UNIVERSITY OF Waterloo

Department of Mechanical and Mechatronics Engineering

MTE 119: Statics

Project 2 Report: Design of a Robotic Manipulator With Final Torque of 32.87 Nm

A Report Prepared For:
Prof. Naveen Chandrashekar, PhD, PEng
Department of Mechanical and Mechatronics Engineering,
The University of Waterloo
200 University Ave. W.,
Waterloo, ON, N2L 3G1

Prepared By:
Group 8
Joey Maillette (21007958)
Karthigan Uthayan (21003984)
Camron Sabahi-Pourkashani (21009799)

April 10, 2023

Table of Contents

1. Introduction

Optimization is not only necessary in engineering, but crucial to many different fields and aspects of various sectors across the world. In mechatronics engineering it matters not what niche one associates themselves with (software, electrical, mechanical, or other), the end goal will always in some shape or form relate back to optimization. In interviews for example, many frequently asked questions are along the lines of reducing algorithm runtime, cost in a mechanical design, minimizing power loss, or one of the other thousand applications of optimization in engineering. The constant idea of optimization is what brings such important value to this project especially as a mechatronics engineer.

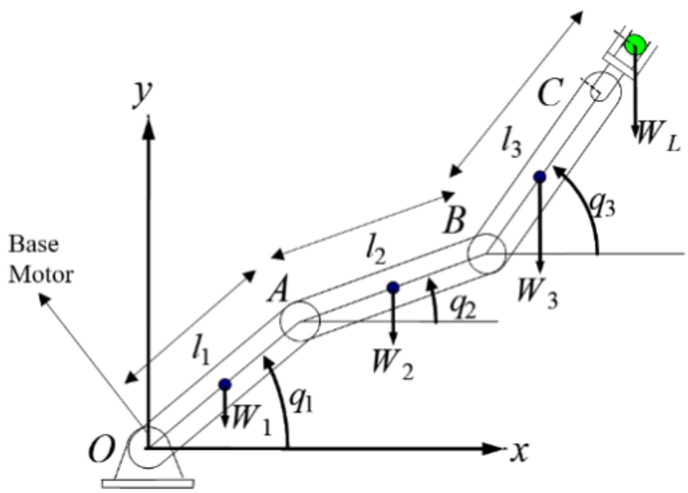

Figure 1: 3-DOF Robotic Manipulator Schematic

This project itself is created to find the most effective lengths of the arms of a three degree of freedom (3-DOF) arm (shown in Figure 1 above) with the given constraints. Effectiveness in this context means that the lengths of these arms should minimize the torque as much as possible for each of the 3 positions provided.

In the real world, many of the tools and techniques used in this project can be extended to future workforce problems, not in just the optimization sector, but to problem solving in general, as this amplifies said core skills. This project overall allowed for a chance to work around constraints to create an optimized designed that gave both a deeper intuition in problem solving and various aspects of the design cycle.

1.1.Objective

The objective of this report is to discuss the results of the groups findings when attempting to minimize the torque of the 3-DOF arm as much as possible. The report additionally considers 3 design iterations, detailing values for torques and respective positions of all points for such iterations.

2. Design Methodology

Based on the constraints presented in the project, as well as a practically infinite combinations of lengths and angles, the best course of action decided was to develop an algorithm which takes in the lengths of each arm and optimizes for the lowest torque within the constraints. Another crucial assumption made was that there is **no loading space required** for the gripper (i.e., an arm could be infinitesimally close to the gripper)**.** For the source code of the algorithm, refer to Appendix A.

Due to the complexity of the torque calculation function, using the SciPy minimize library [1] proved to be useful. Utilizing the “trust-constr” method of optimization removed the need for derivatives such as Jacobian and Hessian matrices.

Using the point of the gripper, angle q_3, length l_3, point B was calculated using basic trigonometric algebra. Point A was determined by finding the intersection of two circles centered at the origin and point B with the radii, l_1 and l_2 respectively [2]. However, majority of times there will be two points of intersections between two circles, which is why the algorithm considers the two points and chooses the point which is creates the least amount of torque while within the constraints.

After all the points are calculated, the function will continue with the torque calculations, which include calculating the mass of each arm based on the given mass per length values as well as calculating the midpoints of each arm since the gravitational force will act upon the centroid which is in the middle of rectangles.

Due to the constraints that had to be put for the SciPy optimization to abide by the stated criteria, a supplementary function was developed to fine-tune the length values initially supplied. This auxiliary function utilizes the NumPy random sampling library [3] to optimize the lengths by testing a range of values within a specified range away from the initial values using the uniform distribution function. The range tested is within ± 0.02 of each length value, and the optimization process is repeated for a specific number of iterations to find the optimal length values. Through this approach, the optimal torque values are determined by identifying the point at which the length values converge.

