An-Najah National University

Faculty of Engineering and Information Technology

Electrical and Telecommunication Department

Graduation Project 2

Design and Wireless Control of a 7-Axis Robotic Arm

Hashem Alshami 11926161

Baraah abualreesh 11822790

Supervisor : Dr. Omar Tamimi

Contents

Dedication

Abstract

Contents

2.3 Implementation of the D-H algorithm

List of figures

Disclaimer Statement

Students from An-Najah National University's Faculty of Engineering and IT's Telecommunications Engineering Department wrote this study. It may have grammatical and content mistakes because it has not been amended or corrected, save from editorial adjustments, as a consequence of the assessment. The opinions expressed in it, along with any conclusions and advice, are all purely the students. An-Najah National University disclaims all liability and responsibility for the results of using this study for purposes other than those for which it was commissioned

Abstract:

Abstract: This project employs a central Arduino-based controller and a specialized set of drivers to enable the development, assembly, and manipulation of a 6-axis robotic arm. The arm design includes the addition of a modified claw clutch operated by a high-performance servo motor. Although operating in an open-loop mode, the system is fitted with limit switches on each axis to guarantee both safety and accuracy. Additionally, each axis is equipped with a stepper motor that is linked to a gear system, so facilitating seamless movement. The main aim of this project is to conceptualize and construct a mechanically robust robotic arm capable of seamless control through the utilization of a wireless joystick. The purpose of this arm is to perform precise and fluid movements, facilitating effective manipulation and relocation of objects.

Chapter 1

Introduction

1.1 Overview

The robotic arm is a programmable mechanical arm that works similar to a human arm. Robotic arms played a significant role in the process of industrial automation. The human-like dexterity of these robotic arms makes them efficacious in diverse applications in a variety of industries - manufacturing, atomic power plants, space exploration, material handling, painting, drilling, agriculture deployments,assistive robotics applications and numerous other applications. The robotic arm typically comprises an end effector that is designed to manipulate and govern with the surroundings. The 7 axis robot arm is designed to manipulate and govern with the surroundings. The 6 Axis is to pivot in 7 different ways that mimic a human arm. The major issues concerned in an industrial robotic arm are its mechanical structure and the control mechanism. The control mechanisms can be effectuated by 3 options: keyboard, joystick and slider-based control. Design of a lightweight robotic arm which can be compatible with any kind of robotic system. In the research, all the proffered control mechanisms adopted inverse kinematics, which makes it easier to control. The proposed control mechanisms are compatible with any other complex robotic systems of the same degrees of freedom. The dexterity of the robotic manipulator depends on the degree of freedom

1. 2 PROPOSED METHODOLOGY

A robotic arm sometimes referred as to as an industrial robot is often describe as a mechanical arm it is device that operation in a similar way to a human arm with number joint that either move along an axis as can rotate in certain direction.Robotics is the branch of technology that deal with design construction operation structural disposition manufactures & application of robot.A robotic arm is a robotic manipulator usually programmable with similar function to a human arm. Robotics is related to the sciences of electronics engineering mechanics and software. A project model should be beneficial to the society & reduce this problem. The basic methodology of praposed methodology is explain below

F. Axis 1 This axis allows the rotation of wrist in angle of 240. It powering the movment of the lower arm

B. Axis 2 This axis allows the lower arm of the robot to extend forward and backward. It is the axis powering the movement of the entire lower arm.

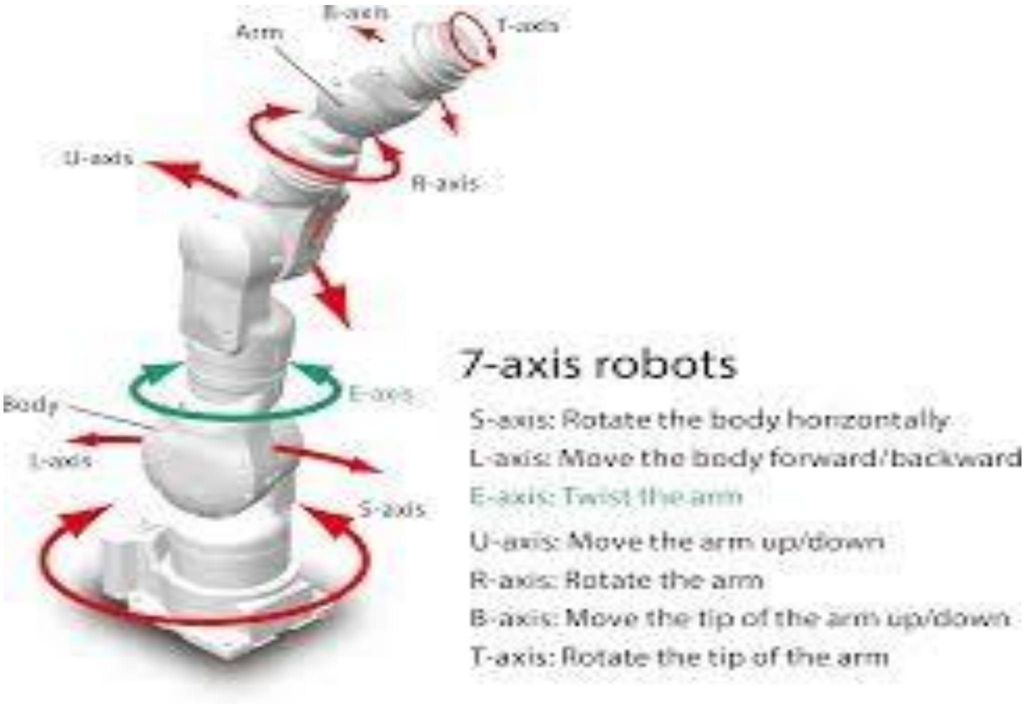
C. Axis 3 The axis extends the robot's vertical reach. It allows the upper arm to raise and lower. On some articulated models, it allows the upper arm to reach behind the body, further expanding the work envelope. This axis gives the upper arm the better part access.

D. Axis4 Working in conjunction with the axis 5, this axis aids in the positioning of the end effector and manipulation of the part. Known as the wrist roll, it rotates the upper arm in a circular motion moving parts between horizontal to vertical orientations.

E. Axis 5 This axis allows the wrist of the robot arm to tilt up and down. This axis is responsible for the pitch and yaw motion. The pitch, or bend, motion is up and down, much like opening and closing a box lid. Yaw moves left and right, like a door on hinges

F. Axis 6 This is the wrist of the robot arm. It is responsible for a twisting motion, allowing it to rotate freely in a circular motion, both to position end effectors and to manipulate parts. It is usually capable of more than a 360 degree rotation in either a clockwise or counterclockwise direction

H.Axis7 Tool or Gripper Rotation This axis rotates the tool or gripper around its own axis It enables fine adjustments in the orientation of the tool or gripper relative to the end effector, ensuring optimal positioning for tasks such as assembly or machining.



