Development of digital twin of articulated robotic arm

A thesis submitted in partial fulfillment of the requirement for the award of the degree

of

Master of Engineering

to

J. C. Bose University of Science and Technology, YMCA, Faridabad, Haryana

by

Adarsh Aggarwal
(Roll No. 20050521001)

Under the guidance of

Prof. S. S. Dhami

and

Mr. TarikDeep Singh

Department of Mechanical Engineering
National Institute of Technical Teachers Training and Research
Chandigarh

Chandigarh - 160019, India

August 2022

CANDIDATE'S DECLARATION

I,Adarsh Aggarwal, student of M.E. (Robotics), Department of Mechanical Engineering, National Institute of Technical Teachers Training and Research, Chandigarh, bearing Roll Number 20050521001 submit this thesis work entitled "Development of digital twin of articulated robotic arm" to J. C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, India for the award of Master of Engineering degree in Mechanical Engineering (Robotics) and declaring that the work done is genuine and produced under the guidance of Dr. S. S. Dhami, Professor, Mechanical Engineering, NITTTR, Chandigarh and Mr Tarikdeep Singh.

I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or university.

Place: NITTTR, Chandigarh

Date: 15 July 2022

Adarsh Aggarwal

CERTIFICATE

This is to certify that, this pre-thesis report titled **“Development of IoT based control and condition monitoring of articulated robotic arm”** submitted by **Adarsh Aggarwal** bearing Roll Number 20050521001 is a bonafide record of the research work carried out by her in partial fulfillment for the requirement of the award of Master of Engineering, Mechanical Engineering (Robotics) degree from **J. C. Bose University of Science and Technology, YMCA, Faridabad, Haryana, India** at **National Institute of Technical Teachers Training and Research, Chandigarh**. This pre-thesis has not been submitted to any other University or Institution for the award of any degree.

Dr. S. S. Dhami
Professor
Mechanical Engineering Department
National Institute of Technical Teachers
Training and Research, Chandigarh
Chandigarh - 160019, India

Acknowledgements

I would first like to thank my thesis advisor Dr. S. S. Dhami, Professor, Mechanical Engineering at NITTTR, Chandigarh for the continuous support of my thesis, for his/her patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my thesis study. He consistently allowed this thesis to be my work but steered me in the right direction whenever he thought I needed it.

Besides my advisor, I would like to thank Mr.Tarikdeep Singh, Competency Head - Iot and Digital at Tech Mahindra who gave me a chance to work as an inter as IoT developer which helped me to develop a foundation in IoT and made concepts easier and the rest of my faculty members: Dr. Rupinder Singh, Professor, Mechanical Engineering, Dr. B. S. Pabla, Professor, Mechanical Engineering, Dr. Niraj Bala, Professor, Entrepreneurship Development and Industrial Coordination for their encouragement, insightful comments, and hard questions.

Finally, I must express my very profound gratitude to my parents, fiance and my best friend Jiya Jindal for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Adarsh Aggarwal
Master of Engineering
Mechanical Engineering Department
Roll Number - 5120021001

Abstract

In 2020, the global market for articulated robots was worth US$7.91 billion. One of the main drivers propelling the market's expansion is the new trend of factory automation, more especially Industrial IoT. Between 2021 and 2026, the market for articulated robots is anticipated to expand at a compound yearly growth rate of around 9%. As a result, tools for designing, creating, programming, simulating, installing, and monitoring the potential health of articulated robot manipulators are required. A recent Internet revolution is the Internet of Things (IoT). By utilising wireless communications and affordable sensors, processing, and storage devices, it can link dispersed and mobile objects, machines, or assets.The Internet is evolving from a network of computers to a network of things. The Internet of Things (IoT) offers a wide range of industrial applications such as remote controlling and condition monitoring. As long as the robot is online, this thesis demonstrates how to leverage Internet of Things (IoT) for such purpose.

