SCARA
MAE C163A/C263A Team Brickbots

Team BrickBots
Tony Zong, Shirley Shah, Matthew Hutchinson,
Andrew Tran, Mihir Baviskar, Blake Dee

December 17, 2023

Contents

1 Introduction

The goal of this project was to design, build and test a robot arm that can build a simple shape out of LEGOs^{TM}. The team went through the entire design process of brainstorming, computing the forward and inverse kinematics and finally assembling and testing the final product.

2 Design

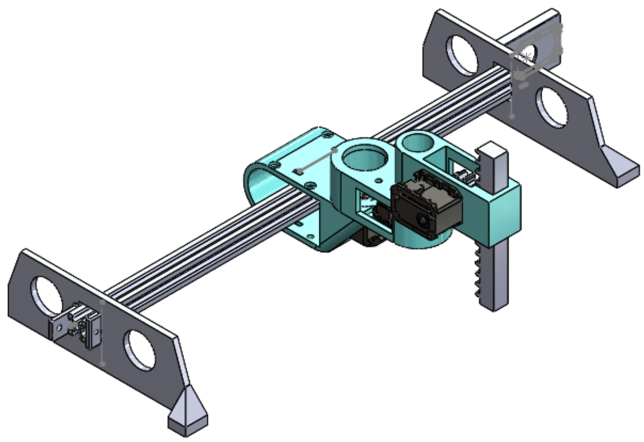


Figure 1: Robotic Arm

2.1 Manipulator Configuration

The hybrid SCARA design implemented in our robotic arm features a combination of SCARA (Selective Compliance Articulated Robot Arm) characteristics and an additional rail for an extra degree of freedom (DOF). The SCARA architecture provides selective compliance along the vertical axis, allowing the arm to be flexible and adapt to different heights while maintaining rigidity in the horizontal plane. The additional rail introduces a linear motion component, enhancing the arm's versatility and expanding its operational workspace. This hybrid design strikes a balance between the benefits of SCARA, such as precision and adaptability, and the advantages of linear motion for extended reach and positional flexibility.

2.2 Design Process

Figure 2 illustrates the design process that the team went through to develop the robotic arm. The team was held back due to numerous 3D printing failures and design oversights. However, once the robot was finally assembled, the team was able to integrate the motor code and test it.

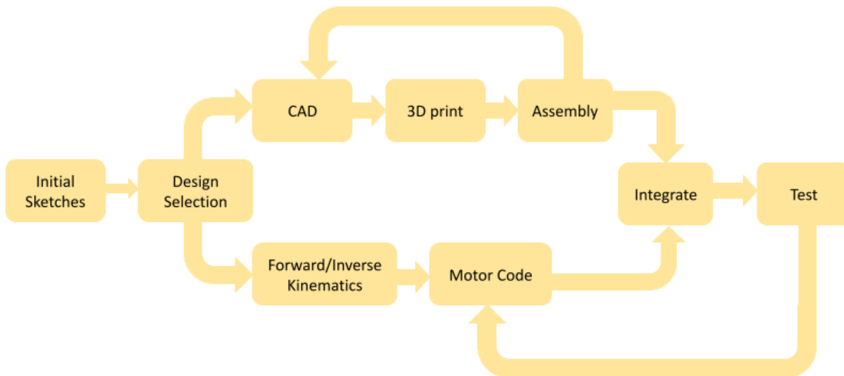


Figure 2: Design Process

2.3 Design Highlights and Difficulties

2.3.1 Link Iterations

Our project initiated with a significant shift from a 6DOF (Degrees of Freedom) to a 4DOF design, primarily to enhance operational efficiency. The reduced degrees of freedom vastly simplified both the inverse kinematics solutions, workspace and task, and mechanical design. Much less 3D printing was required for a simpler, more structurally capable arm with less joint play and drift. The choice of MX28-AR servo motors was due to their proven effectiveness in smaller, more precise mechanical systems. This transition required a careful redesign to ensure that, despite having fewer degrees of freedom, the robot arm could still accomplish the desired tasks effectively.

Initially, in the 6DOF configuration, the links of the robot arm were arranged in an interweaved pattern, allowing for strong parallel mounted links that were supported from both the top and bottom. However, to increase the system's overall height and increase range of motion for each joint, we shifted to a stacked arrangement with each link stepping up to the end effector. Although this increased compliance and play in the system, the design choice for a stacked configuration allowed for a much larger workspace.

2.3.2 Gripper Design

The original gripper design involved a female Lego end effector, but faced with challenges such as a failed 3D print, we adapted by attaching a piece of duct tape to the female Lego. The use of tape proved effective, as it increased the surface area, thereby enhancing grip. To mitigate lateral movement during the downward press, a counterweight was strategically added on the opposite end, ensuring stability and an even application of force on the LEGO.

