****

**MSE 222: Kinematics and Dynamics of**

**Rigid Bodies and Mechanisms**

**Course Project Phase 2: Robot Dynamics**

Group 13:

Jenieva Musico 301394261 jmusico@sfu.ca

Kate Cartier 301453357 kpc8@sfu.ca

Qusai Doctor 301429127 qda13@sfu.ca

SCHOOL OF MECHATRONIC SYSTEMS ENGINEERING

SIMON FRASER UNIVERSITY

**Executive Summary**

Using kinematics of position, velocity, and acceleration, along with joint kinematics, we are able to gather essential information on the end-effector path planning. This process gives understanding to the end-effectors movement and performance in order to optimize its design and control. To design the path trajectory, we used the trapezoidal trajectory function in MATLAB of the end-effector to ensure the robot arm can move accurately and efficiently to perform the needed tasks, in our case, writing unique letters. We obtained the equation of motion with Newton's law showing how the joint motor torques move the end-effector. Inverse kinematics determined the joint angles required to achieve our desired end-effector position and orientation. Forward kinematics allows us to determine the position and orientation of the end-effector, which is necessary for tasks such as path planning and manipulation of our robot arm. It is also a fundamental component of inverse kinematics, which is used to plan the robot's joint movements given a desired end-effector pose. Then we used the jacobian matrix to relate the end-effector velocity to the joint velocities and used it to determine the joint velocities required to achieve a desired end-effector velocity. By taking the derivative of the Jacobian matrix with respect to time, we also calculated the end-effector acceleration as a function of the joint velocities and accelerations. Plotting these paths in MATLAB generated a simulation software for reference to find a source that is required for material properties. Analyzing the designed path and its required dynamics to create two unique letters, we were able to determine the most suitable motors our robot arm will run sufficiently with.

This project is being completed by two, second-year mechatronics students and one third year mechatronics student, all with relatively little background in robotics. Our group has equally contributed to this project. Specific technical requirements were divided and others were completed collaboratively to ensure the understanding of each topic. Certain requirements the group was unfamiliar with needed more discussion and collaboration to complete.

**Table of Contents**

[**Introduction 3**](#_f2t1dlty8r01)

[**Goals and Objectives 3**](#_fnvq0g54o42y)

[**Procedure 3**](#_evaqqmnz7ve4)

[Design Criteria 3](#_22gbk6yh9sp9)

[Kinematics 4](#_izddci9gx9ig)

[Forward Kinematics Derivation: 4](#_zepotjbwu90k)

[Inverse Kinematics Derivation: 5](#_2crbwai56qh7)

[Trajectory Planning 7](#_y3moybsh2rgr)

[Material Selection 8](#_ucgfiuq3w14)

[Dynamics 10](#_qoatf1309heu)

[**Description of Software Used 12**](#_mrn63sg6resu)

[MATLAB Plots 13](#_74rkgx6yzcos)

[Torque vs Time Plot 13](#_t9fzrmfk9rd8)

[**Discussion 14**](#_kyhf675wmmwx)

[DC Motor Selection 14](#_rga8ekd6h4up)

[System Dynamics 14](#_48oi4pyfpfts)

[**Conclusion 14**](#_7jfaap1tb602)

[**References 15**](#_x4odb9os4yqh)

# 

# Introduction

This project is intended to explore the dynamical aspects of the end-effector including the kinematics of the position, velocity, and acceleration to become familiar with the technical aspects of a robot before applying it mechanically later on. This robot has parameters of two-degrees of freedom with two revolute joints and an end-effector. In our stage of education, we are able to address enough information to assemble the necessary dynamic steps to help guide our future courses of the building processes in robotics. Obtaining complete details regarding the joint kinematics allows us to apply those concepts to the essential topic of end-effector path planning. The progression of the end-effector requires the vital application of kinematics knowledge to select a suitable motor for our design. This determines the specific torque needed to generate specific motion while also withstanding the material of the robot arm throughout its motion and orientation.

# Goals and Objectives

The main goal of this project is to design a 2D planar robot for milling two unique capital letters at a constant speed. The letters we have chosen for our design are A and O. Starting from the “Home” position, the robot arm will travel to the planned path and complete the letter. The end-effector then proceeds to accelerate to the beginning of the first letter before hitting a desired constant velocity to complete the designed path. The project is broken down into a number of sub-objectives, each set to accomplish a necessary task to reach our goal. In doing so, providing graphs of the kinematics and kinetics with respect to time that is required to successfully complete the motion and path design. From this, the goal is to determine the joint torque of link one and two in order to provide the necessary motor for the system to work efficiently and effectively based on its material composition and dynamic demands.