Overall, this methodology allows for a more precise and effective optimization process while adhering to the given constraints.

Design Iteration 1

This iteration was made on the basis that the lengths should be as close to 1m as possible, since (at this point) it was proposed that the weight of the beams to be minimized, and since the weight of these beams are depended on the length, the length should be minimized as a result. Hence, the total lengths were made 1.1m. The torque for these lengths resulted in the following set of data.

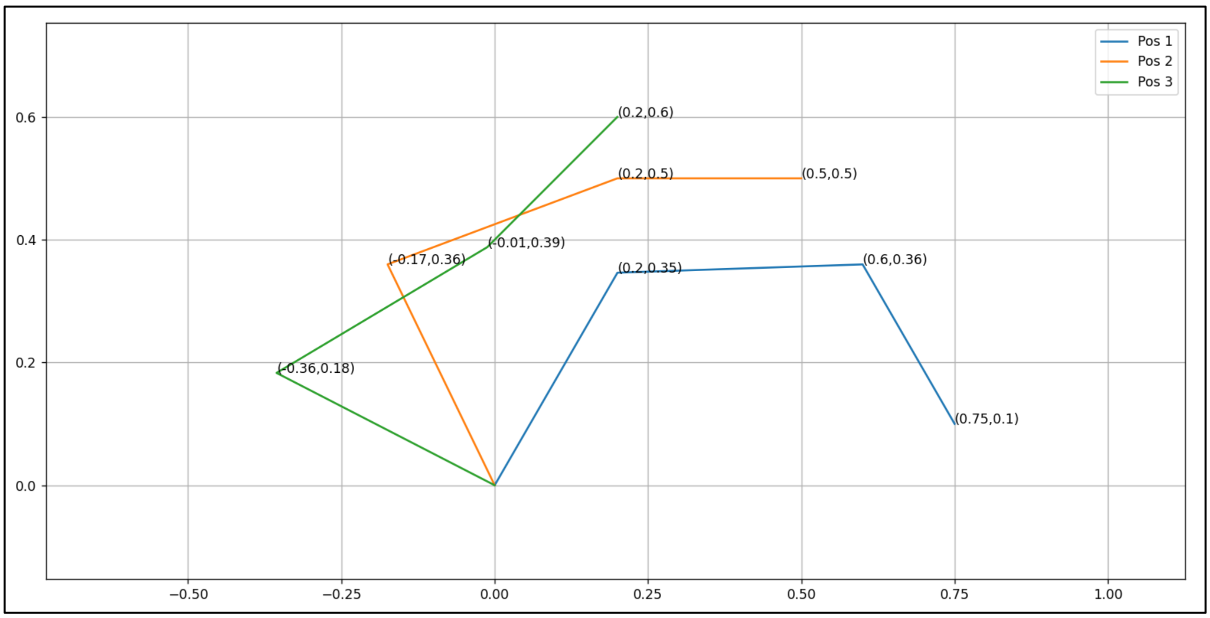

Figure 2: Design 1 Position Diagram

This design has lengths of 0.4m, 0.4m, 0.3m (Total Length = 1.1m) and a total calculated torque of 50.15 Nm.

Table 1: Specifications for each position (Design 1)

Position Number	Torque required (Nm)	Joint #1 (x, y)	Joint #2 (x, y)	Joint #3 (x, y)
1	43.48	(0.2, 0.35)	(0.6, 0.36)	(0.75, 0.1)
2	24.28	(-0.17, 0.36)	(0.2, 0.5)	(0.5, 0.5)
3	5.85	(-0.36, 0.18)	(-0.01, 0.39)	(0.2, 0.6)

Design Iteration 2

In this iteration, it was realized that there is potential for torques to act in opposite directions based on the pivot point (0,0), where torques created by the weight of a beam left of the y-axis would produce a counter-clockwise torque, and vice versa for torques on the right of the y-axis, negating each other. Here, the length parameters were increased so that the torques produced by each beam were balanced across the x-axis, reducing the overall sum of torques. The following results were produced in response to this change.

