

**Research Article****Design and Analysis of a Four DoF Robotic Arm with Two Grippers Used in Agricultural Operations****Basheer Altawil^{a,*} , Fatih Cemal Can^b** ^a*İzmir Katip Çelebi University, Robotics Engineering Department, 35620 İzmir, Türkiye*^b*İzmir Katip Çelebi University, Mechatronics Engineering Department, 35620 İzmir, Türkiye*

ARTICLE INFO

Article history:

Received 12 December 2022

Accepted 7 April 2023

Keywords:

ROS

IoT

Mechanism

Robot Manipulators

Agricultural Robots

Lagrangian mechanics

ABSTRACT

Both academic and commercial interest in agricultural robots has increased recently. This is due to the fact that agricultural robots address significant issues such as seasonal labor shortages during harvest and the rising concern over environmentally friendly practices. Because of these, several distinct agricultural robots have already been created for a variety of purposes, with varying degrees of success, including monitoring, spraying, harvesting, transport, etc. As a result, agriculture automation became unfeasible and unprofitable. The purpose of this study is to provide a new methodology for multitasking in performing agriculture operations by designing a 4 Degrees of freedom (4DoF) robotic arm with a new mechanism that has a different configuration with 2 grippers. We did kinematics and kinetics calculations using Denavit-Hartenberg (D-H) method with Lagrangian mechanics, and with the help of Christoffel Symbols of the First Kind, also We used Robot Operating System (ROS) to provide potential solutions using it. It is easy to be paired with other open-source technologies, such as android or IoT technologies. The robot arm can work synchronously with other hardware, sensors, cameras, and agricultural machines concerning farming operations. In conclusion, we believe that this new configuration will open a door for agricultural tasks to be easily automated and achieved using robotic technologies.

This is an open access article under the CC BY-SA 4.0 license.
(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Introduction

Agriculture-related tasks are now very expensive, time-consuming, and challenging to complete. Robot farmers are one of the cutting-edge technologies that will transform the agriculture industry and deal with its problems. Therefore, combining agricultural and technological instruments is now required to enhance the effectiveness and output of any agriculture system, making it easier for farmers to produce a large quantity of food that can satisfy market demands. As a result, by 2050, it is anticipated that the combined cost of agriculture and technology will be almost \$240 billion[1].

Agri-robotics can adapt to changes in farm scales, crop varieties, soil moisture content, growth patterns, greenhouse chambers (GHC), glasshouses, vertical farms, and hydroponic closed and open systems. Moreover, robotic technologies are taking a huge part in these

operations, especially, robotic arms that will be implemented on mobile robots, and it is becoming more difficult to manage all of these operations, and it's becoming more complex to cover all of the details with limited resources of robotics software and hardware[2]. As a result, using the Robotic Operating System (ROS) is preferred, as it can simplify all complex operations in one package. In recent years, research on agricultural robot arms has advanced quickly. Motion planning and control are vital to study areas for the collaborative operations of dual-arm robots [3].

The quick verification of novel prototypes, algorithms, and/or applications of motion in robotic sectors, as well as the optimization of system performance, have all been made possible by simulation approaches in the development of robotics. Without relying on a real complete hardware system, a robotic simulator can model the motion of robots and all objects in a virtual work

* Corresponding author. E-mail address: basheeraltawil@gmail.com
DOI: 10.18100/ijamec.1217072

envelope, saving time and money. This has lately led to an increase in interest in robotic simulation as it is a necessary case to design any robotic system, especially for agricultural operations purposes, see Figure 1 [4][5].

In our study, we used RVIZ, GAZEBO, and MOVEIT to mimic the motion of the robot arm concerning real environment task-performing. Path planning, real-time control, and all kinematic, dynamic solutions were all imitated using ROS. For path planning, the ROS framework was employed to avoid abrupt changes during fertilizing and harvesting operations. Using the SOLIDWORKS CAD application, a 4-DoF manipulator with 2 distinct end effectors was developed to harvest and manipulate the objects. Then, we installed the Unified Robotics Description Format (URDF) as a CAD plugin using SOLIDWORKS. This plugin assists in converting CAD-designed robots into a format that ROS can read and show on RVIZ and Gazebo in integration with MOVEIT.



Figure 1. Gazebo and RVIZ simulator

With our 4 degrees of freedom (4-DoF), see Figure 2, Figure 3, which can be easily integrated with a mobile robotic platform, it will be easy and simple to be able to implement and achieve the tasks. This paper will cover the design of the robot arm parts and carry on every analysis required for the motion and control of a robotic arm.

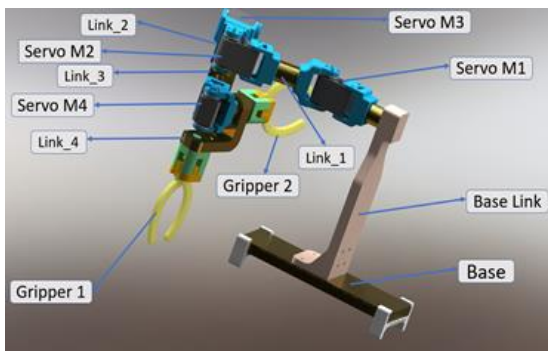


Figure 2. 4DoF Manipulator CAD Design

As a result, kinematic analysis, and kinetic analysis are completed to prepare the robot arm for ROS platform control and simulation. Numerous CAD programs can be used to create the articulated robot arms so we used both

SOLIDWORKS and Fusion360, 3D computer-aided design programs to prepare the parts that will be used later.

