- CLASSY COLONOSCOPY -

[1] Supporting Calculations

The aim is to find angles θ_0 and θ_1 throughout the given positions of the end-effector (x,y).

For finding the angle of θ_1, the first step is finding the length of the straight-line distance from the base joint A to the end-effector C (x, y):

$$z = AC = \sqrt{x^2 + y^2},$$

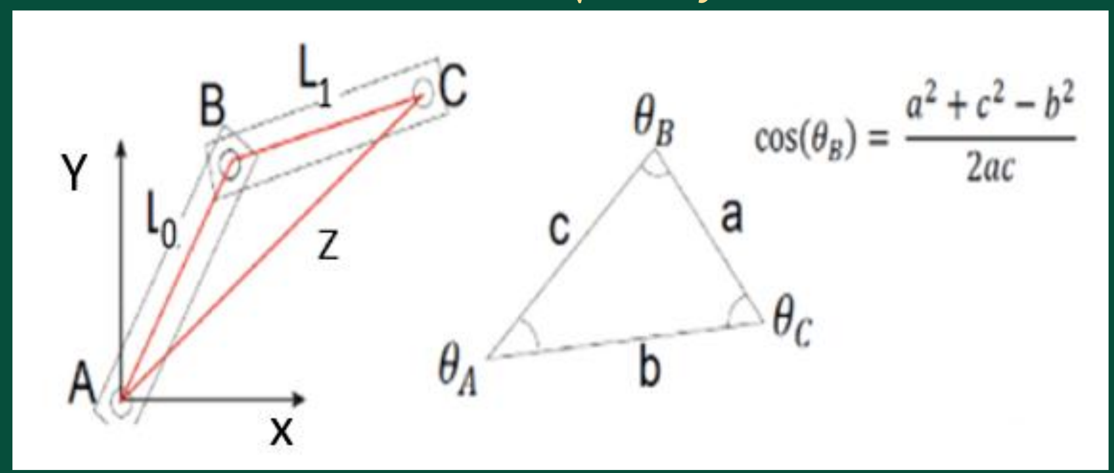


Figure 1: Diagram of 2-link robot arm and Triangle model used for applying the cosine theorem [1]

The cosine theorem required is as follows:

$$\cos\theta_b = \frac{a^2 + c^2 - b^2}{2ac} = \frac{{L_1}^2 + {L_0}^2 - z^2}{2L_1L_0}$$

Since θ_1 and θ_b are supplementary angles, their sum is 180 degrees so:

$\theta_1 = \pi - \theta_b$, and $\cos\theta_1 = \cos(\pi - \theta_b) = -\cos\theta_b$

Thus, the formula for $\cos\theta_1$ can be derived:

$$\cos\theta_1 = -\cos\theta_b = \frac{z^2 - {L_1}^2 - {L_0}^2}{2L_1L_0}$$

Finally, the angle of θ_1

$$\theta_1 = cos^{-1}\left(\frac{b^2 - {L_1}^2 - {L_0}^2}{2L_1L_0}\right)$$

Since the robot's movement has two modes: elbow-down and elbow-up, this results in two possible solutions for the final calculated angle, namely positive and negative.

For finding the angle of θ_0, the first step is considering the position of the end-effector C (x, y) relative to the base point A. The direction from the base to the target point which represents the absolute orientation of the vector AC can be written as:

$$\theta = atan2(y, x)$$

The position of the end-effector can be expressed using vector addition:

$$\vec{AC} = \vec{AB} + \vec{BC} = L_0 + L_1$$

In a coordinate frame aligned with link AB, the components of vector AC are:

$$V_1 = L_0 + L_1\cos\theta_1,$$
$$V_2 = L_1\sin\theta_1,$$

Where the V_1 and V_2 are the vectors responding in the horizontal direction and vertical direction.