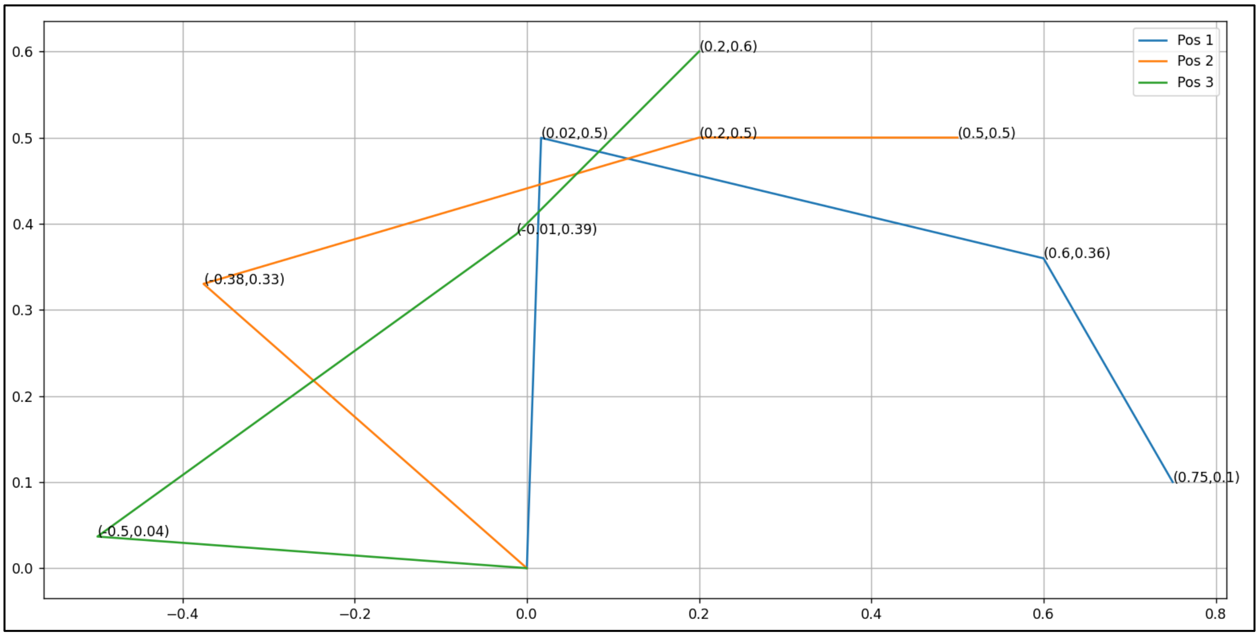

Figure 3: Design 2 Position Diagram

This design has lengths of 0.5m, 0.6m, 0.3m (Total Length = 1.4m) and a total calculated torque of 47.44 Nm.

Table 2: Specifications for each position (Design 2)

Position Number	Torque required (Nm)	Joint #1 (x, y)	Joint #2 (x, y)	Joint #3 (x, y)
1	42.57	(0.02, 0.50)	(0.6, 0.36)	(0.75, 0.1)
2	20.84	(-0.38, 0.33)	(0.2, 0.5)	(0.5, 0.5)
3	2.19	(-0.5, 0.04)	(-0.01, 0.39)	(0.2, 0.6)

Design Iteration 3

The third iteration was constructed by incorporating ideas brought by both iteration 1 and iteration 2. With the combination of low lengths which corresponding to low weights, and the torques balancing each other, the following results were produced.