2.3.3 Leg Iterations

The initial leg height posed issues, as it was too small, causing the prismatic joint to hit the ground. Additionally, the pulley pinion system experienced slipping problems, resulting in inaccuracies in movements and necessitating frequent recalibration of the offset.

2.3.4 Task Design

Task design underwent several modifications to enhance the success rate of the pressing operation. Initially, a simple downward press proved insufficient, leading to inconsistent outcomes and an impractical margin of error. Consequently, we introduced a two-step approach, incorporating an initial firm downward press followed by a lateral jiggle motion to align with the Lego pegs. This adjustment significantly improved the stability and effectiveness of the pressing action.

2.4 Static Force Analysis

Static force analysis was conducted on the LeBAR robotic arm to ensure that the structure and actuators could withstand operational loads.

Our SCARA-like robotic arm experiences forces and moments which are primarily due to gravitational effects on the arm's links and the actuators' reaction forces. The arm's end effector is designed to handle a maximum payload of M kg, which dictates the forces experienced by the manipulator. Equilibrium Equations We have employed the principles of static equilibrium to derive the necessary equations.Mathematically, these conditions are expressed as:

$$\sum F_x = 0, \quad \sum F_y = 0, \quad \sum M_z = 0$$

Table 1: Calculated Reaction Moments at the Joints

Joint	Reaction Moment $\mathrm{N} \cdot \mathrm{m}$
N_3	0.098
N_2	0.241
N_1	0.455
N_0	0.49

Solidworks was used to model material mass and center of mass. Our 20% infill PLA was assumed to have a density of $250\,\mathrm{kg/m}^3$.

The reaction moment N_3 corresponds to the interaction between the final prismatic joint and the third link. N_2 reflects the moment at the first revolute joint, taking into account the combination of the two subsequent links and the prismatic end effector. N_1 represents the moment between the rail and the rest of the arm, indicating the load transferred through the arm's base to the rail system. Finally, N_0 is the moment between the robotic arm as a whole and the ground. Because MX-28AR servo motors are rated for a radial load of $0.3\,\mathrm{N} \cdot \mathrm{m}$[1], the load N_2 reduces our factor of safety to an insufficient $FOS = 1.24$. As such, each link was widened to distribute the load on the next link, instead of on the servo horn and shaft.

The implementation of horizontal revolute joints indicates that the corresponding servo motors only need sufficient torque to overcome the friction between links. Referring to Appendix C, a quick calculation indicates to us that the frictional moment at the link interfaces is negligible. The coefficient of friction of PLA was obtained online[2].

The rail itself was assumed to frictionless. The end effector experienced similarly negligible fricitonal forces. Our target force of 4.8 N [3] online is achieved easily as at a radius of 0.02m and stall torque of 2.5 N $\cdot$ m allows for 125 N of force.

2.5 Derivation of Kinematics

This section describes the derivation of forward and inverse kinematics for the two designs that were considered.

2.5.1 4DOF Configuration

Table 2: Denavit-Hartenberg Parameters

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	0
2	90°	A_1	D_2	θ_2
3	0	A_2	D_3	θ_3
4	0	A_3	0	0
f	180°	0	D_f	0

The DH parameters of the 4DOF system are summarized in Table 2. The frame assignment is seen in Figure 3 Using forward kinematics, we get the following solution

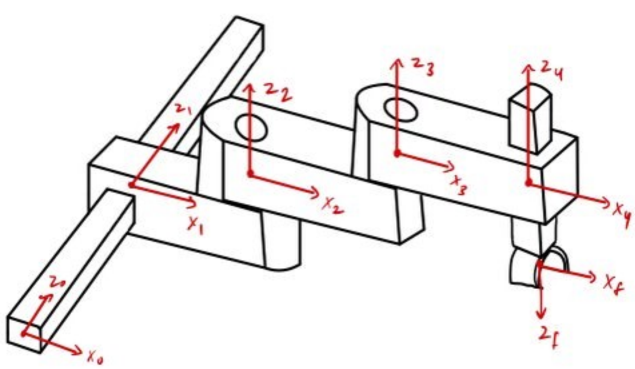


Figure 3: 4DOF Frame Assignment

$$T_0^5 = \begin{bmatrix} \cos(t_2+t_3) & \sin(t_2+t_3) & 0 & A_1 + A_3\,\cos(t_2+t_3) + A_2\,\cos(t_2) \\ 0 & 0 & 1 & -D_3 \\ \sin(t_2+t_3) & -\cos(t_2+t_3) & 0 & d_1 + A_3\,\sin(t_2+t_3) + A_2\,\sin(t_2) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The inverse kinematics was first solved using algebraically, with the resulting equations shown in Eqn. 1 to Eqn. 3. This was done by taking the inverse of each individual transformation matrix and multiplying it to the goal matrix . This process is illustrated here:

