Development of automatic teeth-cleaning robot

Anthony Rezkalla*, Abdelmonem Abdelrahman*,
Paula Magdy*, Joseph Maher* and Salma Essam*
*German University in Cairo (GUC), Egypt
Emails: anthony.rezkalla@student.guc.edu.eg, abdelmonem.eissa@student.guc.edu.eg, paula.kousa@student.guc.edu.eg,
joseph.estawro@student.guc.edu.eg, salma.abass@student.guc.edu.eg

***Abstract*—This paper showcases the CAD model of a right robotic arm, details its individual components, and presents a comprehensive circuit diagram. Additionally, a DH convention table is constructed to elucidate the arm's kinematic configuration. Furthermore, we conduct simulations using Simulink in various locations. These elements collectively offer an in-depth exploration of the robotic arm, providing valuable insights for researchers and practitioners in the field of robotics and automation.**

***Keywords*-Robotic; DH-Convention; Circuit Diagram; Kinematic; Simulation..**

I. Introduction

In the dynamic landscape of technological advancements over the past two decades, there has been a notable surge in the development of robotic devices dedicated to improving the lives of individuals facing mobility challenges. This proposal pioneers a groundbreaking concept—the teeth-cleaning robotic arm—a visionary device meticulously crafted to foster oral health and autonomy for diverse disability groups. Targeting conditions such as cerebral palsy, motor neuron disease, muscular dystrophy, multiple sclerosis, strokes, and degenerative illnesses prevalent among the elderly, this innovative robotic arm aims to redefine personal care by providing an accessible and efficient solution to a crucial aspect of daily hygiene. By combining cutting-edge robotics with a compassionate focus on addressing specific needs, this proposal envisions a transformative leap towards empowering individuals with disabilities and advancing the frontier of inclusive healthcare technologies. Section II illustrates the methodology focusing on the hardware design and circuit implementation showcasing the utilization of Proteus software, as demonstrated in 2, and the robot forward and inverse kinematics. Section III introduces the simulation environment employed to simulate the robot's motion, namely MATLAB/Simulink. Additionally, the section outlines the specific parameters utilized, such as the length, width, and thickness of each link. The simulation results are presented in the form of figures and visualizations, accompanied by detailed analysis and comments on each response.

Finally, in Section IV, we draw conclusions based on the findings from the preceding sections. This section encapsulates the key takeaways from the study and provides insights into potential future recommendations for further improvement and exploration. Additionally, a comprehensive list of references is provided.

II. Methodology

The methodology section of this report provides a comprehensive overview of the systematic approach employed in designing and implementing the robotic system. This section is divided into two key components: Hardware Design and Implementation, and Robot Kinematics. The Hardware Design and Implementation segment focuses on the various components that constitute the physical infrastructure of the robotic system. It encompasses hardware components, the circuit diagram detailing their interconnections, and the manufacturing process that transforms these designs into tangible robotic hardware.

A. Hardware Design and Implementation

1) Hardware Components: The specific physical elements that constitute the robotic system are illustrated in the following table I. It delves into the intricacies of the mechanical, electrical, and electronic components that form the backbone of the hardware.

Figure 1: Robot Limbs

Type	Amount
Servo Motors	5
3D Printed Limbs	4
Power supply	1
Arduino	1
Gripper	1

Table I: Hardware components

2) Circuit Diagram: The circuit diagram is a visual representation of the electrical connections and relationships between various hardware components. This subsection elucidates the design and layout of the circuit, providing insights into the electrical architecture that enables seamless communication and functionality. The circuit diagram is ellaborated in the following figure 2.

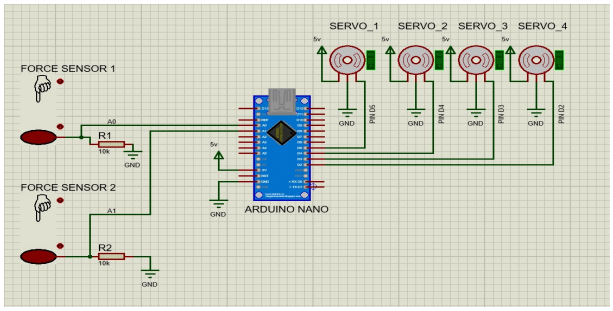

Figure 2: Proteus Circuit Diagram

B. Robot Kinematics

1) Robots' Frame Assignment: Understanding the frame of reference is pivotal in the analysis of robot motion. This subsection outlines how frames are assigned to the robotic system, setting the stage for subsequent kinematic computations. This will be illustrated in fig.3, fig.4, fig.5, fig.6, fig.7, and fig.8

