MCEN90028 Robotics Systems

Chess Player Robot Arm

Zixuan Liu 1057135

Zhenlin Liang 900611

Hui Zhang 1086533

May 2021

Contents

1 Introduction of Task

The purpose of the project is to use robotic manipulator instead of humans to move chess. Some basic work has been done in previous assignments. The final specific tasks to be executed of the project can be divided into two. The first one is that the robot manipulator should complete 25 specified movements, and it needs to move one piece to the target position during each movement. The chessboard has 64 squares and the size of each is $3.5cm \times 3.5cm$. The ideal performance of task 1 is not only completing all steps but also accurately put pieces inside the square. The second task requires the robot arm to remain only one direction movement when the human user manually push the end-effector. An advance requirement for task 2 is that it demands robot manipulator to track the moving displacement and change the moving direction. This report will explain what works the team have done and describe how the team integrate them to achieve goals.

2 Robot Design

2.1 Integration from Previous Assignments

In order to achieve the tasks, the team completed some design work in previous assignments:

- In Assignment 1, the team decides that the robot manipulator has five degree of freedom and derives the DH table, Forward Kinematics and Inverse Kinematics. Subsequently, the team initially decides the length of each link which can meet the task space by using the forward kinematics expression. However, when we built our first prototype, we find the links are too heavy for the motor to operate precisely. To improve the performance of our robot, we redesign the link lengths and adopt can weight deduction techniques(Table 1). The initial pose of the robot is shown in figure 1.

- In Assignment 2, the Jacobian Matrix for the joint velocity related to the end-effector velocity has been derived. After verifying the Jacobian matrix through its definition, it will be used to achieve the controller in the further design. Besides, it is necessary for the team to find the maximum torque needed for each joint because the motors have

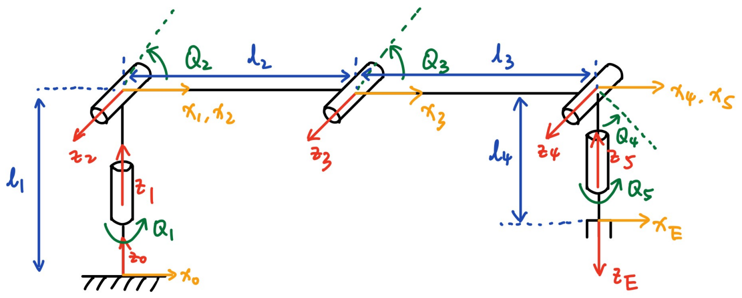


Figure 1: A schematic diagram of the initial position of the robot

Link	Previous Length (cm)	Modified Length (cm)
L_1	30	18
L_2	20	18
L_3	30	20
L_4	15	15

Table 1: The numerical length of links

torque limitation. The team obtains the theoretical maximum torque requirements for each joint in table 2.

Joint	Max Torque (Nm)
$\tau_{1,max}$	0
$\tau_{2,max}$	-1.0081
$\tau_{3,max}$	-0.4645
$\tau_{4,max}$	0
$\tau_{5,max}$	0

Table 2: The maximum torque need for each link

- In Assignment 3, the team generate 3 trajectory using start and final velocities in task space. There is 1 via point in the first trajectory from the ready pose to the original pose. Trajectory 2 from the original pose to the target pose have two via points. Trajectory 3 from the target pose back to the ready pose has 1 via point.

In terms of the operation of the robot, the position vector in the form of a matrix will be used in MATLAB as the reference. It can help the controller compare the current position with it and adjust the motor angle or velocity. More details on the algorithm will be explain in the controller part.

2.2 Hardware

In the design, it requires 5 motors because of the five degrees of freedom. The team has two types of the servo motor. Compare the specification in Table2 and Table3, it can conclude that $FeeTech\,SCS15$ motors is more suitable for installation at the first four joint because their max torque is $1.65Nm$ which is meet the joint 2 max torque requirement. The first degree of freedom supplies the torque for the robot arm horizontal rotating. The responsibility of the second and the third motor is to actuate the link in the vertical plane. The motor in joint 4 will ensure that the end-effector is perpendicular to the ground. $FeeTech\,SCS009$ motor is used in joint 5 to control the gripper opening and closing.