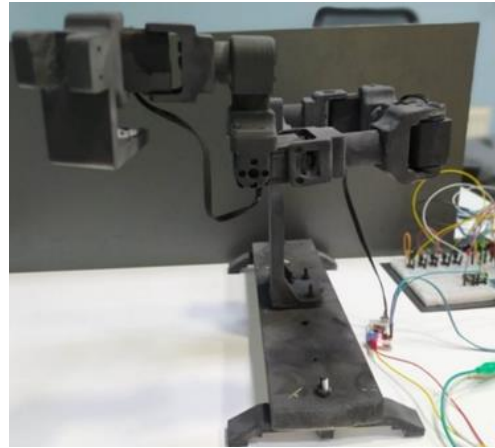


Figure 3. 4-DoF Manipulator Assembly

Analysis for robotics, particularly robotic manipulators, is extremely tough, and difficult. Therefore, many well-known scientists are striving to minimize the problems and make them more adaptable for those who are interested in robotics science. We used the four Denavit–Hartenberg parameters (D-H parameters) in Figure 4, which are used for attaching reference frames to the links of a spatial kinematic chain, or robot manipulator.

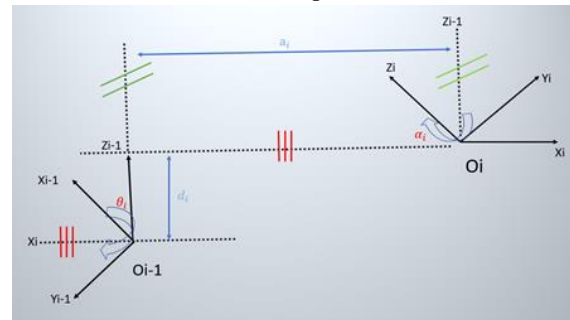


Figure 4. D-H Link Parameters

This convention was established in 1955 by Jacques Denavit and Richard Hartenberg to standardize the coordinate frames for spatial links [6].

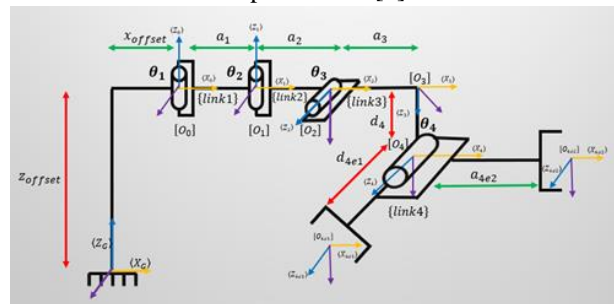


Figure 5. Robotic Arm Kinematic Labelling

The relationship between the joints that work as connecting parts and the links that make up the kinematic chain is determined by the kinematic modeling and

analysis of a robot manipulator [7] which basically depend on Denavit–Hardenberg parameters that we used in this paper.

After deciding the D-H parameters, arm kinematic labeling, see Figure 5 is presented in this paper which will help us to obtain homogenous transformation matrices to carry out forward kinematic (FW), inverse kinematic (IK), and Jacobian calculations that will be used to select the joint actuators and control the position and linear velocity of both first and second end effectors of the robot manipulator.

2. Kinematics and Kinetic Analysis

As previously noted, Denavit-Hardenberg was the method we employed for our analysis. This methodology comes in a variety of forms, which the scientist briefly discussed. The strategy we employed follows the Denavit-Hardenberg classical convention. As shown in Fig. 4, and Fig. 5, we follow this mythology to prepare DH parameters, and kinematic labeling respectively to obtain the DH table shown in Table 1 which will be necessary for the coming analysis.

Table 1. Denavit-Hardenberg classical Convention Table

Link, i	a_i	α_i	d_i	θ_i	Joint rotation limits (Degree)
1	a_1	0	0	θ_1	0-170
2	a_2	$-\frac{\pi}{2}$	0	θ_2	0-170
3	a_3	$-\frac{\pi}{2}$	0	θ_3	0-170
4	0	$-\frac{\pi}{2}$	d_4	0	Fixed
4e1	0	0	d_{4e1}	θ_4	0-170
4e2	a_{4e2}	0	0	θ_4	0-170

The set of parameters that are mentioned in table I and Figure 4, and Figure 5, twist angle α_i , offset distance a_i , translation distance d_i , and joint angle θ_i . For articulated mechanisms, some of these parameters are constant, and some of them are variable. 4e1, and 4e2 in our study, they are representing the end effector 1, and end effector 2 respectively.