In this work, a system is developed to connect the virtual and actual robot so that the articulated robotic arm can be controlled through a virtual robot remotely. The virtual robot and controller of the physical robotic arm were built using ROS. The virtual robot is imported in Gazebo for visualization and simulation purpose.The ROS node connects a virtual robotic arm with a real robotic arm for bi-directional communication. Raspberry pi is used with sensors to collect the acceleration and temperature data and publish it on a cloud database for further analysis for the real-time condition monitoring of robot.

The input joint angles were given to the virtual robot through the virtual controller which is superimposed on the physical robot with help of `write_node`.The actual joint position of the robot is fetched using python script along with the temperature and acceleration to monitor the condition of the robot. The absolute percentage error of actual joint position and input joint position is calculated. Acceleration and temperature data of the robot is gathered at different speed at same payload.

Keywords: Digital Model, ROS, Gazebo, IoT, Industry 4.0 , Data Acquisition

Contents

List of Figures

List of Tables

Nomenclature

$\Im(\cdot)$	Imaginary part of a complex variable
$\Re(\cdot)$	Real part of a complex variable
μ/μ_j	Linear mass density of string $/j = 1$ and 2 for contact and messenger wires, respectively (average dropper mass included)
ω	Circular frequency
$\mathbf{I}_n$	Identity matrix of dimension $n \times n$
c_j	Phase velocity of string $/j = 1$ and 2 for contact and messenger wires, respectively (calculated using μ_j)
w/w_j	Transverse displacement of a continuum $/j = 1$ and 2 for contact and messenger wires, respectively
x	Spatial coordinate

CHAPTER 1

INTRODUCTION

Chapter 1

INTRODUCTION

Robots have become an intelligent source of aid to humans, and are used in many fields of industry and everyday life. In the last four decades, a great deal of research and development has been carried out to equip robots with a force sensor, vision, and computerized control, to improve their intelligence and ability to interact with the environment, therefore the development of digital twin allows analysis of data and monitoring of systems to head off problems before they even occur, prevent downtime, develop new opportunities and even plan for the future by using simulations.
The main goal of this thesis is to develop a digital twin of the articulated robotic arm. This thesis also aims to create an open-source framework to develop a digital twin and gather data on the physical health of a robot for predictive maintenance by doing FE analysis using real-time data.

1.1 INDUSTRIAL ROBOT

In this section, the Industrial Robot and its design process is described. It is important for the reader to understand what is being tested during this work and the reason why this model was selected.

1.1.1 Definition

According to the International Organization for Standardization the definition of Industrial Robot is the following: "automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes". Another defined an Industrial Robot as complex mechatronic system which integrates several areas of knowledge. It has a strong dependence between geometric, dynamic performance, structural strength, functionality and cost. The general commercial and technological advantages of robots are listed below:

- Robots are good substitutes to human beings in hazardous or uncomfortable work environments.

- A robot performs its work cycle with consistency and repeatability which is difficult for human beings to attain over a long period of continuous working.
- Robots can be reprogrammed. When the production run of the current task is completed, a robot can be reprogrammed and equipped with the necessary tooling to perform an altogether different task.
- Robots can be connected to the computer systems and other robotics systems. Nowadays robots can be controlled with wire-less control technologies. This has enhanced the productivity and efficiency of automation industry.

There are several classes of robots: robotic aircraft, robotic ships, mobile robots and others. An important application of robots is in industry – for machine tending, welding, painting, assembly and etc. These “industrial robots” can be viewed as consisting of a mechanical portion “the manipulator” controlled by a microprocessor.

1.1.2 ANATOMY OF INDUSTRIAL ROBOTS

The manipulator of an industrial robot consists of a series of joints and links. Robot anatomy deals with the study of different joints and links and other aspects of the manipulator’s physical construction. A robotic joint provides relative motion between two links of the robot. Each joint, or axis, provides a certain degree-of-freedom (dof) of motion. In most of the cases, only one degree-of-freedom is associated with each joint. Therefore the robot’s complexity can be classified according to the total number of degrees-of-freedom they possess.