Specification	FeeTech SCS15	FeeTechSCS009
$MaxTorque(Nm)$	1.65	0.076
$MaxSpeed(RPM)$	55	110
$FeedbackRange$(degree)	215	300

Table 3: The motor specifications

Figure 2: FeeTech SCS15

Figure 3: FeeTechSCS009

2.3 Mechanical Design

The material of the robotic arms is wood. Firstly, the team builds a CAD model based on the five degrees of freedom and arm length designed in assignment 1. In order to reduce weight, some links adopt hollowed-out design. Team members mainly use *SolidWorks* and website *OnShape* to sketch and share model parts.

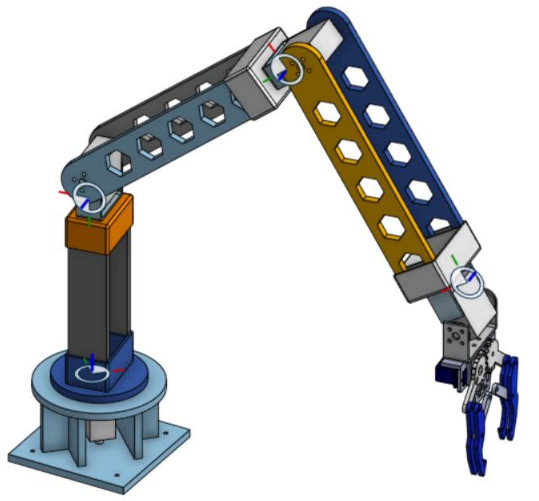


Figure 4: The CAD model of the robot in *OnShape*

Once the team has drawings of links and the basement, it can use laser cut to cut 3mm thick MDF (Medium Density Fiberboard) in workshop. Considering that the chessboard needs to be covered by the task space , its cutting uses 6mm MDF and padded 10cm higher than the base plane. Color connectors are created in 3D printed. The gripper designed a gear structure transmission. In order to increase the friction of grasping chess pieces, a sponge was added at the end of the grip.
It is worth emphasizing that because joint 1 is subjected to the gravity caused by other parts, the team used thrust bearing (Figure5) in basement design to protect motor. In this way, the weight of the robot will not acting on the motor shaft at joint 1. The motors and links connections at Joint 2 and Joint 3 are also installed angle ball bearings to ensure that the robot arms does not stutter when they are rotating.