2.1. Homogeneous Transformation Matrices and Kinematic Analysis.

To reach the formulation presented in (2), we must first construct homogeneous transformation matrices (HTM) by vectorially multiplying the four matrices by going on every singular axis respectively as shown in (1) that we got from each parameter in Table 1. The matrices for each unique frame have been created depending on rotation and translation principles [8]. To determine the last point in which we are interested, we can multiply them individually.

$${}^{i-1}T = Rot_{z(i-1)}(\theta_i) \cdot Trans_{z(i-1)}(d_i) \cdot Trans_{x(i)}(a_i) \cdot Rot_{x(i)}(\alpha_i) \tag{1}$$

$${}^{i-1}T = \begin{pmatrix} C(\theta_i) & -C(\alpha_i)S(\theta_i) & S(\alpha_i)S(\theta_i) & a_n C(\theta_i) \\ S(\theta_i) & C(\alpha_i)C(\theta_i) & -S(\alpha_i)C(\theta_i) & a_n S(\theta_i) \\ 0 & S(\alpha_i) & C(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

The parameters expressed in (2) are $C(\theta_i)$, $S(\theta_i)$, $S(\alpha_i)$, $C(\alpha_i)$ which represents the abbreviation of $\cos(\theta_i)$, $\sin(\theta_i)$, $\sin(\alpha_i)$, and $\cos(\alpha_i)$ respectively and they will be used with all formulations in all kinematic and dynamic analyses. From equations 1 and 2 we can relate all joints to the reference frame like in the following expressions. In (3), we obtain the transformation matrix that relates joint 1 to the base frame.

$${}^0_1T = \begin{pmatrix} C(\theta_1) & -S(\theta_1) & 0 & a_1 C(\theta_1) \\ S(\theta_1) & C(\theta_1) & 0 & a_1 S(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

In (4), we obtain the transformation matrix that relates joint 2 to the base frame. Moreover, the expression θ_{12} is representing the algebraic summation of θ_1, θ_2 .

$${}^0_2T = \begin{pmatrix} C(\theta_{12}) & 0 & -S(\theta_{12}) & a_1 C(\theta_1) + a_2 C(\theta_{12}) \\ S(\theta_{12}) & 0 & C(\theta_{12}) & a_1 S(\theta_1) + a_2 S(\theta_{12}) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$\left. \begin{aligned} \beta &= C(\theta_3)C(\theta_{12}) \\ \gamma &= S(\theta_3)C(\theta_{12}) \\ \delta &= C(\theta_3)S(\theta_{12}) \\ \varepsilon &= S(\theta_3)S(\theta_{12}) \\ P3_x &= a_1 C(\theta_1) + a_2 C(\theta_{12}) + a_3 \beta \\ P3_y &= a_1 S(\theta_1) + a_2 S(\theta_{12}) + a_3 \beta \\ P3_z &= -a_3 S(\theta_3) \\ P4_x &= a_1 C(\theta_1) + a_2 C(\theta_{12}) + a_3 \beta - d_4 \gamma \\ P4_y &= a_1 S(\theta_1) + a_2 S(\theta_{12}) + a_3 \beta - d_4 \varepsilon \\ P4_z &= -a_3 S(\theta_3) - d_4 C(\theta_3) \end{aligned} \right\} \text{Equations group 1}$$

In (5), we obtain the transformation matrix that relates joint 3 to the base frame. We generate new parameters which are $\beta, \gamma, \delta, \varepsilon, P3_x, P3_y, P3_z, P4_x, P4_y,$ and $P4_z$ that take expressions into them to minimize the size of the matrix as it is shown in equations group 1.

$${}^0_3T = \begin{pmatrix} \beta & S(\theta_{12}) & -\gamma & P3_x \\ \delta & -C(\theta_{12}) & -\gamma & P3_y \\ -S(\theta_3) & 0 & -C(\theta_3) & P3_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

In (6), we obtain the transformation matrix that relates

joint4 to the base frame and we substitute the new parameters generated in group 1 like in (5).

$${}^0_4T = \begin{pmatrix} \beta & \gamma & S(\theta_{12}) & P4_x \\ \beta & \epsilon & -C(\theta_{12}) & P4_y \\ -S(\theta_{34}) & C(\theta_{34}) & 0 & P4_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

In (7), we obtain the transformation matrix that relates end effector 1 to the base frame and we substitute the new parameters that we obtained in equation group 2, like in (5), and (6). Moreover, the expression θ_{34} is representing the algebraic substruction of θ_3, θ_4 .

$$\begin{aligned} \forall &= C(\theta_{12})C(\theta_{34}) \\ \rho &= C(\theta_{12})S(\theta_{34}) \\ \epsilon &= S(\theta_{12})C(\theta_{34}) \\ \sigma &= S(\theta_{12})S(\theta_{34}) \\ Pe1_x &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta - d_4\gamma + d_{4e1}S(\theta_{12}) \\ Pe1_y &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta - d_4S(\theta_3)S(\theta_{12}) - d_{4e1}C(\theta_{12}) \\ Pe1_z &= -a_3S(\theta_3) - d_4C(\theta_3) \\ Pe2_x &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta + a_{4e2}\beta - d_4\gamma \\ Pe2_y &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta + a_{4e2}\delta - d_4\epsilon \\ Pe2_z &= -a_3(\theta_3) - a_{4e2}S(\theta_3) - d_4C(\theta_3) \end{aligned} \quad \left. \vphantom{\begin{aligned} \forall &= C(\theta_{12})C(\theta_{34}) \\ \rho &= C(\theta_{12})S(\theta_{34}) \\ \epsilon &= S(\theta_{12})C(\theta_{34}) \\ \sigma &= S(\theta_{12})S(\theta_{34}) \\ Pe1_x &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta - d_4\gamma + d_{4e1}S(\theta_{12}) \\ Pe1_y &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta - d_4S(\theta_3)S(\theta_{12}) - d_{4e1}C(\theta_{12}) \\ Pe1_z &= -a_3S(\theta_3) - d_4C(\theta_3) \\ Pe2_x &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta + a_{4e2}\beta - d_4\gamma \\ Pe2_y &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta + a_{4e2}\delta - d_4\epsilon \\ Pe2_z &= -a_3(\theta_3) - a_{4e2}S(\theta_3) - d_4C(\theta_3) \end{aligned}} \right\} \text{Equations group 2}$$

