Kai Bing Goh, William Delaporte, Yi Gao & Yixiang Gao
981436, 915100, 1208446, 990803
Department of Mechanical Engineering
The University of Melbourne
VIC 3010, Australia
kaibingg@student.unimelb.edu.au
wdelaporte@student.unimelb.edu.au
yigao4@student.unimelb.edu.au
yixiang1@student.unimelb.edu.au

Robotics Systems Assignment 5
Final Report

MCEN90028 : Dr. Justin Fong & Dr. Tomislav Bacek
Due –*Friday, May 27, 2022*

Table of Content

1 Introduction

This report details the process of designing a robot arm to perform the task of moving chess pieces.
The arm has two primary functions:

1. The designed robot arm is able to pick up a chess piece from the board and place on another square or to the side of the board, in the case that the piece is taken. No touching of other chess pieces is allowed during the movement process.

2. After picking up a chess piece, the robot arm should allow manual movement of the end-effector to intended positions. The direction of movement allowed depends on the type of piece that is picked up. For example, a bishop should be able to move diagonally while a knight should be able to move in it's distinctive L-shaped pattern.

In the previous 3 assignments, we built a DH table, determined the dimensions for each of the robot's arm to ensure that the workspace is covered, derived Jacobian matrices to relate velocities and forces in joint space and working space, and generated trajectories to move the chess pieces from one position to another. This paper will mainly focus on the physical implementation of previous tasks and evaluation of the performance via real world testing.
For ease of understanding, the prototype and working space are shown in figures below. The DH table is shown in table 1.
A chess board is manufactured with a square width of $3cm \times 3cm$, and there is a $2cm$ border around the board. The base of the robot is positioned in the center (y=0) and $10cm$ below the border of the board. This gives the workspace a total length of $L = 40cm$, and a total width $W = 38cm$. The board is assumed to have a thickness of $3mm$, and the maximum height of a chess piece is measured to be a maximum of $6cm$.

a_{i-1}	α_{i-1}	d_i	θ_i
0	0	d_1	Q_1
0	90	0	Q_2
a_2	0	0	Q_3
a_3	0	0	Q_4
a_E	0	0	0

Table 1: DH table with parameters

2 Robot and Task integration

2.1 PID Controller

In task 1, the method is task space position control, so we directly use the function"setPID" to set the parameters for the motor's in-built position PID controller. These parameters are shown in table 2.