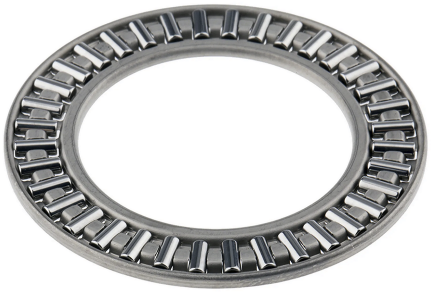


Figure 5: The thrust bearing for the basement design

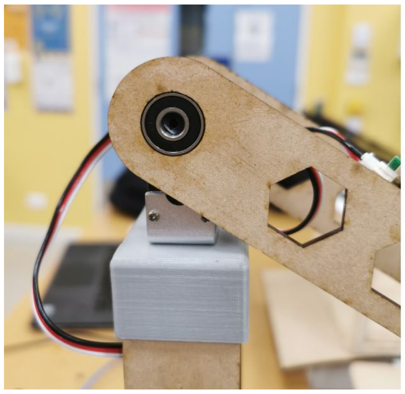


Figure 6: The angle ball bearing in Joint 2 and Joint 3

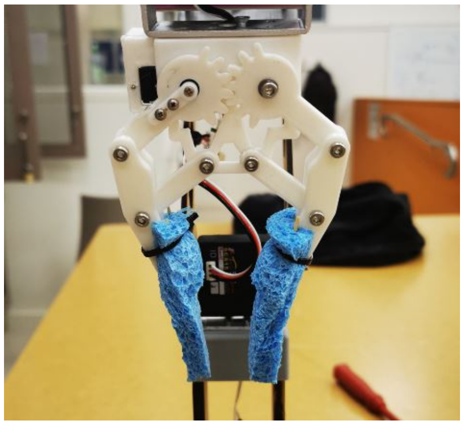


Figure 7: The front view of the gripper

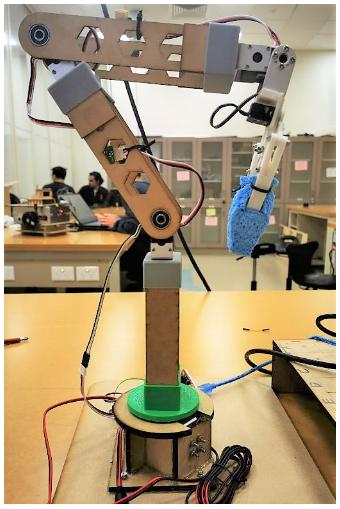


Figure 8: The side view of the robot

2.4 Controller

2.4.1 Open Loop Position Control

In task 1 of the project, the robot is required to pick up a chess piece from one given location and precisely place it at another given location. An open loop position control strategy is chosen to complete this task. When a initial position and a final destination is specify, we can generate a cubic polynomial expression of the trajectory as per described in assignment 3. With a sampling time of 0.02 seconds, we can obtain a set of 3D coordinates which can represent the generated trajectory. Afterward, we can apply the inverse kinematics, as discussed in assignment 1 to find the angular position of each joint corresponds to each paired of 3D coordinate. The calculated angular positions will be sent to the motors every 0.02 seconds to control their motions.

The reason we choose to use position control instead of velocity control is because the SCS15 motor has a poor accuracy on velocity mode. According to its specification, it can only be controlled to operate in a certain levels of speed, from 0 to 9, which is not a specific amount, such as 1 radius per second. Although a stable close-loop controller can ensure the robot arm can converge to a certain position, this may take a large amount of control effort and a relatively long time. Meanwhile, using close loop velocity control may end up with an elbow up position of the robot which is also something we want to avoid. Besides, the SCS15 motor has an embedded PID controller to regulate its rotation when operating in position mode. Therefore, using a simple open loop position control technique can yield a good performance. By trial and error, we set the gain of the PID controller as $K_p = 9$, $K_I = 5$, $K_P = 9$.

2.4.2 Task Space Velocity Control

Task 2 of the project requires the robotic arm can only be free to move in one specific direction and resist the movement in other directions. To carry out the task, a task space velocity control scheme with a P controller is applied to the system. The control law can be written as

$$\dot{x}_c = \dot{x}_{ref} + K_P \, e_c$$

By trial and error, the controller has the best performance when

$$K_P = \begin{bmatrix} K_{Px} & 0 & 0 \\ 0 & K_{Py} & 0 \\ 0 & 0 & K_{Pz} \end{bmatrix} = \begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}$$

If we want the robot be able to move in one specific direction, x, y or z, we can set the gain of that corresponding direction to zero. After obtaining $\dot{x}_c$, we can find $\dot{Q}_c$ which can be directly fed in the motor as a input to control its motion, using the following formula.

$$\dot{Q}_c = J^{-1} \, \dot{x}_c$$

The problem with a simple P controller is that there may be a steady-state error. The reason we did not choose a PI controller which can solve this issue is because we try to avoid winding up. We did try to implement the robot with a PI controller. The motor will produce a unnecessarily large speed when we move the robot in a direction that is resisted by it and keep the position for a while. The large speed will cause a damage to other parts of the robot. Thus, we finally decided to use P instead of PI even it has a slight steady-state error.

2.5 Calibration

After assembling out robot, two tests were done to examine the precision. Firstly, we sent our robot to the initial (fully stretched) position, as shown in figure 1. Because of some installing error, link 2 and 3 were not perfectly parallel to the ground. Therefore, we added a constant bias to each joint to compensate the error. Secondly, we chose five coordinates randomly and sent the robot to these positions one by one. Since our robot can reach all of these positions with an acceptable precision, we decided to take the designed link length as the actual length.

3 Automation

The robot is controlled by the given Arduino Compatible Mega and the FE-URT-1 signal converter. All commands are sent to the motor through Arduino IDE and all calculation are done in MATLAB.

In task 1, a pre-built loop will be ran to perform the 25 required movements listed in assignment 4. For each movement, an initial location and a destination will be sent to a MATLAB function which will generate a sequence of angular positions Q that corresponds to the cubic polynomial trajectory. And the angular positions Q will be sent to each motor respectively by Arduino as a position command. In every movement, the end effector will move in following sequence:

$$Ready\,Pose \Rightarrow Initial\,Location \Rightarrow Destination \Rightarrow Ready\,Pose$$

when all 25 movements are finished, the function will automatically stop. However, the robot will remain power on and in position control mode. Without further command, the robot will hold still at the ready pose with end effector released.