# Procedure

## Design Criteria

The first objective of this phase of the project consists of deciding which letters the robot will be designed to mill. The letters A and O were selected as they provide an opportunity to clearly lay out the process of milling four line segments with different constraint equations.

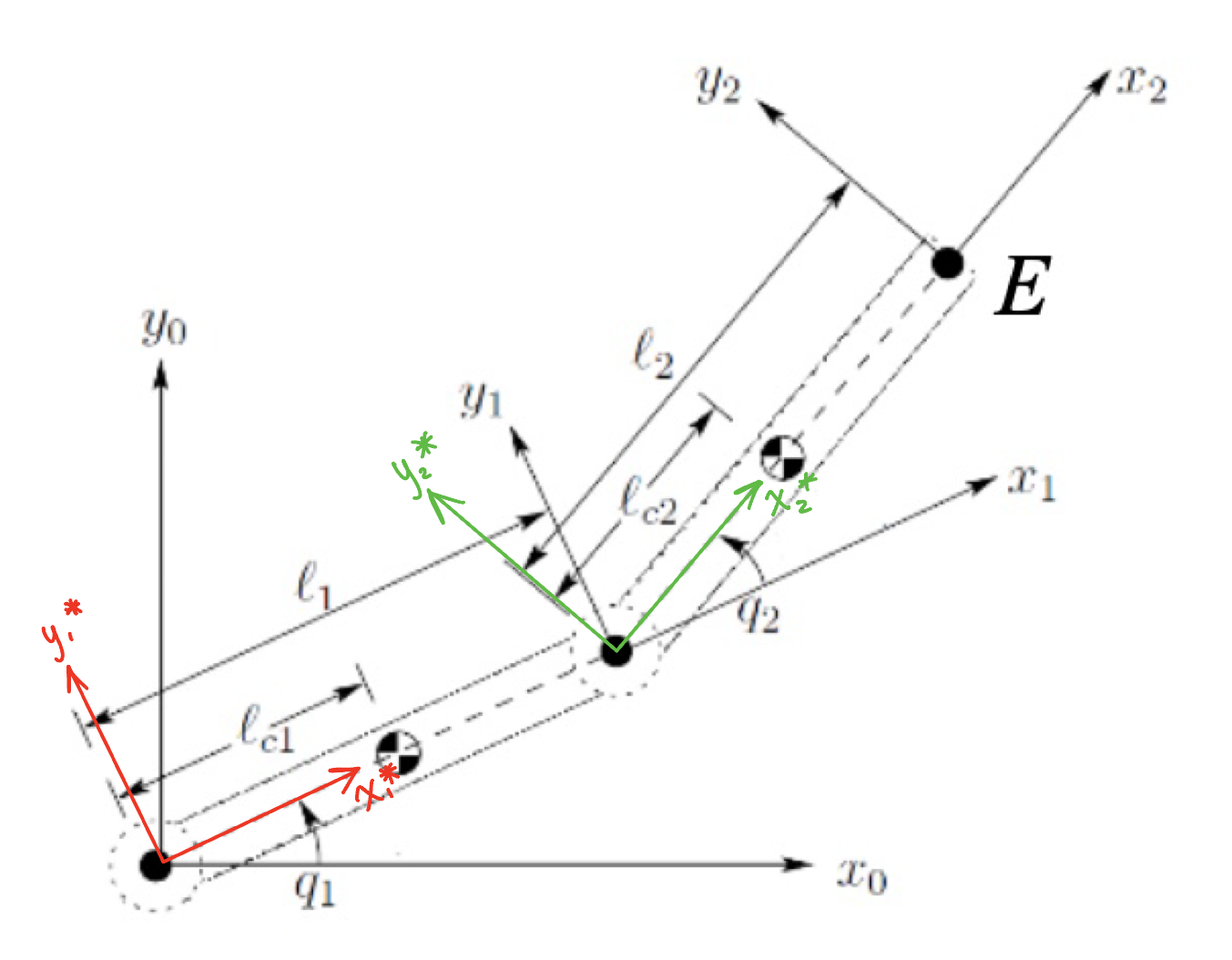
Constraint Equations for A:

y = ax + b , y = constant , y = -ax + b

Constraint Equation for O:

(x – a)2 + (y – b)2 = r2

With the constraint equations for each letter determined, the fixed and moving frames are selected and the home position is defined as the robot arms positioning when the joint angles are zero. Assuming a two-link, rotational joint arm as shown in Figure 1, the fixed frame is defined to be parallel with the ground and a moving framing is set at each joint, as well as at the end of rod two, at the end-effector.