7-axis robots

S-axis: Rotate the body horizontally
L-axis: Move the body forward/backward
E-axis: Twist the arm
U-axis: Move the arm up/down
R-axis: Rotate the arm
B-axis: Move the tip of the arm up/down
T-axis: Rotate the tip of the arm

1.3 BLOCK DIAGRAM AND WORKING

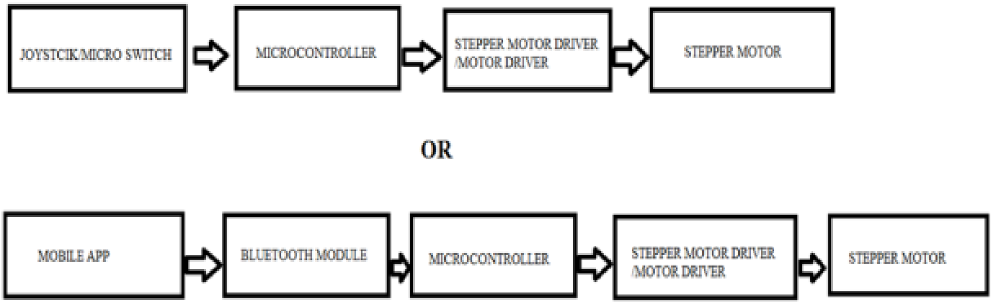


Fig. 2 Block Diagram of ARM

1.4 Literature Review:

Robots have their verifiable past however they came into presences just in 1961 when Unimation Inc, USA presented the principal servo-controlled new robots. Early advancement tracing back to 500B.C shows that the Egyptians, Indians, the Chinese, and the Romans fabricated numerous automatics manikins that emulate the development of creatures and birds. The Chinese fabricated many interesting gadgets that portrayed successive movements. Additionally, the early men found numerous components and showed their advancement expertise in building ships and acquainted weavers weave. This leds to the new revolution. During the 1940s, far-off transported ace slave controllers were created. Afterward, power criticism and sensation tangible components were added to them to work with better control. Tele-worked gadgets were utilized in damages Investigation in 1976. In 1948, the progress was created at Bell labs U.S.A. IBM"s first commercial PC IBM 701 was presented to world in 1952. At that point came mathematically controlled apparatuses in which different slides of machines were uprooted by mathematical orders through reasonable equipment. The improvement NC (mathematically controlled) machine apparatuses has, along these lines, been a defining moment in the advancement of mechanical technology.

The planet partnership in 1959 presented a pick and spot robot. In 1961, the primary mechanical robot was marketed by Unimation Inc. Chip innovation was brought by INTEL In 1961. The genuine robot advancement measure proceeded somewhere in the range of 1968 and 1982 when different models of robots were created by driving robot researchers in various colleges, public labs and distinctive mechanical houses in the USA, Japan, France, UK, and other European countries. A portion of the robot models of authentic interest are the Versatran by AMF, created in (1963) and Cincinnati Milacron presented in (1974), Irb-6 by ASLA in (1978). The

Kawasaki and Hitachi bunches in Japan have likewise contributed a ton in creating different sensors to make robots think" intensely. [1]

Above was just the general background of evolution of robots. Coming back to our topic, we will now discuss the methods adopted by different researchers to achieve their goals.

1.5 Literature Review 7-DOF Robotic Arms

The development of 7-DOF (degrees of freedom) robotic arms represents a significant advancement in robotic manipulation, providing increased flexibility and precision compared to traditional robotic arms with fewer degrees of freedom. These robotic arms have found applications in various fields, including manufacturing, healthcare, research, and personal assistance. This literature review examines the design, control methods, applications, and future trends of 7-DOF robotic arms.

Design and Architecture

7-DOF robotic arms are designed to mimic the range of motion of a human arm, offering a high level of dexterity and adaptability. The additional degrees of freedom allow for more complex and nuanced movements, making them suitable for tasks requiring precision and flexibility.

- **Mechanical Design**: The UFACTORY xArm 7 is a prime example, featuring a lightweight construction using carbon fiber and aluminum, which reduces its weight by at least 50% compared to traditional industrial robots . This design enhances mobility and ease of use, making the arm suitable for a variety of applications.

- **Actuation and Sensors**: Advanced robotic arms incorporate sophisticated actuation mechanisms, such as harmonic drives and brushless motors, to achieve precise movements. Sensors, including force-torque sensors and encoders, provide feedback for accurate control and adaptability.

Control Methods

The control of 7-DOF robotic arms is a complex task due to the need to manage multiple joints and ensure coordinated movements. Two main categories of controllers are commonly used:

1. • **Model-Based Controllers**: These controllers rely on mathematical models of the robot's dynamics and kinematics. They are designed to handle the uncertainties and variations in the robotic system by using adaptive algorithms. Examples include adaptive filtering and admittance control, which help in achieving smooth and precise movements .
2. • **Model-Free Controllers**: These controllers do not require detailed models of the robot and instead use learning-based approaches. Techniques such as reinforcement learning and iterative learning control have been applied to develop controllers that can adapt to different tasks and environments without extensive pre-programming .

Applications

The versatility of 7-DOF robotic arms allows them to be used in various applications:

1. **Industrial Automation**: These robotic arms are widely used in manufacturing for tasks such as assembly, welding, and material handling. Their flexibility enables them to handle complex and varied tasks efficiently).
2. **Healthcare and Surgery**: In the medical field, 7-DOF robotic arms are used in minimally invasive surgeries, providing surgeons with enhanced precision and control. They are also used in rehabilitation to assist patients with mobility exercises .
3. **Research and Development**: In research, these robotic arms are employed in experiments requiring high precision and repeatability. They are also used

4. in developing advanced human-robot interaction systems and artificial intelligence applications

Service Robotics: 7-DOF robotic arms are increasingly being used in service applications, such as personal assistance robots and customer service robots, where their ability to perform complex tasks is highly valued

Future Trends

The future of 7-DOF robotic arms is promising, with several trends expected to drive further advancements:

1. **Integration with AI**: The integration of artificial intelligence and machine learning will enable robotic arms to become more autonomous and capable of learning new tasks without extensive reprogramming.
2. **Enhanced Human-Robot Collaboration**: Future developments will focus on improving safety and interaction, allowing humans and robots to work together more seamlessly and efficiently.
3. **Miniaturization and Portability**: Advances in materials and manufacturing technologies will lead to more compact and portable robotic arms, expanding their applications in various fields.
4. **Increased Accessibility**: As technology advances and costs decrease, 7-DOF robotic arms will become more accessible to small and medium-sized enterprises and individual users, broadening their impact across different sectors

1.6 A Review on Robot Arm Using Haptic Technology

Robots are assuming a vital part in our everyday life. The primary thought behind putting forth any robot is to decrease human attempt. Contingent on the prerequisites, various kinds of robotized and self-controlled robots are being carried out. As numerous robots which are as of now existing in the market experiencing controlling issues and to beat this one of the arising innovation named Haptic Technology is talked about in this paper. This paper additionally examines a methodology for show of a robot arm for genuinely incapacitated individuals who can't walk. For those individuals who can only with significant effort lift straightforward things (for example lifting a glass of water). So the improvement of a Robot Arm which is constrained by the remote medium is the subject of the paper and to make this done haptic innovation will assume a vital part all things considered. Basically, a robotic arm is divided into two sections one is haptic glove and the other is automaton arm. The region where robot arm moves, relies on the scope of Zig-honey bee module and robot arm is constrained by the haptic glove. So by keeping all Imperative and constraints this paper will talk about a survey on the execution of a minimal effort robot arm utilizing Haptic innovation which plays out different errands utilizing less number of assets and it is not difficult to utilize.

1.7 Current situation on the Market

The current status of 7-DOF (degrees of freedom) robotic arms in the market shows significant advancements in their design, functionality, and applications. Here are some notable aspects:

1. **Design and Flexibility**: The UFACTORY xArm 7 is a prominent example of a 7-DOF robotic arm, designed for high flexibility and precision. It features a lightweight construction made from carbon fiber and aluminum, reducing its weight by at least 50% compared to traditional industrial robots. This

design ensures human-like flexibility and ease of movement ().

2. **Applications**: These robotic arms are used in various fields, including AI and computer vision research, machine tending, packing, sorting, and even complex operations such as filming and precise object manipulation. The xArm 7, for instance, can handle a payload of up to 3.5 kg with a repeatability of ±0.1 mm, making it suitable for tasks requiring high accuracy .

3.**Control and Programming**: Advances in control systems have made 7-DOF robotic arms more user-friendly and versatile. The xArm 7 can be programmed without code using its GUI platform, xArm Studio. This platform supports multiple operating systems and includes features like manual teaching, where the arm learns movements by being guided by hand . Furthermore, both model-based and model-free controllers are being developed and refined to improve the performance and adaptability of these robots in various environments .

4.**Market Availability and Support**: The UFACTORY xArm 7 is available for purchase at around $10,000 to $11,500, depending on the specific configuration and optional add-ons like grippers or vacuum systems. It comes with support options such as free training, a one-year warranty, and local support in some regions .