Starting task 2 by running the task 2 execution function in MATLAB. In this function, a position control which similar to task 1 will be used firstly to pick up the chess piece and the end effector will stop when reaching the first via point. Secondly, the motor will be switched to velocity control mode and the P controller, as per discussed in controller section, will start operating with all three directions constrained. The coordinate of the via point will be used as the reference of the controller. During the mode switching process, the motors will lose power for a few seconds but it will get back to the reference position once the P controller starts functioning. There are three MATLAB sections which corresponding to the three movements that listed in the requirement of task 2. The MATLAB codes need to be manually run to perform the task. Each time the codes have been run, the P controller will operate for about 15 seconds.

4 Video

The video is available on the YouTube: https://youtu.be/LNQqpI5VklY

5 Discussion

5.1 The manipulator design

The End effector is designed to utilise Mechanical Pinch way to pick up and place the chess piece from the original location to the destination. The design of gear is used to control the clamping and loosing process of the End effector by the servo motor.

A relatively thick claws is helpful to hold the chess piece tightly. Sponge slice is added to the gripper to increase friction and contact area with the chess piece. In addition, the small stick is added to the sponge slice to hold the shape while it extruding each other.

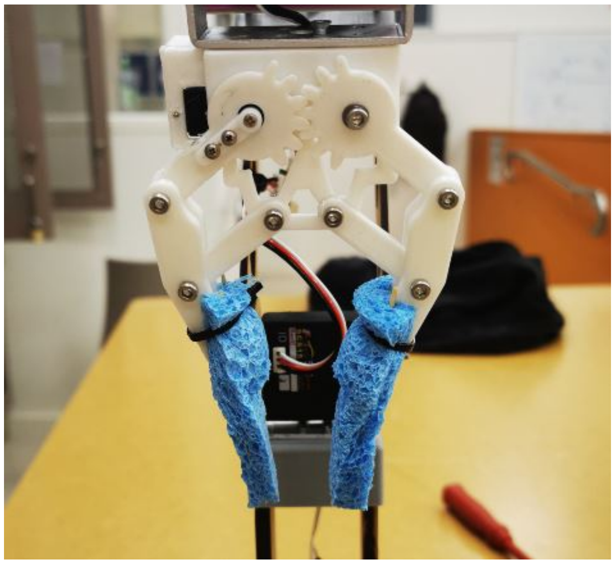


Figure 9: The design of End effector

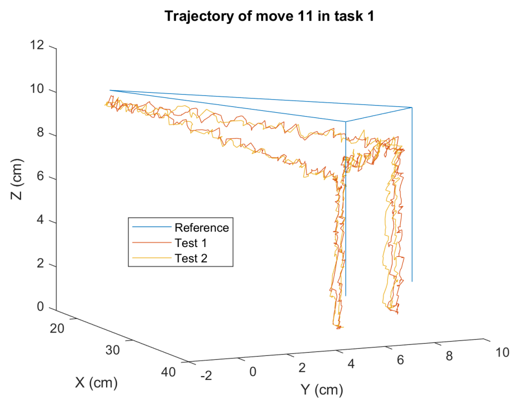
5.2 Performance

- **Chess Piece Pick-up Accuracy**: The definition of successful attempts is that the End effector is directly above the chess piece and hold it until it is ready to place. In the process of demonstration and tests, our chess piece player robot arm perform high accuracy on picking up the chess pieces. For task1, 25 out of 25 successful pick up attempts in the first test run and 25 out of 25 successful pick up attempts within the second test run.

- **Chess Piece Placement Accuracy**: The definition of successful attempts is that the End effector is directly move above to the destination and place it in the square without touch on the lines. In the process of demonstration and tests, our chess piece player robot arm perform high accuracy on placing down the chess pieces. For task1, 23 out of 25 successful placement attempts in the first test run, the piece was put on the edge of square in the two failed cases, and 25 out of 25 successful placement attempts within the second run.