**Figure 1.** *Two-Link Rotational Joint Arm.*

The joint angles are represented as q1= θ1, and q2 = θ2, rod lengths are l1 and l2, and the distances from centers of mass are lc1 and lc2. The fixed frame is pictured as x0, y0 and the joint moving frames are defined as x1\*, y1\* and x2\*, y2\*. The axis x1, y1 shows the x1\*, y1\* axis at the end of the first rod, making it easy to visualize the joint angle derivation. In this case, our end-effector E is at the origin of x2, y2.

## Kinematics

This model is used to simulate the movement required for tracing a letter by controlling the joint positions to place the end effector in a specific 2D position. To accomplish this, forward kinematics can be used to determine the position of any point on the robot based on the joint positions. Then, inverse kinematics is used to work backwards and calculate the joint angular velocities and accelerations based on the end-effectors position, velocity, acceleration.

To derive the velocity and acceleration of the end-effector as a function of the joint angles (θ1, θ2), measured from the joint moving frames (x1\*, y1\* and x2\*, y2\*), the following steps were followed.

### Forward Kinematics Derivation:

(1)

(2)

Forward kinematics requires that the end-effector velocity and acceleration are found with respect to the angles θ1 and θ2. It was determined before that the position of the end-effector is given by equations (1) and (2). These equations are found by simply using trigonometry on the planar 2R robot diagram. These equations are then differentiated with respect to time to find the velocity:

(3)

(4)

For the acceleration, the velocity functions, equations (3) and (4), are differentiated again giving equations (5) and (6):

(5)

(6)

### Inverse Kinematics Derivation:

Similarly, inverse kinematics requires that the end-effector angular velocity and acceleration are found, but with respect to the position of the end-effector x and y. In the project description the equations are provided as:

(7)

(8)

Equation (7) is obtained by squaring equations (1) and (2), adding them together, and using the trigonometric identities ‘sin2(θ) + cos2(θ) = 1’ and ‘cos(θ1 - θ2) = cos(θ1)cos(θ2) + sin(θ1)sin(θ2)’ to simply the expressions [3]. In this case, θ1 was set to (θ1 + θ2) and θ2 was set to θ1. Equation (8) is a bit more complex and is obtained by first applying the trigonometric identity for compound angles on equations (1) and (2). Each equation is then multiplied by a secondary factor, and the equation for x will be subtracted from the equation for y. Simplify the expression and write in terms of θ1 to obtain equation (8).

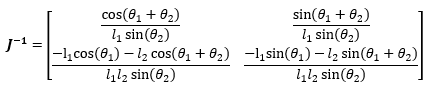
In order to find the angular velocity and acceleration of the inverse kinematic formulas, the Jacobian **J** must be used. From forward kinematics, equations (3) and (4) can be written in matrix form as:



This correlates directly to the equation = **J**. By finding the inverse of the Jacobian, the angular velocity and could be found. The inverse of a 2x2 matrix can be found by multiplying the determinant of the matrix by the altered matrix as shown below:



Doing so for the Jacobian gives the inverse:

 (9)

Now, multiplying the Jacobian with the velocity vector **Ẋ** gives the equation for the angular velocity and .

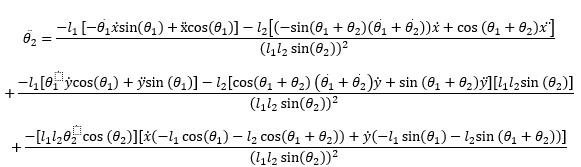
 (10)

 (11)

Finding the derivative of the above equations with respect to time gives the following angular acceleration equations:



 (12)

(13)

## Trajectory Planning

To plan the trajectory of the end-effectors position and orientation, a description for how the design will follow a specific path over time, given constraints such as position, velocity, and acceleration must be defined. This path includes a set of waypoints directing the end effector from its initial position, through a sequence of desired intermediate points and ending at the desired final position. Two methods of trajectory planning are explored in the process of this design to achieve two different forms of motion; task space trajectory, and joint space trajectory.

Task space trajectory is represented in terms of the end-effector's position and orientation in the robot's workspace. It specifies the desired path that the end-effector should follow, and is defined using waypoints. Task space trajectory planning involves calculating the joint angles required to move the end-effector along the desired path, taking into account the robot's kinematics and any constraints on its motion. Task space trajectory is used in designing the milling of the letter A as it follows a straighter trajectory since the end effector is moving smoothly with respect to the environment even if the joints are not.