Each joint is connected to two links, an input link and an output link. Joint provides controlled relative movement between the input link and output link. A robotic link is the rigid component of the robot manipulator. Most of the robots are mounted upon a stationary base, such as the floor. From this base, a joint-link numbering scheme may be recognized as shown in figure 1.7 [**?**]

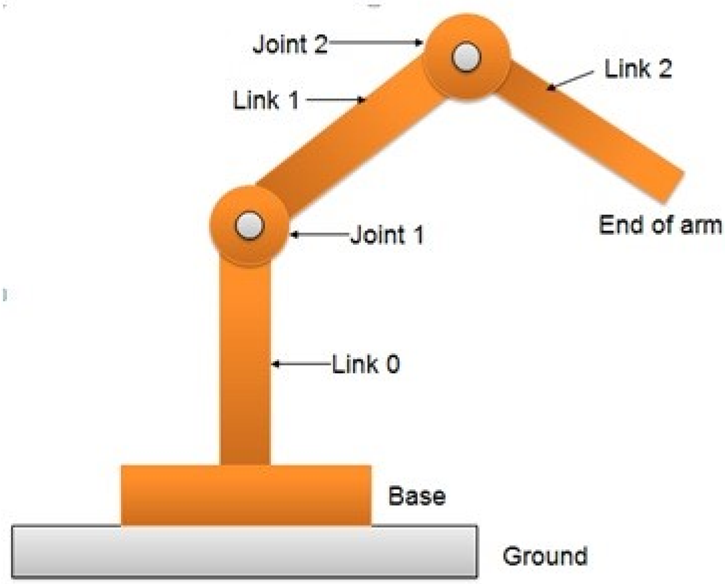


Fig. 1.1: Joint-Link Scheme For Robot Manipulator

Nearly all industrial robots have mechanical joints that can be classified into following five types as shown in figure 1.2

(a) Linear Joint (type L joint) - The relative movement between the input link and the output link is a translational sliding motion, with the axes of the two links being parallel.

(b) Orthogonal joint (type U joint) - This is also a translational sliding motion, but the input and output links are perpendicular to each other during the move.

(c) Rotational joint (type R joint) - This type provides rotational relative motion, with the axis of rotation perpendicular to the axes of the input and output links.

(d) Twisting joint (type T joint) - This joint also involves rotary motion, but the axis or rotation is parallel to the axes of the two links.

(e) Revolving joint (type V-joint) - In this type, axis of input link is parallel to the axis of rotation of the joint. However the axis of the output link is perpendicular to the axis of rotation.