$$T_{Goal} = T_0^5 = T_0^1 T_1^2 T_2^3 T_3^4 T_4^5$$

$$T_{Goal} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \text{px} \\ r_{21} & r_{22} & r_{23} & \text{py} \\ r_{31} & r_{32} & r_{33} & \text{pz} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[T_0^1]^{-1} T_{Goal} = [T_0^1]^{-1} T_0^1 T_1^2 T_2^3 T_3^4 T_4^5$$

$$[T_0^1]^{-1}[T_1^2]^{-1} T_{Goal} = [T_0^1]^{-1}[T_1^2]^{-1} T_0^1 T_1^2 T_2^3 T_3^4 T_4^5$$

$$etc...$$

$$\text{atan2}(r_{31}, \sqrt{1 - r_{31}^2}) = b$$

$$\text{atan2}(r_{31} - \cos(b), sin(b) - r_{11}) = \theta_2 \tag{1}$$

$$\text{atan2}(r_{31}, \sqrt{1 - r_{31}^2}) - \theta_2 = \theta_3 \tag{2}$$

$$A_1 \tan(\theta_2) + \text{pz} - \text{px} \tan(\theta_2) - A_3 \frac{\sin(\theta_3)}{\cos(\theta_2)} = d_1 \tag{3}$$

However, we found that these equations resulted in many solutions, making it difficult to solve for a specific position. From this point, the team decided to take a more geometric approach. Since LEGO's must be attached at at 90° rotations, we determined that the end effector must be in one of 3 orientations: A_3 must be parallel to x or parallel to z (two possible orientations), so that the LEGO is aligned with the base grid. From here it's easy to dissect things case by case and solve the inverse kinematics. In case 1, where A_3 must be parallel to x. By our design, this sets a constraint $\theta_2 = -\theta_3$. This makes it so that the min P_x this specific orientation can reach is $A_1 + A_3$, but makes the inverse kinematics extremely easy to solve with simple geometry. It is also important to note, that these solutions only care about the z and x values of the end effector. This takes out the effect of $d4$, which is offset created by the last prismatic. That value ended up being hard-coded since the height of the LEGO remains constant.:

$$\cos(\theta_3) = \frac{(P_x - A_3 - A_1)}{A_2} \quad sin(\theta_3) = \sqrt{1 - \cos(\theta_3)^2}$$

$$\theta_3 = atan2(\sin(\theta_3), \cos(\theta_3))$$

$$\theta_2 = -\theta_3$$

$$d_1 = P_z - A_2 \cdot \sin(\theta_3)$$

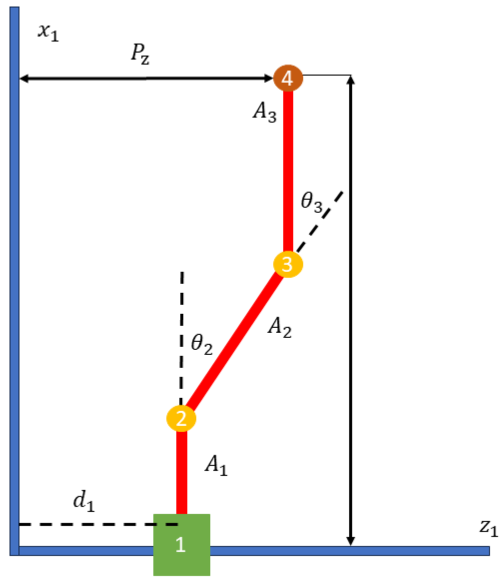


Figure 4: $P_x > A_1 + A_3$

Figure 5 shows the two cases where the end effector is parallel to the Z axis. These are similarly trivial to solve since they constrain the angles to $\theta_2+\theta_3 = \pm 90°$. These orientations allow us to reach P_x less than $A_1 + A_3$.

$$cos(\theta_2) = \frac{(P_x - A_1)}{A_2} \;\; sin(\theta_2) = \sqrt{1 - cos(\theta_2)^2}$$

