DESIGN AND IMPLEMENTATION OF AN AUTOMATED ROBOTIC ARM USING IOT

Adhiban Siddarth. V,
Electronics and Instrumentation Engineering,
Arunai Engineering College,
Tiruvannamalai,
siddhu@gmail.com

Sathish. S,
Electronics and Instrumentation Engineering,
Arunai Engineering College,
Tiruvannamalai,
sathishsrnivasan22@gmail.com

Surya. M,
Electronics and Instrumentation Engineering,
Arunai Engineering College,
Tiruvannamalai,
suryagokul306@gmail.com

Project guide: Mr. V. Velmurugan
AP/EIE,
Electronics and Instrumentation Engineering,
Arunai Engineering College,
Tiruvannamalai,
velmurugan@arunai.org

***Abstract*— A robotic arm sometimes referred to as an industrial robot, is a device that runs in an equivalent way to a human arm, with several joints that either move along an axis or can rotate in certain directions. Some robotic arms are anthropomorphic and try and imitate the exact movements of huma arms. They are, in most cases programmable and used to perform specific tasks, most commonly for manufacturing, fabrication, and industrial applications. Robotic arms were originally designed to help in mass production factories, most famously in the manufacturing of cars. They were also implemented to mitigate the risk of injury for workers, undertake monotonous tasks, and free workers to concentrate on the more complex production elements.**

Keywords—6 DOF, Robotic arm, IoT, Servo motor (*key words****)***

I. INTRODUCTION

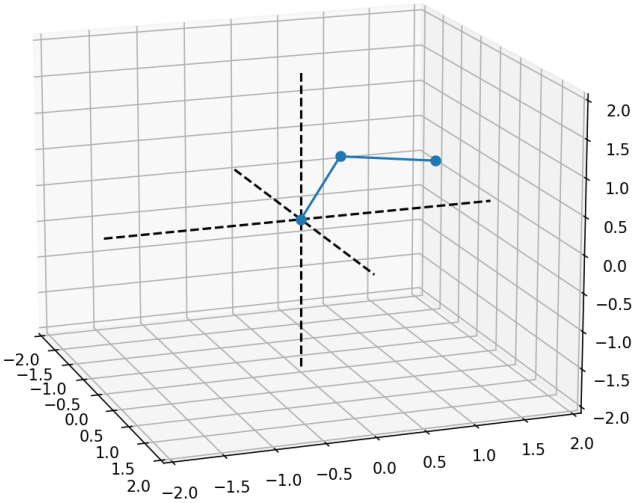
In today's world, human work is increasing, and become tough to manage a lot of tasks in less time, this is the reason robotics and automation come into usage which reduces the human effect. Robotics is the joint work of mechanical engineering, electrical engineering, and computer of science. A robot is a machine designed to perform a particular job coded with some programming language. The Robotic arm is a copy of the human arm that can do rotational motion and translation motion as the human arm based on IoT used for doing dangerous tasks such as Welding, Gripping, Spinning, etc. The Robotic arm types are a Cartesian robot, cylindrical robot, Spherical robot, Articulated robot, Parallel robot, SCARA robot. The number of degrees of freedom is equal to the total number of independent displacements or aspects of motion. A machine may work in two or three dimensions but have more than three degrees of freedom. The term is widely used to define the motion capabilities of robots. Our robotic arm consists of Six-DOF which increases the motion capabilities.

The main goal of this project is to control the Robotic Arm manually and automatically by using the ESP32 Microcontroller in IOT to pick, move & rotate the object by giving its 3D coordinates. In Industries highly advanced robots are used and controlled manually or processors like Arduino, Raspberry Pi, Microprocessors, etc. Here we are using C++ to program ESP32. This project focuses to create and build more compact, useful, and cheaper robotic arm to perform various functions where human is proven too dangerous to perform a specific task and to eliminate human errors to get more precise work and to do repeating tasks. This robotic arm can be controlled by using Buttons and IoT. We can control this robotic arm from a remote location wirelessly. Anyone can operate this robotic arm easily. It can work using battery power or else AC power.