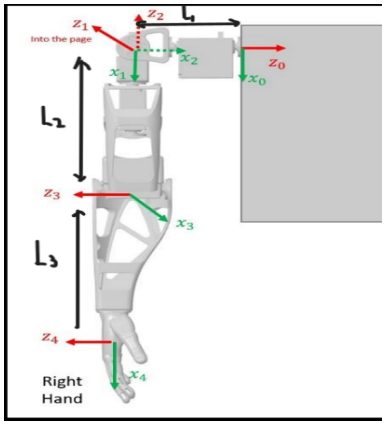

Figure 3: Frames assignment on paper

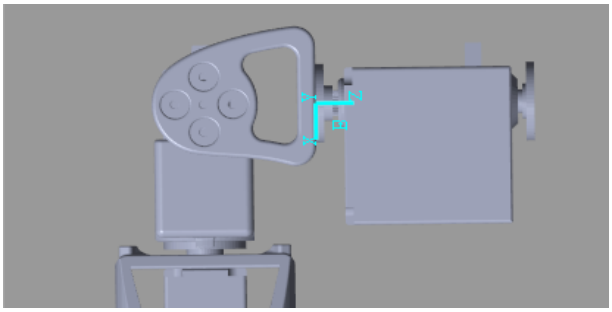

Figure 4: Revolute Joint 1

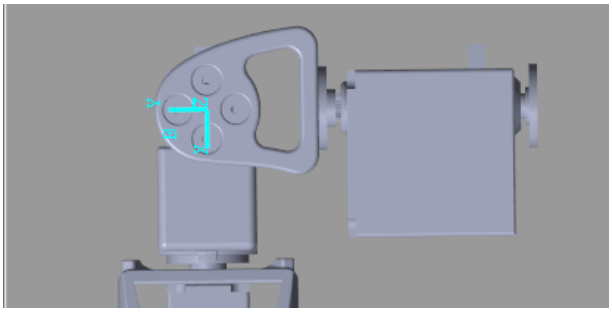

Figure 5: Revolute Joint 2

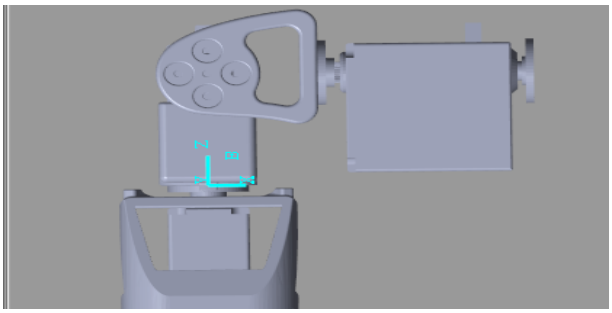

Figure 6: Revolute Joint 3

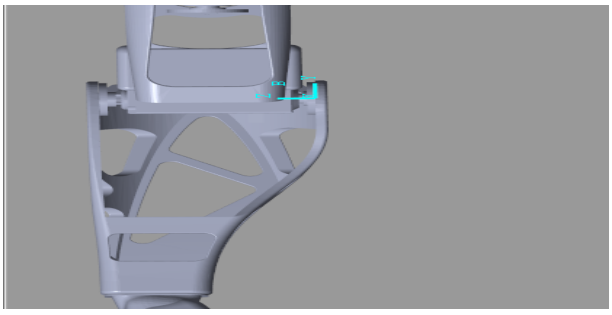

Figure 7: Revolute Joint 4

2) DH Convention: The Denavit-Hartenberg (DH) Convention is a crucial aspect of robot kinematics. This subsection provides an explanation of the DH parameters, elucidating how they are assigned and utilized in the kinematic analysis of the robotic system. These parameters are shown

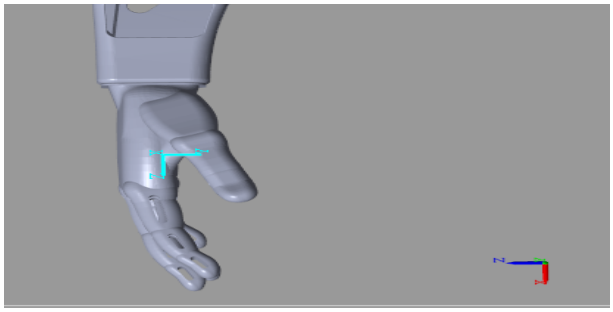

Figure 8: End Effector frame

in the following table II.