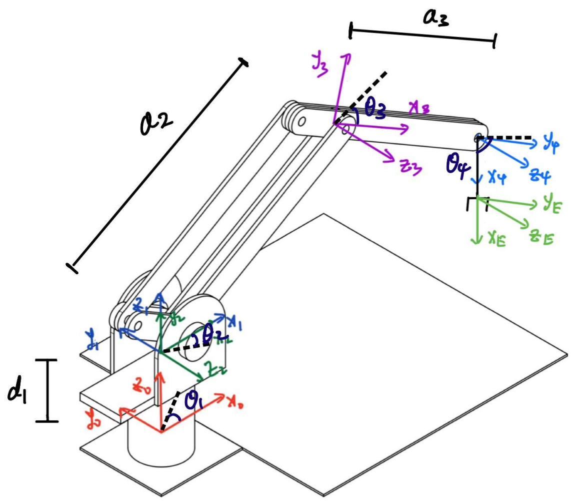


Figure 1: Schematic for the Prototype of the Robot Arm

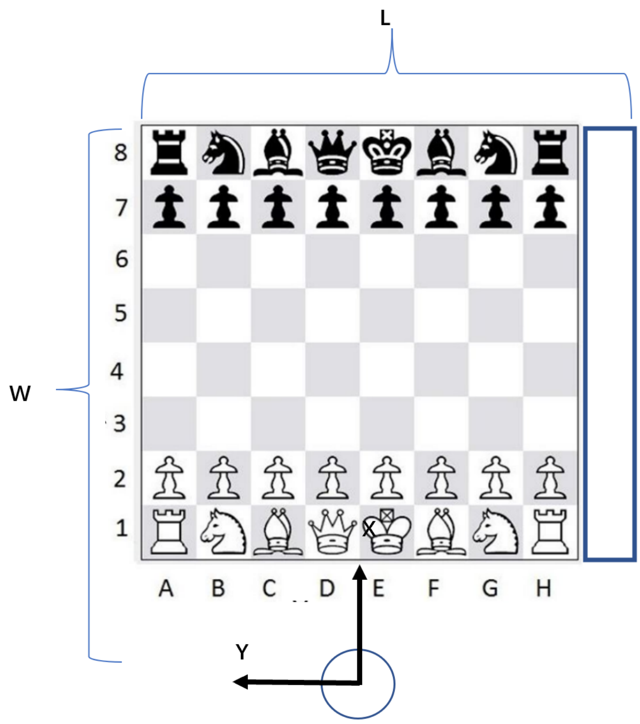


Figure 2: Workspace of Manufactured Board

Motor ID	K_P	K_I	K_D
1	10	10	10
2	10	3	1
3	10	3	1
4	10	1	1

Table 2: PID controller parameters

For the 5th and last motor, which will actuate the gripper, simply rotates between two specific angles which corresponds to its releasing state and grabbing state, the PID controller does not need not be tuned for accuracy. Furthermore, a PID controller that is too accurate could lead to the motor oscillating slightly about the desired value, which could negatively impact the gripper from holding the piece properly.

2.2 Task 1 design

The joints are controlled by sending the desired angles to the motors. The first step is to calculate the position we need to reach. Figure 3 shows a list of the chess moves the robot must accomplish.

Robot Moves			Chess Notation	
Move #	From	To	White	Black
1	e2	e4	e4	e5
2	g1	f3	Nf3	Nc6
3	f1	b5	Bb5	Nf6
4	b1	c3	Nc3	Bc5
5	e1	g1	0-0	d5
6	h1	f1		
7	d5	captured pool	exd5	Nxd5
8	e4	d5		
9	d5	captured pool	Nxd5	Qxd5
10	c3	d5		
11	c6	captured pool	Bxc6+	bxc6
12	b5	c6		
13	c2	c3	c3	0-0
14	f3	g5	Ng5	e4
15	d2	d4	d4	exd3(ep)
16	d1	f3	Qf3	d2
17	d5	captured pool	Qxd5	dxc1=Q
18	f3	d5		
19	c1	captured pool	Raxc1	cxd5
20	a1	c1		
21	g1	h1	Kh1	Bb7
22	f2	f4	f4	Rfe8
23	g5	h3	Nh3	Rad8
24	g2	g3	g3	Be3
25	c1	d1	Rcd1	f6
26	f1	e1	Rfe1	d4++

Figure 3: list of Movement

As the chess moves are represented by the name of the squares, for example 'a1' represents the first column and first row counting from the bottom left corner of the board when playing as white. From the letter and number, we wrote a simple function called "coord_generator" to convert the chess square to a task space coordinate. Then, these XYZ-positions can be used with the trajectory generation derived in assignment 3 to calculate the intermediate positions. These are then used as inputs for the inverse kinematic function created in assignment 1 to generate the corresponding rotation angle for each joint. Finally, these angles can be used to find the correct position signal to send to each motor. The details of this process are covered in the next section.

2.3 Task 2 design

In task 2, we use a velocity controller to implement the manual movement of a rook (X or Y direction),a bishop(diagonal direction) and a knight(L shape).In this task, we also need the transformation between joint space and task space using forward and inverse kinematics.The reference position of the end-effector is firstly calculated using joint angles.During the process of manual movement of end-effector, the task space positions are calculated at every sample time.The Jacobian matrix is then used for transforming task space velocity command to joint space velocity command. More details about implementation will be discussed in next section.