In this formulation, we have new extra parameters which are like it shown in equation group 2.

$${}^0_{4e1}T = \begin{pmatrix} \forall & \rho & S(\theta_{12}) & Pe1_x \\ \epsilon & \sigma & -C(\theta_{12}) & Pe1_y \\ -S(\theta_{34}) & C(\theta_{34}) & 0 & Pe1_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

In (8), we obtain the transformation matrix that relates end effector 2 to the base frame and we substitute the new parameters that we prepared in equations group 2.

$${}^0_{4e2}T = \begin{pmatrix} \forall & \rho & S(\theta_{12}) & Pe2_x \\ \epsilon & \sigma & -C(\theta_{12}) & Pe2_y \\ -S(\theta_{34}) & C(\theta_{34}) & 1 & Pe2_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8)$$

A manipulator's forward kinematics determines the end effector's position and orientation in cartesian space using the input joint angles. The aforementioned matrix expresses the end-effector position, as can be seen by glancing at it. As shown in Matrix 7 and 8, we expressed the first-end effector and the second-end effector onto the base frame respectively[8]. Therefore, we can get the rotation and translation of them so that the forward kinematics analysis will be like the following. Equations group3, and group4, show the forward kinematics for the

first gripper, and second gripper respectively. Therefore, X1, Y1, Z1, and X2, Y2, Z2 are representing the position of gripper one and gripper two on space concerning the base frame.

2.1.1. Forward kinematics analysis for the first gripper

$$\begin{aligned} X1 &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta - d_4 + d_{4e1}S(\theta_{12}) \\ Y1 &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta - d_4S(\theta_3)S(\theta_{12}) - d_{4e1}C(\theta_{12}) \\ Z1 &= -a_3S(\theta_3) - d_4C(\theta_3) \end{aligned} \quad \left. \vphantom{\begin{aligned} X1 &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta - d_4 + d_{4e1}S(\theta_{12}) \\ Y1 &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta - d_4S(\theta_3)S(\theta_{12}) - d_{4e1}C(\theta_{12}) \\ Z1 &= -a_3S(\theta_3) - d_4C(\theta_3) \end{aligned}} \right\} \text{Equations group 3}$$

2.1.2. Forward kinematics analysis for the second gripper

$$\begin{aligned} X2 &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta + a_{4e2}\beta - d_4\gamma \\ Y2 &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta + a_{4e2}\delta - d_4\epsilon \\ Z2 &= -a_3(\theta_3) - a_{4e2}S(\theta_3) - d_4C(\theta_3) \end{aligned} \quad \left. \vphantom{\begin{aligned} X2 &= a_1C(\theta_1) + a_2C(\theta_{12}) + a_3\beta + a_{4e2}\beta - d_4\gamma \\ Y2 &= a_1S(\theta_1) + a_2S(\theta_{12}) + a_3\delta + a_{4e2}\delta - d_4\epsilon \\ Z2 &= -a_3(\theta_3) - a_{4e2}S(\theta_3) - d_4C(\theta_3) \end{aligned}} \right\} \text{Equations group 4}$$

2.2. Serial Manipulator Kinetic Analysis

Robotics kinetic calculations assess and explain a robot's motion and external forces using mathematical models and equations. This may involve computations in the fields of dynamics and kinematics, which both deal with the forces acting on the robot and how they affect its mobility. The robot's movement can be controlled, its behavior predicted, and its stability and safety can be guaranteed using these calculations. Inverse kinematics, forward kinematics, and motion planning are a few methods frequently utilized in dynamic computations for robotics [9]. In robotics, dynamic computations can be carried out in a variety of ways, including Lagrangian mechanics, Newton-Euler methods, Kane's method, Featherstone's algorithm, and Hybrid Dynamics. In this article calculations to analyze the 4-DoF serial robot manipulator, we used Lagrangian mechanics in combination with Christoffel Symbols of the First Kind to simplify the dynamical calculations.

Depending on the Lagrangian principle that aims to derive the governing general formulation of the dynamical equation of motion will be like it shown in (9).

$$\tau = M\ddot{\theta} + V + G, \tau = Q - J^T F_e + f_r \quad (9)$$

The first term is called the inertia forces, the second term is called the Coriolis and centrifugal force, and the third term is representing the gravitational effects forces Where $\tau, \ddot{\theta}, J^T F_e, f_r$ are representing joint torque, joint acceleration, Jacobian transpose, the force exerted on the joint, and friction force respectively. (10) is show the first term of (9) which is representing the inertia matrix.

$$M(\theta) = J_{v1}^T m_1 J_{v1} J_{v2}^T m_2 J_{v2} + J_{v3}^T m_3 J_{v3} J_{v4}^T m_4 J_{v4} + J_{w1}^T I_1 J_{w1} + J_{w2}^T I_2 J_{w2} + J_{w3}^T I_3 J_{w3} + J_{w4}^T I_4 J_{w4} \quad (10)$$