The direction of AC is equal to the base rotation θ_0 plus the internal angle of the vector.

$$\theta = \theta_0 + atan2(V_2, V_1) \Rightarrow \theta_0 = \theta - atan2(V_2, V_1)$$

Finally, substituting the formula of the end-effector C (x, y):

$$\theta_0 = atan2(y, x) - atan2(L_1\sin\theta_1, L_0 + L_1\cos\theta_1)$$

Using these equations, we calculated the joint angles for desired positions as seen below for the bottom right corner:

Note: The lengths of the robot arm are 250cm for L_0 and 200cm for L_1 and we went for an elbow up design for our robot arm.

$$\theta_1 = cos^{-1}\left(\frac{310^2 + 60^2 - 200^2 - 250^2}{2 \times 250 \times 200}\right) = 91.6°$$
$$\theta_0 = a\tan2(60,\ 310) \ldots$$
$$-a\tan2(200 \times sin(91.6),\ 250 + 200 \times cos^{-1}(91.6))$$
$$\theta_0 = -28.3°$$

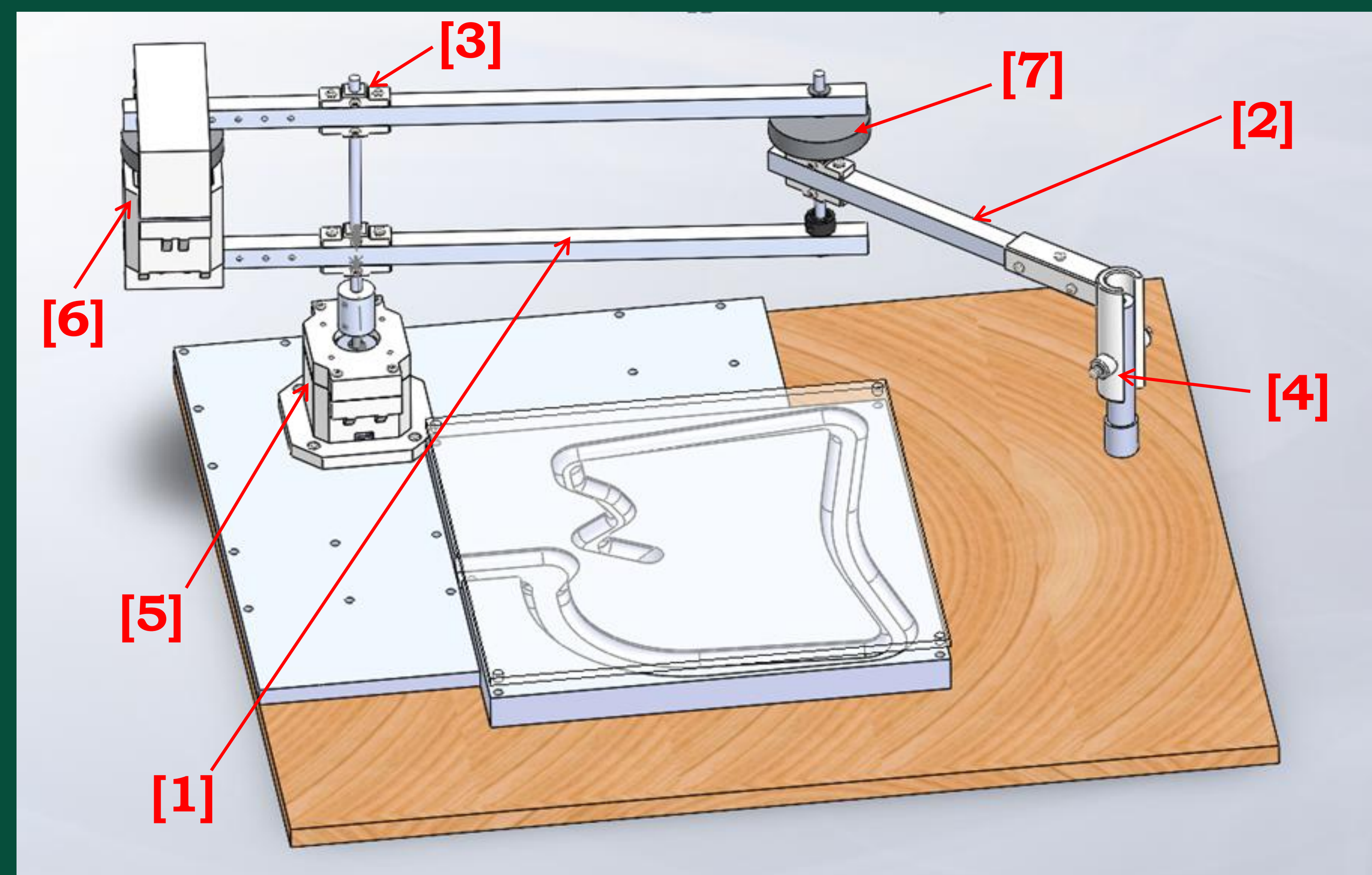
Please see Table 1 for joint angles for all four corners.