- **Reference Trajectory Error**: Reference trajectory error can be measured by the distance between the actual and reference position of the end effector at the same instance of time. By collecting the angular position feedback provided by the motors, we can generate the actual trajectory of the end effector using inverse kinematics. Figure 10 shows a comparison of the reference and actual trajectory of movement 11 and 18 in task 1. And figure 11 illustrates the error in distance of both movements versus time. According to these figures, generally the actual trajectory match the reference. At the end of each trajectory, the offset is less than 1 cm which can still ensure the chess piece will be placed inside the square box. Figure 13 shows the instantaneous angular position of each joints in both two movements. According to this figure, mainly the error is caused by joint 2 and 3. Because the motors at these two joints need to support larger gravitational torque comparing to other joints.

- **Repeatability of Chess Piece Placement**: Repeatability represents how well the robot can repeat its performance. Regarding to our robot, it can be measured by the difference between the actual trajectories of the same movement of multiple runs. In generate, the robot has a good repeatability. It has a 95 percent rate of successfully completing task 1. To scientifically examine the repeatability, we randomly selected two movements that listed in assignment 4 (movement 11 and 18); let the robot to execute each movement twice; recorded the sequence of the angular position of each joint; compared the actual trajectory of both runs. The four trajectories are shown in figure 10 and figure 12 shows the offset between two runs in different time instance. These two figure suggest that the robot has a good repeatability. The maximum offset between two runs in both movements is less than 1.4 cm. In particular, the offset at the end of trajectory is less than 0.4 cm. The dimension of each square of our chess board is 3.5cm * 3.5cm. Thus, the uncertain (offset) of different runs will not greatly affect the performance.



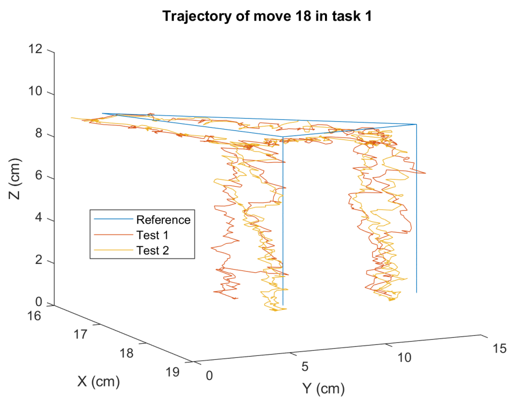
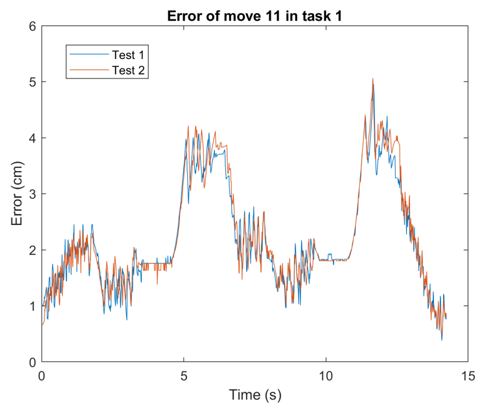


Figure 10: Reference and actual trajectories of move 11 (left) and move 18 (right) in task 1



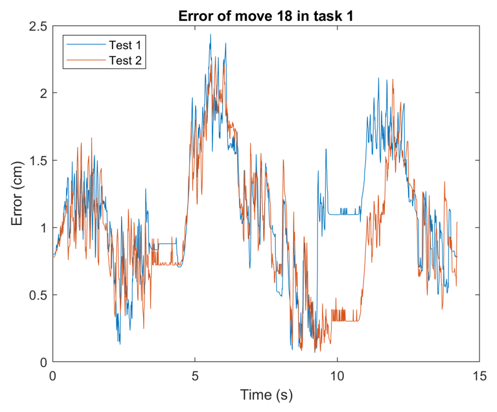
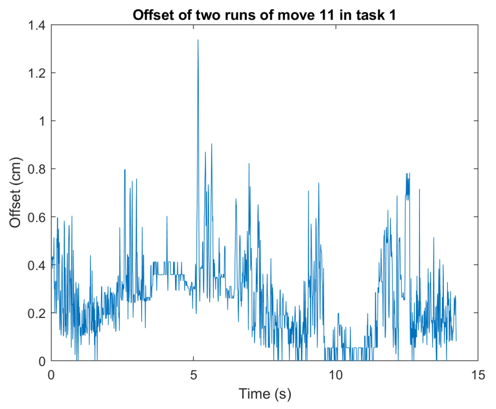