Joint space trajectory is represented in terms of the joint angles of the robot over time. It specifies the desired path that the robot's joints should follow to move the end-effector along the task space trajectory. Joint space trajectory planning involves calculating the joint angles directly, without necessarily specifying the end-effector's position and orientation at every point in time. Joint space trajectory is used in the design of milling the letter O as it follows a smoother rounded trajectory as it is not solving the inverse kinematics between intermediate points.

To generate a smooth trajectory, trapezoidal velocity trajectory or polynomial trajectory motion profiles are used to determine the motion of the robot throughout its trajectory.

A trapezoidal velocity trajectory is a type of profile that consists of a constant acceleration phase, a constant velocity phase, and a constant deceleration phase. The acceleration and deceleration phases are ramped up and down respectively, to ensure that the robot moves smoothly and does not exceed its maximum speed or acceleration limits. A trapezoidal velocity profile is used for the design of milling the two letters to produce a precise movement along the trajectories.

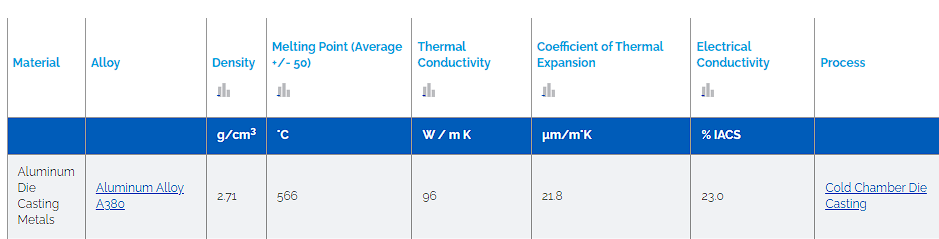
A polynomial trajectory, on the other hand, is a type of profile that uses polynomial functions to describe the robot's motion over time. Polynomial trajectories can take on many different shapes, such as linear, quadratic, or cubic, and can be customized to fit specific applications or requirements. Polynomial trajectories can provide smoother and more continuous motion than trapezoidal velocity profiles, however the loopy resultant trajectories are why this method is not used in the robot’s design.

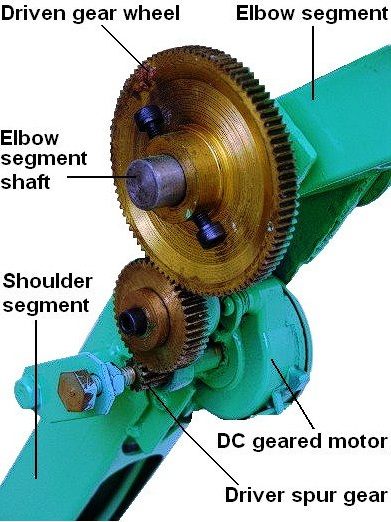
## Material Selection

After using kinematics to map the joint positions of the robot design to the specific positions and orientation of the end effector, dynamics is used to map the required joint forces and torques to their position, velocity, and acceleration. Moving to dynamics, additional information regarding the robot design is required. Specifically; mass, moment of inertia, and center of mass.

The material chosen for the robot arms is aluminum A380 (material properties shown in **Table 1**). This is because aluminum is a softer and less heavy metal, which makes it easier to die cast the robot arms [1]. This also means that the arms would be able to be lifted up more easily and carry more weight, as aluminum is less heavy. The arms would have a small width in comparison to the length, in order to reduce the amount of mass and avoid deflection, as the robot is only drawing letters and doesn’t need the stability to hold heavy objects. One motor would then be placed on the shoulder through spur gears and a shaft with a key to fit into the keyseat of the arm and shoulder connection, and the other motor would be placed on the elbow, connecting both the arms with its shaft. **Fig. 2** shows how a spur gear would connect the motor to the elbow, although in this case, the motor would be turned 90 degrees in order for the motion to be in only the x-y plane. For simplicity, calculations for the centers of mass and moments of inertia will only require the mass of the arms.