Internet of things research often involves collecting network traffic sent or received by IoT devices, in [1] they used Arduino that has no built-in Wi-Fi, we are using Esp32, it has built-in Wi-Fi. Python script is employed in the Raspberry Pi to develop a program that will be able to control the robotic arm and to receive the commands from the smart phone [2], C++ is used to program ESP32, there are 4 DOF is used in [2], in our project we are using 6 DOF. A six degree of freedom robotic arm is the most widely used mechanical device in the field of robotics. It can work in the various complex environment and has the accuracy and precision that the human arm cannot achieve [3], they used ESP8266 microcontroller, but we are using ESP32, ESP32 has more features than ESP8266. The program was loaded into ESP32 hardware because of its integrated Wi-Fi and dual-mode Bluetooth [4], so we are using ESP32 for Wi-Fi, [4] is the robot used for writing on the paper, not pick and drop. [5] This also the 6 DOF Robotic Arm, but each motor is controlled individually, but our servo motors are working simultaneously. [6] is the pick and place robotic arm, it has 5 DOF, it uses Arduino, they used MATLAB. In [7] they used keypad to control the robotic arm, we referred to use keypad from here, but they used 8051 microprocessor and stepper motor. In [8] they used Wi-Fi to control the robotic arm, so we referred it from them, but they used PS2 controller. In [9] they also used servo motor to control the robotic arm, but they used Arduino UNO and Ethernet Shield to get internet connection. In [10] they built robotic arm with 6 DOF and using servo motor, but they don't use internet to control their robotic arm

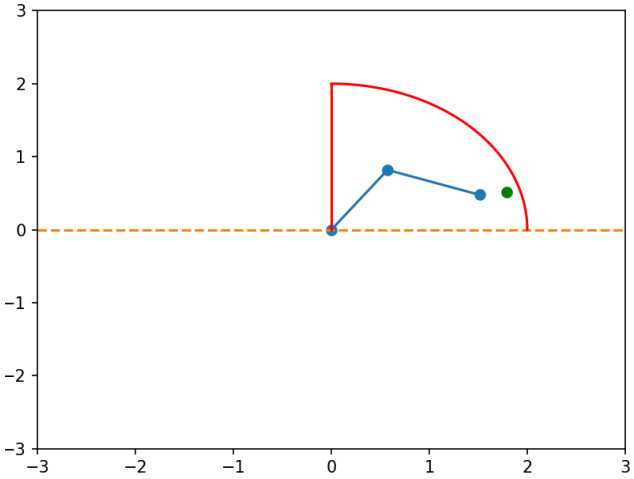
This paper first presents GRAPHICAL REPRESENTATION in section II. KINAMATIC MODEL of robotic arm is presented in III. WORKSPACE ANALYSIS is presented in IV. WORKFLOW is presented in V. CONCLUSION is presented in VI.

II. GRAPHICAL REPRESENTATION



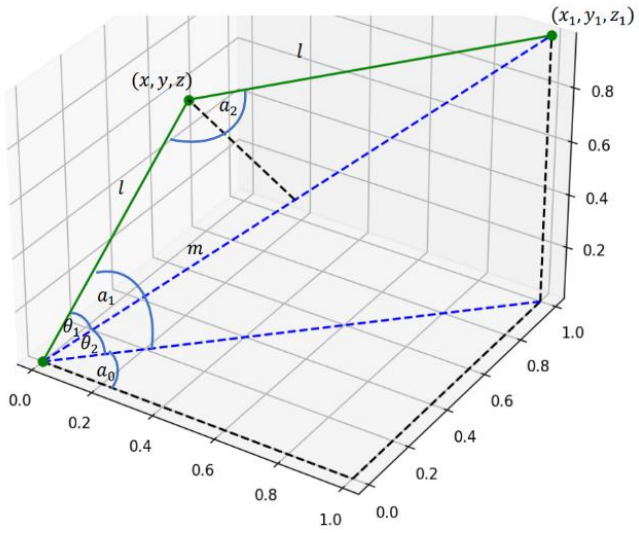
The Robotic arm 3D graph

The Robotic arm works in 3D space as per the Coordinates given to it. This 3D graph simulates the exact working of the Robotic arm.
This is six-axis Robotic-arm which has 6 servos to operate the robot arm in six-axis. The basement of the robot arm is assumed as the origin (0,0,0). The coordinates are given to the robotic arm concerning the origin (0,0,0). It is more precise to the coordinates.