3 Task Specific Motion Planning

3.1 Task 1

The steps to picking up a chess piece are shown in figure 4. This section will describe in detail how each part can be implemented to complete a chess move and how to improve accuracy. As explained in section 2.2, all the initial and final positions are fixed. Where we can improve the performance is trajectory generation, inverse kinematics and adjusting.

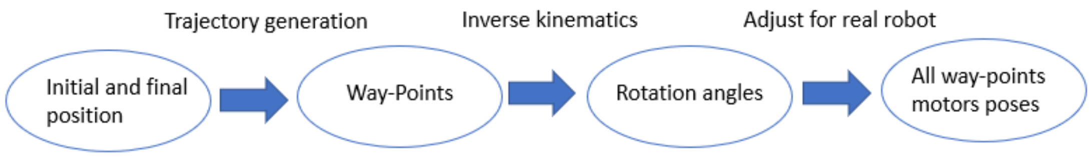


Figure 4: Process of control motor

In theory, more way-points implies greater accuracy and control over the end effector pose. However, since our approach needs to calculate the inverse kinematics for each point in the trajectory, the process may take a significant amount of time. By reducing the number of way-points calculated via inverse kinematics and simply plotting a straight line between them, the computation time can be greatly decreased. With our trajectory algorithm, the number of way-points required to plot a course can be reduced to 7. Even with this number, the calculations can still takes more than 3 seconds to complete. Since the motors are being used in position control mode, their speed cannot be set to make the motions smooth. To fix this, we use "linspace" to generate more intermediate points between two way-points in joint space. Using this command, the speed of the movement increases for way-points that are further apart. We compared the size of the two adjacent pose of each joints by dividing the largest value by 0.01 to get the number needed for the "linspace". This method produces a negligible increase in calculation time and allows for smooth movement between way-points

Secondly, in order to prevent the gripper from touching the chess, we altered the inverse kinematics function to use a gripper length 1.6cm longer than the actual manufactured length. This is a simple way to ensure the gripper never gets too close to the board.

Finally, there is a need to convert the calculated joint angles into appropriate motor input angles. Due to the placement of some of the motors within the physical robot, these do not always match the angles derived in previous assignments. The position shown in figure 5 is the initial position, where the motors 1,2,3 and 4's angles are set to 0. $\theta_{m1} = \theta_1$ is identical to the calculations, but θ_{m2}, θ_{m3} and θ_{m4} are very different. Motor 2 controls the angle between link 2 and the vertical, but in our inverse kinematic calculations, θ_2 is the angle from the horizontal. Therefore, it takes as input the angle that is complementary to θ_2. That is, $\theta_{m2} = \theta_2 - \pi$.

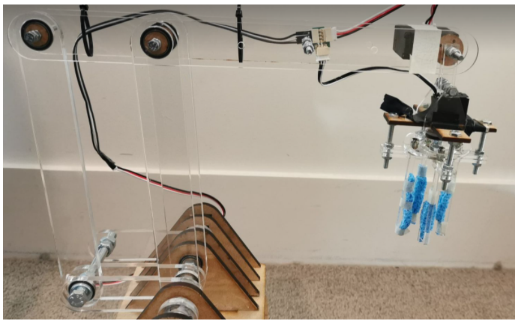


Figure 5: Initial position

Our robot arm is not designed to have motor 3 installed on the arm. Instead, link 3 is controlled with motor 3 placed on the base and controlled via the parallel linkage, as shown in figure 6. Due to the design, the rotated angle of θ_3 is different. The actual signal sent to the motor should therefore be the sum of desired θ_3 and θ_2 from our inverse kinematics. Also, due to the way the motor is oriented, link 3 is lowered by decreasing the motor angle. This requires that the angle also be inverted to achieve the desired motion. Therefore, the motor input angle is given by $\theta_{m3} = -(\theta_2 + \theta_3)$