Chapter 2

Control using matlab

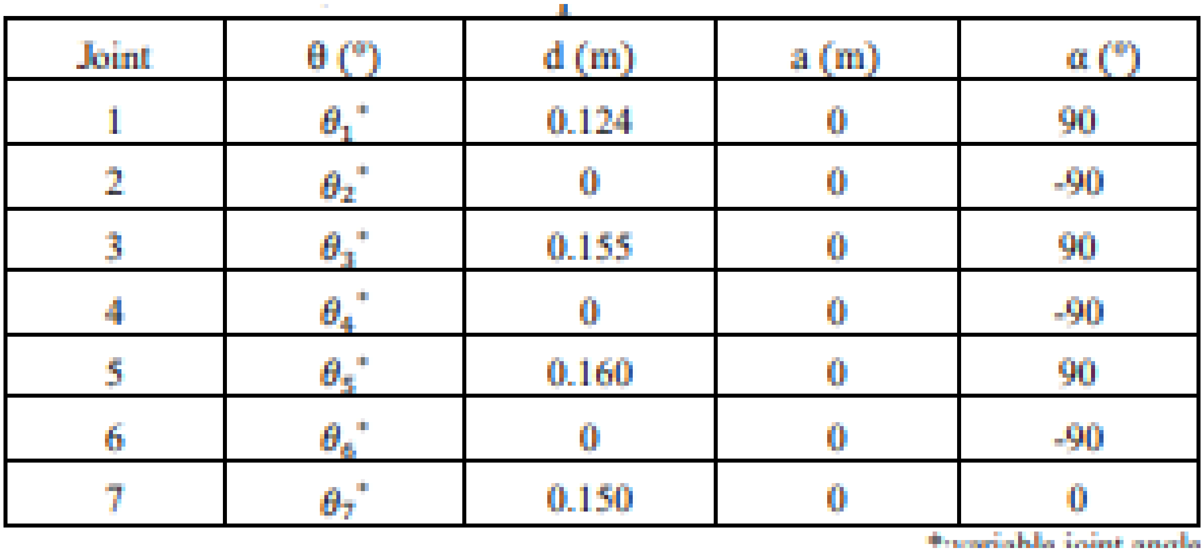
2.1 INTRODUCTION

For the control simulation of a robotic arm, most studies focus on the implementation of a single axis. In other words, the control method is simulated based on the dynamics of one axis, ignoring the coupling effect from other axes and the timevariant of parameters.To improve performance and correctness of control simulation of the multi-axis system, this paper uses MATLAB robotics toolbox to simulate control of 7-DOF manipulator based on sliding model control. The Robotics Toolbox is a software package that allows a MATLAB user to readily create and manipulate datatypes fundamental to robotics such as homogeneous transformations, quaternions and trajectories. Functions provided, for arbitrary serial-link manipulators, include forward and inverse kinematics, Jacobians, and forward and inverse dynamics

2.2 METHOD

The toolbox provides a serial-link manipulator object based on the D-H algorithm, so it is quite handy to implement kinematics [2] of the manipulator. Firstly, after acquiring the D-H parameters, the Link function provided by the toolbox can be used to build a link object that contains all information of a single axis, then the Serial-Link function combines all link objects to create a 7-DOF robotics arm object which is the basis for the subsequent simulation. In this case, the D-H parameters are given in table 1.

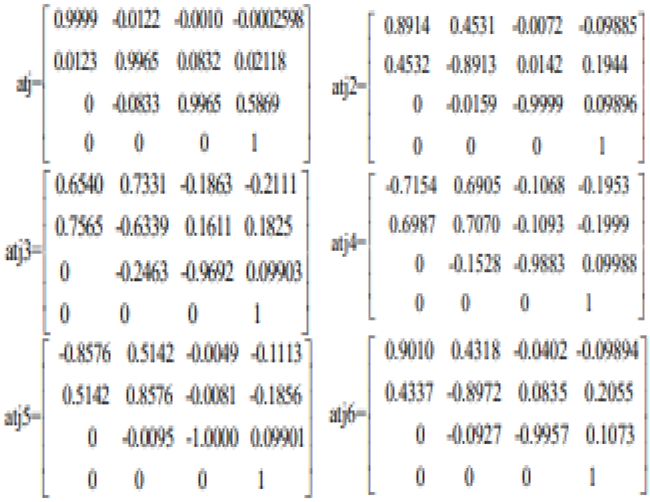
Table 1. 7-Link D-H parameter table



Joint	θ (°)	d (m)	a (m)	α (°)
1	θ_1^*	0.124	0	90
2	θ_2^*	0	0	-90
3	θ_3^*	0.155	0	90
4	θ_4^*	0	0	-90
5	θ_5^*	0.160	0	90
6	θ_6^*	0	0	-90
7	θ_7^*	0.150	0	0

*variable joint angle

In the toolbox, several functions are given to solving inverse kinematics problem of arbitrary serial-link manipulators. Briefly, the algorithms it used can be divided into two classes, the Jacobian iteration method and the optimization toolbox method using the libraries that provide solvers to nonlinear optimization problems and developed by MATLAB. , the Jacobian method [4] ,[5] is utilized for getting the inverse kinematics result of trajectory points. Thereby, the correspondence between the joint space and the operation space is obtained,



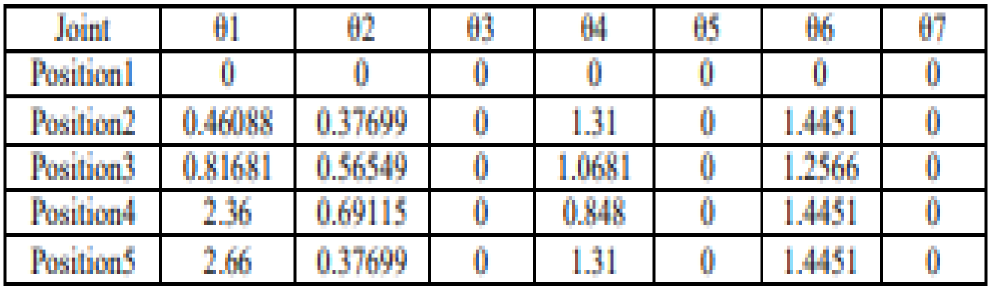
$$atj=\begin{bmatrix} 0.9999 & -0.0122 & -0.0010 & -0.0002598 \\ 0.0123 & 0.9965 & 0.0832 & 0.02118 \\ 0 & -0.0833 & 0.9965 & 0.5869 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad atj2=\begin{bmatrix} 0.8914 & 0.4531 & -0.0072 & -0.09885 \\ 0.4532 & -0.8913 & 0.0142 & 0.1944 \\ 0 & -0.0159 & -0.9999 & 0.09896 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$atj3=\begin{bmatrix} 0.6540 & 0.7331 & -0.1863 & -0.2111 \\ 0.7565 & -0.6339 & 0.1611 & 0.1825 \\ 0 & -0.2463 & -0.9692 & 0.09903 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad atj4=\begin{bmatrix} -0.7154 & 0.6905 & -0.1068 & -0.1953 \\ 0.6987 & 0.7070 & -0.1093 & -0.1999 \\ 0 & -0.1528 & -0.9883 & 0.09988 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$atj5=\begin{bmatrix} -0.8576 & 0.5142 & -0.0049 & -0.1113 \\ 0.5142 & 0.8576 & -0.0081 & -0.1856 \\ 0 & -0.0095 & -1.0000 & 0.09901 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad atj6=\begin{bmatrix} 0.9010 & 0.4318 & -0.0402 & -0.09894 \\ 0.4337 & -0.8972 & 0.0835 & 0.2055 \\ 0 & -0.0927 & -0.9957 & 0.1073 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where, atj is position and orientation of end-effector in operation space. The next step is to decide some trajectory points or middle points where the robot should definitely arrive. Because the toolbox has the advantage of easily visualizing the state of the robot, some certain points including orientation and position information of the end-effector can be got directly from the rendered graph. The chosen points are shown as homogeneous transformation matrices in atj and the inverse kinematics results are in table 2