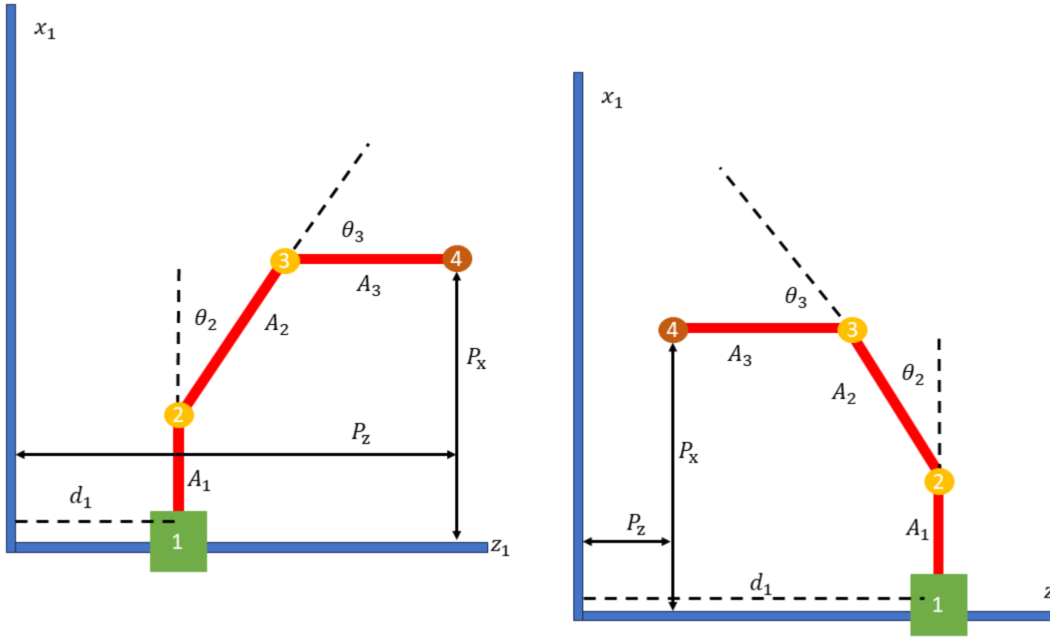
$$\theta_2 = atan2(sin(\theta_2), cos(\theta_2)) \;\; \theta_3 = 90° - \theta_2$$

$$d_1 = P_z - A_3 - A_2 \cdot sin(\theta_2)$$

$$cos(\theta_2) = \frac{(P_x - A_1)}{A_2} \;\; sin(\theta_2) = \sqrt{1 - cos(\theta_2)^2}$$

$$\theta_2 = -atan2(sin(\theta_2), cos(\theta_2)) \;\; \theta_3 = \theta_2 - 90°$$

$$d_1 = P_z + A_3 + A_2 \cdot sin(\theta_2)$$



(a) $P_x \leq A_1 + A_3$ and $P_z > \frac{Z_{max}}{2}$ (b) $P_x \leq A_1 + A_3$ and $P_z \leq \frac{Z_{max}}{2}$

Figure 5: Parallel to Z-axis case

2.5.2 6DOF Configuration

The DH parameters for the 6DOF configuration are seen in Table 3

Table 3: DH parameters for 6 DOF configuration

i	a_{i-1}	α_{i-1}	θ_i	d_i
1	0	0	0	d_i
2	0	$-\pi/2$	θ_2	0
3	0	0	$-\pi/2$	d_3
4	A_4	0	θ_4	D4
5	A_5	0	θ_5	D5
6	0	π	0	d_6

Using these parameters, the forward and inverse kinematics are as follows

$$T_0^6 = \begin{bmatrix} s_{245} & -c_{245} & 0 & A5s_{24} + A4s_2 + A6s_{245} \\ 0 & -1 & 0 & D4 + D5 - D6d3 \\ 0 & 0 & -1 & d1 + A5c_{24} + A4c_2 + A6c_{245} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$b = \text{atan2}(\left(r_{11}, \sqrt{(1 - r_{11}^2)} \right)) = \theta_2 + \theta_4 + \theta_5$$

$$\text{atan2}((c_b - r_{31}, r_{11} - s_b)) = \theta_2$$

$$\text{atan2}((r_{31}c_b + r_{11}s_b - 1, r_{11}c_b - r_{31}s_b)) = \theta_5$$

$$\text{atan2}\left(r_{11}, \sqrt{(1 - r_{11}^2)}\right) - \theta_2 - \theta_5 = \theta_4$$

$$p_y - D4 - D5 + D6 = d3$$

$$\frac{A6s_5}{s_2 4} - \frac{p_x}{t_{24}} - \frac{A4s_4}{s_{24}} + p_z t_{24} = d1$$

2.5.3 Numerical Derivation

In parallel with the analytical derivation of inverse kinematics, we used the Matlab Robotics Toolbox to numerically solve for our desired end effector positions. This redundancy helped ensure a kinematic solution in the event that our analytical solution failed, which seemed probable in the early, 6 DOF stages of the LEGO robot.