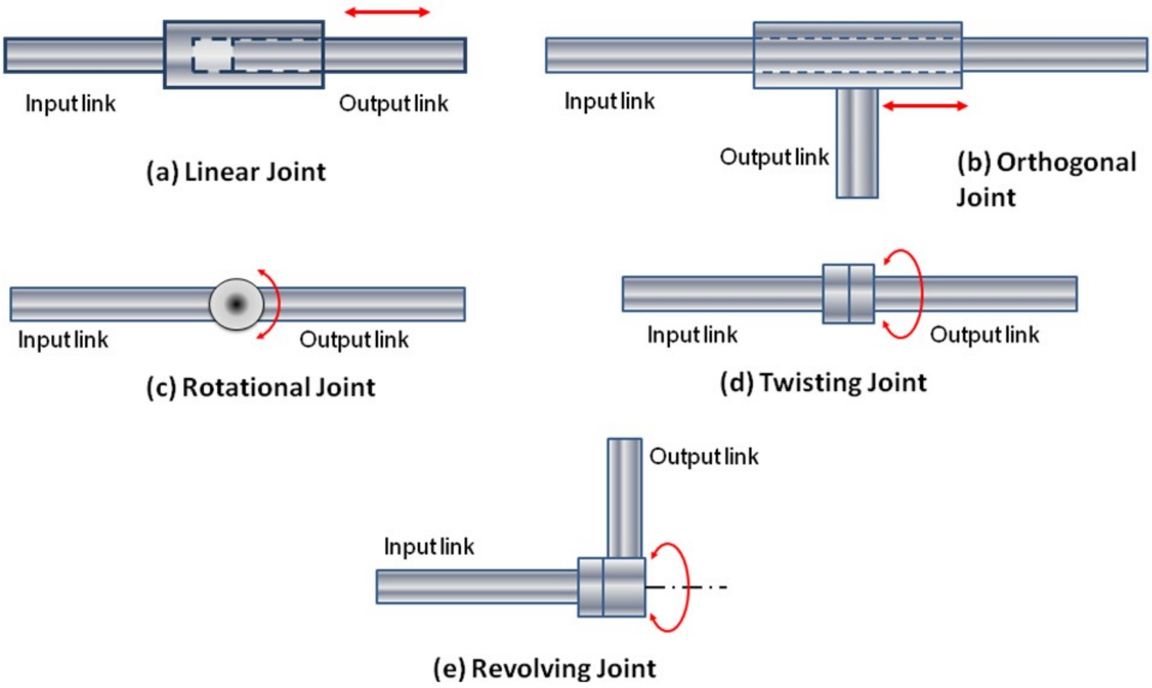


Fig. 1.2: Types of Joints

1.1.3 Common Robot Configurations

Basically the robot manipulator has two parts viz. a body-and-arm assembly with three degrees-of-freedom; and a wrist assembly with two or three degrees-of-freedom. For body-and-arm configurations, different combinations of joint types are possible for a three-degree-of-freedom robot manipulator. Five common body-and-arm configurations are outlined in Figure 1.3

(a) Polar Configuration - It consists of a sliding arm L-joint, actuated relative to the body, which rotates around both a vertical axis (T-joint), and horizontal axis (R-joint).

(b) Cylindrical configuration - It consists of a vertical column. An arm assembly is moved up or down relative to the vertical column. The arm can be moved in and out relative to the axis of the column. Common configuration is to use a T-joint to rotate the column about its axis. An L-joint is used to move the arm assembly vertically along the column, while an O-joint is used to achieve radial movement of the arm.

(c) Cartesian co-ordinate robot - It is also known as rectilinear robot and x-y-z robot. It consists of three sliding joints, two of which are orthogonal O-joints.