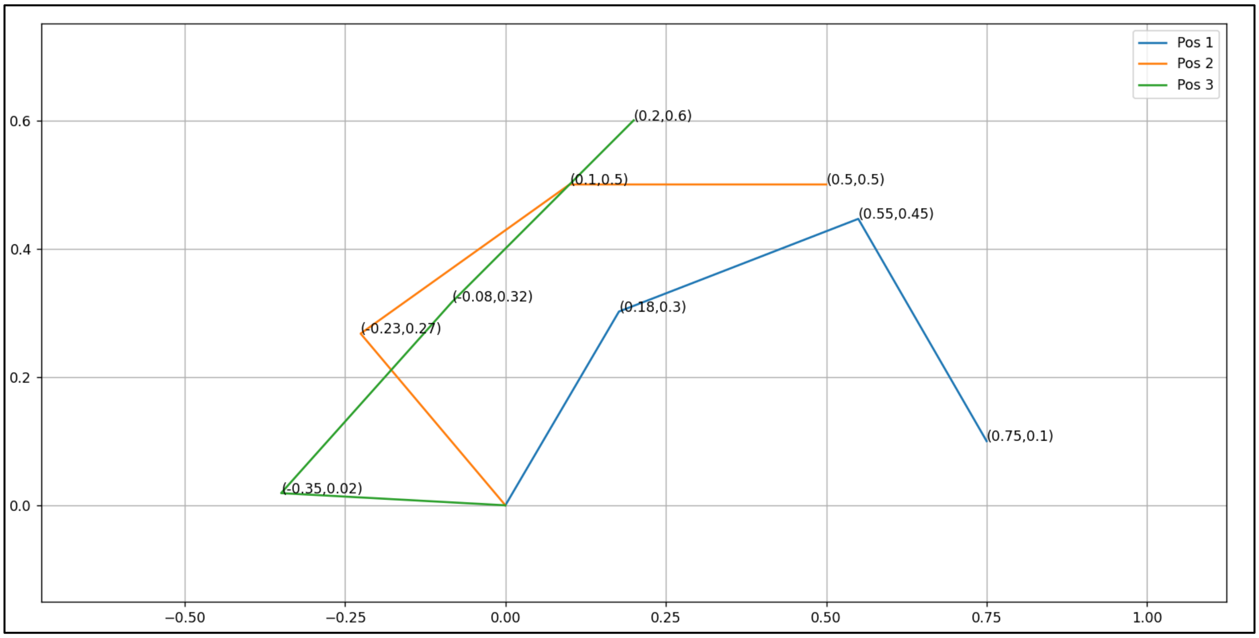

Figure 4: Design 3 Position Diagram

This design has lengths of 0.35m, 0.4m, 0.4m (Total Length = 1.15m) and a total calculated torque of 49.79 Nm.

Table 3: Specifications for each position (Design 3)

Position Number	Torque required (Nm)	Joint #1 (x, y)	Joint #2 (x, y)	Joint #3 (x, y)
1	43.41	(0.18, 0.3)	(0.55, 0.45)	(0.75, 0.1)
2	23.66	(-0.23, 0.33)	(0.1, 0.5)	(0.5, 0.5)
3	5.94	(-0.35, 0.02)	(-0.08, 0.32)	(0.2, 0.6)

3. Final Design

There are some notes to be kept in mind throughout the final design section. Firstly, the sign conventions, the downwards direction was set as positive for the gravitational forces. Furthermore, for moments, the clockwise direction was set as positive, these directions were chosen to accurately reflect the calculations of the optimization script. Lastly, the reason for slight discrepancies between the hand calculated final torque and the torque outputted by the optimization script is due to differences in significant digits. The optimization script does not round or truncate any numbers throughout its calculations, eliminating rounding error. However, in the hand calculation the number of significant digits was reduced for simplification.

Alongside this, a new component in the algorithm was implemented to check if there were intersection between segments of arms. This was done with a function created based on the shapely library [4] and ensured that some cases were not deleted when running the algorithm. This allowed for arms that were very close to, but not exactly intersecting, making the code far more effective.