Table II: DH-Parameters Table

i	θi	di	ai	αi
0-1	q_0	$-l_1$	0	$-\pi/2$
1-2	$-\pi/2+q_1$	0	0	$\pi/2$
2-3	$-\pi/2+q_2$	$-l_2$	0	$\pi/2$
3-4	$-\pi/2+q_3$	0	l_3	0

3) Total Homogeneous Transformation Matrix : Building upon the DH Convention, this subsection explores the concept of the Total Homogeneous Transformation Matrix, a key mathematical tool used to describe the pose of the robot in its operational space.

$$T_{EE} = \begin{bmatrix} A & B & C & D \end{bmatrix} \tag{1}$$

Then by substituting the different columns A, B, C, D which explained in the appendix into the Total Homogeneous Transformation Matrix with:

$$q_0 = 0, q_1 = 0, q_2 = 0, q_3 = 0 \tag{2}$$

$$T_{EE} = \begin{bmatrix} 1 & 0 & 0 & 0.2930 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.0660 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

4) Forward Position Kinematics: This subsection addresses the forward kinematics problem, providing insights into how the end-effector's position and orientation are determined based on the joint variables. After conducting simulations, the 4 Degree of Freedom (DOF) robot was analyzed using Simscape Multibody. The system was subjected to two sets of actuation angles: the first with all angles set to 0 (0,0,0,0), and the second with angles of π/2, π/2, π/2, π/2, respectively. Additionally, the results obtained from the simulations were compared with those derived from custom MATLAB functions designed to compute the end effector's position through forward kinematics.

The outcomes of these experiments provided valuable insights into the accuracy and performance of the Simscape Multibody simulation. Notably, when actuated with 0000 angles, the simulation results closely matched the expected outcomes based on the kinematic equations. This demonstrated the reliability of the simulation tool for predicting robot behavior in a controlled scenario, which can be illustrated in the following fig. 9.

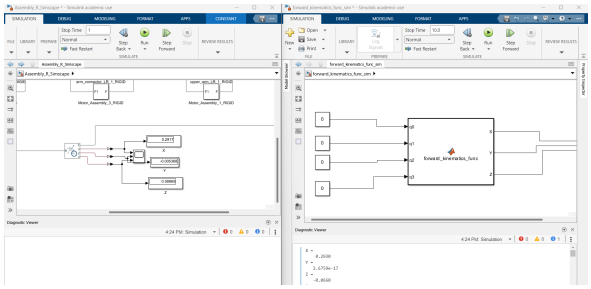

Figure 9: Case Study 0

However, when subjected to the $\pi/2$, $\pi/2$,$\pi/2$,$\pi/2$ angles, some discrepancies were observed. These differences were attributed to potential limitations in the model, such as joint limits, singularities, or numerical approximations. Additionally, the MATLAB functions were utilized to cross-verify the results. Interestingly, the functions produced nearly identical results to those obtained through Simscape Multibody, further validating the simulation. This can be seen in the following figure 10.

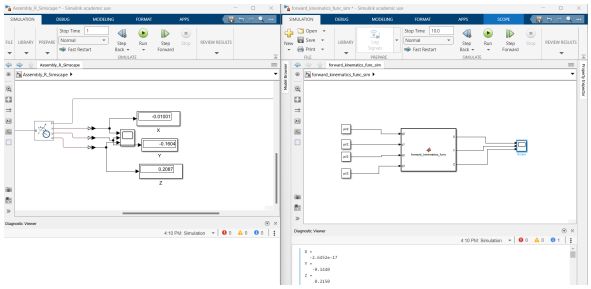

Figure 10: Case Study $\pi/2$

5) Inverse Position Kinematics: In contrast, the inverse kinematics problem is explored in this subsection. It delves into the methodologies employed to calculate the joint variables necessary to achieve a desired end-effector pose. In order to solve the Inverse Kinematics using Newton Raphson, we use the following equation 19:

$$q_{n+1} = q_n + \left[\frac{\partial F}{\partial q}\right]^{-1} F(q_n) \tag{4}$$

where a Matlab function is used to obtain the pseudo-inverse (since the system is under-actuated) of the matrix representing (dF/dq).

$$\left[\frac{\partial F}{\partial q}\right] = \begin{bmatrix} E \\ F \\ G \end{bmatrix} \tag{5}$$

Then by substituting the different rows A, B, C which explained in the appendix and taking into consideration matlab's computational capabilities are restricted when attempting to obtain the symbolic pseudo-inverse of equations. so when we substitute into the equation with $q = [\pi/6, \pi/5, \pi/6, \pi/2]$

$$\left[\frac{\partial F}{\partial q}\right] = \begin{bmatrix} -0.069 & 0.13 & 0.099 & -0.1 \\ -0.13 & 0.073 & -0.026 & -0.058 \\ 0 & -0.078 & -0.1 & -0.085 \end{bmatrix} \quad (6)$$

$$\left[\frac{\partial F}{\partial q}\right]^{-1} = \begin{bmatrix} 2.6 & -7.2 & 1.8 \\ 1.3 & 1.9 & -2.9 \\ 4.1 & 4.2 & -2.0 \\ -6.1 & 3.2 & -6.8 \end{bmatrix} \quad (7)$$