The Robotic arm 2D graph

III. KINAMATIC MODEL



d_1 is the distance between (0,0,0) and (1,1,0)

$$d_1 = \sqrt{x_1^2 + y_1^2}$$

$$a_0 = \cos^{-1}\left(\frac{x_1}{d_1}\right)$$

d_2 is the distance between (0,0,0) and (1,1,1)

$$d_2 = \sqrt{x_1^2 + y_1^2 + z_1^2}$$

$$m = \frac{d_2}{2}$$

$$\theta_1 = \cos^{-1}\left(\frac{m}{l}\right)$$

$$\theta_2 = \cos^{-1}\left(\frac{d_1}{d_2}\right)$$

$$a_1 = \theta_1 + \theta_2$$

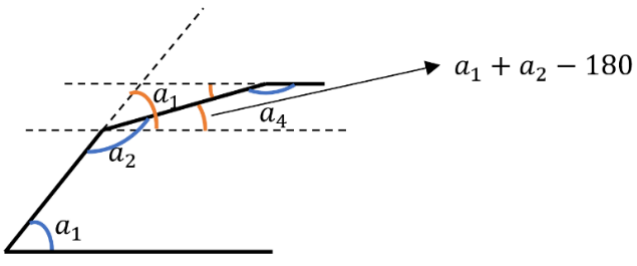
$$a_2 = \pi - 2 * \theta_1$$

$$\boldsymbol{a_0 = cos^{-1}\left(\frac{x_1}{\sqrt{x_1^2+y_1^2}}\right)}$$

$$\boldsymbol{a_1 = cos^{-1}\left(\frac{\sqrt{x_1^2+y_1^2+z_1^2}}{2*l}\right) + cos^{-1}\left(\frac{\sqrt{x_1^2+y_1^2}}{\sqrt{x_1^2+y_1^2+z_1^2}}\right)}$$

$$\boldsymbol{a_2 = \pi - 2 * cos^{-1}\left(\frac{\sqrt{x_1^2+y_1^2+z_1^2}}{2*l}\right)}$$

Here $a_0, a_1, and\ a_2$ are the angles of servo motors in the Robotic arm. a_0 is the angle of the servo motor which is placed in the basement of the Robotic arm, this servo rotates the robotic arm along the Z-axis at the range of 0 to 180 angles (anti-clockwise).



$\because Here,\ 2\pi\ radian = 360\ degree$

$\pi\ radian = 180\ degree$

$a_4 = \pi - (a_1 + a_2 - \pi)$

$a_4 = 2\pi - a_1 - a_2$

For our project, the wrist angle is calibrated for our convenience by removing $\pi/2$ in a_4.

$a_4 = 2\pi - a_1 - a_2 - \pi/2$

$$\boldsymbol{a_4 = 3\pi/2 - a_1 - a_2}$$

This a_4 angle is used to keep the wrist parallel to the ground surface for all the movement of the robotic arm. The Robotic arm moves anywhere in its range and the wrist is always parallel to the ground surface. We can change the angle of the Robotic Arm's wrist at any angle in between the manual control or in automation.

The Robotic arm can reach any point within its range using the above equations. But the robotic arm needs a smooth move from one point to another point. So, the points in between the two points are used to move the robotic arm smoothly.

We must find points in between two points.

Assume $(x_1, y_1, z_1)\ and\ (x_2, y_2, z_2)$ are the two points.

$\frac{x-x_1}{x_2-x_1} = \frac{y-y_1}{y_2-y_1} = \frac{z-z_1}{z_2-z_1} \ldots (1)$

(1) is the Equation of straight line in 3D.