3.1. Overview of Design

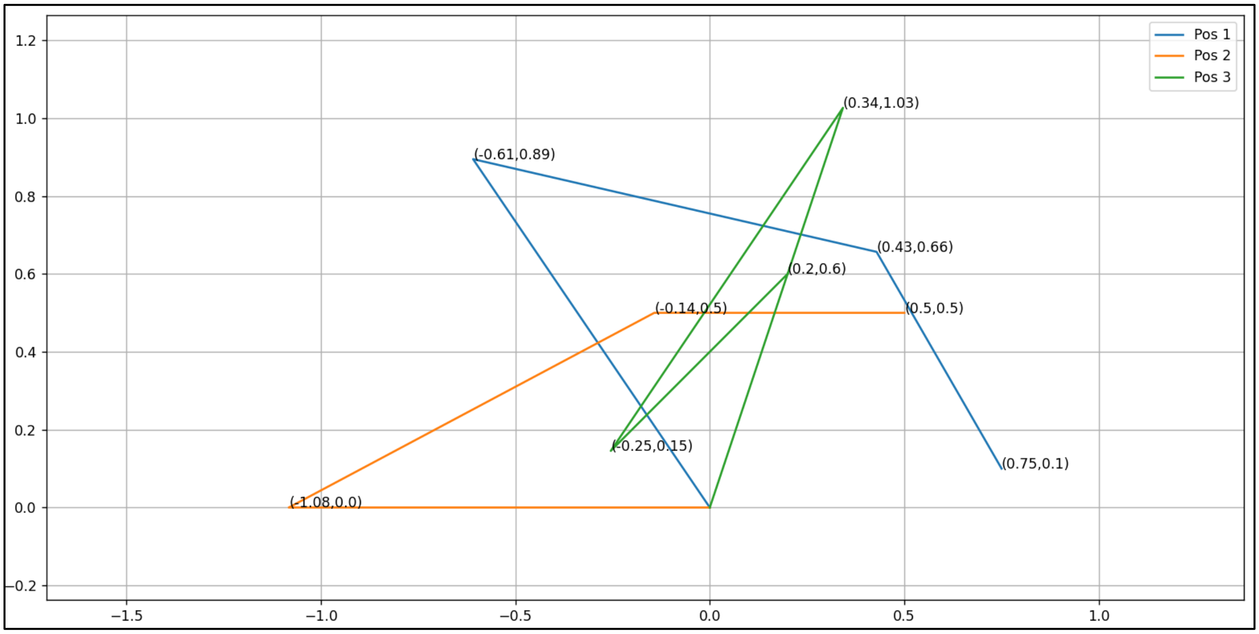

Figure 5: Final Design Position Diagram

Table 4: Final design link lengths

Link 1 length, l_1 [m]	**Link 2, l_2 [m]**	**Link 3, l_3 [m]**
1.082	1.064	0.643

Table 5: Final design link angles for each position

Position Number	**Link 1 Angle, q_1 [deg]**	**Link 2 Angle, q_2 [deg]**	**Link 3 Angle, q_3 [deg]**
1	124.43	347.53	-60
2	0	28.01	0
3	71.73	236.16	45

3.2.Force Analysis

3.2.1. Weight Calculations

Using the lengths from above, earths gravitational acceleration, and the respective densities for each beam, the weight of each beam was calculated, shown below.

$$W_1 = 9.80665\frac{m}{s^2} \times 4\frac{kg}{m} \times 1.082\text{m} = 42.443\text{N}$$

$$W_2 = 9.80665\frac{m}{s^2} \times 2\frac{kg}{m} \times 1.064\text{m} = 20.869\text{N}$$

$$W_3 = 9.80665\frac{m}{s^2} \times 1\frac{kg}{m} \times 0.643\text{m} = 6.306\text{N}$$

$$W_L = 9.80665\frac{m}{s^2} \times 5kg = 49.03\text{N}$$

Table 6: Final design weight values

Weight 1, W_1 [N]	**Weight 2, W_2 [N]**	**Weight 3, W_3 [N]**	**Load Weight, W_L [N]**
42.443	20.869	6.306	49.03