SCAN ME:
Website Containing Additional Information

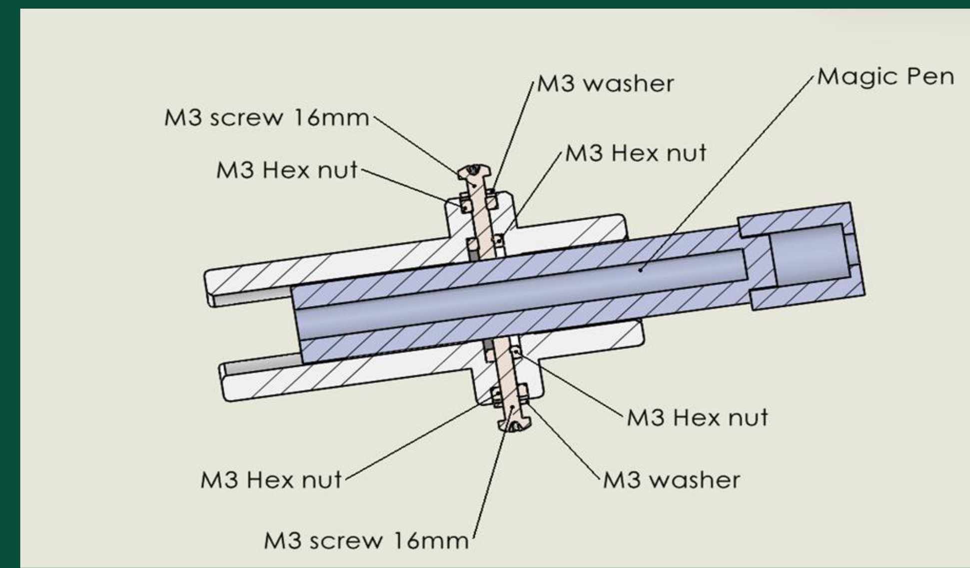
Position of corners, Origin is bottom left corner of workpiece	Position of corners, Origin is the centre of our motor position	θ_1	θ_0
(250, 0)	(310, 60)	91.6°	−28.3°
(250, 250)	(310, 310)	26.2°	33.4°
(0, 250)	(60, 310)	91.6°	39.8°
(0, 0)	(60, 60)	162.3°	−0.6°