$(1) \Rightarrow y = (y_2 - y_1)\frac{x-x_1}{x_2-x_1} + y_1 \ldots (2)$

$(1) \Rightarrow z = (z_2 - z_1)\frac{x-x_1}{x_2-x_1} + z_1 \ldots (3)$

Now, we take 10 to 100 numbers between x_1 and x_2. Apply these numbers in (2) and (3).

Then we can get the points in between $(x_1, y_1, z_1)\ and\ (x_2, y_2, z_2)$.

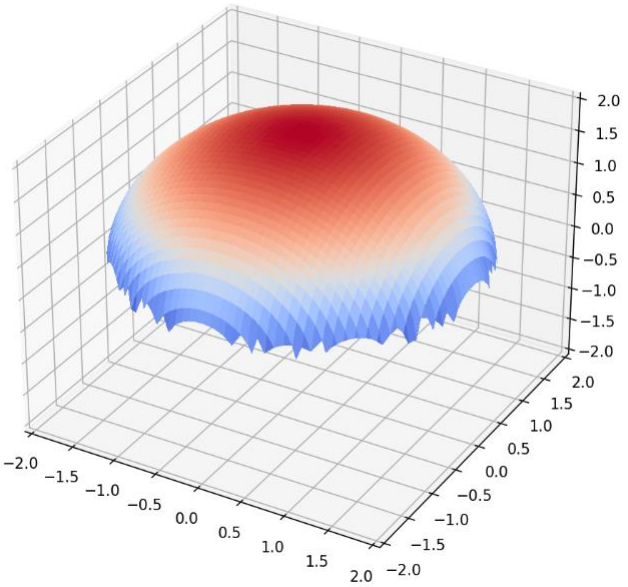
When $x_1 = x_2$, we cannot find numbers between x_1 and x_2.

Then replace x instead of y and y instead of x in (2) and (3).

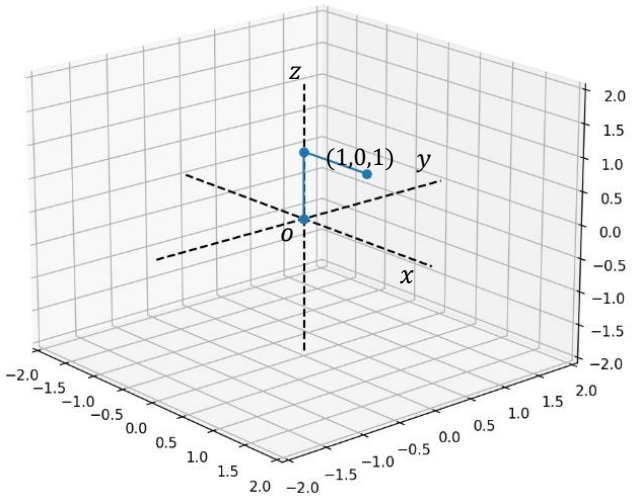
When $y_1 = y_2$, we cannot find numbers between y_1 and y_2.

Then replace y instead of z and z instead of y in the earlier two equations.

IV. WORKSPACE ANALYSIS



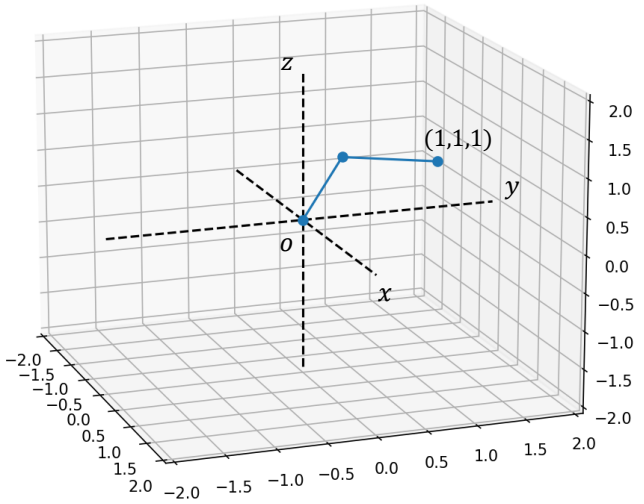
Equation of sphere: $x^2 + y^2 + z^2 = r^2$
r value is the radius of the sphere. Here $r = 2.l$.
$\Rightarrow x^2 + y^2 + z^2 = (2.l)^2$
$\Rightarrow x^2 + y^2 + z^2 = 4.l^2$
The robotic arm range is within this sphere above the XY plane.
Considering $(x_1, y_1, z_1) = (1,0,1)$ the position and orientation of the end-effector will be shown below.