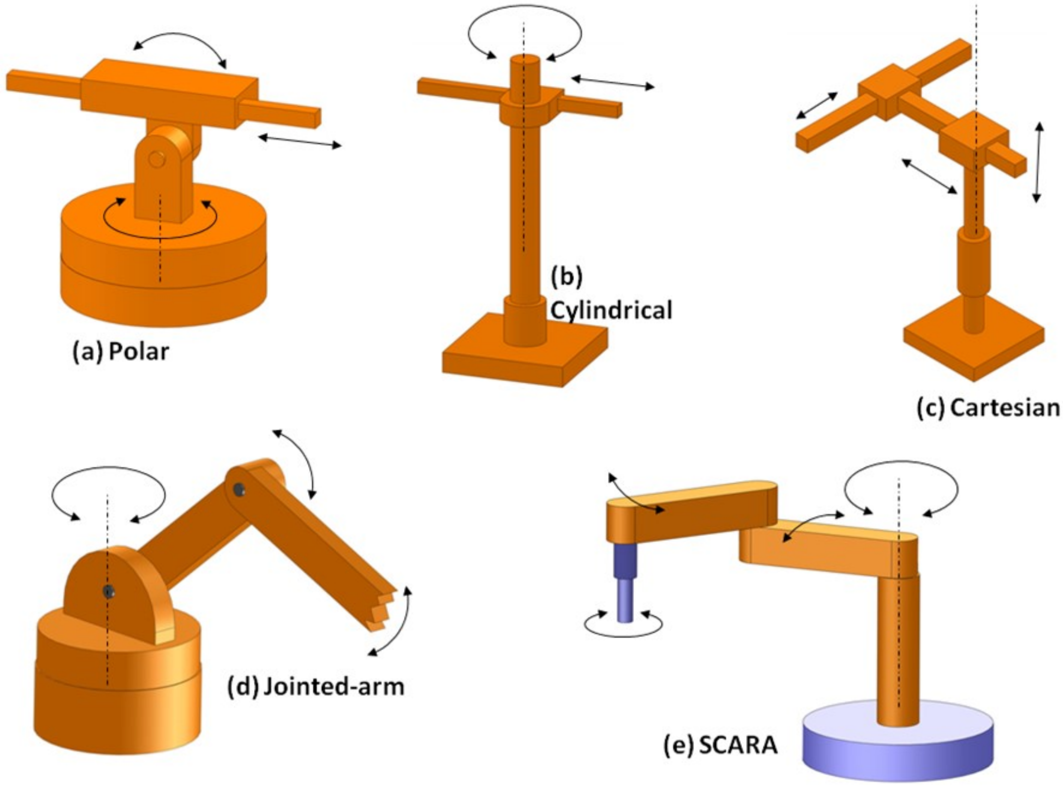


Fig. 1.3: Common body-and-arm configurations

(d) Jointed-arm robot - It is similar to the configuration of a human arm. It consists of a vertical column that swivels about the base using a T-joint. Shoulder joint (R-joint) is located at the top of the column. The output link is an elbow joint (another R joint).

(e) SCARA - Its full form is 'Selective Compliance Assembly Robot Arm'. It is similar in construction to the jointer-arm robot, except the shoulder and elbow rotational axes are vertical. It means that the arm is very rigid in the vertical direction, but compliant in the horizontal direction.

Robot wrist assemblies consist of either two or three degrees-of-freedom. A typical three-degree-of-freedom wrist joint is depicted in Figure 1.4. The roll joint is accomplished by use of a T-joint. The pitch joint is achieved by recourse to an R-joint. And the yaw joint, a right-and-left motion, is gained by deploying a second R-joint. The SCARA body-and-arm configuration typically does not use a separate wrist assembly. Its usual operative environment is for insertion-type assembly operations where wrist joints are unnecessary. The other four body-and-arm configurations more-or-less follow the wrist-joint configuration by deploying various combinations of rotary joints viz. type R and T.

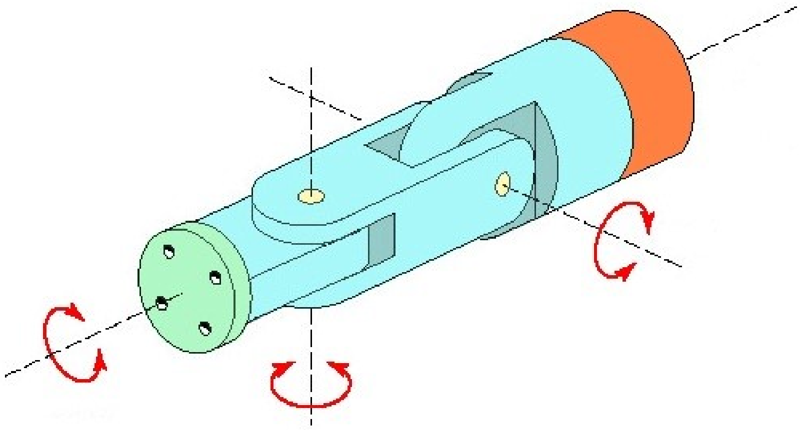


Fig. 1.4: Robotic Wrist Joint

1.1.4 End Effector

An end effector is usually attached to the robot39;s wrist, and it allows the robot to accomplish a specific task. This means that end effectors are generally custom-engineered and fabricated for each different operation. There are two general categories of end effectors viz. grippers and tools. Grippers grasp and manipulate the objects during the work cycle. Typically objects that grasped are the work parts which need to be loaded or unloaded from one station to another. The robot end effector may also use tools. Tools are used to perform processing operations on the work part.