In order to accomplish this, we started by downloading the ROS-Solidworks add-on which allowed us to export a URDF file of the full robot assembly. This URDF file contained the relevant DH parameter information directly from the CAD, as well as joint origin and types. We then created a rigid body tree model in Matlab using the URDF file, which we could then call the IK solver onto. To test the solver and model, we generated a struct containing random configurations of the relevant joint angles, forward kinematic ally solved to the 4x4 transformation matrix, and ran the solver on that matrix to see if we accomplished the initial joint configurations. The random configuration as well as IK solution - successfully achieved - are shown below. Finally, we uploaded the solved joint angles to the servos and configured our robot in the random configuration, successfully.

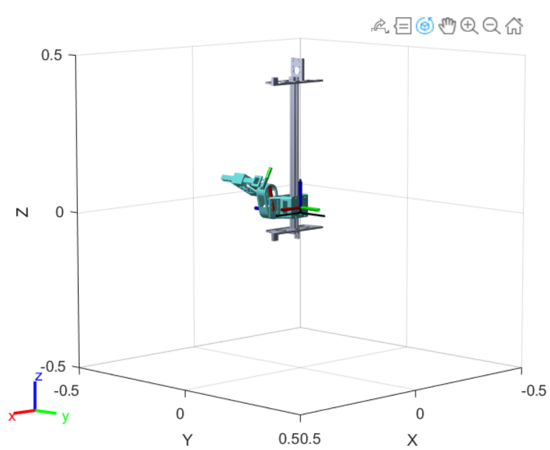


Figure 6: Depicts the Matlab inverse kinematic solver solution, which mirrored the original randomly generated configuration.

Although the inverse kinematic solver successfully solved the robotic configuration, we ultimately decided to implement the analytical solution for the LEGO pick-and-place task as it allowed us to quickly iterate the end effector position to match the error in the robot arm play and LEGO placement. The next step would be to use the numerical solver in the live robot arm and compare the two methods.

2.6 Singularity Discussion

One potential singularity in our design arises from the fact that Link 3 is longer than Link 1. Specifically, when both θ_2 and θ_3 are set to 90 degrees or -90 degrees, the end effector could potentially collide with the prismatic joint. To mitigate this issue, we have implemented a proactive approach in our control logic.

Prior to executing commands involving joint movements, our system incorporates a check on the positional parameter P_x. If P_x is negative or in close proximity to zero, we refrain from executing the corresponding command. Instead, we opt to output an "invalid x value" message. This preemptive measure ensures that the end effector does not inadvertently collide with the prismatic rail, safeguarding the integrity and functionality of the robotic system.

2.7 Workspace Analysis

The workspace of the final configuration is shown in the Figure 7. The area shaded in the darker green is the full theoretical workspace. However, it is observed that the since the last link is longer then the first link, it will coincide with the the prismatic joint. Thus the workspace was limited so that when $\theta_2 = \pm 90$, θ_3 can from 0 to 180 degrees as opposed to -90 to 90.