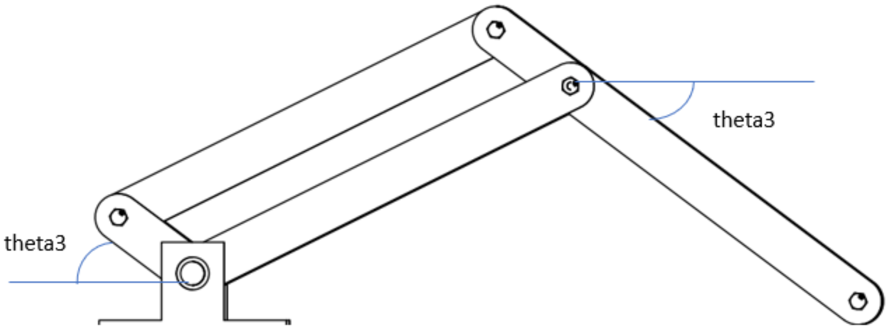


Figure 6: Link 3 side view

Since, θ_4 is dependent on the values of θ_2 and θ_3 in the inverse kinematics, changing those also requires a change to this. Since the parallel linkage sets the angle of link 3 relative to the ground, the angle θ_{m2} has no effect on the required angle θ_{m4}. Instead, to keep the gripper perpendicular to the ground, motor should be set to the same angle as motor 3. However, the motor is also oriented in the opposite direction, so the angle also needs to be inverted. Therefore, $\theta_{m4} = -\theta_{m3}$.

3.2 Task 2

In task 2, a velocity controller is required which will allow for manual movement of the end-effector while following the game rules for chess. For example, if a rook is being held, the arm must allow movement in a straight line in the X or Y direction. Thus, when moving a rook in the X direction, there should be no control effort in this direction, but with velocity controlled in all other directions. The velocity controller is shown in figure 7.
In this task, we use a proportional controller K_p and we assume $\dot{\vec{X}}_{ref}$ equals to 0.

The velocity command in task space is given by: $\dot{\vec{X}}_c = K_p * \vec{e}$=$K_p * (\vec{X}_{ref} - \vec{X})$
The velocity command in joint space is given by: $\dot{\vec{q}_c}$=$J^{-} * \dot{\vec{X}}_c$

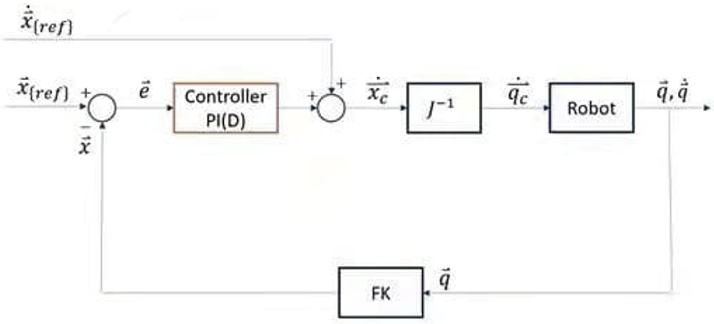


Figure 7: Velocity Control

To implement the function of manual movement for a rook, K_p is set to 0 in the direction we move. Through trial and error, we get $K_p = [0, 2, 2]$ (3 values for x,y,z translation) when moving in the X direction and $K_p = [2, 0, 2]$ when moving in the Y direction.

At each sample time, the robot reads the current position and subtracts it's reference position, which is set using the initial position, to get the position error e. Then using the equations above to calculate the joint space velocities, values are sent to the motor to implement velocity control. As K_p is set to 0 in the direction we move, there is no control effort in this direction. However, in other directions, the controller would force the end-effector to track the reference.

The pseudo-code is shown in below:

```
While(manually moving rook)
        read current position
        calculate reference error
        calculate task space velocity command
        calculate joint space velocity command
        send joint space velocity command to
          motor to execuate velocity control
```

Figure 8: pseudo-code for velocity control

3.3 Calibration

The results in reality are not, as a rule, the same as the simulated ones. It can be influenced by various aspects. For example, the angle of rotation can be randomly off because of tolerances between the motor's teeth and its coupling disc, or because of friction within the bearing. Therefore, we set offset the motor angles to deal with this problem. From multiple rounds of testing, the offset is chosen to be -0.32 and if the y-axis value is less then 1.49 the offset, it should be -0.325 to account for the room to move in a different direction. Similarly, θ_3's offset is -0.02 and θ_4 is -0.43.