By implementing the calculations shown in (10), we could end up with the 4X4 matrix for the first term of the general formulation of the dynamic equation, see (11).

$$M_{ij}(\theta) = \begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{pmatrix} \quad (11)$$

Table 2. Christoffel Symbols c_{ijk} values Table

Column-1		
ROW1	C1R1	C111+C121+C131+C141
ROW2	C1R2	C112+C122+C132+C142
ROW3	C1R3	C113+C123+C133+C143
ROW4	C1R4	C114+C124+C134+C144
Column-2		
ROW1	C2R1	C211+C221+C231+C241
ROW2	C2R2	C212+C222+C232+C242
ROW3	C2R3	C213+C223+C233+C243
ROW4	C2R4	C214+C224+C234+C244
Column-3		
ROW1	C3R1	C311+C321+C331+C341
ROW2	C3R2	C312+C322+C332+C342
ROW3	C3R3	C313+C323+C333+C343
ROW4	C3R4	C314+C324+C334+C344
Column-4		
ROW1	C4R1	C411+C421+C431+C441
ROW2	C4R2	C412+C422+C432+C442
ROW3	C4R3	C413+C423+C433+C443
ROW4	C4R4	C414+C424+C434+C444

In addition, in the second term of (9) which is representing Coriolis, Centrifugal forces will be calculated the *Christoffel Symbols of the First Kind*, see (12), making the calculations simple and more accurate when we will use them in torque computation. Coriolis force.

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k} \right\} \quad (12)$$

This kind of force is generated in a reference frame that is rotating relative to an inertial frame. Moreover, a "fictitious" force known as the centrifugal force affects things traveling in a circular motion in a frame of reference that is not inertial.

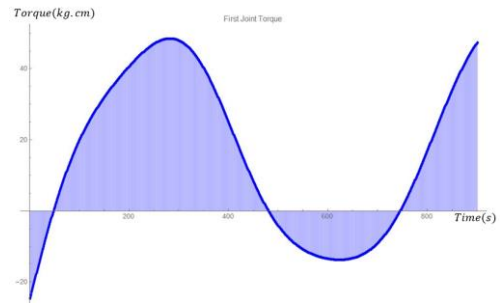


Figure 6. First Joint Torque

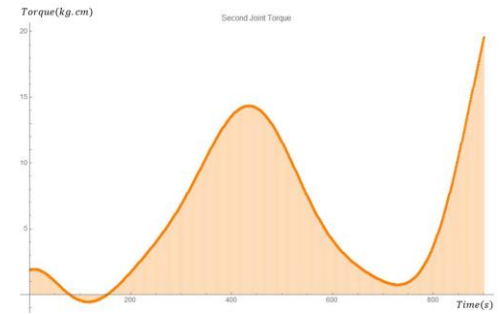


Figure 7. Second Joint Torque

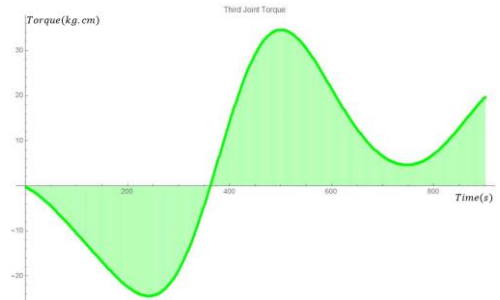


Figure 8. Third Joint Torque

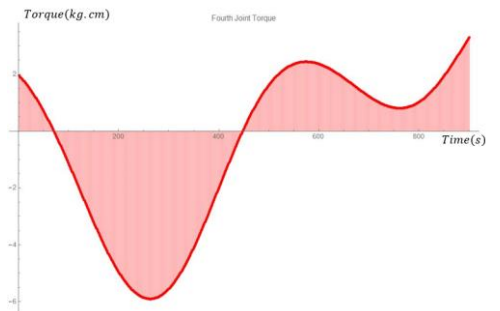


Figure 9. Fourth Joint Torque

c'_{ijk} s calculations totally depend on i, j, k factors. Therefore, the N elements of it calculated using the equation $(c_{ijk})_N = 4^3 = 64$ elements, because, we have three factors which are i, j, k , and we have 4 cases of each of them in total regarding robot arm joints. After using (12), we could generate 64 values for c_{ijk} , then organize them as shown in Table 2. After implementing the principles shown in Table 2, we could obtain the final 4X4 Coriolis, Centrifugal forces Matrix, see (13) that will be used in torque computation.

$$V_{ij}(\theta, \dot{\theta}) = \begin{pmatrix} V_{11} & V_{12} & V_{13} & V_{14} \\ V_{21} & V_{22} & V_{23} & V_{24} \\ V_{31} & V_{32} & V_{33} & V_{34} \\ V_{41} & V_{42} & V_{43} & V_{44} \end{pmatrix} \quad (13)$$

Also, we re-formalize the general mechanical equation to be suitable for final torque computation, see (14). This equation is the governing equation for torque computation, we sum two 4X4 matrices which are inertia M_{matrix} , and V_{matrix} which represent the inertia matrix, and Coriolis, Centrifugal forces Matrix respectively. The final result of torque will be the robotic arm links length, and masses in terms of θ , $\dot{\theta}$, and $\ddot{\theta}$, and the joint torques will be drawn regarding these variances concerning time-variant.

$$\sum_j m_{kj}(\theta)\ddot{\theta}_j + \sum_{i,j} c_{ikj}(\theta)\dot{\theta}_i \dot{\theta}_j + g_k(\theta) = \tau_k \quad (14)$$

$k = 1, 2, \dots, n$ and n is the number of robot arm joints. On the right-hand side, is the final torque of the joints. Finally, we used **Mathematica** platform to compute these calculations and draw the final torques of joint1, joint2, joint3, and joint4 as shown in Figure 6, Figure 7, Figure 8, and Figure 9, respectively. All codes can be found in our Github repository link under the name of **4-DoF Robotic Arm Analysis and Control**.