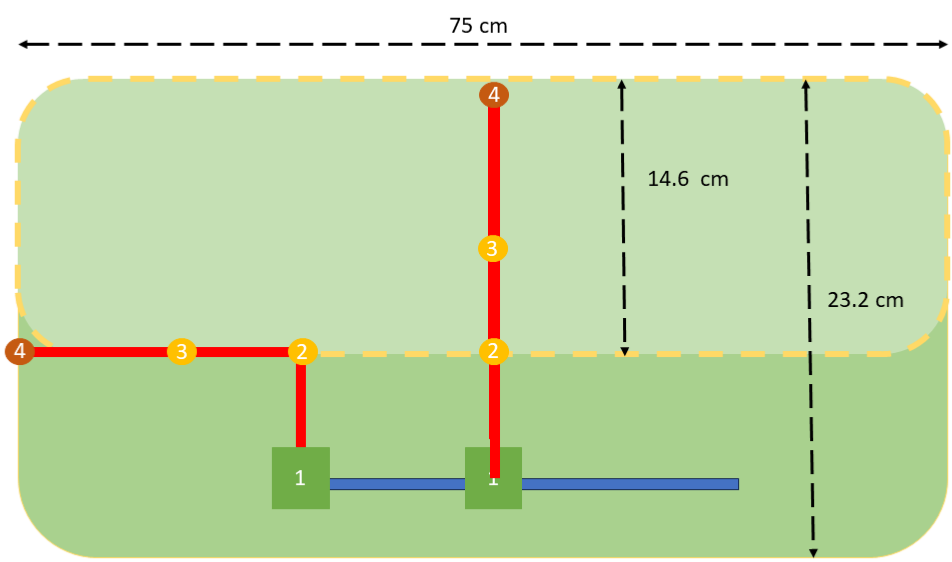


Figure 7: Workspace Analysis: Theoretical vs Actual

2.8 Task Simulation

A simulation of the 4-DOF configuration was done using Matlab. Using the constrained geometric kinematics equations derived in Section 2.5.1 was used. The simulation was set to perform a simple task. It the arms would start fully extended at its home position. Home position is defined to be $z = 0$ with respect to the base frame and the $x = A_1 + A2 + A3$. The end gripper prismatic would move up, pick up the LEGO, and then move back up. Then the arm would start moving to the inputted x and z. After the desired z and x position is achieved, the end prismatic is actuated again to place the LEGO on top of the other LEGO. The robot would then move back to home position to pick up the next LEGO.

2.9 Performance Analysis

In comparing HadrianX and the LEGO Pick-and-Place Robot, notable differences emerge in their structures, mechanisms, and applications. HadrianX, a sophisticated bricklaying robot, boasts a 32-meter telescoping boom arm and a versatile multi-degree of freedom system tailored for swift construction with large masonry blocks. Its specialized end effector prioritizes efficiency over precision in large-scale tasks. Conversely, the LEGO Pick-and-Place Robot, with a 4-degree of freedom manipulator, emphasizes precision for handling individual LEGO bricks with extreme accuracy during picking and placing maneuvers. Compliance is introduced through a jiggle maneuver in the software to handle slight errors during placement. HadrianX excels in speed, laying up to 300 masonry blocks per hour, ideal for large-scale projects, while the LEGO robot focuses on accuracy, operating at a slower pace to ensure precise LEGO manipulation. In summary, HadrianX prioritizes speed and efficiency for large-scale construction, while the LEGO robot emphasizes precision and accuracy, showcasing the adaptability of robotics in addressing diverse industry needs.

2.10 Constructive Comments

In reflection, our project encountered substantial challenges due to our ambitious design. The precision required in LEGO manipulation, especially concerning the end gripper, resulted in significant setbacks and a prolonged development timeline. Valuable mechanical insights and future steps include the recommendation for a Mechanical FEA analysis to refine the arm's structure, a shift to a lead screw mechanism for enhanced precision, transforming the end effector into a compliant gripper, and the repositioning of the first revolute for improved gantry access. On the software side, a more modular and functional approach would be more user-friendly and adaptable. In essence, an iterative design process, coupled with mechanical enhancements and software streamlining, could significantly elevate the project's efficiency and success in future iterations.

3 Conclusion

In our engineering journey with the BrickBot, we went through different phases to refine our design. We started with a 6-DOF setup but ended up streamlining it to a more efficient 4-DOF system.

When dealing with kinematics, we used analytical, geometric, and numerical methods to tackle the challenges of complex robotic configurations. To avoid potential issues like singularities, we implemented checks on positional parameters in real-time, ensuring the stability of the system. Adding compliance in the software, driven by the need for precise LEGO placement, demonstrated our ability to adapt to unexpected challenges and optimize the system.

In the end, we successfully stacked one brick on top of another using our kinematic equations. It's a significant accomplishment that highlights not just a project's storyline but the practical execution of an iterative engineering process. Our success lies in creating a functional and efficient robotic system through adaptive design and effective problem-solving.

A Forward Kinematics Intermediary Matrices

A.1 FK for 4 DOF

$$T_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = \begin{bmatrix} \cos(t_2) & \sin(t_2) & 0 & A1 \\ 0 & 0 & -1 & 0 \\ \sin(t_2) & \cos(t_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} \cos(t_3) & -\sin(t_3) & 0 & A2 \\ \sin(t_3) & \cos(t_3) & 0 & 1 \\ 0 & 0 & 1 & D3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & A3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A.2 FK for 6 DOF

$$T_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} c_4 & -s_4 & 0 & A4 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & D4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^5 = \begin{bmatrix} c_5 & -s_5 & 0 & A5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & D5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^6 = \begin{bmatrix} 1 & 0 & 0 & A6 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -D6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B Inverse Kinematics Derivation of 4DOF config