Figure 11: Error in distance of move 11 (left) and move 18 (right) in task 1



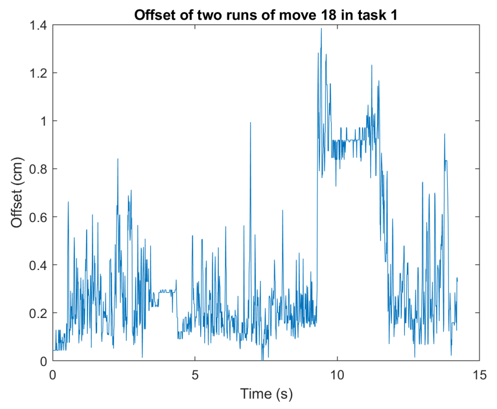


Figure 12: Offset of two runs of move 11 (left) and move 18 (right) in task 1

5.3 Limitation and Improvement

- **Reachable space is not cover the whole chess board:** There is a trade-off between the reachable space and performance. Because of the torque of our servo motor generated is relatively limited, if the link is too long or too heavy that our motor would have relatively low accuracy in rotating the links and cause the inevitable misalignment of End effector between designed location and the location in the real world. Therefore, we sacrifice a little reachable space of the robot arm to decrease the length of link which also meets the requirements of task1 for the high-performance.

  Solution:

  - High-Torque or more robust motor: High-Torque motor is able to generate more torque to rotate the link to the right pose which increases the accuracy of our robot arm.
  - High Controller Gain: After the calculation in the Assignment 2, the torque generated by the servo motors is enough for our robot, if we redesign the inbuild PID Controller of the motors which would greatly help the motor reach the desired angle.
  - Lighter links: The material of our link is 3mm wood provided in the workshop, we can choose another low density material with high structural integrity to replace the wood, such as PLA material.

- **Chess piece falling while moving:** There are two reasons related to the chess falling. One is the End effector cannot hold the chess piece tight; the other one is the friction between the gripper and the chess piece.

  Solution:

  - Redesign the End effector: One valid improvement is thicken the part of claws and adjust the angle of gear connection.
  - Increase friction of gripper: Sponge slice is added to the gripper to increase the friction contact point with the chess piece. In addition, the small stick is added to the sponge slice to hold the shape while it extruding each other.

- **Weak base while high speed rotation:** Heavy robot arm requires a strong base (which is the Joint1 in our design in Figure 1) to support its weight during rotation along z-axis. However, we use a 3D-printed base connected with the joint1 which has low structure integrity and makes the entire robot unstable. High speed rotation around the base would result in lower accuracy.

  Solution:

  - Rebuild the base: A larger base plane have advantage on sharing the weight of the whole robot arm and relief pressure of the thrust bearing.

- Heavy of the whole robot: The weight of Link occupies more than half of the weight of the robot, which has a certain degree of influence on the accurate realization of the predetermined task.

  Solution:

  - Redesign the link shape: Cutting off part of the link and maintain structure integrity, we can perform finite element analysis on the redesigned link shape and material to determine the final version.

- **Excessive time spent on the robot movement:** The travel time for each segment is set to be 3 seconds, the whole process of task1 consumes approximately 8 minutes to finish. **Solution:**

  - Personalize the running time: While maintaining stability and accuracy, a unique running time is adopted for each movement.