3.2.2. Position 1

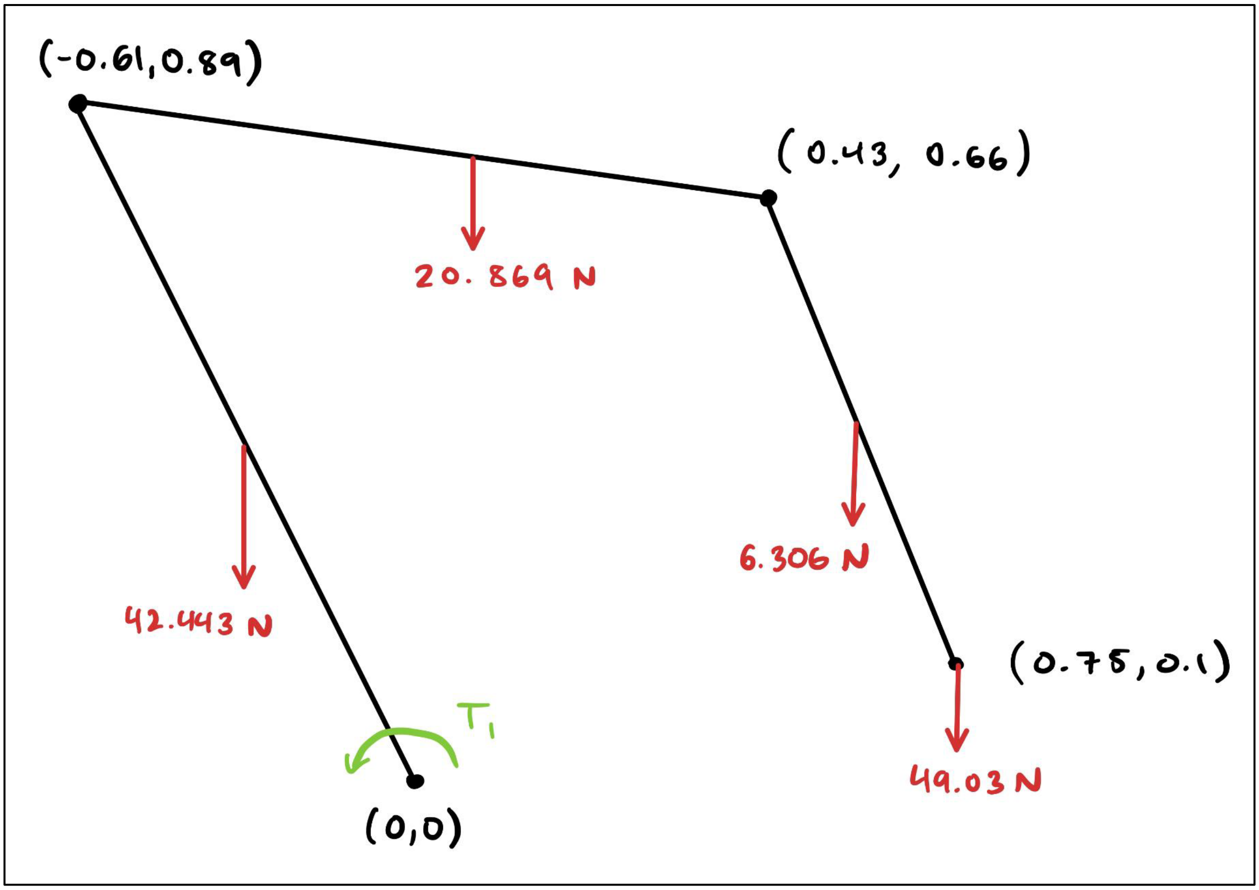

Figure 6: FBD of Position #1

$$T_{1.1} = (42.443N)\left(\frac{(-0.61) + 0}{2}m\right) = -12.945Nm$$

$$T_{1.2} = (20.869N)\left(\frac{-0.61 + (0.43)}{2}m\right) = -1.878Nm$$

$$T_{1.3} = (6.306N)\left(\frac{0.43 + 0.75}{2}m\right) = 3.721Nm$$

$$T_{1.4} = (49.03N)(0.75m) = 36.773Nm$$

$$\Sigma M_o = 0 = -12.945 - 1.878 + 3.721 + 36.773 - T_1$$

$$T_1 = 25.670Nm$$

3.2.3. Position 2

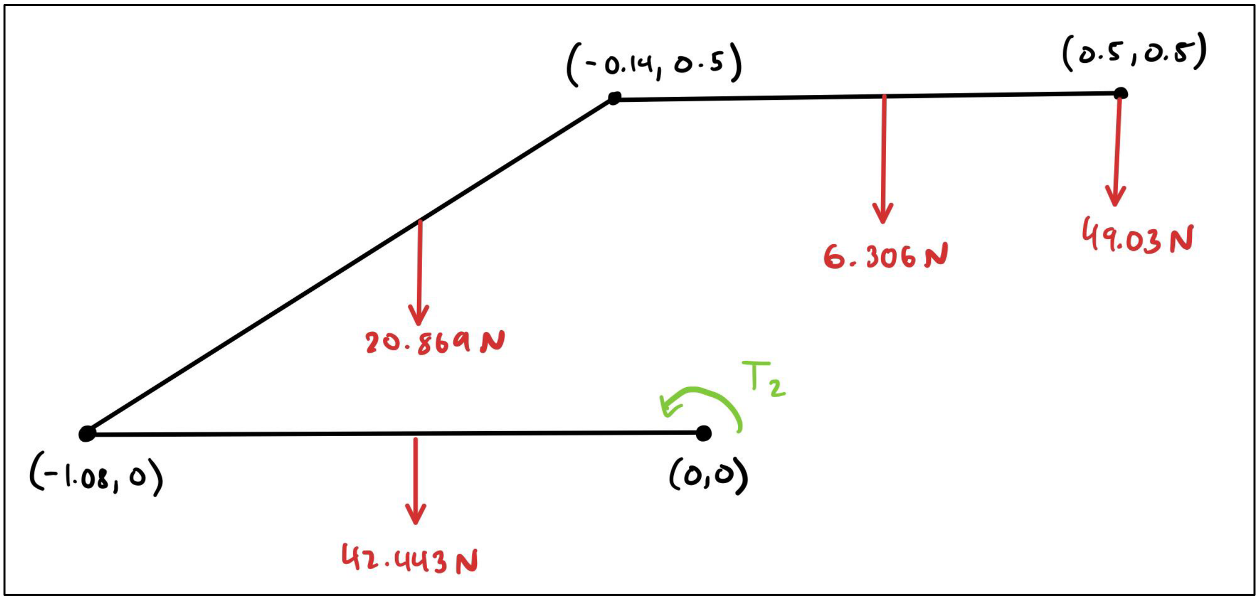

Figure 7: FBD of Position #2

$$T_{2.1} = (42.443N)\left(\frac{(-1.08)+0}{2}m\right) = -22.920Nm$$

$$T_{2.2} = (20.869N)\left(\frac{-1.08+(-0.14)}{2}m\right) = -12.730Nm$$

$$T_{2.3} = (6.306N)\left(\frac{-0.14+(0.5)}{2}m\right) = 1.135Nm$$

$$T_{2.4} = (49.03N)(0.5m) = 24.515Nm$$