After 20 iterations:

$$q_{n+1} = \begin{bmatrix} 0.6289 & 0.788 & 0.4069 & 1.571 \end{bmatrix} \quad (8)$$

6) Trajectory Design: The trajectory design phase is pivotal in planning and controlling the robot's motion. This subsection explores various methodologies employed in designing trajectories, ensuring precise and efficient robot movements. For the trajectory design for the brushing teeth robot, the robot arm executes a task-space trajectory along a circular track. Notably, the center of this circular trajectory is intentionally offset, resulting in a linear shift of the brushing track along the circular path.

The starting point $X_0 = (0, 0.149, 0.078)$ and the end-point $X_f = (0, 0.149, 0.083)$.

To perform five circles of radius $r = 0.02$ m in a duration of ten seconds:

$$\theta = \omega t$$

$$5(2\pi) = \omega(10)$$

$$\omega = \pi$$

The trajectory equation in the x direction:

$$x(t) = r\cos(\omega t) + c = 0.02\cos(\pi t) + c \quad (9)$$

At $t = 0$, $c = -0.02$:

$$x(t) = 0.02\cos(\pi t) - 0.02 \quad (10)$$

The y value is constant since it starts at 0.149 m and ends at the same point. The trajectory equation in the z direction:

$$z(t) = r\sin(\omega t) + c_1 + c_2 t = 0.02\sin(\pi t) + c_1 + c_2 t \quad (11)$$

At $t = 0$, $c_1 = 0.078$:

$$z(t) = 0.02\sin(\pi t) + 0.078 + c_2 t \quad (12)$$

where c_2 is a factor multiplied by t to shift the center of the circle along the z-axis.

III. Simulation Results

The simulation results presented in this section offer a comprehensive evaluation of the robotic system's performance, providing invaluable insights into its capabilities and functionality. Divided into several key subsections, this segment begins with an exploration of the Initial Simulation, shedding light on the preliminary assessments and foundational characteristics of the simulated model. Subsequently, the Trajectory Design Results unveil the system's responsiveness and precision in executing predefined paths. Finally, the culmination of these simulations is manifested in the examination of the Final Hardware, where the virtual assessments are translated into real-world applicability, bridging the gap between theoretical simulations and practical implementation.

A. Initial Simulations

This section include several simulations for the robotic arm with different configurations on simscape and the CAD model for the robotic arm.

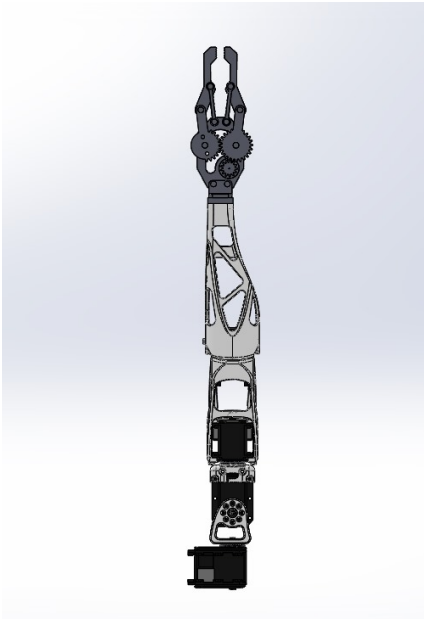

Figure 11: CAD Model

The following figure 12 that each joint is set on zero degrees.

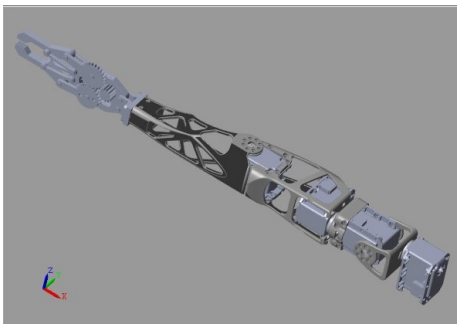

Figure 12: Zero Angles Each Joint

The following figure 13 that each joint is set on 45 degrees.

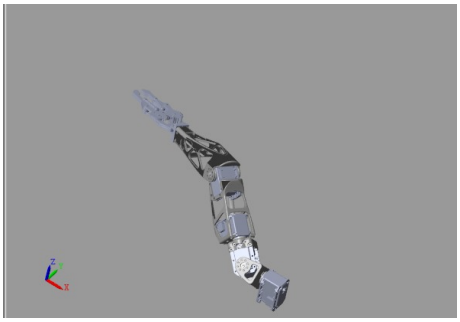

Figure 13: 45 Degrees Each Joint

The following figure 14 that each joint is set on 90 degrees.