Table 1: Joint angles for each corner of workpiece

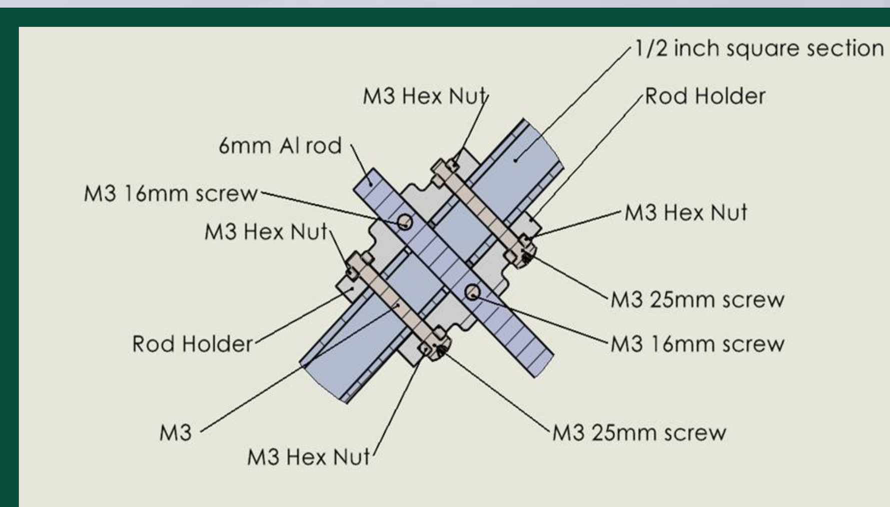
[2] Design Features

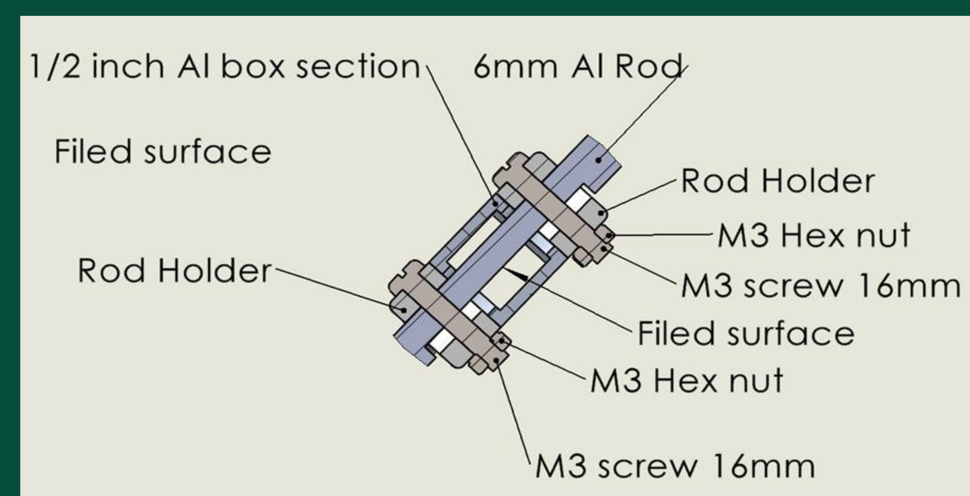


[1]	Arm 1
[2]	Arm 2
[3]	Rod Holder
[4]	Pen Gripper
[5]	Stepper Motor 1
[6]	Stepper Motor 2
[7]	Pulley System

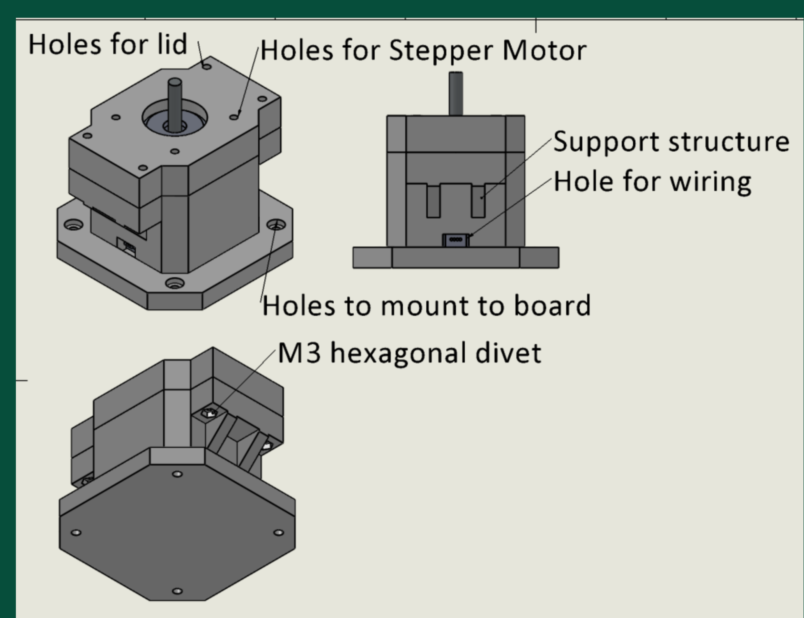


The pen gripper uses the friction of two screws held in place by two hex nuts each within a 3D printed "Lego hand" housing to hold the pen in place. The two hex nuts prevent the screws from moving whilst providing a thread to tighten against the pen.