**Table 1.** *Material properties of aluminum alloy A380 [1].*





**Figure 2.** *Visualization of how the spur gear will connect with the motor [4].*

With the density now given, the mass of the arm and elbow can be found by multiplying the density by the volume of each link. In this case, the arms are assumed to be of similar length, 0.3m, and the width and depth are made smaller in comparison so that the center of mass and moment of inertia calculations can ignore the 3D case. The width is therefore assumed to be 0.03m for mass reduction and the depth/height is assumed to be 0.005m in order to ignore the 3D plane. Although this would cause some error in potential calculations requiring the moment of inertia, the error would likely be small for a theoretical robot arm. The mass can therefore be calculated for future equations as:

Based on the arm and elbow design discussed, the center of mass calculations for link 1 and link 2 are determined as:

(14)

(15)

(16)

(17)

Where the center of mass is measured from the x0 - y0 axis. If the center of mass of the entire system is required, the equations would become:

(18)

(19)

Next, the moment of inertia about the mass centers, and about the joints 1 and 2 are determined. The moment of inertia about the joints can be found by using the parallel axis theorem, with the moment of inertia from the center of mass. In this case, the links are assumed to be in a 2D-plane, so the depth of the arm and elbow is not considered.

## Dynamics

Equations of Motion:

*Link 1:*





 (20)

 (21)

*Link 2:*





 (22)

 (23)

*Eliminating constraint forces and separate them from the torque joints using:*

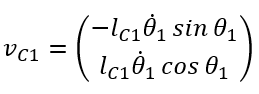
,  (24)

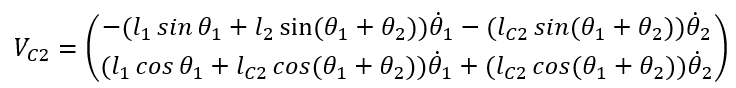
*Plug [24] into [20] and [22] and plug [23] into [20]and [22] to get the expression [25] and [26]:*

 (25)

 (26)

*Sub in link 1 and 2 linear velocities and joint displacements into [25] and [26] (the angular velocity, , is measured relative to ground where is measured relative to link 1)*









*Torque 1 expression:*

 (27)

*where*









*Torque 2 expression:*

 (28)

*where*









# Description of Software Used

MATLAB was used for the design of this project as its extensive toolboxes made it the ideal software for modeling the complex kinematics and dynamics of a 2D planar robot's motion. To implement the design of a two-link rotational arm in MATLAB, the RigidBodyTree class was used to represent the connectivity of rigid bodies with joints.

## MATLAB Plots

|  |  |
| --- | --- |
| **Figure 3.** *Desired (X, Y) trajectory plot of letter O.* | **Figure 4.** *Desired (X, Y) trajectory plot of letter A.* |
| **Figure 5.** *Position, velocity, and acceleration of end-effector throughout desired (X, Y) trajectory of the letter O.* | **Figure 6.** *Position, velocity, and acceleration of end-effector throughout desired (X, Y) trajectory of the letter A.* |
| **Figure 7.** *Joints (1,2) angles and angular velocities over time for the trajectory of the letter O, based on end effector kinematics.* | **Figure 8.** *Joints (1,2) angles and angular velocities over time for the trajectory of the letter A, based on end effector kinematics.* |

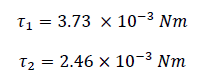
The MATLAB plots shown in **Fig. 3** **and 4** display the desired trajectory of the end-effector for each letter, as found using a set of 2-D waypoints for the letter and then connecting them using a trapezoidal profile where each segment has a duration of 1 second. **Fig. 5 and 6** display plots of the end effectors position, velocity, and acceleration over time respectively, for each letter, as determined through trapezoidal trajectory planning.

The MATLAB plots shown in **Fig. 7 and 8** show the joints (1,2) position, and velocity over time as calculated by using a trapezoidal trajectory profile, followed by inverse kinematics to find the joint angles in terms of the end effectors position, velocity, and acceleration over time. The angular acceleration is not pictured as there were issues calculating the values using MATLAB.

# Discussion

## DC Motor Selection

Due to time limitations and the angular acceleration not working in MATLAB, joint torques from a similar looking model found online are used [2]. In this design, the maximum torques are calculated based on the motor and part specifications, but for this project the torque found will be used in order to find a motor. If spur gears are used, the torque from the motors could increase or decrease depending on the sizing of the gears used, and the angular acceleration from the motors could change, however for simplicity they will be ignored.



For joint 1, a 24V dc motor was found [6], with a power rating of 350 W and RPM of 1000. Using the formula found online [8], a torque very close to the wanted value is found. Similarly, for joint 2, a 12 V dc motor with 1000 RPM [7] was found, which seems ideal for our case.