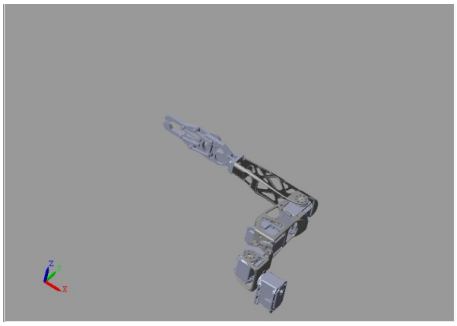

Figure 14: 90 Degrees Each Joint

The following figure 13 that the first and second joint are set on zero degrees, the third and the fourth joint is set on 90 degrees.

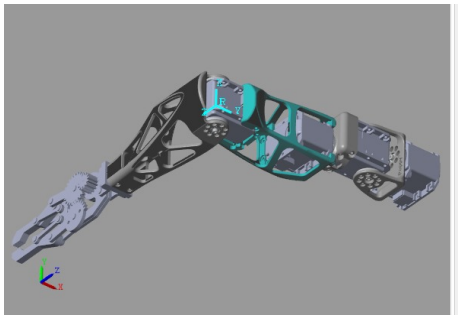

Figure 15: Zero,Zero,Ninety,Ninety

B. Trajectory Design results

In the implementation, the joint angles, represented by the q array, were obtained from the task traj function. This function generates a circular trajectory with a shifting center along the z-axis, determined by parameters such as sampling time, trajectory time, and initial and final positions. Subsequently, these joint angle values were transcribed into arrays and seamlessly integrated into the Arduino IDE. The Arduino code was designed to iterate through each array index, converting the angles to degrees, and transmitting them to the respective servo motors of the robotic system. This iterative loop facilitated the complete execution of the array of angles, ensuring the intended movement of the robot's joints. Notably, a Simulink Simscape simulation corroborated the circular trajectory, validating the expected behavior of the implemented control scheme.

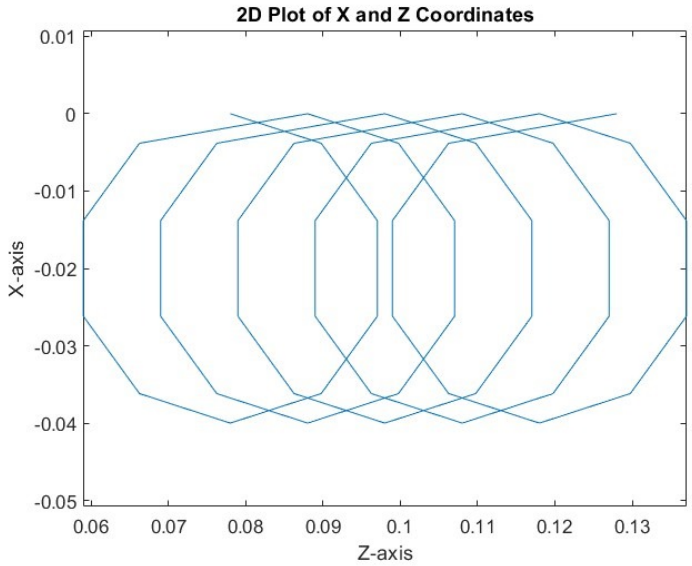

Figure 16: Circular Trajectory

C. Final Hardware

The Hardware Design and Implementation and Robot Kinematics sections collectively provide a comprehensive

understanding of both the physical and computational aspects of the robotic system, laying the groundwork for subsequent sections of this report. Fig. 17 shows the manufactured Hardware Fixation with Organized Circuitry and fig. 18 shows the overall setup connecting both arms together.