On the other hand, because of its own gravity and motor power, the front end will sink slightly when the arm is extended forward. So we force the end up by adding gripper to the inverse kinematics function. The original gripper is 15.5cm and we set to 16 cm such that it does not touch the chessboard. The bigger of the x-axis value, the larger of gripper should be. When x bigger than 16.5 and less than 22.5 the gripper we set 16.4cm. If x bigger than 22.5, the gripper is 16.9cm.

Finally, PID controller can make the angle as close as possible to the encoder's measured value. However, it need some time for reference tracking, which demonstrated back and forth self-adjustment movements. Therefore, we did not set K_D to be very large, which may slow down the settling time, but still ensuring to prevent the pieces from swing and hitting the others. Of course, the trade-off for this decision is a decrease in accuracy by a small amount.

4 Mechanical Design

In terms of the materials involved in building our robot manipulator, we mainly considered 6mm MDF for the base, and thickness of 3mm and 6mm acrylic for the arms due to its availability in the Telstra Creator Space. At the very bottom where the robot sits on the manufactured chess board, there are nuts which are tightly friction fitted onto the chess board, so that the bolts securing the robot can be easily taken out when needed, which was very helpful throughout the manufacturing process. Due to this, we chose MDF as our primary material because we have experienced cracking of acrylic during friction fitting. Moreover, the bolts are supported by layers of 6mm MDF circular cutouts as shown in figure 9 below.

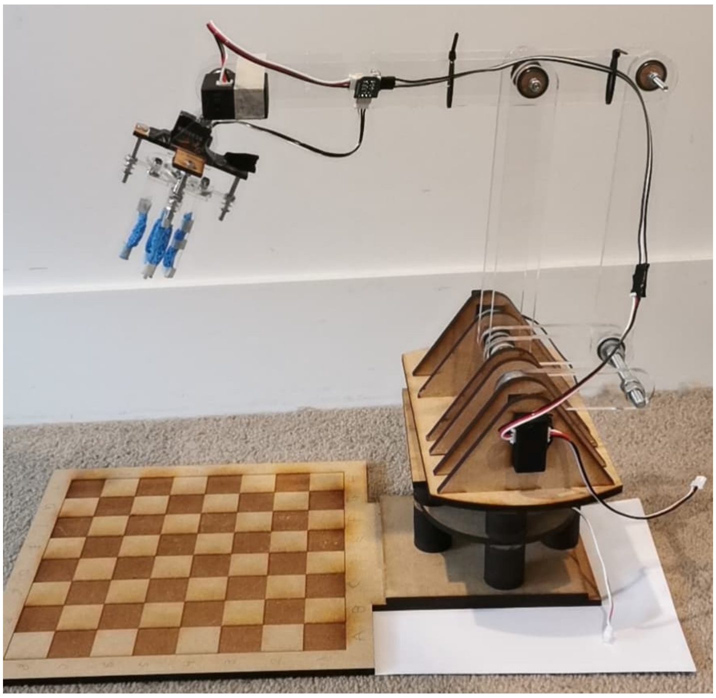


Figure 9: Final Iteration of the Robot

For the arms, initially we had set out to use MDF with multiple small cut outs at different areas of the arm. However, this had created major issues where the robot had visible wobbling effects and had severely affected the end effector. The other chess pieces around the targeted piece would get knocked over due to the oscillations of the end-effector in the x-y plane when it was travelling down vertically. This behaviour was unacceptable and in order to mitigate this shortcoming, simulations to calculate torque required were carried out to prepare for the next iteration. As a result, it was decided to remove the cut out areas and use 3mm acrylic for link 3, and 6mm acrylic for the rest of the links of the parallel linkage excluding the end-effector. From our observations, this iteration provided a more robust motion which do not suffer from major oscillations, and was able to perform the predicted motions in a more consistent and robust manner to the simulations that are carried out prior to manufacturing.