Here we consider $l = 1$. The angles $a_0,\ a_1$ & a_2 will be 0.0, 1.570, 1.570 in radian. Then these angles are converted into degrees as 0, 90, and 90 degrees. The return value of the acos() in C++ will be radian only. Here acos means cos^{-1}.

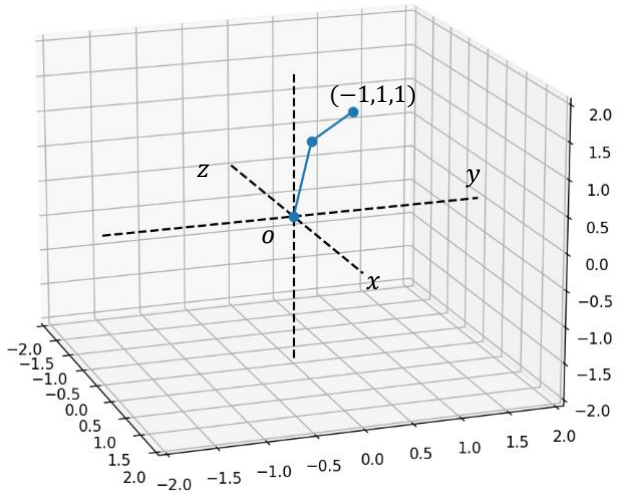
```
a0 = acos(x1 / d1);
```

The value given to the servos should be in degrees, so we should convert the radian into degrees. If we consider $(x_1, y_1, z_1) = (2,2,2)$ the result will be invalid, because (2,2,2) is outside the sphere $x^2 + y^2 + z^2 = 4$.
Considering $(x_1, y_1, z_1) = (1,1,1)$ the position and orientation of the end-effector will be shown below.



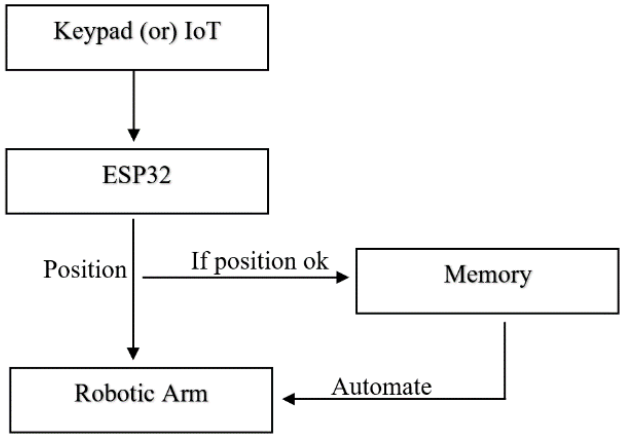
The angles a_0, a_1 & a_2 will be 0.785, 1.139, 2.094 in radian, 45, 65.26, 119.99 degrees.

Considering $(x_1, y_1, z_1) = (-1,1,1)$ the position and orientation of the end-effector will be shown in below.



The angles a_0, a_1 & a_2 will be 2.356, 1.139, 2.094 in radian, 135.0, 65.264, 119.999 degrees.