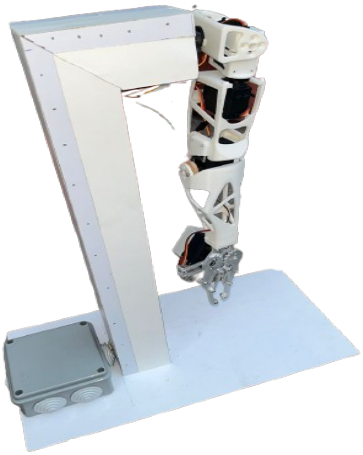

Figure 17: Manufactured Hardware Fixation with Organized Circuitry

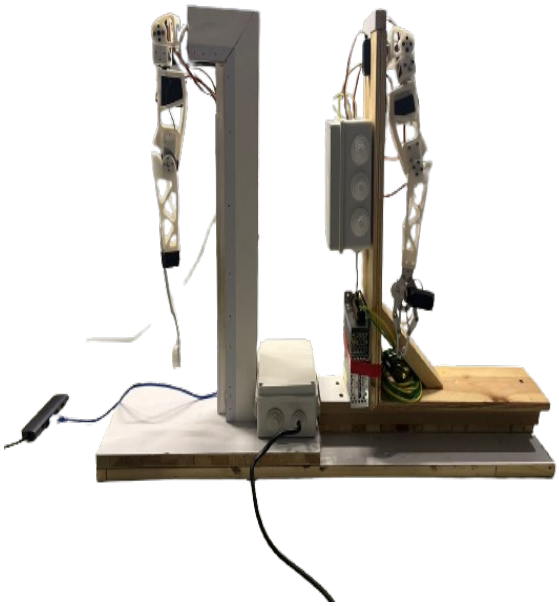

Figure 18: Overall setup connecting both arms together

The following figure 19 is a QR code that directs to a Simscape simulation validating the chosen circular trajectory for the teeth-cleaning robotic arm.

Figure 19: Simscape simulation QR Code

The following figure 20 is a QR code granting direct access to a video showcasing the fully fabricated hardware. The video captures the essence of the teeth-cleaning robotic arm with a circular trajectory, featuring the common stand shared by the two robots

Figure 20: Fully fabricated hardware

IV. Conclusions and Future Recommendations

In conclusion, our extensive exploration has afforded us valuable insights into the intricacies of the robotic arm project. The SimScape simulations served as a powerful tool, allowing us to observe and analyze diverse arm configurations, enriching our comprehension of its dynamic behavior. The construction and analysis of the Denavit-Hartenberg (DH) convention table have laid a robust foundation for understanding the arm's kinematic characteristics, providing a critical framework for subsequent developments. The identification of necessary components has not only demystified the theoretical aspects but has also illuminated the practical considerations inherent in constructing and operating the robotic arm.

The culmination of our efforts lies in the examination of the Computer-Aided Design (CAD) model, offering a tangible and visual representation of the arm's design. This holistic approach, incorporating simulation data, kinematic insights, practical considerations, and visual representation,

forms a comprehensive understanding of the robotic arm. By effectively bridging the gap between theory and practical implementation, this research serves as a significant stride towards the advancement of the broader field of robotics and automation.

V. Appendix

$$A = \begin{bmatrix} cos(q_3 - \pi/2) * cos(q_2 - \pi/2) * cos(q_0) * cos(q_1 - \pi/2) - cos(q_0) * sin(q_1 - \pi/2) \\ cos(q_1 - \pi/2) * sin(q_0) + sin(q_0) * sin(q_1 - \pi/2) \\ sin(q_3 - \pi/2) * cos(q_1 - \pi/2) - cos(q_3 - \pi/2) * cos(q_2 - \pi/2) * sin(q_1 - \pi/2) \\ 0 \end{bmatrix} \tag{13}$$

$$B = \begin{bmatrix} -sin(q_3 - \pi/2) * cos(q_2 - \pi/2) * cos(q_0) * cos(q_1 - \pi/2) - sin(q_2 - \pi/2) * sin(q_0) - cos(q_0) * sin(q_1 - \pi/2) \\ sin(q_0) * sin(q_1 - \pi/2) + cos(q_1 - \pi/2) * sin(q_0) + sin(q_2 - \pi/2) * cos(q_0) \\ sin(q_3 - \pi/2) * cos(q_2 - \pi/2) * sin(q_1 - \pi/2) + cos(q_3 - \pi/2) * cos(q_1 - \pi/2) \\ 0 \end{bmatrix} \tag{14}$$

$$C = \begin{bmatrix} sin(q_2 - \pi/2) * cos(q_0) * cos(q_1 - \pi/2) \\ sin(q_1 - \pi/2) * sin(q_0) - cos(q_2 - \pi/2) * cos(q_0) \\ sin(q_1 - \pi/2) * sin(q_2 - \pi/2) \\ 0 \end{bmatrix} \tag{15}$$

$$D = \begin{bmatrix} 149cos(q_0) * sin(q_1 - \pi/2)/1000 + 19cos(q_3 - \pi/2) * cos(q_0) * cos(q_1 - \pi/2) - sin(q_2 - \pi/2) - cos(q_0) * sin(q_1 - \pi/2) \\ cos(q_1 - \pi/2) * sin(q_0) - 149sin(q_0) * sin(q_1 - \pi/2)/1000 + sin(q_0) * sin(q_1 - \pi/2) \\ 19sin(q_3 - \pi/2) * cos(q_1 - \pi/2) - 19cos(q_3 - \pi/2) * cos(q_2 - \pi/2) * sin(q_1 - \pi/2) - 149cos(q_1 - \pi/2)/1000 \\ 1 \end{bmatrix} \tag{16}$$