Table 2. 7 Joint space table



Joint	θ1	θ2	θ3	θ4	θ5	θ6	θ7
Position1	0	0	0	0	0	0	0
Position2	0.46088	0.37699	0	1.31	0	1.4451	0
Position3	0.81681	0.56549	0	1.0681	0	1.2566	0
Position4	2.36	0.69115	0	0.848	0	1.4451	0
Position5	2.66	0.37699	0	1.31	0	1.4451	0

when these points are determined, the path between the two points should be explicit [6] in this experiment, A quintic polynomial is used with default zero boundary conditions for velocity and acceleration with 0.01s sampling time

$$\theta(t)=a0+a1t+a2t+a3t+a4t+a5t \qquad (1)$$

A 5-order polynomial has 6 coefficients so he can satisfy the six constraints given by the expression six constraints:

$\theta0=a0 \qquad \theta f=a1t_f+a2t_f+a3t_f+a4t_f+a5t_f$

Where, θ0 , θ f Constraints between the starting point and the final point position

$\theta = at \qquad \theta f=a1+2a2t_f+3a3t_f+4a4t_f + 5a5tf$

$\theta = 0 \qquad \theta f=0$

Where, θ 0 andθ f Constraints between the starting point and the final point accelerated velocity, default zero boundary conditions, These constraints determine the solution of six equations and six unknown linear equations

$$a_0 = \theta_0 \quad (5) \qquad a_1 = \dot{\theta}_0 \quad (6) \qquad a_2 = \frac{\theta_0}{2} \quad (7)$$

$$a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)\mathrm{t}_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)\mathrm{t}_f^{\,2}}{2t_f^{\,3}} \quad (8)$$

$$a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)\mathrm{t}_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)\mathrm{t}_f^{\,2}}{2t_f^{\,4}} \quad (9)$$

$$a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)\mathrm{t}_f - (\ddot{\theta}_0 - \ddot{\theta}_f)\mathrm{t}_f^{\,2}}{2t_f^{\,5}} \quad (10)$$

Where, 0 a , 1 a , 2 a , 3 a , 4 a , 5 a , 6 a ,Eq.(5)-Eq.(10) are solutions of 6 constraints and put them in Eq.(1) can find the fifth-order polynomial parameter of any joint angular position to the desired end position

2.3.Implementation of the D-H algorithm

At this stage it is proposed to simulate the mathematical model in Matlab, the programming codes establish to replace the dimensions of the links , and thus obtain a numerical result when validating the prototype of the mathematical model in . When simulating the mathematical model in Matlab we will obtain the coordinates of the final effector based on the joints configurations where its workspace is defined. The manipulators were also simulated in an industrial environment using the CoppeliaSim software. This simulation will be done to achieve the results of four tests (avoiding obstacles, different orientations, velocity, acceleration, and productivity

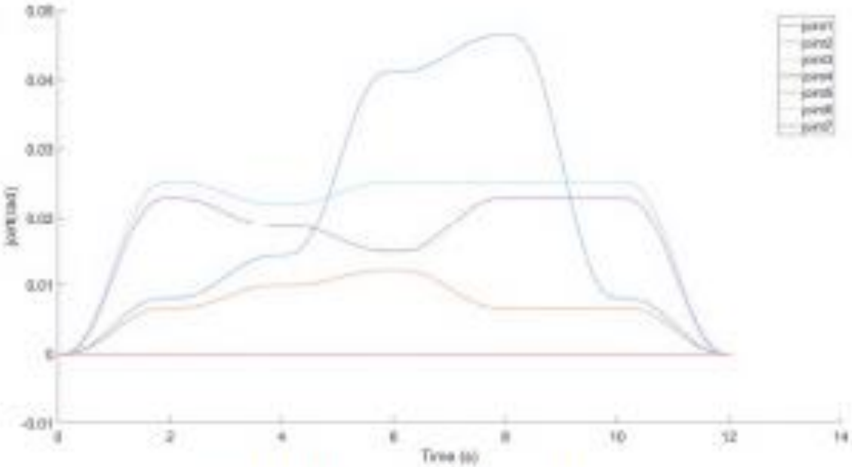


Fig.1. Designed Joint Variable

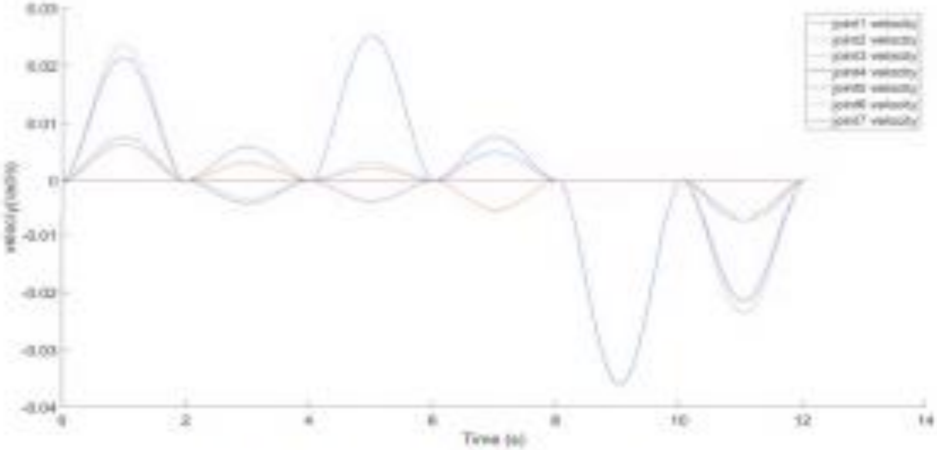


Fig.2. Designed Velocity Variable

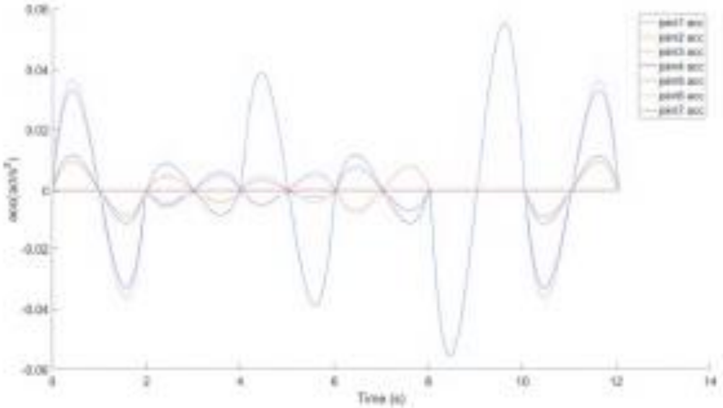


Fig.3. Designed Acceleration Variable

By solving the equation to obtain the coefficients, numerous transition points between middle points can be calculated with 0.01s sampling time. As shown in fig.1, fig.2 and fig.3, the designed joint angle, velocity and acceleration have the characteristic of smoothness and continuity, and the whole trajectory is shown in fig.4