For the designs of the joints, radial bearings were utilised where necessary, including the additional joints of the parallel linkage. This had prevented any friction or backlashes from during the rotational movements of the joints. Due to incorrect size of the bearings bought, discs which were friction fitted between the shaft and the inner diameter of the bearing were added. To secure the arms on the robot, 3D CAD models were designed as shown in the generated drawing in figure 1, as a result walls were manufactured with rectangular or circular cutouts to mount the motors and the radial bearings which will hold the shaft (a bolt). In order to align it, nuts were used to secure the components as shown in figure 10, i.e. the MDF between bearing and shaft, arms for link 2 which are tightly bitten by nuts and rotates when the shaft/motor rotates. Due to its symmetrical arrangement, the shaft length is measured and used to design the platform to install the innermost left and right walls.

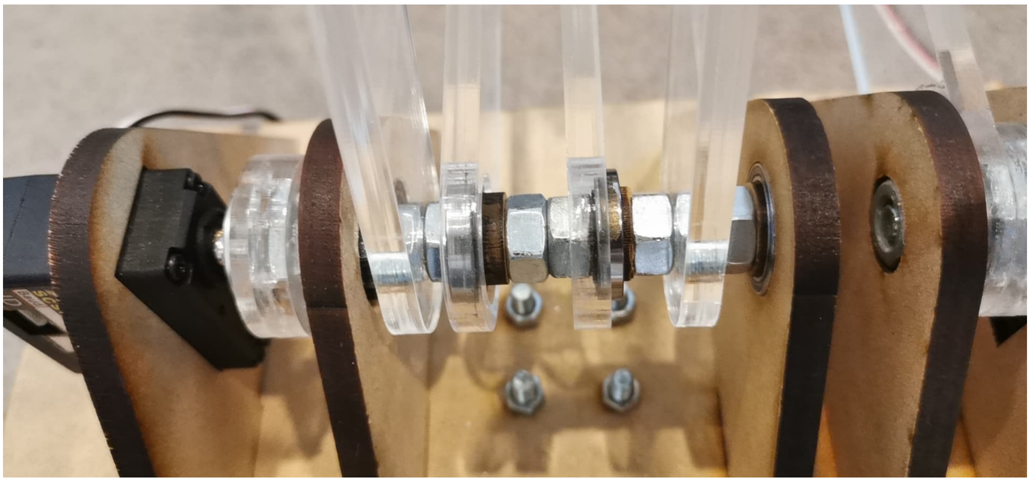


Figure 10: Aligned Position of Bolts and Walls Fitted to Hold Motors

With further measurements and iterations in Fusion360, the resulting design allows for a perfect alignment of not only the positions of arms used for the parallel linkages on the shaft, but also the parallel axis of rotation of joint 2 and 3 on the platform.

Figure 11: Same Axes of Rotation of Joint 2 and 3 Observed From the Side

Beneath this platform shown in the figures above, a thrust bearing was installed and can be demonstrated in figure 12(b). The thrust bearing is able to prevent any friction between the platform and the topmost surface of the base, additionally it also supported an axial loads of the entire weight of all components sitting above the bearing. Provided that the friction caused from the weight of the entire arm and the rough MDF surface is not negligible, this bearing is undeniably a necessary component because any minor backlash caused in this joint can ultimately result in large inaccuracies and deviations in terms of the precision of the end-effector relative to the desired position. In terms of the mechanism involved in rotating the platform, 4 bolts with their heads attached to the motor's coupling disc were secured using nuts from the top surface of the platform , thus allowing the any weight from above to sit on the thrust bearing.