Using Matlab and the forward kinematics soltution we get:

$$\mathrm{atan2}(r_{31}, \sqrt{1 - r_{31}^2}) = \theta_2 + \theta_3$$

We can manipulate this equation to be:

$$\mathrm{atan2}(r_{31}, \sqrt{1 - r_{31}^2}) - \theta_2 = \theta_3$$

In addition, for the sake of clarity:

$$\mathrm{atan2}(r_{31}, \sqrt{1 - r_{31}^2}) = b$$

Solving this algebraically

$$r_{11}\ \cos(\theta_2) + r_{31}\ \sin(\theta_2) = \cos(\theta_3)$$

$$r_{11}\ \cos(\theta_2) + r_{31}\ \sin(\theta_2) = \cos(b - \theta_2)$$

$$r_{11}\ \cos(\theta_2) + r_{31}\ \sin(\theta_2) = \cos(b)\cos(\theta_2) + \sin(b)\sin(\theta_2)$$

Divide by $\cos(\theta_2)$

$$r_{11}\tan(\theta_2) + r_{31}\ \ = \cos(b) + \sin(b)\tan(\theta_2)$$

Bring the $\tan(\theta_2)$ values to the other side

$$r_{31} - \cos(b) = \sin(b)\tan(\theta_2) - r_{11}\tan(\theta_2)$$

$$\frac{r_{31} - \cos(b)}{sin(b) - r_{11}} = \tan(\theta_2)$$

Finally,

$$\mathrm{atan2}(r_{31} - \cos(b), sin(b) - r_{11}) = \theta_2$$

From this we can determine θ_3:

$$\mathrm{atan2}(r_{31}, \sqrt{1 - r_{31}^2}) - \theta_2 = \theta_3$$

Finally, we need to determine d_1

$$\mathrm{px}\ \cos(\theta_2) - A_1\ \cos(\theta_2) - d_1\ \sin(\theta_2) + \mathrm{pz}\ \sin(\theta_2) = A3\sin(\theta_3)$$

$$A_1\tan(\theta_2) + \mathrm{pz} - \mathrm{px}\tan(\theta_2) - A_3\frac{\sin(\theta_3)}{\cos(\theta_2)} = d_1$$