$$\Sigma M_o = 0 = -22.920 - 12.730 + 1.135 + 24.515 - T_2$$

$$T_2 = -10.000Nm$$

3.2.4. Position 3

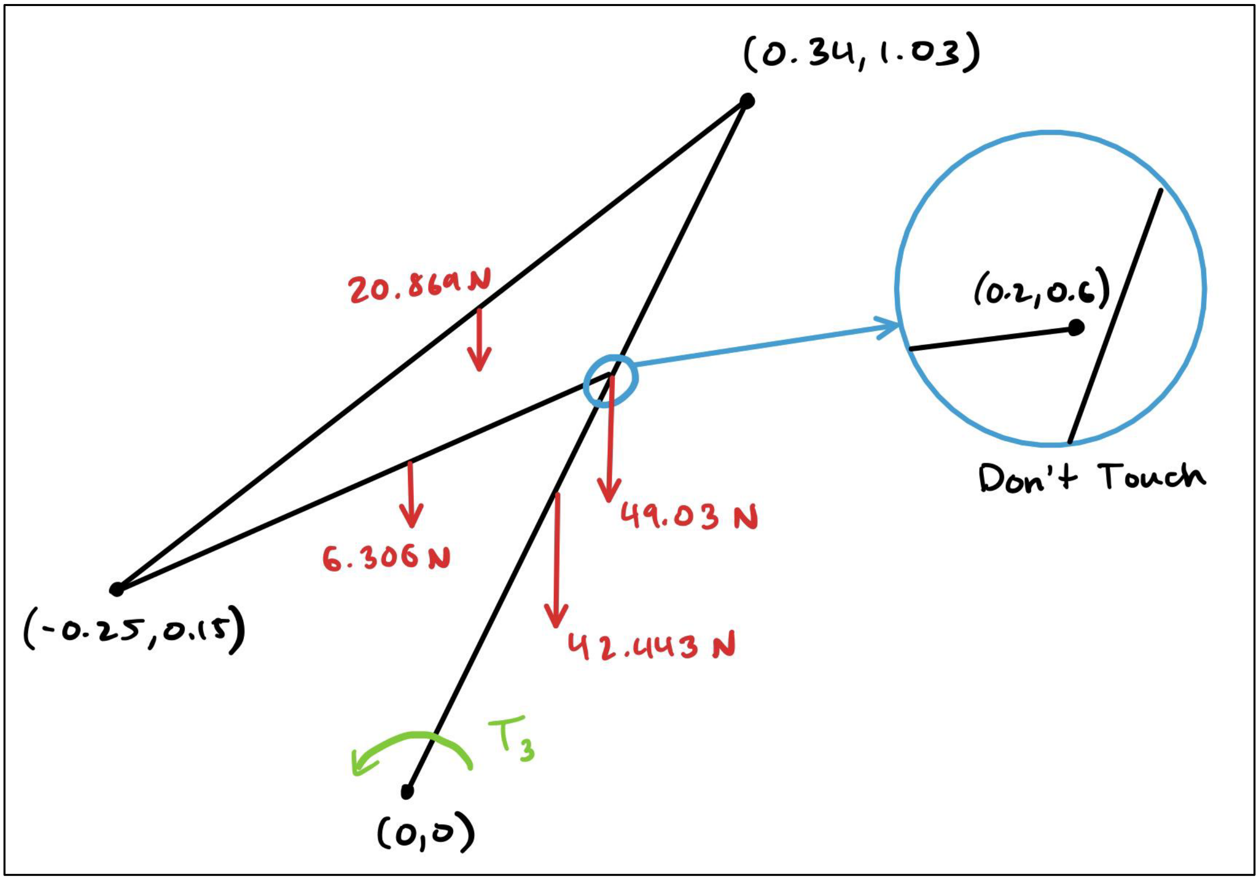

Figure 8: FBD of Position #3

$$T_{3.1} = (42.443N)\left(\frac{0.34 + 0}{2}m\right) = 7.215Nm$$

$$T_{3.2} = (20.869N)\left(\frac{0.34 + (-0.25)}{2}m\right) = 0.939\text{Nm}$$

$$T_{3.3} = (6.306N)\left(\frac{-0.25 + 0.2}{2}m\right) = -0.158Nm$$

$$T_{3.4} = (49.03N)(0.2m) = 9.806Nm$$

$$\Sigma M_o = 0 = 7.215 + 0.939 - 0.158 + 9.806 - T_3$$

$$T_3 = 17.802Nm$$

3.2.5. Final Torque

The final torque value shows slight deviation from the actual calculated torque value, this was due to compound errors created when rounding the lengths and points used to calculate all the values above. See Appendix A for exact length and torque values.