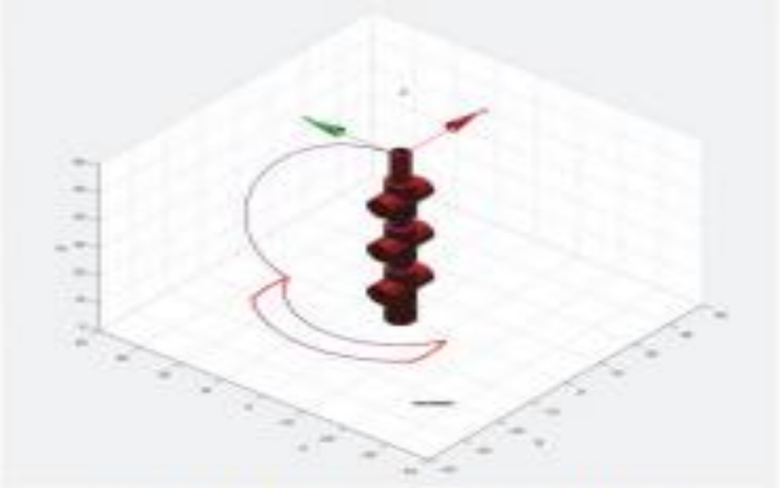
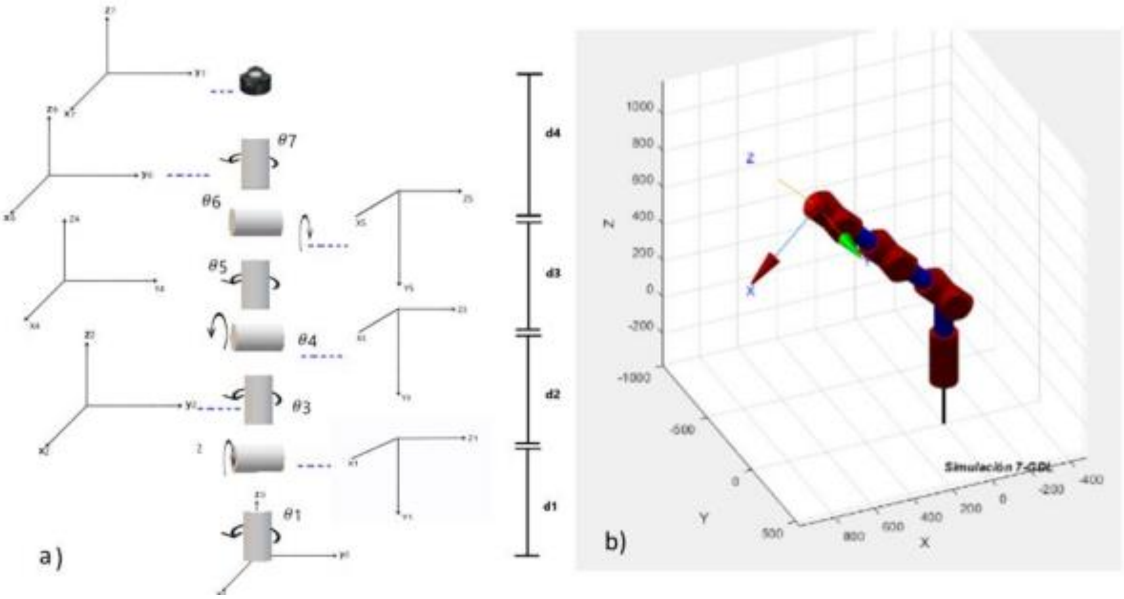


Fig.4. Trajectory of Simulation



Chapter3:

Component and Programming

3.1 Fabrication of Robotic Arm

As simulations of proposed Robotic arm is showing promising results, the next step is to go ahead and fabricate it. For implementing the actual design, we need several gadgets or tools helpful for making the design successful.

Required Components

1-Stepper Motors

2-driver

3-Controller (Arduino mage2560)

4-Mechanical Claw (Gripping tool)

5-Links (brackets)

6-Strong Bottom Plate

7-HC-05 (Bluetooth Module)

8-Bearing

9-Screws and Nuts

10-Power Adapter

11-PCB

12-Headers

13-Jumper wires

stepper motor

A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotation.Its normal shaft motion consists of discrete angular movements of essentially uniform magnitude when driven from sequentially switched DC power supply.

The stepper motor is a digital input-output device. It is particularly well suited to the type of application where control signals appear as digital pulses rather than analog voltages. One digital pulse to a stepper motor drive or translator causes the motor to increment one precise angle of motion. As the digital pulses increase in frequency, the step movement changes into continuous rotatio
Some industrial and scientific applications of stepper motors include robotics,

stepper motor work

Every revolution of the stepper motor is divided into a discrete number of steps, in many cases 200 steps, and the motor must be sent a separate pulse for each step. The stepper motor can only take one step at a time and each step is the same size.

Since each pulse causes the motor to rotate a precise angle, typically 1.8°, the motor's position can be controlled without any feedback mechanism. As the digital pulses increase in frequency, the step movement changes into continuous rotation, with the speed of rotation directly proportional to the frequency of the pulses.

a Stepper Motor Works in a 7-DOF Robotic Arm

Stepper motors are commonly used in robotic arms due to their precision, reliability, and ability to control angular position without the need for feedback systems. Here is an overview of how stepper motors work and their role in a 7-DOF robotic arm.

Stepper Motor Basics

A stepper motor moves in discrete steps, allowing for precise control of the motor's position. This is achieved through the following components and principles:

1. **Electromagnets**: Inside a stepper motor, there are multiple electromagnets arranged around a central rotor. These electromagnets are activated in a specific sequence to rotate the rotor in steps.
2. **Rotor**: The central part of the motor that rotates. It is typically made of a magnetic material or has permanent magnets attached to it.
3. **Stator**: The stationary part of the motor that surrounds the rotor. It contains the electromagnets.
4. **Driver Circuit**: Controls the current to the electromagnets, dictating the sequence and timing of the steps.
5. **Control Signals**: Stepper motors receive step and direction signals from a controller (like a microcontroller or a dedicated robot controller). The controller sends pulses, where each pulse corresponds to a single step of the motor.

Operation in a 7-DOF Robotic Arm

In a 7-DOF robotic arm, stepper motors are used to control the movement of each joint. Here's how they are integrated and operate:

1. **Joint Movement**: Each joint in the robotic arm is equipped with a stepper motor. The motors are responsible for the precise angular positioning of the joints, allowing for controlled and repeatable movements. For example:
    - **Base Rotation (Axis 1)**: A stepper motor rotates the base of the arm, allowing it to turn left or right.
    - **Shoulder Joint (Axis 2)**: A stepper motor moves the shoulder up and down.
    - **Elbow Joint (Axis 3)**: Another stepper motor bends and straightens the elbow.
    - **Wrist Pitch, Yaw, and Roll (Axes 4, 5, and 6)**: Stepper motors control the up-and-down, left-and-right, and rotational movements of

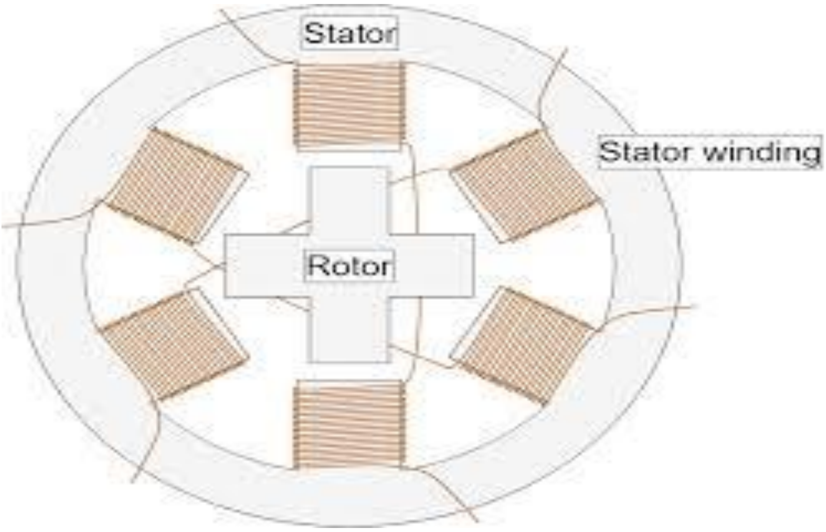
the wrist.

   - **End-Effector Rotation (Axis 7)**: A stepper motor provides additional rotational capability to the end-effector.

2. **Control and Coordination**: The robotic arm's control system sends step and direction signals to the driver circuits of each stepper motor. The control system can be a microcontroller running specific algorithms to manage the coordinated movements of all the joints. For instance, using ROS (Robot Operating System) allows for sophisticated control algorithms that ensure smooth and coordinated movements.
3. **Precision and Repeatability**: Stepper motors offer high precision and repeatability, crucial for tasks requiring exact positioning. They can move to a specific position and hold that position without drift, making them ideal for robotic arms performing repetitive tasks.
4. **Torque and Speed Considerations**: The torque and speed requirements for each joint vary based on the arm's design and the tasks it performs. Stepper motors are chosen based on their torque capabilities to handle the loads at different joints. High-torque stepper motors might be used for the base and shoulder joints, while smaller stepper motors can be used for the wrist joints.
5. **Feedback Systems**: While stepper motors inherently do not require feedback for basic operation, adding encoders can enhance their functionality. Encoders provide feedback on the actual position of the motor, allowing for closed-loop control, which can correct any missed steps and improve accuracy.