(a) (b)

Figure 12: (a) Thrust Bearing Purchased (b) Installed Thrust Bearing

Another important design is using discs with hexagon cutouts (shown in figure 13) to be friction fit the hexagonal shaped heads of the bolts, where these discs are screwed together with the provided motor's metal discs. Due to different sizes of bolts used depending on the joints, i.e. smaller radius bolts for arms further from the base to reduce torque, many iterations have been made to correctly achieve a secure fit of the nuts to the discs. Glue was additionally applied when the design was finalised after many testing

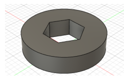
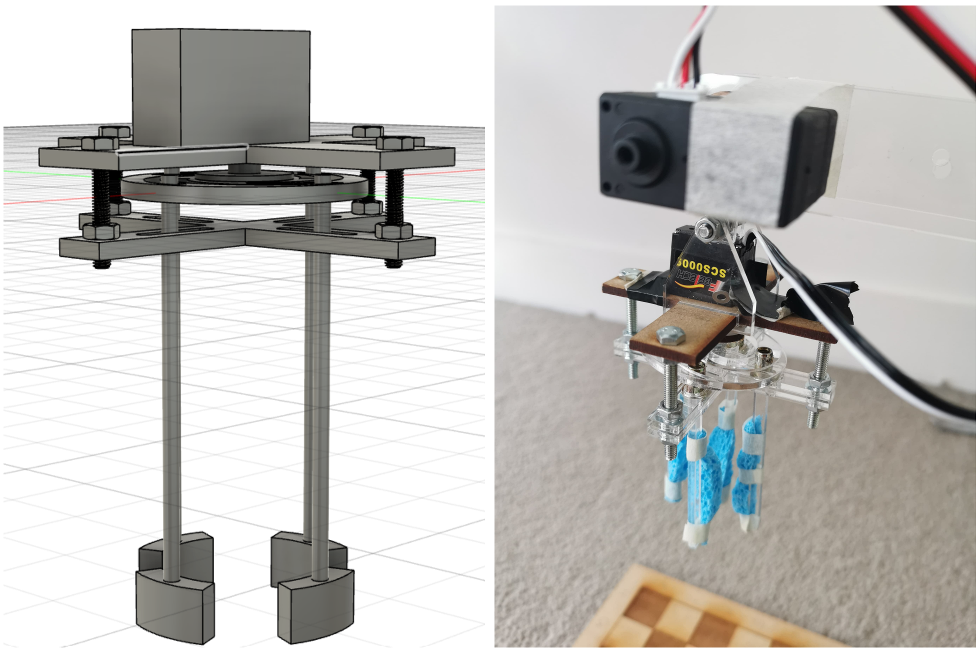


Figure 13: Disc for Driving the Shaft (Hex Bolt)

End Effector Gripping Mechanism

Figure 14(b) shows the final end product of the gripper, where this design is based around the mechanism of a rotating disc with spiral paths cut outs to drive 4 vertical beams inwards and outwards as the disc rotates, but each beam are constrained with only movements in the x or y axis depending on the beams. Below paragraph breaks down the structural aspects of the gripper and details its operating mechanism.



(a) (b)

Figure 14: (a) Prototype Gripper Ideation in CAD (b) Final product of manufactured Gripper

First, the SCS009 motor was friction fitted onto a top plate, which also had a purpose to hold all other plates that would be required below. Below the top plate is where the provided disc for the motor is attached, and this disc is tightly secured with the manufactured disc with spiral paths via bolts. Beneath the discs are 2 ”+” shaped plates, which are held by bolts and nuts connected to the top plate, and the 2 plates sandwiches the flat part of the semi-tubular rivet with a tolerance of the thickness of M4 washers to allow for movement. The rivet is generously tolerance fitted through the spiral disc, and acrylic beams were manufactured to be attached to the other end of the rivet. When everything is assembled, as the spiral disc rotates, this will drive the rivets inwards (for example), the acrylic beams connected below, due to its rectangular dimensions, are only allowed to move inwards of the x/y paths on the "+" shaped plates. The sandwiched flat part of the rivet would ensure that those beams will always stay perpendicular to the chess board when torque is applied (from chess pieces) at the bottom end of the acrylic beams. To prevent friction, we chose 3mm acrylic for most of the components as shown in this image and 3mm MDF for non-crucial mounting parts to reduce weight. Finally, sponges were utilised to hold the chess pieces due to the sufficient friction it provides and its elastic properties, where it solved our issue of deciding thresholds on how wide or narrow the gripper should operate from grabbing the varying chess pieces' radii.