$$E = \begin{bmatrix} 9.1e-18*cos(q0)+0.14*sin(q3-1.6)*(6.1e-17*cos(q0)+6.1e-17*cos(q2-1.6)* \\ (1.0*cos(q0)-3.7e-33*cos(q1-1.6)*cos(q0)+6.1e-17*sin(q1-1.6)*sin(q0)) \\ +6.1e-17*sin(q2-1.6)*(cos(q1-1.6)*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0)) \\ +6.1e-17*cos(q1-1.6)*cos(q0)-1.0*sin(q1-1.6)*sin(q0))+0.14*cos(q3-1.6)* \\ (1.0*sin(q2-1.6)*(1.0*cos(q0)-3.7e-33*cos(q1-1.6)*cos(q0)+6.1e-17*sin(q1-1.6)* \\ sin(q0))-1.0*cos(q2-1.6)*(cos(q1-1.6)*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0))) \\ +9.1e-18*cos(q1-1.6)*cos(q0)-0.15*sin(q1-1.6)*sin(q0), 0.15*cos(q1-1.6)* \\ cos(q0)-0.14*sin(q3-1.6)*(6.1e-17*cos(q2-1.6)*(6.1e-17*cos(q1-1.6)*cos(q0) \\ -3.7e-33*sin(q1-1.6)*sin(q0))-6.1e-17*sin(q2-1.6)*(6.1e-17*cos(q1-1.6)* \\ sin(q0)+1.0*sin(q1-1.6)*cos(q0))-1.0*cos(q1-1.6)*cos(q0)+6.1e-17* \\ sin(q1-1.6)*sin(q0))-0.14*cos(q3-1.6)*(cos(q2-1.6)*(6.1e-17*cos(q1-1.6) \\ *sin(q0)+1.0*sin(q1-1.6)*cos(q0))+1.0*sin(q2-1.6)*(6.1e-17*cos(q1-1.6)*cos(q0) \\ -3.7e-33*sin(q1-1.6)*sin(q0)))-9.1e-18*sin(q1-1.6)*sin(q0), 0.14*sin(q3-1.6) \\ *(6.1e-17*sin(q2-1.6)*(3.7e-33*cos(q1-1.6)*sin(q0)-1.0*sin(q0)+6.1e-17* \\ sin(q1-1.6)*cos(q0))-6.1e-17*cos(q2-1.6)*(cos(q1-1.6)*cos(q0)-6.1e-17* \\ sin(q1-1.6)*sin(q0)))-0.14*cos(q3-1.6)*(1.0*cos(q2-1.6)*(3.7e-33* \\ cos(q1-1.6)*sin(q0)-1.0*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0))+1.0*sin(q2-1.6)* \\ (cos(q1-1.6)*cos(q0)-6.1e-17*sin(q1-1.6)*sin(q0))), 0.14*sin(q3-1.6)* \\ (1.0*sin(q2-1.6)*(3.7e-33*cos(q1-1.6)*sin(q0)-1.0*sin(q0)+6.1e-17*sin(q1-1.6)* \\ cos(q0))-1.0*cos(q2-1.6)*(cos(q1-1.6)*cos(q0)-6.1e-17*sin(q1-1.6)*sin(q0))) \\ +0.14*cos(q3-1.6)*(6.1e-17*sin(q0)-6.1e-17*cos(q2-1.6)* \\ (3.7e-33*cos(q1-1.6)*sin(q0)-1.0*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0))- \\ 6.1e-17*sin(q2-1.6)*(cos(q1-1.6)*cos(q0)-6.1e-17*sin(q1-1.6)*sin(q0))+6.1e-17* \\ cos(q1-1.6)*sin(q0)+sin(q1-1.6)*cos(q0)) \end{bmatrix} \tag{17}$$