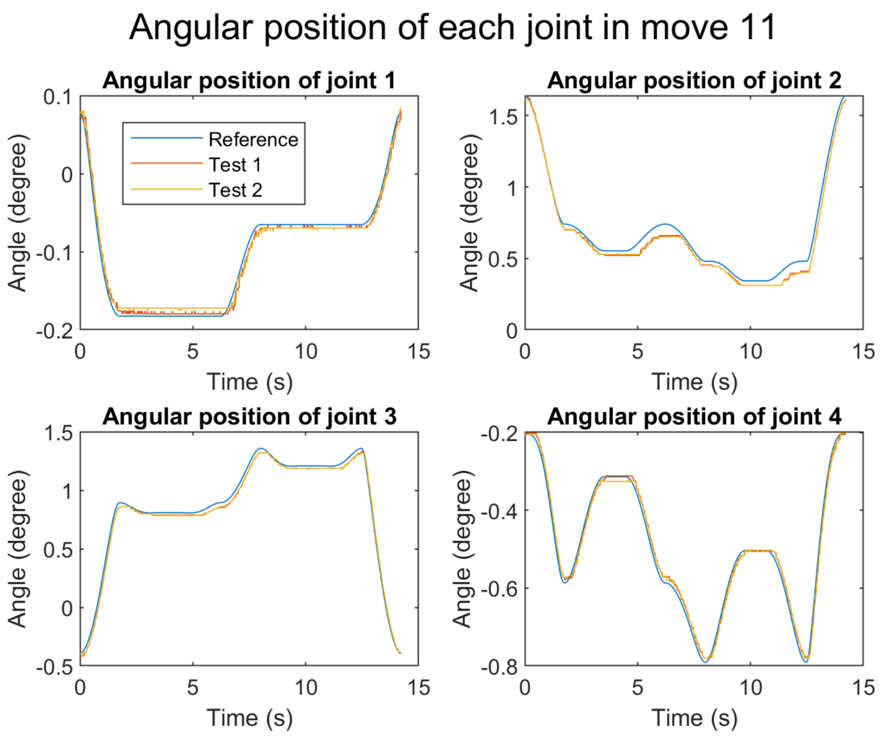
6 Conclusion

In conclusion, the whole design process of the robot arm for chess player is presented, including the calculation of DH parameter, the determination of link length and reachable space by forward kinematics and inverse kinematics, the CAD model designed by Onshape, the usage of thrust bearing and angular ball bearings, the wood link cut by laser cutter, the connection between links printed by 3D-printer.

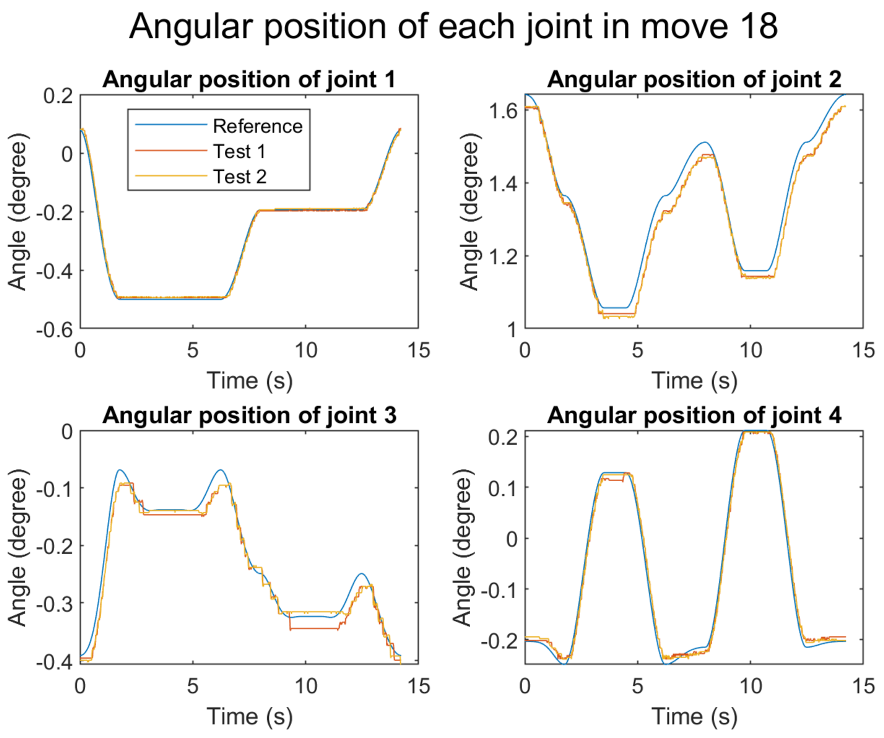
The whole implementation of specific tasks is also introduced, including the generation of trajectory, the determination of via points, the algorithm of different tasks, the evaluation of the performance, the discussion of the limitation and the potential improvement in the further design.

After the demonstration, the designed robot is able to pick up the chess in the desired location and placed it at the destination along the specification in the task1 correctly. Subsequently, the robot also able to move under the limit that listed in the task2 relatively accurate.

Furthermore, several limitations and constrains are presented and the potential improvement and innovations in our further design related to them are also introduced.



(a)



(b)

Figure 13: Angular position of each joint in move 11 (a) and move 18 (b) of task 1