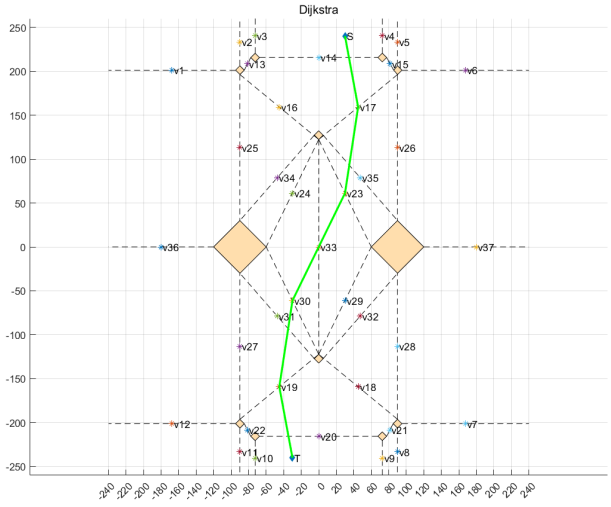

Fig. 17. Optimal path

parameter is imported into the inverse kinematics equation analyzed above for solving. Finally, the angle change of the motor at any time is obtained. By importing the motor, the manipulator can start to move.

C. Simulation

To better Verify the feasibility of the system, the Multibody tool of Simulink is used in our project. Because our machanism is a kind of serial mechanism, the RBT package is not adaptive to it. Consequently, Simscape Multibody, which provides a multi-body simulation environment for 3D mechanical systems, including robots, is chosen. In this tool, we can model the robot with blocks representing bodies, joints, constraints, and force elements. Simscape Multibody formulates and solves the equations of motion for the complete mechanical system.

Here is the system that we build. There are base, motors, links, and joints settled.

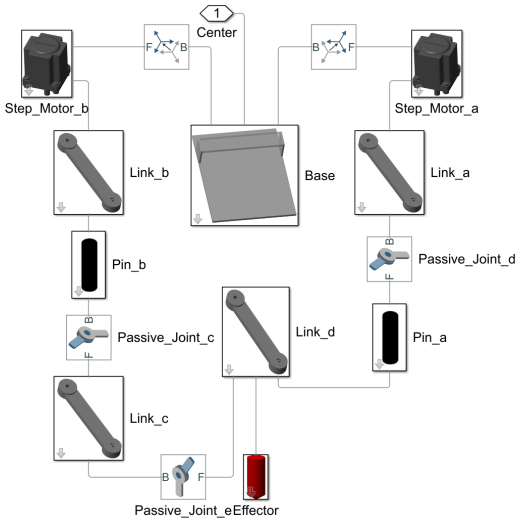

Fig. 18. frame

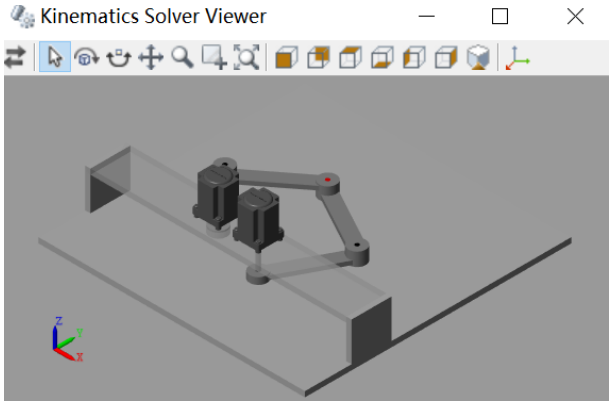

Fig. 19. visual model

Then in the code, we traverse the two angular input and get the workspace in the simulation. Besides, we also give the trajectory for the end effector to simulate the motion.

D. Potential Field Methods

This section mainly introduces potential field methods to avoid singularities. After the robot arm been built, it is found that there are some singularity points, which we hope to avoid.Due to the parallel structure of our manipulator, there are only two motors, so the position of the end actuator can be well represented by the angle of the motors in a plane. The potential field method can easily solve the problem of avoiding obstacles on the plane. In our working condition, there are no obstacles, but it is also the same to avoid the singularity points. Artificial potential field method is a classical path planning algorithm for robots. In this algorithm, the object and the obstacle are regarded as objects with gravitational and repulsive forces on the robot respectively, and the robot moves along the resultant force of all other forces. The potential field

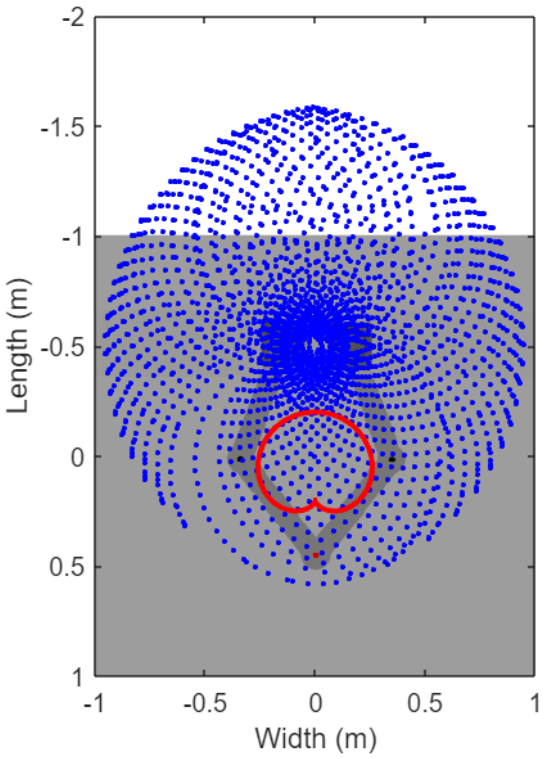

Fig. 20. workspace and trajectory

method generates a force field according to the movement environment, and then the robot starts from the "top of the mountain" with the guidance of the force field, avoids the "small peak" formed by obstacles on the plane, and runs all the way to the "foot of the mountain" where the target position is located.