$$F = \begin{bmatrix} 9.1e-18*sin(q0)-0.14*cos(q3-1.6)*(1.0*sin(q2-1.6)*(3.7e-33*cos(q1-1.6)*sin(q0) \\ -1.0*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0))-cos(q2-1.6)*(cos(q1-1.6)*cos(q0) \\ -6.1e-17*sin(q1-1.6)*sin(q0)))+0.14*sin(q3-1.6)*(6.1e-17*sin(q0)-6.1e-17* \\ cos(q2-1.6)*(3.7e-33*cos(q1-1.6)*sin(q0)-1.0*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0)) \\ -6.1e-17*sin(q2-1.6)*(cos(q1-1.6)*cos(q0)-6.1e-17*sin(q1-1.6)*sin(q0))+6.1e-17* \\ cos(q1-1.6)*sin(q0)+1.0*sin(q1-1.6)*cos(q0))+9.1e-18*cos(q1-1.6)*sin(q0) \\ +0.15*sin(q1-1.6)*cos(q0), 0.15*cos(q1-1.6)*sin(q0)-0.14*sin(q3-1.6)* \\ (6.1e-17*cos(q2-1.6)*(6.1e-17*cos(q1-1.6)*sin(q0)+3.7e-33*sin(q1-1.6)*cos(q0)) \\ +6.1e-17*sin(q2-1.6)*(6.1e-17*cos(q1-1.6)*cos(q0)-1.0*sin(q1-1.6)*sin(q0))- \\ 1.0*cos(q1-1.6)*sin(q0)-6.1e-17*sin(q1-1.6)*cos(q0))+9.1e-18*sin(q1-1.6)*cos(q0) \\ +0.14*cos(q3-1.6)*(cos(q2-1.6)*(6.1e-17*cos(q1-1.6)*cos(q0)-1.0*sin(q1-1.6)*sin(q0)) \\ -1.0*sin(q2-1.6)*(6.1e-17*cos(q1-1.6)*sin(q0)+3.7e-33*sin(q1-1.6)*cos(q0))), \\ -0.14*cos(q3-1.6)*(1.0*sin(q2-1.6)*(cos(q1-1.6)*sin(q0)+6.1e-17*sin(q1-1.6)* \\ cos(q0))+1.0*cos(q2-1.6)*(cos(q0)-3.7e-33*cos(q1-1.6)*cos(q0)+6.1e-17*sin(q1-1.6)* \\ sin(q0)))-0.14*sin(q3-1.6)*(6.1e-17*cos(q2-1.6)*(cos(q1-1.6)*sin(q0) \\ +6.1e-17*sin(q1-1.6)*cos(q0))-6.1e-17*sin(q2-1.6)*(cos(q0)-3.7e-33 \\ *cos(q1-1.6)*cos(q0)+6.1e-17*sin(q1-1.6)*sin(q0))), -0.14*cos(q3-1.6)*(6.1e-17*cos(q0) \\ +6.1e-17*sin(q2-1.6)*(cos(q1-1.6)*sin(q0)+6.1e-17*sin(q1-1.6)* \\ cos(q0))+6.1e-17*cos(q2-1.6)*(cos(q0)-3.7e-33*cos(q1-1.6)*cos(q0)+6.1e-17* \\ sin(q1-1.6)*sin(q0))+6.1e-17*cos(q1-1.6)*cos(q0)-1.0*sin(q1-1.6)*sin(q0)) \\ -0.14*sin(q3-1.6)*(cos(q2-1.6)*(cos(q1-1.6)*sin(q0)+6.1e-17*sin(q1-1.6)*cos(q0)) \\ -1.0*sin(q2-1.6)*(cos(q0)-3.7e-33*cos(q1-1.6)*cos(q0)+6.1e-17*sin(q1-1.6)*sin(q0))) \end{bmatrix} \tag{18}$$

$$G = \begin{bmatrix} 0, 0.15 * sin(q1 - 1.6) + 0.14 * cos(q3 - 1.6) * (1.0 * cos(q1 - 1.6) * cos(q2 - 1.6) - 6.1e - 17 * \\ sin(q1 - 1.6) * sin(q2 - 1.6)) - 0.14 * sin(q3 - 1.6) * (6.1e - 17 * cos(q1 - 1.6) * \\ sin(q2 - 1.6) - 1.0 * sin(q1 - 1.6) + 3.7e - 33 * cos(q2 - 1.6) * sin(q1 - 1.6)), -0.14 * cos(q3 - 1.6) * \\ (1.0 * sin(q1 - 1.6) * sin(q2 - 1.6) - 1.0 * cos(q2 - 1.6) * (6.1e - 17 * cos(q1 - 1.6) + 6.1e - 17)) \\ -0.14 * sin(q3 - 1.6) * (6.1e - 17 * cos(q2 - 1.6) * sin(q1 - 1.6) + 6.1e - 17 * sin(q2 - 1.6) * \\ (6.1e - 17 * cos(q1 - 1.6) + 6.1e - 17)), 0.14 * cos(q3 - 1.6) * (cos(q1 - 1.6) + 6.1e - 17 * sin(q1 - 1.6) * \\ sin(q2 - 1.6) - 6.1e - 17 * cos(q2 - 1.6) * (6.1e - 17 * cos(q1 - 1.6) + 6.1e - 17) - 3.7e - 33) \\ -0.14 * sin(q3 - 1.6) * (cos(q2 - 1.6) * sin(q1 - 1.6) + sin(q2 - 1.6) * (6.1e - 17 * cos(q1 - 1.6) + 6.1e - 17)) \end{bmatrix} \tag{19}$$