5 Discussion

5.1 Results

The results are as expected. There is a pause of almost 2 seconds for each step, because the machine holds the current pose while computing the inverse kinematics of the way-points. This time delay is inevitable, hence the visible slower performance, and the computation of these steps despite using the laptop with the fastest processor within the group merely reduces computation time taken from 3s to 2s. Secondly, some pieces cannot be placed on the center of the square too consistently, because of the accumulating small errors from offsets and other components. Thirdly, there is a slight tremor that cannot be ignored, the main reason is that the robot is not strong enough and the speed control algorithm does not generate a sufficiently smooth trajectory.

5.2 Limitations and Further Improvement

Gear Teeth and Coupler Disc Backlash

Despite efforts in reducing friction to a negligible extent via bearings and strict alignment design, one discovered limitation suffered is the backlash from the gears which could affect the precision of the end-effector. From attempting to rotate the platform manually while the motors were powered to stay in position, it was observed that there were room for some rotations, specifically where the provided motor's coupling disc can wiggle by a small amount with respect to the stationary motor gears. It was suspected that this deterioration was due to damage to the gear teeth from the initial stages of our testing. For instance, when selecting the motor trajectory parameters, the arm was accidentally moved in an unacceptable fast speed and halted to a stop from a relatively large moment, where this torque could easily damage to the motor teeth. As for the other joints' motors that are also facing the same issue, there were some cases where the arms hit the chessboard to a stop due to changes in the height of the end effector or gripper length from their theoretical values. In addition to that, the tolerances in precision of the motor angles were also considered to contribute to justify this undesired behaviour.

In order to solve this, one precaution to prevent this issue is to test the behaviour of all motors without anything attached to the motor. For instance, attaching a tape where one end sticks onto the gear's center and the other end pointing outwards, such that changes in position or velocity can be observed without the presence of damaging torque. If the same behaviour persists after performing the above procedure, this could imply that the motors do not have sufficient precision for this project, an alternative is to purchase more powerful and precise motors, however the trade-off of more expensive costs which may not be necessary.

5.3 Gripper Inaccuracies

During demonstration, it is observed that some chess pieces had fallen off after being picked up as it reached the first via point, as opposed to what was observed in our preparation prior to the demonstration. By performing inspection on the gripper post-demonstration, it is clearly observed that the sponges had been compressed and lost some of its elasticity. This may also be caused to the unevenness when installing the sponge pads using tape on top of losing elasticity, and this was due to the team running out of time towards the deadline.

Easy fixes can be made where the sponges can be replaced with other materials providing high elasticity, compressibility and sufficient friction necessary for the task. For instance, Poron foam or rubber pads can be considered as an alternative material. Additionally the mounting method of the cushioning material can be replaced by using glue, with careful alignment to prevent any unevenness.

6 Conclusion

This report reviewed the the process of implementing motion planning and control methods in order to perform tasks 1. Position control was chosen as the method to navigate the robot's end-effector to the targeted location on the chessboard, pick up the chess piece, transport it to the desired location and place it down. Calibrations involving identifying offsets were vital to compensate for any inconsistencies in the manipulator's motion which may had arisen from and manufacturing process. The mechanical designs of major components, such as the linkage arms, base, platform, shaft and motor installment, use of bearings, and gripper mechanism were explained in detail. During testing, as mentioned in the discussions, uncontrollable backlash were identified due to suspected deterioration of the gears within the motor. Possible improvements have been proposed to prevent currently discovered issues for better performance.