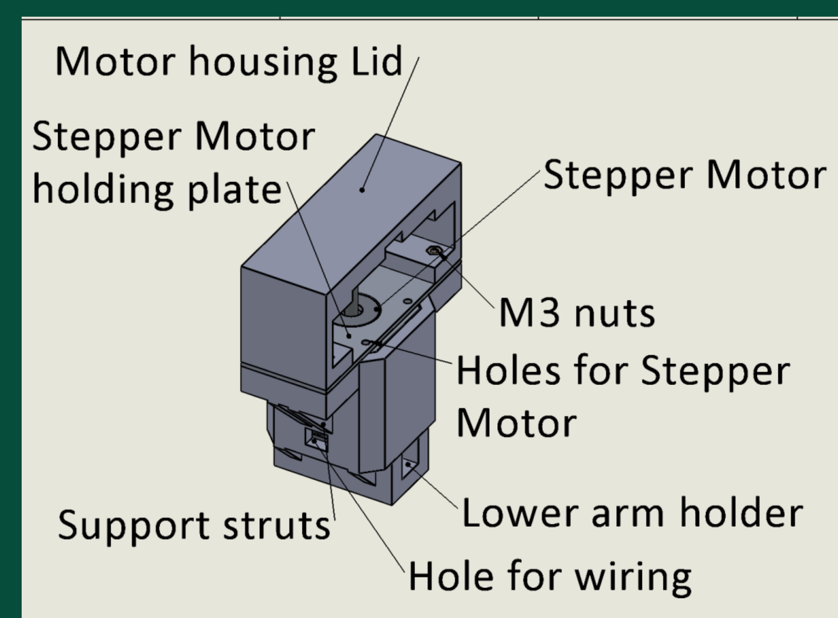


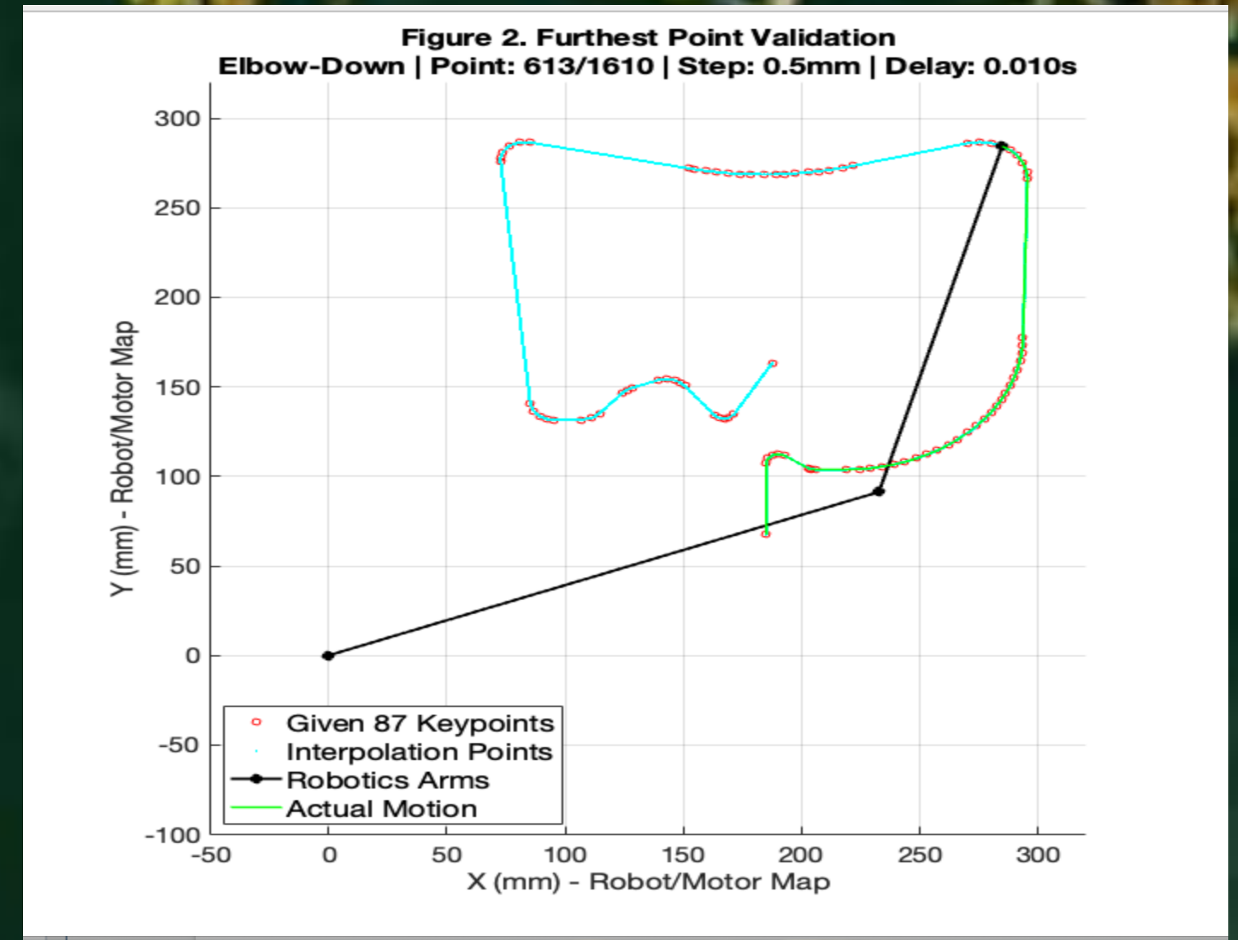
The purpose of the rod holders is to connect the 6mm Al rod to the ½ inch Al box section, so the components do not rotate relative to each other. To do this the two rod holders will be mounted to the ½ inch square sections using two screws and a hex nut. Following this two other screws will be threaded through the rod holder and the 6mm Al rod to reduce play and increase precision.