V. WORKFLOW



We give input to ESP32 using Keypad or IoT as changing the 3D Coordinate position. Then ESP32 converts the input into the position and then positions into the angles for the servo motors of the robotic arm. Then by pressing the 'Record' button we can store the desired positions in the ESP32. After all the positions are stored, we should press the 'Automate' button to automate the Robotic arm. Then Robotic arm moves to the stored positions repeatedly in a continuous loop. Pause the automated Robotic arm, we should press the 'Pause' button. After that, we should press the 'Run' button to resume the automation. Again, we press the 'Record' button to erase the stored positions, and then we can record a new position.

VI. CONCLUSION

The mathematical Inverse kinematics equation of our model gives the correct value of the angles for our robotic arm. The robotic arm is working as per the getting angles for the servo motors. End-effector of the robotic arm moves the desired location. When the record button is pressed the coordinate position is successfully recorded. When the automation button is pressed the robotic arm's end-effecter is moving its correct coordinates as per the recorded coordinate positions. The robotic arm is successfully working in IoT. The robotic arm can be controlled by Mobile or Computer using IoT, the ESP32 successfully connected to the internet and getting data from the cloud service provider, when the position is given through IoT the coordinates value is successfully changed in the cloud and ESP32 is getting the changed values of variables in IoT Cloud. Then those values are given to the robotic arm and robotic arm moves our desired position successfully. The recording of the robotic arm positions and the automation of the robotic arm through IoT works successfully.

VII. REFERENCE

[1] Xi Jiang, Noah Apthorpe, "Automating Internet of Things Network Traffic Collection with Robotic Arm Interactions", Networking and Internet Architecture (cs.NI); Cryptography and Security (cs.CR); Robotics (cs.RO), 30 Sep 2021, doi:2110.00060.

[2] M. K. Ishak and N. M. Kit, "Design and Implementation of Robot Assisted Surgery Based on Internet of Things (IoT)," International Conference on Advanced Computing and Applications (ACOMP), 2017, pp. 65-70, doi: 10.1109/ACOMP.2017.20.

[3] S. Fu and P. C. Bhavsar, "Robotic Arm Control Based on Internet of Things," 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2019, pp. 1-6, doi: 10.1109/LISAT.2019.8817333.

[4] Victor H.Benitez, RodrigoSymonds, David E.Elguezabal, "Design of an affordable IoT open-source robot arm for online teaching of robotics courses during the pandemic contingency" , HardwareX, Volume 8, October 2020, e00158, doi: 10.1016/j.ohx.2020.e00158.

[5] N. K. Agrawal, V. K. Singh, V. S. Parmar, V. K. Sharma, D. Singh and M. Agrawal, "Design and Development of IoT based Robotic Arm by using Arduino," Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 776-780, doi: 10.1109/ICCMC48092.2020.ICCMC-000144.

[6] K. Jahnavi and P. Sivraj, "Teaching and learning robotic arm model," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017, pp. 1570-1575, doi: 10.1109/ICICICT1.2017.8342804.

[7] Jegede Olawale, Awodele Oludele, Ajayi Ayodele, and Ndong Miko Alejandro Babcock University, Ilisan- Remo, Nigeria, "Development of a Microcontroller Based Robotic Arm", January 2007.

[8] Mohd Ashiq KamarilYusoff, Reza EzuanSamin, Babul Salam KaderIbrahim, "Wireless Mobile Robotic Arm", Volume 41, 2012, doi.org/10.1016/j.proeng.2012.07.285.

[9] Wan Muhamad Hanif WanKadir, Reza EzuanSamin, Babul Salam KaderIbrahim, "Internet Controlled Robotic Arm", 2012, doi.org/10.1016/j.proeng.2012.07.284.

[10] Jamshed Iqbal, Raza ul Islam, and Hamza Khan, "Modeling and Analysis of a 6 DOF Robotic Arm Manipulator", Canadian Journal on Electrical and Electronics Engineering Vol. 3, No. 6, July 2012.

[11] https://www.arduino.cc/reference/en/

[12] https://esp32io.com/tutorials/esp32-hello-world

[13] https://en.wikibooks.org/wiki/C%2B%2B_Language

[14] https://docs.platformio.org/en/stable/tutorials/espressif32/arduino_debugging_unit_testing.html