## System Dynamics

In the absence of external forces or interactions with a workpiece, the resistance forces experienced by the robot arm are due to its own inertia and the effects of gravity. The inertia of the links causes them to resist changes in motion, while the force of gravity pulls down on each link, creating a torque that must be counteracted by the motors driving the joints.

Additionally, frictional forces between the moving parts of the robot arm can also create resistance, leading to energy losses and decreased efficiency. These frictional forces can arise from a variety of sources, including bearing friction, motor friction, and Coulomb friction between surfaces in contact.

Overall, the resistance forces experienced by a two-link, 2 DOF robot arm when not interacting with any workpiece are primarily due to its own inertia and the effects of gravity, as well as frictional forces between its moving parts. Understanding and managing these resistance forces is essential for optimizing the performance and efficiency of the robot arm in various applications.

# Conclusion

In conclusion, through the analysis of the kinematics and kinetics of our system, we were able to utilize trajectory planning, dynamic equations, and MATLAB to graph the resulting motions but unfortunately not provide the correct joint torque forces of our system. Therefore, we relied on other sources to provide link one and two joint torque in order to provide a motor for the system to the best of our ability due to the circumstances. With these joint torque values, we were able to select the 24VDC [6] motor and 12VDC [7] motor due to the magnitude of the joint torques of link one and link two.

References

1. “Aluminum alloy A380 properties: Aluminum die casting,” *Dynacast*. [Online]. Available: <https://www.dynacast.com/en/knowledge-center/material-information/aluminum-die-casting-metals/aluminum-alloy-a380>. [Accessed: 10-Apr-2023].
2. G. Bourke, “2 DOF Articulated Pen Plotter,” *blogger.com*, 01-Feb-2014. Available: <http://2dofpenplotter.blogspot.com/2014/02/2-dof-articulated-pen-plotter-beng.html#more>. [Accessed: 8-Apr-2023].
3. RoboAnalyzer IIT Delhi, *Video 7: MATLAB Program for Inverse Kinematics of 2R Planar Manipulator*. Available: <https://www.youtube.com/watch?v=4r5CIhNALAQ&list=PLYForsr6emXvwHhd4Nu-sapoIX5IkAOfU>. YouTube, 2021. [Accessed: 6-Apr-2023].
4. Zeb, Jahan. (2007). Design and Development of a Radiation Protection Assistant Robot. <https://www.researchgate.net/figure/DC-geared-motor-with-spur-gear-in-Elbow-module-assembly_fig9_44163596>. [Accessed: 10-Apr-2023].
5. “Chapter 7 dynamics,” ocw.mit.edu, 2005. <https://ocw.mit.edu/courses/2-12-introduction-to-robotics-fall-2005/c7caaa2376b8ec01e270328a3b80b029_chapter7.pdf#:~:text=Two%20methods%20can%20be%20used%20in%20order%20to,and%20more%20systematic%20than%20in%20the%20Newton-Euler%20method> (accessed Apr. 03, 2023).
6. “57mm 24V 350W 1000RPM 3.2Nm Geared Brushless DC Motor with Precision Planetary Gearbox,” *FRANKHUMOTOR*. <https://www.frankhumotor.com/57mm-24v-350w-1000rpm-3-2nm-geared-brushless-dc-motor-with-precision-planetary-gearbox.html> [Accessed: 9-Apr-2023]
7. “12VDC 8-1000RPM High Torque Low Rpm DC Motor All Metal Low Noise Gear Motor JGB37-545,” *AliExpress*, 2023. <https://www.aliexpress.com/i/32871194707.html> [Accessed: 11-April-2023]
8. “WEN Technology - Calculators.” <http://wentec.com/unipower/calculators/power_torque.asp> [Accessed: 11-April-2023]
9. “Design Trajectory with Velocity Limits Using Trapezoidal Velocity Profile - MATLAB & Simulink.” <https://www.mathworks.com/help/robotics/ug/design-a-trajectory-with-velocity-limits-using-a-trapezoidal-velocity-profile.html> [Accessed: 5-Apr-2023]
10. “Derive and Apply Inverse Kinematics to Two-Link Robot Arm - MATLAB & Simulink Example.” <https://www.mathworks.com/help/symbolic/derive-and-apply-inverse-kinematics-to-robot-arm.html> [Accessed: 7-Apr-2023]
11. “2-D Path Tracing with Inverse Kinematics - MATLAB & Simulink.” <https://www.mathworks.com/help/robotics/ug/2d-inverse-kinematics-example.html> [Accessed: 3-Apr-2023]