This is the 1st motor housing, to the left, that encompasses the stepper motor that drives the 1st arm. In a similar style of design to the 2nd housing, there are hexagonal cut outs for M3 nuts to slot into for easy and secure assembly, a cut out for wiring, holes to secure the motor rigidly to the housing and supporting struts. These supporting struts for 1st motor housing are to ensure that the inertial forces from the whole arm assembly do not cause the housing lid to shear from the Motor casing. Both motor housings have been designed with efficient assembly in mind therefore using a modular approach to each housing allows simultaneous production.

This is the 2nd motor housing, to the right, that holds the Stepper motor that drives the second arm. The Lower arm holder slides across the lower arm and bolts on, to ensure a secure motor. If this timing belt stretches over time due to continuous use, the model is designed to allow for simple adjustment to extend distance, to ensure the timing belt is always taught and never skips a tooth. The Stepper motor hold plate adds extra support to the stepper motor, this hold plate ensures that even the smallest rotation within the housing doesn't occur so all parts of rotation from the motor is transferred to the 2nd arm, ensuring exact and precise movement.





[3] Interpolation Of Points & Reachability

The MATLAB algorithm computes the Euclidean distance $dist = \sqrt{\Delta x^2 + \Delta y^2}$ between each pair of the 87 given points. It dynamically assigns a resampling density $N = \lceil dis\,t/\Delta s \rceil$ to maintain a uniform spatial resolution across the entire trajectory. This transforms sparse input data into a dense array of approximately 1600 coordinates (when $\Delta s = 0.5mm$), ensuring the end-effector adheres strictly to straight-line paths between critical "corners" of the colon map.

The algorithm also performs a Reachability Check using the triangle inequality. It validates every newly generated point to ensure it lies within the physical working area defined by the arm lengths L_0 and L_1. Any point that falls outside this boundary is automatically tagged as "Unreachable Point" to maintain safety. The Figure above shows the Reachability and Flexibility of our design at Furthest Points.

(See Video and Code in QR code)

[4] Chosen Motors

The usage of 1/8th micro-stepping represents a balance between the high mechanical torque of 1/4 stepping and the smoothness of 1/16th stepping. While 1/4 stepping offers a 38.3% incremental torque, it often suffers from audible "cogging" and resonance that can cause the arm's end effector to oscillate at low speeds. On the other hand, a 1/16th stepping provides near-silent motion, however its incremental torque drops to 9.8%, which is often insufficient to overcome the motor's internal magnetic detent torque and or static friction of the arm's joints. By choosing a 1/8th stepping, the system retains a 19.5% incremental torque (so the motor reliably moves the rotor when commanded) while providing a theoretical resolution of per pulse of 0.225°.

Team 4: Ebony Cope, Imogen Dolan, Harry Marsden, Yecheng Wang, Ali Waseem, Andy Wills, Jiaoxiao Wu