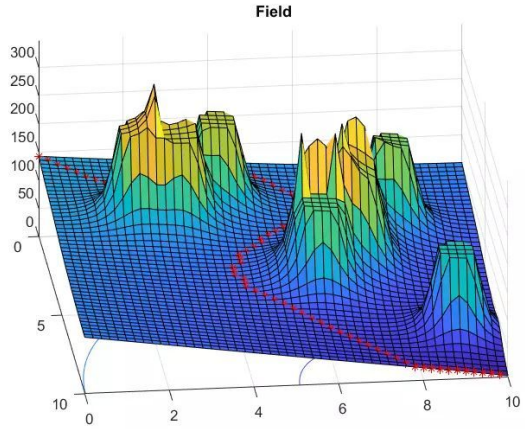

Fig. 21. Potential Field

In the code, we determine the existing position of the actuator and the position to be reached, as well as the points to be avoided and the influence range of the point position on the surrounding area. By calculating the direction of the sum of the virtual gravitation and repulsive force at the current position of the actuator, the next direction of operation is determined. After each short distance, repeat the calculation, and then again, until the final destination is reached.

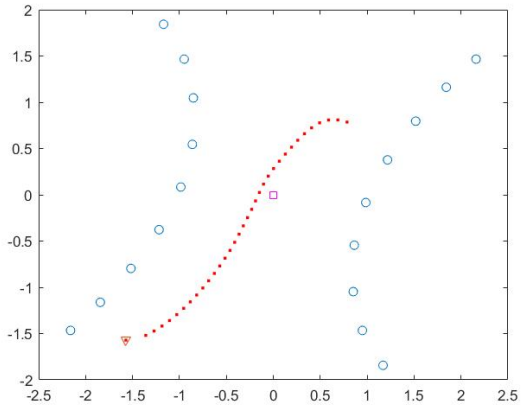

Fig. 22. Path simulation

In Figure 22, you can see that the blue circle is the obstacle point we need to avoid, and the small red triangle is the destination. The red dot is the position of actuator after each movement, and the red dot together is the obstacle avoidance path that we need.

V. Experiments

A. Task-Space Trajectory Following

To validated the correctness of the derived inverse kinematics, we performed task-space trajectory following experiment in diffrent configurations. Due to the link's mechanical design, we were able to perform in multiple configurations such as ++,+-,– and -+. We command the robot to track a circle of 3-cm radius in various configurations, the pictures of the experiments are shown in.

B. Cross-Configuration Pick and Place

To test the robustness of the algorithm and the hardware, we performed the pick-and-place task 10 times and recorded the . The robot succeeded 9 out of 10 times, the failure was due to the insufficient adhesion by the vacuum pump, which led to the object flying when switching between configurations.

VI. Conclusion and Future Work

This paper recorded the development of a direct-drive five-bar-linkage SCARA robot, and the corresponding kinematics and the ability to work across configurations without failing at singularities. For more robust and swift pick-and-place performance, future improvement can be made in the computer vision algorithm, real-time trajectory planning and more compliant end-effector. On the hardware, the potential of a high-speed direct-drive robot can be further explored, such as performing

impedance force control in the task space. With dynamics considered, the robot has potential to perform more tasks than pick-and-place, such as physical human-robot integration in upper-limb rehabilitation robot, robot teleportation joystick with force feedback and more. We hope the hardware we developed can be further exploit by future students taking ME331.

ACKNOWLEDGMENT

This paper summarizes the course project of <ME331 Robot Modeling and Control> in 2022 Fall.

References

[1] H. Asada and K. Youcef-Toumi, "Analysis and Design of a Direct-Drive Arm With a Five-Bar-Link Parallel Drive Mechanism," Journal of Dynamic Systems, Measurement, and Control, vol. 106, no. 3, pp. 225–230, 09 1984. [Online]. Available: https://doi.org/10.1115/1.3149676

[2] N. Hata, F. Ohara, R. Hashimoto, M. Hashizume, and T. Dohi, "Needle guiding robot with five-bar linkage for mr-guided thermotherapy of liver tumor," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004, C. Barillot, D. R. Haynor, and P. Hellier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 161–168.

[3] I. Hassanzadeh, A. Harifi, and F. Arvani, "Design and implementation of an adaptive control for a robot," American Journal of Applied Sciences, vol. 4, pp. 56–59, 2006.

[4] M. R. Khan and J. S. Dai, "Mathematical analysis of finger positions of a spherical linkage metamorphic robot palm," Journal of Mechanical Engineering, vol. 36, p. 18–26, Apr. 2008. [Online]. Available: https://www.banglajol.info/index.php/JME/article/view/807

[5] H. Asada, K. Youcef-Toumi, and Y. Koren, "Direct-Drive Robots, Theory and Practice," Journal of Dynamic Systems, Measurement, and Control, vol. 111, no. 1, pp. 119–120, 03 1989. [Online]. Available: https://doi.org/10.1115/1.3153012