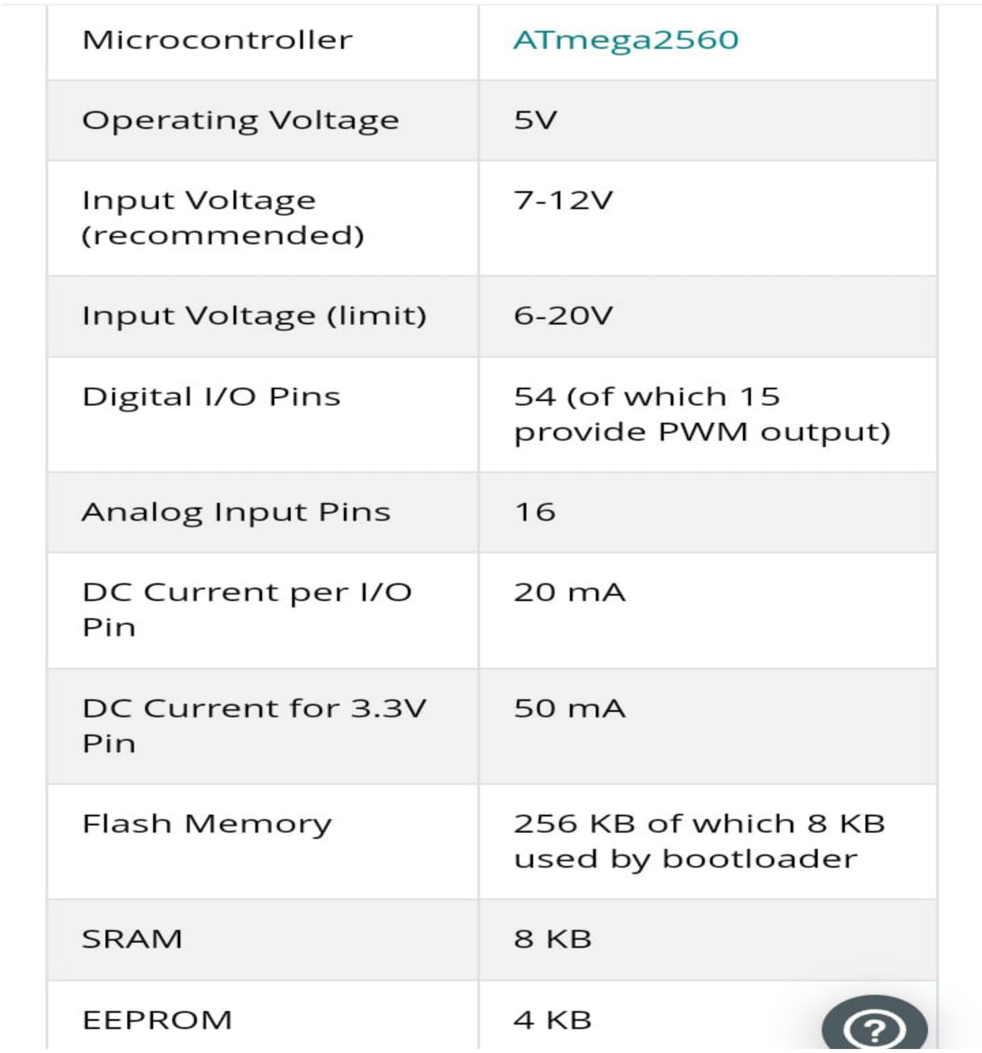
Example Configuration

- **Base and Shoulder Motors**: High-torque stepper motors (e.g., NEMA 23 or larger) to support the weight of the arm and provide robust movement.
- **Elbow and Wrist Motors**: Medium-torque stepper motors (e.g., NEMA 17) to handle the intermediate joints.
- **End-Effector Motor**: Small stepper motor for precise control of tools or grippers.



Arduino mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila. The Mega 2560 is an update to the Arduino Mega, which it replaces.



Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB

power supply

is an electrical device that supplies electric power to an electrical load. The main purpose of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load. As a result,

power supplies are sometimes referred to as electric power converters
A power supply is an electrical device that supplies electricity to those components that use electric power. A power supply is different from a power source. The main function of a power supply is to receive the current from a source and convert it to accurate voltage, frequency, or format to that component that is called power load. The power supply can be of different types.

DC power

DC power supply flows electric charge in one direction, so it supplies energy with fixed polarity. This power supply can obtain power from an AC or DC source. When we need a large power supply, this DC can be used for processes like the smelting of aluminum and other electrochemical processes. The battery is a very common example of a DC power supply.

AC power

AC power supply flows electric charge periodically in a reverse direction. AC power supply provides variable current and frequency to a load. Commonly we use AC power sources for electrical testing in aviation, lighting, laboratory testing, military, and factory production, and our normal electricity supply is AC supply. From an AC power source, we can generate AC supply from 45 to 500 Hz.

Linear-regulated power

The linear-controlled power supply is named because it uses linear nonswitching technology to control the voltage output from the power supply.

Power Supply in a 7-DOF Robotic Arm

Overview

The power supply in a 7-DOF (degrees of freedom) robotic arm is crucial for its operation, providing the necessary electrical power to the motors, sensors, and control systems. Ensuring a stable and sufficient power supply is essential for the smooth and precise functioning of the robotic arm.

Components of the Power Supply System

1. **Power Source**:
    - **AC Power**: For stationary robotic arms used in industrial settings, an AC power source is common. This typically involves connecting to the mains electricity supply.
    - **DC Power**: For mobile or portable robotic arms, DC power sources like batteries are used.
2. **Power Supply Unit (PSU)**: Converts the raw input power from the source into the specific voltages and currents required by the robotic arm's components. PSUs can be AC-DC converters or DC-DC converters depending on the power source.
3. **Voltage Regulators**: Ensure a stable output voltage, protecting sensitive electronic components from fluctuations and providing a consistent power supply.
4. **Battery Management System (BMS)**: For battery-powered systems, the BMS monitors and manages the battery's state of charge, health, and performance, ensuring efficient and safe operation.
5. **Distribution System**: Distributes power to various components of the robotic arm, including motors, sensors, and control electronics. This system often includes fuses or circuit breakers for safety.

Power Requirements and Considerations

1. **Motors**:
    - **Stepper Motors**: Typically require a specific voltage and current to operate efficiently. The power supply must match these requirements to ensure optimal performance.
    - **Servo Motors**: These may have different power requirements, often needing higher peak currents for dynamic movements.
2. **Control Electronics**:
    - **Microcontrollers and Processors**: Usually operate at lower voltages (e.g., 3.3V or 5V). The power supply must provide these voltages with minimal noise to ensure reliable operation.
    - **Communication Modules**: Wireless modules (e.g., Wi-Fi, Bluetooth) also have specific power needs and are sensitive to power quality.

3. **Sensors**:
    - **Encoders, Force-Torque Sensors**: These sensors require a stable power supply to provide accurate measurements. Voltage regulators are crucial to maintain consistency.

Example Power Supply Configuration

1. **AC Power Source**:
    - **Input**: 110V/220V AC from mains.
    - **PSU**: AC-DC converter providing 24V DC output.
    - **Voltage Regulators**: Step-down regulators to provide 12V, 5V, and 3.3V outputs.
2. **DC Power Source**:
    - **Battery Pack**: Li-ion battery pack providing 24V DC.
    - **BMS**: Manages the battery pack, ensuring safe charging and discharging.
    - **DC-DC Converter**: Steps down 24V to required voltages (12V, 5V, 3.3V).
3. **Distribution System**:
    - **Power Bus**: Centralized distribution system with fused connections to each component.
    - **Wiring**: Appropriate gauge wires to handle the current demands of motors and other components.