1.2 SUBSYSTEM OF INDUSTRIAL ROBOTS

The various sub systems of industrial robots are described below-:

1.2.1 Actuators

Actuators are basically prime movers providing both force and motion. Basically three types of drive systems are commonly used to actuate robotic joints. These are electric, hydraulic, and pneumatic drives. Electric motors are the prime movers in robots. Servomotors or steeper motors are widely used in robotics. Hydraulic and pneumatic systems such as piston-cylinder systems, rotary vane actuators are used to accomplish linear motions, and rotary motions of joints respectively.

Pneumatic drive is regularly used for smaller, simpler robotic applications; whereas electric and hydraulic drives may be found applications on more sophisticated industrial robots. Due to the advancement in electric motor technology made in recent years, electric drives are generally favoured in commercial applications. They also have compatibility to computing systems. Hydraulic systems, although not as flexible as electrical drives, are generally used where larger speeds are required. They are generally employed to carry out heavy duty operations using robots.

1.2.2 Transmission System

The transmission system used in robot to transmit power and motion consists of chains, timing belts, metal belts, cables and pulleys and linkages. Gear boxes and harmonic drives serve to provide speed reduction. Ball screws are used with suitable mechanisms to convert rotary motion to linear motion and if needed back to oscillatory motion. Drive stiffness is an important consideration in robotics and so also is backlash.

1.2.3 Power supplies and power storage system

Hydraulic and Pneumatic power packs: These consist of a motor driving a positive displacement pump or compressor to generate the high pressure fluid flow. In using hydraulic systems the necessity of having an oil tank increases the weight of the system, additionally the issue of ensuring that the oil is free of contaminants is to be handled. In pneumatic power pack dry air is desired.

1.2.4 Sensors

The sensors for feedback in robots consists of tachometers and encoders and pentameter's to sense motor motions, simple switches, force sensors, acceleration sensors, optical systems, special cameras and vision systems.
There are generally two categories of sensors used in robotics. These are sensors for internal purposes and for external purposes. Internal sensors are used to monitor and control the various joints of the robot. They form a feedback control loop with the robot controller. Examples of internal sensors include potentiometers and optical encoders, while tachometers of various types are deployed to control the speed of the robot arm.
External sensors are external to the robot itself and are used when we wish to control the operations of the robot. External sensors are simple devices, such as limit switches that determine whether a part has been positioned properly, or whether a part is ready to be picked up from an unloading bay.

1.2.5 Microprocessors and Controllers

There are a host of electronic circuits, motor controllers, analog to digital converters and digital to analog converters, frame grabbers, and so on utilized to handle sensors and vision systems and convert the inputs from them into a form usable by the processor for control of the entire system in conjunction with the algorithms and software developed specifically for the purpose.

1.2.6 Softwares

The software used consists of several levels. Motor control software consists of algorithms that help the servo to move smoothly utilizing the data from feedback units. At the next level, there is software to plan the trajectory of the end effector and translate the same into commands to individual motor controllers. The output of sensors is also to be interpreted and decisions made. At the highest level, there is software that accepts commands from the user of the robot and translates it into appropriate actions at the lower level.

The combination of drive system, sensors, and feedback control system determines the dynamic response characteristics of the manipulator. Speed in robotic terms refers to the absolute velocity of the manipulator at its end-of-arm. It can be programmed into the work cycle so that different portions of the cycle are carried out at different velocities. Acceleration and deceleration control are also important factors, especially in a confined work envelope. The robot's ability to control the switching between velocities is a key determinant of the manipulator's capabilities. Other key determinants are the weight (mass) of the object being manipulated, and the precision that is required to locate and position the object correctly. All of these determinants are gathered under the term 'speed of response', which is defined as the time required for the manipulator to move from one point in space to the next. Speed of response influences the robot's cycle time, which in turn affects the production rate that can be achieved.