$$T = \sqrt{T_1^2 \ + T_2^2 \ + T_3^2}$$

$$T = \sqrt{(25.670)^2 + \ (-10.000)^2 + (17.802)^2}$$

$$T = 32.800Nm \ \approx 32.87Nm$$

4. Summary of Design and Conclusions

A design for the 3-DOF arm was constructed in such a way that the torque about the arm were minimized through making its length as effective as possible while simultaneously meeting the constraints described in the assignment description. Through the aforementioned methodology that was followed throughout the project, three iterations that had relatively low torque values (compared to initial guesses) were created. After reiteration and modification to the optimization python script, the torque was slowly refined as much as possible, and a final design was reached. This design was sufficient in meeting the constraints described in the project criteria and had a torque of 32.87 Nm with lengths 1.082m, 1.064m, and 0.643 for the l_1, l_2, and l_3 respectively (see Figure 5). Position 1 had points (-0.61,0.89), (0.43, 0.66), and (0.75, 0.1) for A, B and C respectively. Similarly, Position 2 had points (-1.082, 0), (-0.14, 0.5), (0.5, 0.5) and Position 3 had points (0.34, 1.02), (-0.25, 0.15), (0.2, 0.6) for said joints, which, when combined created a low optimized torque value. Overall, this project was successful in determining an optimized torque design for the 3 degrees of freedom when placed at the 3 specified positions.

5. Contributions

Name	Contributions
Joey Maillette	• Contributed one design iteration. • Worked collaboratively on the report sections (final design, initial designs, methodology etc.) • Helped advance optimization software. (NumPy implementation) • Worked on final design
Camron Sabahi-Pourkashani	• Contributed one design iteration. • Helped with optimization and SciPy implementation in software aspect of project. • Worked collaboratively on the report sections (Final Design, methodology, objective etc.) • Worked on final design
Karthigan Uthayan	• Contributed one design iteration. • Helped with calculation and optimization of the software aspect of project. • Worked collaboratively on the report sections (Introduction, Methodology, Objective, Final Design etc.) • Worked on final design

6. References

[1] SciPy, "scipy.optimize.minimize," The SciPy Community, 1 January 2023. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize. [Accessed 28 March 2023].

[2] Tomerikoo, "Finding the intersection of two cicles," Stack Overflow, 29 April 2021. [Online]. Available: https://stackoverflow.com/questions/55816902/finding-the-intersection-of-two-circles#:~:text=If%20distance%20between%20the%20two,other%20and%20so%20not%20intersecting.. [Accessed 30 March 2023].

[3] NumPy Developers, "Random Sampling (numpy.random)," NumPy, 05 April 2023. [Online]. Available: https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html. [Accessed 05 April 2023].

[4] Stack Overflow, "How can I check if 2 lines intersect?," [Online]. Available: https://stackoverflow.com/questions/3838329/how-can-i-check-if-two-segments-intersect. [Accessed 6 April 2023].

[5] S. Gillies, "The Shapely User Manual - Shapely 2.0.1 documentation," 4 April 2023. [Online]. Available: https://shapely.readthedocs.io/en/latest/manual.html. [Accessed 6 April 2023].

[6] The Matplot Development team , "Matplotlib: Visualization with Python," [Online]. Available: https://matplotlib.org/. [Accessed 6 April 2023].

7. Appendix A

In addition to the NumPy [3], and SciPy [1] libraries, the Shapely library [5] was used to find if 2 segments of the 3DOF arm intersected and the MatPlot library [6] was used to plot our 3DOF arm at its 3 respective positions. **The full source code can be found here.**

The exact values for the torques are as following (in Nm):

Position 1 torque: 25.704683479766302

Position 2 torque: -10.11182978772526

Position 3 torque: 17.822003232135337

Final Torque: 32.87253646521882

The exact values for the lengths are as following (in m):

L1: 1.0821468359052653

L2: 1.0641662437491748

L3: 0.6427511332656937