Example of Power Supply Operation

In a 7-DOF robotic arm, the power supply ensures that each motor receives the correct voltage and current for precise movement. For instance, the base motor might require 24V and 2A for rotation, while the smaller wrist motors might need 12V and 1A. The control electronics, operating at 5V, manage the overall operation, receiving sensor inputs and sending movement commands to the motors.


L (AC) N
⏚
-V -V +V +V
V ADJ

Mechanical Claw

Mechanical Claw will be used for gripping different objects. It can pick objects of variable sizes.

Gripper Force

The gripper force computation is accomplished for breaking down the grasping force. The grinding between the gripper and the article ought to be to such an extent that it bear the heaviness of the item. Gripper force is calculated using the following formula:

µxnfxFg W (1)

Where µ = coefficient of friction of the finger contact surface against the part surface

nf = number of contacting fingers

Fg = gripper force

W = object weight being gripped by system µ = 0.75 nf = 2 W = 5 kg Putting in Equation (1) W nfx µ

Fg= 5 2×0.3 8.33N

Brackets

Brackets of variable shapes and sizes will be used which will serve the purpose of links. They will be connected using mechanical screws.

.

Bearing

One large bearing will be used at the base of the robotic arm so that it can rotate freely and will help in reducing frictional effect and ensure smooth operation.

Screws & Nuts:

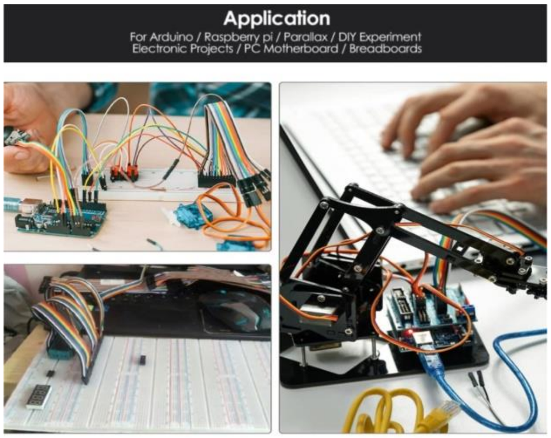
Screws and nuts of various sizes will be used for linking or joining different parts of Robotic Arm.

Power Adapter:

Power Adapter will be used for providing AC power to the system.

Headers and Jumper wires:

Lastly, male Headers and female Jumper wires are used for connection b/w different components



3.2 Software Programming

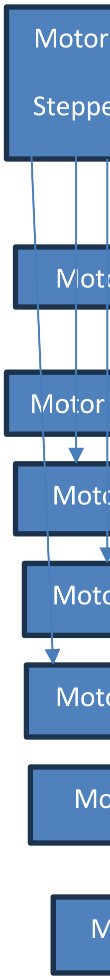
In our project we used software Programming: C programing for Arduino. The code we used shown in Appendix.

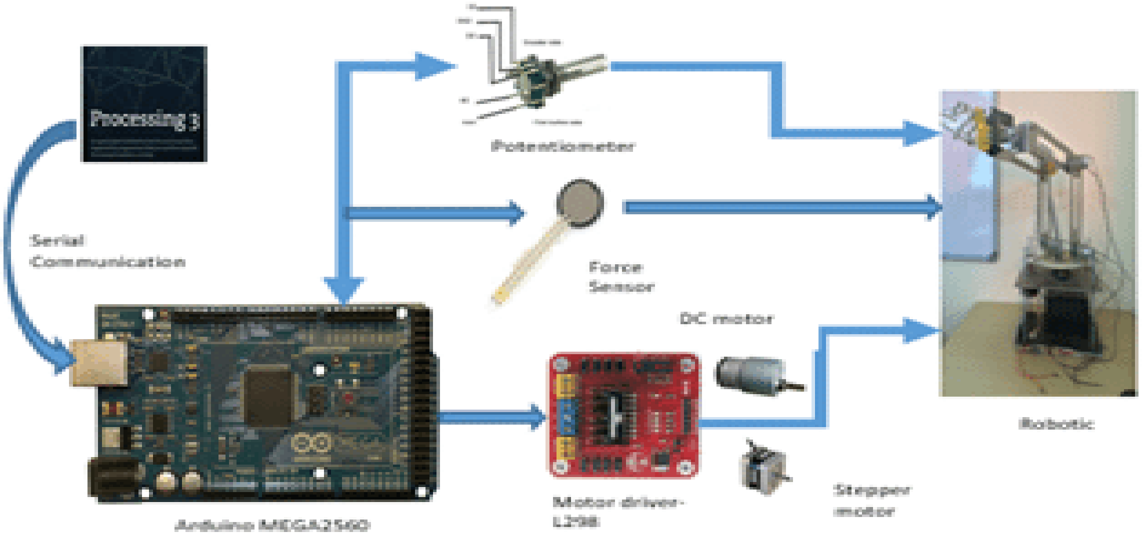
3.2.1 C programmin

Dennis Ritchie developed the general-purpose computer language C at Bell Laboratories in 1972. Despite being an ancient language, it is extremely popular. Since the UNIX operating system was created using C, the two are closely related. It is extremely quick compared to other programming languages, such as Python and Java, and it is very adaptable; it can be used for both technology and applications [20]. The C programming language is a machine-independent language that is primarily used to develop numerous applications and operating systems like Windows, as well as more complex programs like the Git repository, the Oracle database, the Python interpreter, and games. It is regarded as a programming foundation when learning any other programming language. Examples of such applications include operating systems and other application software for computer architectures ranging from supercomputers to PLCs and embedded systems