C Static Force Calculations

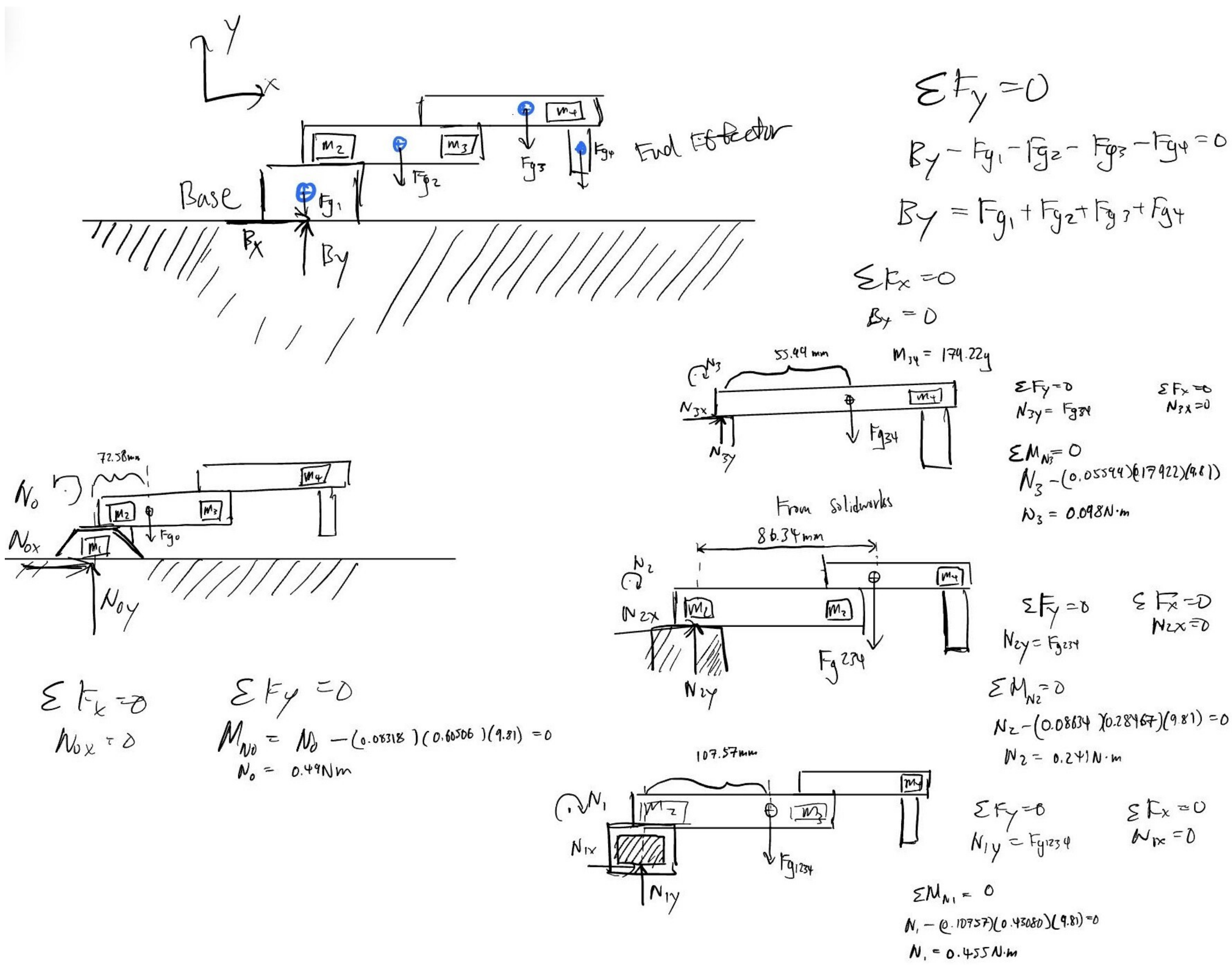


Figure 8: Static Force Analysis Calculations

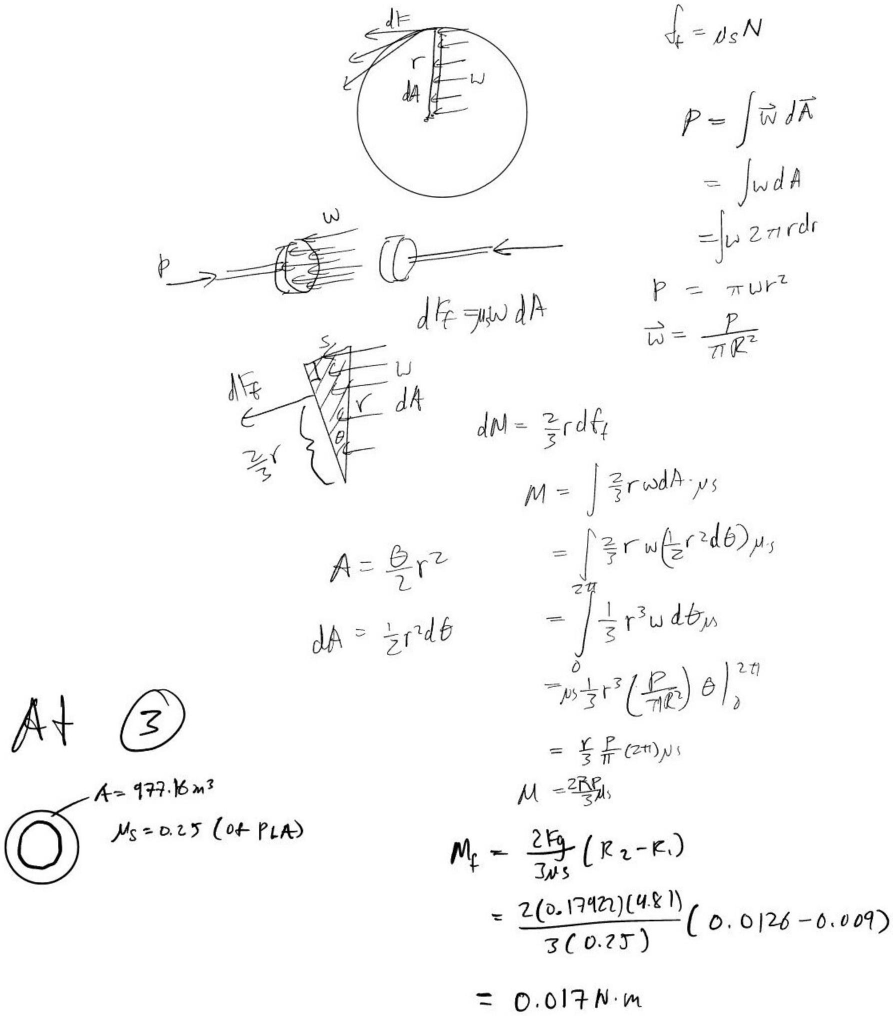


Figure 9: Static Force Analysis Calculations

D References

[1] https://www.matweb.com/search
/DataSheet.aspx?MatGUID=ab96a4c0655c4018a8785ac4031b9278ckck=1
[2]https://emanual.robotis.com/docs/en/dxl/mx/mx-28/
[3]https://josswb.medium.com/lego-lessons-clutch-power-3f63468d702a