3. Manipulator Design and Implementation

This robot is an articulated manipulator with four degrees of freedom, as we mentioned in earlier sections, and revolute joints at every joint. Because the mechanism was made for testing purposes, the mechanical design was chosen based on the requirements that the mechanism may be used on an experiment table. Additionally, it was found that the restricted rotation angle for all joints depends on a range from 0 to 170[24]. Figure 3 shows the assembled side of the robot manipulator after 3D printing, whereas Figure 2 shows the CAD drawing of the developed mechanism designed using SOLIDWORKS.

Manipulator parts are made to be appropriate for prototyping and printing after being converted from CAD to extension STL[10]. The components are made to look like genuine robot arms to users so that we may address aesthetics, industrial design, and formality, just like an established robotics company. Thanks to the SW2URDF plugin for making it simple to convert the design to ROS and begin controlling it.

As shown in Figure 2 from the CAD design illustrates how there are two end effectors, one of which is a peristaltic pump that was purchased on the market and will be in charge of fertilizing the plant. The other device is a gripper with a micro servo motor that will be in charge of harvesting and gripping. The LX-16A servo motor from the HI-WONDER firm is used to power the robot arm's joints. The market has made the controllers available that use the bus servo controlling protocol. We created our connections to retain the servo motor and the links of the

manipulator due to the diverse configurations of the robot's mechanism.

We realized that the actuators attached to the first and third joints, respectively, are bearing large torque because they were chosen based on the joint torque that we determined during the dynamic study of the robot arm. The first joint is responsible for hauling all of the robot's components, and the third joint rotates around the X-axis while defying gravity to account for gravity's influence on the situation. However, as the first and second joints move perpendicular to gravity as part of the SCARA mechanism, we did not include the gravity factor[7].

4. Controlling and Software

The robot arm will move following the requirements of the plant and the plant's readiness based on the feedback we will gather from the system we are using. Because of this, the entire governing algorithm will operate as indicated in Figure 10. The inputs for the system's master can come from a variety of sources. The expert who visits the farm daily, weekly, or even monthly can make use of these resources. Additionally, it can be obtained through the sensors that were mounted on the system during installation.

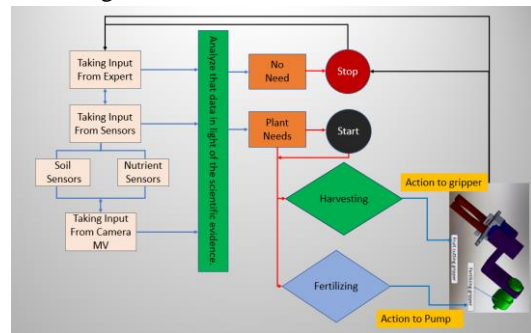


Figure 10. The Entire System Controlling Algorithm

These sensors may be vision sensors, such as cameras, or they may be digital, such as nutrient solution sensors or soil analyzer sensors. After receiving the inputs, we may examine them in the database we set up on the backend of our software frame and provide the instructions for the robot to follow to complete the required work [11].

4.1. Preparing ROS Environment and its operating system.

A robotic operating system (ROS) is a versatile and open-source software framework and we have been using it in this study. It is A collection of software libraries and development tools that are used to create robotic devices and programs. We used Melodic Moriena on Ubuntu 18.04 to create ROS packages for our robot arm. It only works with Ubuntu 18.04 Bionic Beaver when using ROS Melodic which we used in this project, and it is operating with the Linux operating system [12].

The robot arm kinematic Model has to be equivalent to

the real robot. Therefore, the robot model in ROS contains crucial packages and some important nodes that aid in creating the 3D robot models in the virtual frame. These packages employ the Unified Robot Description Format (URDF). So, the manipulator model is described in the URDF, which is an XML specification. To perform robot work, we have to prepare the inertia matrix, collision detection matrix, robot joints, and arms visualization, the transmission of joints actuators, reduction rate, actuators that will be used, gazebo plugin, sensor plugin, and required controllers that we may use during task implementation.