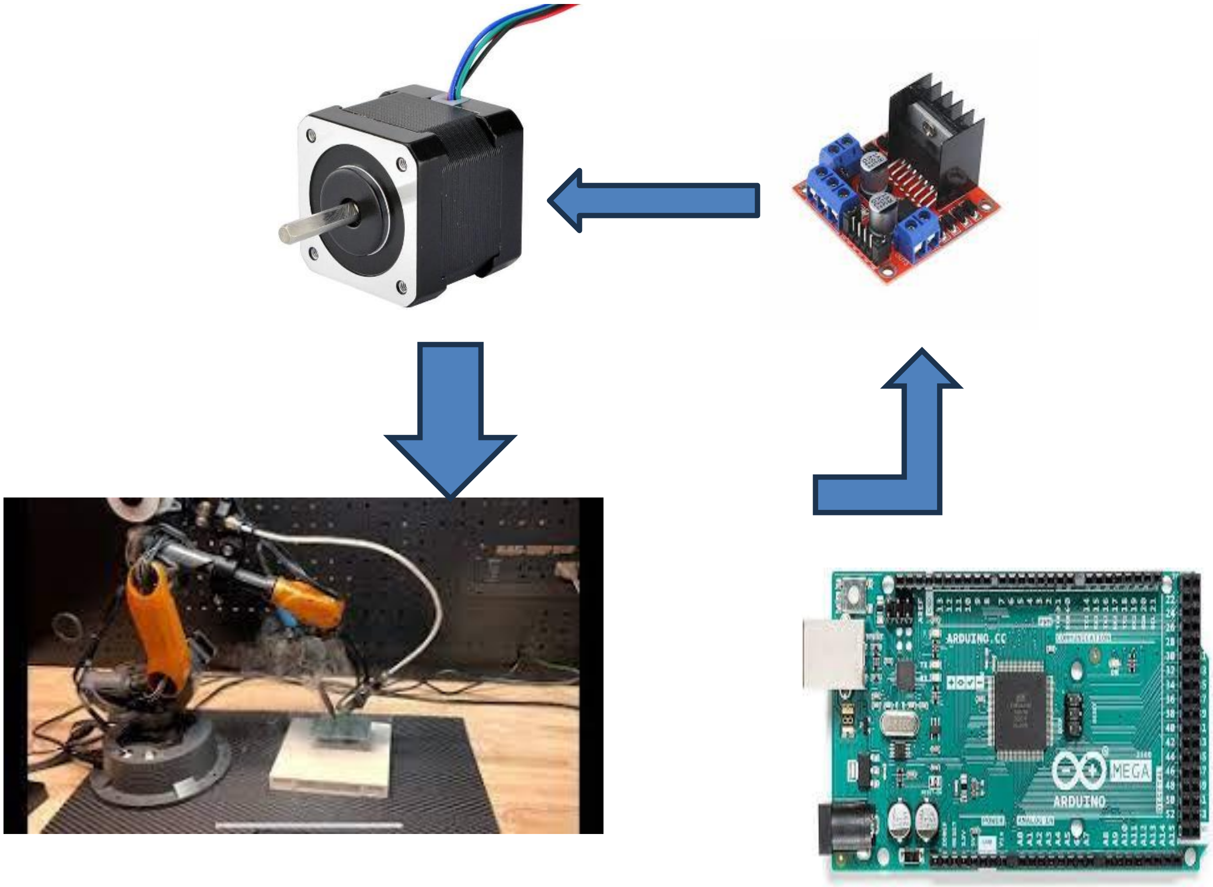
ARDUINO

3.3 Block Diagram for 7 dof robotic arm


Motor
Steppe
Mot
Motor
Moto
Moto
Moto
Mo
M



block diagram

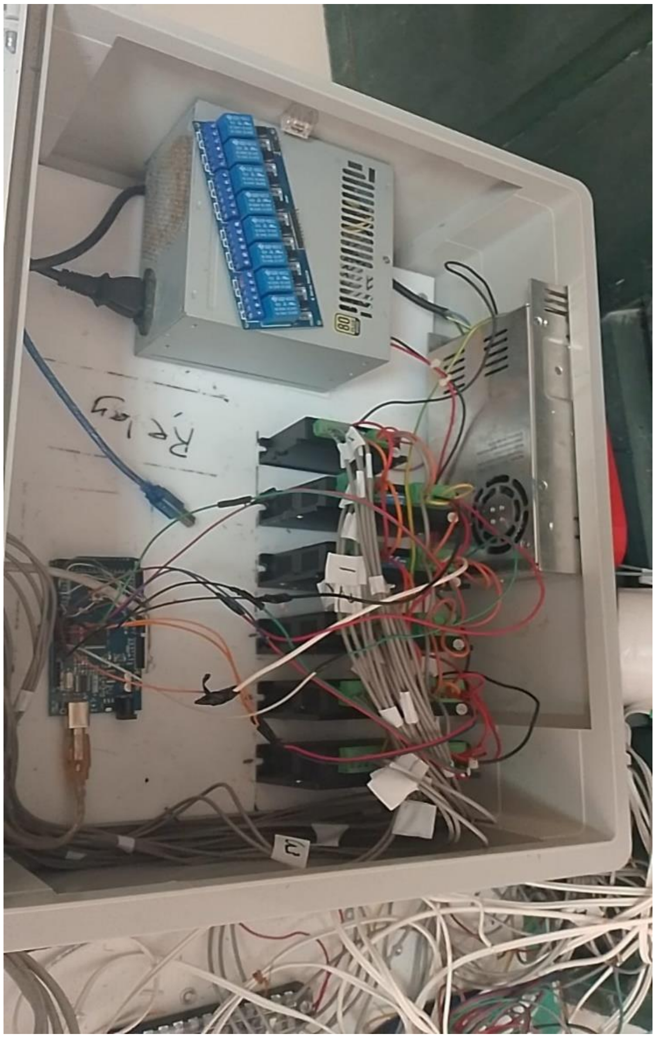


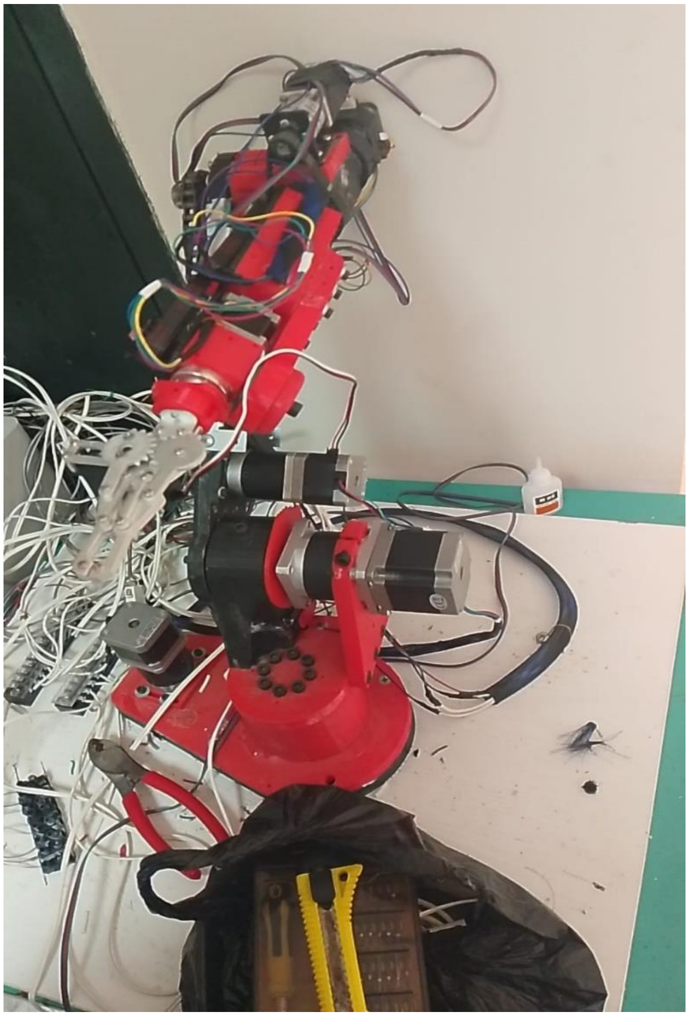
4 Chapter

The structure of the robotics system

shows a whole view for the project, Figure 7.4 shows a top view for the robot with its working space, also it illustrates the zone of the random objects and zones for sorted objects

The whole structure of the robotics system.


Relay



4.2 Methodology

At the beginning of the project, we read several research papers related to the topic of our project and summarized them to see the difference between the researches, and then decided how we would design the 7 dof robotic arm according to what we saw fit.

Then we began writing the report, dividing the tasks between us, and researching the topic of the 7 dof robotic arm in detail.

We have chosen the parts we need and will use, and their quantities, based on what is best for the arm. We also selected the programming language we will use to write the code and program the Arduino. As a result, we will work on operating the arm with the required precision and flexibility. Then, we studied the project in terms of the best way to connect the components correctly and add any missing parts, taking into consideration the sensitivity of the components used.

For the practical part, we purchased the missing parts and calculated the cost. We then tried to connect the components and faced some problems, but we resolved them, thanks to God. We also wrote the code and added it, attempting to operate the arm. We hope that we have succeeded and that our work meets your approval.

4.3 Earlier course work

Through the five years too many courses had been taken, some of them have been used in this project in direct way or not directly

. → English 102.

→ English in Workspace contents were employed in writing professional report and preparing a perfect presentation.

→ Electrical installation and safety system.

→ C++ used to write the code on the Arduino to operate the fountain

. → microprocessors and microcontrollers used it to learn how to control the motors, RGB digital strip led, Arduino

→ Control systems to deal with

4.4 Conclusion

Throughout the project, various challenges were encountered, including issues with component compatibility, signal interference, and precise control of the arm's movements. These challenges were addressed through iterative testing, debugging, and refinement of both hardware and software.

The project was conducted with a focus on cost-effectiveness, ensuring that the selected components provided the necessary performance without excessive cost. This approach makes the design more accessible and feasible for a wider range of users and applications.

In conclusion, the project has successfully demonstrated the design, implementation, and wireless control of a 7-axis robotic arm, highlighting the potential for advanced robotic systems in various applications. The lessons learned and the technologies developed can serve as a foundation for further advancements in robotic manipulation and control.