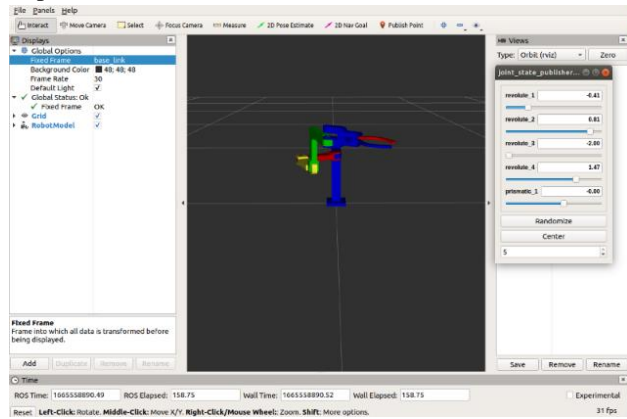


Figure 11. RVIZ with its joint publisher GUI

Following the preparation of the packages we previously discussed, as shown in Figure 11, we could launch the files using a Linux terminal and visualize our robotic arm to interact with it using RVIZ, GAZEBO, and MOVEIT interfaces. There are two packages to deal with to interact with the robot arm which are joint_state_publisher and robot_state_publisher. These packages are responsible for transforming the internal case of the joint of the robot arm so that we can move, stop, and visualize the status of every singular joint by helping these two packages.

4.2. Preparing the controlling packages and their algorithms.

As we have already indicated, there are already created packages available for editing, modifying, and improving. MOVEIT, which enables us to operate robotic arms and create the desired trajectories, is one of the most intriguing packages. The procedure that MOVEIT work depends on robot kinematics and dynamics KDL, and inverse kinematics of the robot arm so that by bringing the end effector of the robot arm to the desired position in different places, we can implement and repeat that action as much as we can, see Figure.12.

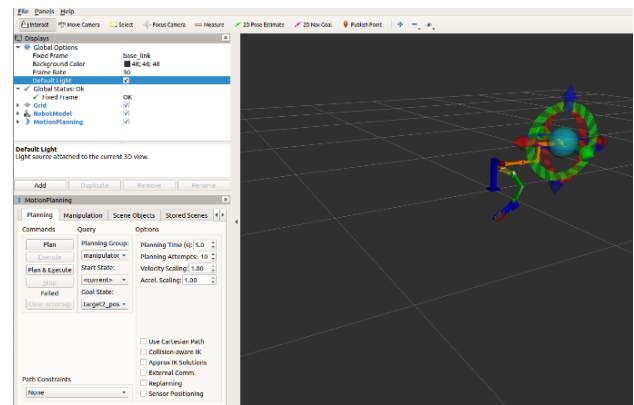


Figure 12. Path Planning with MOVEIT

In addition, we use *rosserial* protocol to enable the serial connection of the servo system of our robot arm for LX-16A servo motor and the extra hardware elements such as Arduino microcontroller to the ROS system.

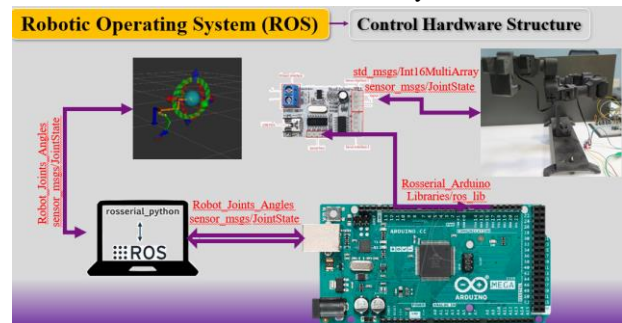


Figure 13. Bus Servo connecting Arduino with LX-16

Through the use of serial libraries, the type of activity (Subscriber/ Publisher or Service) and a message format are directly established on the device. Additionally, a Python node on the host computer relays messages and makes ROS aware of any nodes generated on the device. Devices can have any functionality as long as they communicate utilizing the serial connection and the *rosserial* format[13], see Figure.13.

4.3. Controlling the inputs coming from the agricultural system.

As we mentioned before, we will get some data from some sensors, experts, and/or cameras. These data will help us to decide where the robot arm should go during its work. Therefore, after collecting all these data, the robot should know which plant needs to be getting some operations such as harvesting and/or pesticide spraying as we mentioned before.

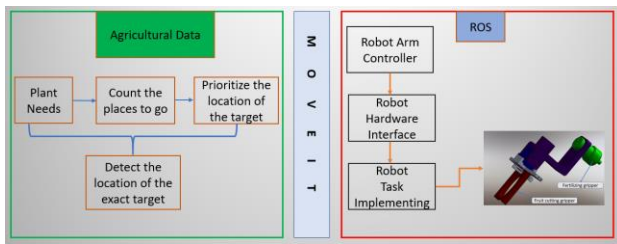


Figure 14. The internal control algorithm of the robot arm

However, firstly, we count the places where the robot should go and prioritize those places one by one depending on the readiness of the plant, the quantity of the plant that needs to be harvested, and the need for the plant that will be sprayed. After considering all the previous parameters, we can decide the exact location of every singular plant, and the MOVEIT group will give the command to the robot controller that will help the robot arm hardware and physical interface to move to the target to implement its task, see Figure 14.

5. Results and Discussion

In this study, we created an articulated robot arm with new configurations that will be better suited for use in agriculture and the execution of its tasks. We perform the mathematical calculations required for the modeling and control of the robot arm. Additionally, we created, produced, and modified the robot's components so that they could be used with ROS APPs. Additionally, we developed a system control method that will aid researchers and developers who wish to carry out further research in this area. By integrating an arm into a mobile robot, the robot will be able to gather information from the agricultural field and determine where it should move. We could not build the complete agricultural system because of the enormous expense needed, so we built just the robot arm using 3D printing and simulated it using ROS interfaces.

We used mathematical modeling that primarily relies on Denavit-Hartenberg principles, which enabled us to create a new labeling table from which we were able to derive the homogenous transformation matrices for the model we created. The serial robotics arms supported by this new model configuration will have fresh images that will aid researchers and developers in finding solutions to global problems.

For mechanical calculations, as can be seen from joint torque graphs Figure 6 through Figure 9, our mechanism's joints' torque varies from joint to joint, necessitating the employment of various motors with various torques. However, we employed the **LX-16A** servo motor in our investigation, which has a 15 kg.cm maximum torque. This caused joint 1 and joint 3 of the manipulators to have various deficiencies and vibrations during experimental work, which can help us choose new actuators. However,

because of the restricted funds available for the project, it was challenging to choose new, more expensive actuators when ROS would have been sufficient to complete and run the simulation.

6. Conclusion

Over the past ten years, there has been a substantial increase in research efforts aimed at creating agricultural robots that can efficiently complete laborious field jobs. A commercial level of robotics has not been attained for agricultural applications and researchers, developers, and robotics companies are intensively trying to commercialize robotics systems to become more helpful to users. Making an agriculture machine that can be integrated with ROS to be ready to be included in the agricultural system is difficult today, so the tasks where we must succeed as developers include developing its description model, ROS sensor drivers, and ROS controller interfaces with the required controllers.

That's why we design a new simple model with a new controlling algorithm that can enable developers to create and put into practice useful agriculture automation, which is another step toward increasing agriculture productivity and efficiency and enhancing food security for lighting future generations' dreams.

Acknowledgment

This work is supported by the Department of Robotics Engineering, Graduate School of Natural and Applied Sciences, İzmir Katip Çelebi University.

Author's Note

Previous version of this paper was presented at 10th International Conference on Advanced Technologies (ICAT'22), 25-27 November 2022, Van, Turkey.

References

- [1] Mitra, Manu. Robotic farmers in agriculture. *Advances in Robotics & Mechanical Engineering*.2019; pp. 91-93,doi: 10.32474/ARME.2019.01.000125.
- [2] Concepcion II, Ronnie S.,and et al. 'Denavit-Hartenberg-based Analytic Kinematics and Modeling of 6R Degrees of Freedom Robotic Arm for Smart Farming. *Journal of Computational Innovations and Engineering Applications* . 5.2 .2021 ; pp.1-7.
- [3] R Shamshiri, Redmond, Cornelia Weltzien, Ibrahim A. Hameed, Ian J Yule, Tony E Grift, Siva K. Balasundram, Lenka Pitonakova, Desa Ahmad, and Girish Chowdhary. Research and development in agricultural robotics: A perspective of digital farming.2018. doi: 10.25165/j.ijabe.20181104.4278.
- [4] Deng, H., Xiong, J., and Xia, Z. Mobile manipulation task simulation using ROS with MoveIt.IEEE International Conference on Real-time Computing and Robotics (*RCAR*) IEEE.2017; ;pp. 612-616. doi: 10.1109/rcar.2017.8311930.
- [5] Cruz Ulloa, C., Krus, A., Barrientos, A., Del Cerro, J., and Valero, C. Robotic fertilisation using localisation systems based on point clouds in strip-cropping fields. *Agronomy*. 11.1.2020 ; pp. 11. doi:10.3390/agronomy11010011.
- [6] Jha, Aparna, Manoj Soni, and Mohd Suhaib. *Simulation and kinematic analysis of KUKA KR5 Arc robot*. Vol. 1149. No. 1 . IOP Conference Series: Materials Science and Engineering.. IOP Publishing ; 2021.

- [7] A. Candemir, F. C. Can, and M. Engineering, "Pick & Place Task Implementation of a Scara Manipulator via Robot Operating System and Machine Vision," vol. 7, no. 6, pp. 420–430, 2022.
- [8] M. Brandstötter, A. Angerer, and M. Hofbauer, "An Analytical Solution of the Inverse Kinematics Problem of Industrial Serial Manipulators with an Ortho-parallel Basis and a Spherical Wrist," 2014. [Online]. Available: <https://www.researchgate.net/publication/264212870>.
- [9] Tsai, Lung-Wen. *Robot analysis : the mechanics of serial and parallel manipulators*, John Wiley and Sons ; 1999.
- [10] Poon, R. J. M. *Design and Control of a Mounted Robotic Arm Tool Changer and Measurement Tools for Agriculture* (Doctoral thesis), Massachusetts Institute of Technology). 2021; pp. 122. <https://hdl.handle.net/1721.1/139075>.
- [11] Ulloa, C. C., Krus, A., Barrientos, A., Del Cerro, J., and Valero, C. Trend Technologies for Robotic Fertilization Process in Row Crops. *Frontiers in Robotics and AI*. 2022; pp. 9. <https://doi.org/10.3389/frobt.2022.808484>.
- [12] J. M. O’Kane, A gentle introduction to ROS. Jason M. O’Kane, 2014.
- [13] Zubrycki, Igor, and Grzegorz Granosik. Introducing modern robotics with ros and Arduino, including case studies. *Journal of Automation Mobile Robotics and Intelligent Systems* 8.1 . 2014 ; pp. 69-75. doi:10.14313/jamris_1-2014/9.