Stability refers to the amount of overshoot and oscillation that occurs in the robot motion at the end-of-arm as it attempts to move to the next programmed location. More oscillations in the robotic motion lead to less stability in the robotic manipulator. However, greater stability may produce a robotic system with slower response times.

Load carrying capacity is also an important factor. It is determined by weight of the gripper used to grasp the objects. A heavy gripper puts a higher load upon the robotic manipulator in addition to the object mass. Commercial robots can carry loads of up to 900 kg, while medium-sized industrial robots may have capacities of up to 45kg.

1.3 DEFINITION OF DIGITAL TWIN

A virtual version of a physical thing that is created to precisely reflect it is called a digital twin. A wind turbine, for example, has a number of sensors that may be used to monitor different elements of its performance. These sensors gather information on the physical object's performance in terms of temperature, energy generation, weather conditions,

and other factors. Then, a processing device receives this information and applies it to a digital copy.
After receiving this knowledge, the virtual model may be used to run simulations, look at performance issues, and make suggestions in order to learn crucial lessons that can then be applied to the original physical entity.

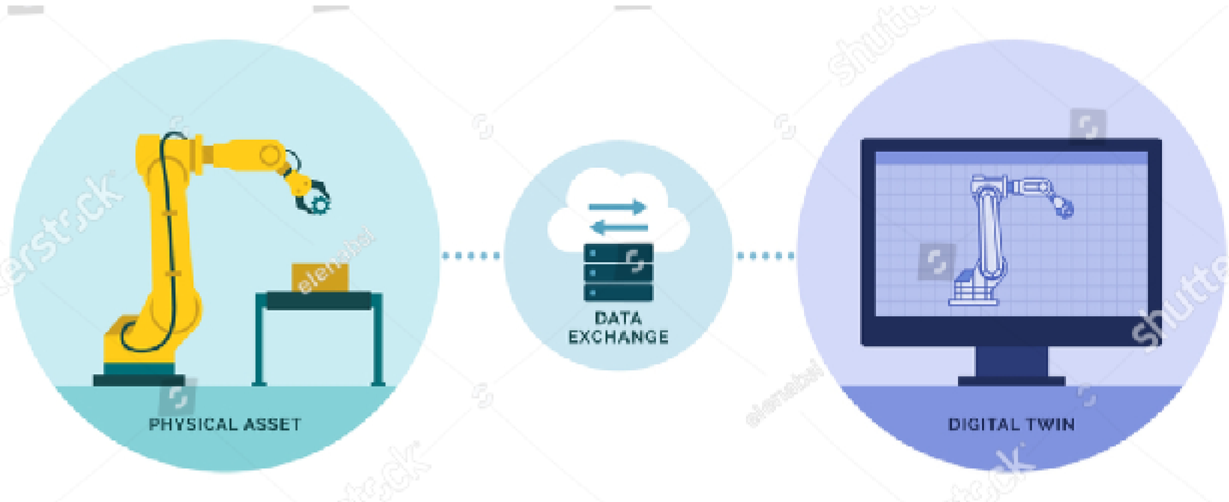


Fig. 1.5: Representation of Digital twin.

1.3.1 Types of Digital Twin

Digital twins come in a variety of forms depending on how magnified the product is. The field of application is where these twins diverge the most. It is typical for many kinds of digital twins to coexist in a system or process.

1.3.1.1 Component twins/Parts twins

Component twins are the basic unit of digital twin, the smallest example of a functioning component. Parts twins are roughly the same thing, but pertain to components of slightly less importance.

1.3.1.2 Asset Twins

When two or more components work together, they form what is known as an asset. Asset twins let you study the interaction of those components, creating a wealth of performance data that can be processed and then turned into actionable insights.

1.3.1.3 System or Unit twins

The next level of magnification involves system or unit twins, which enable you to see how different assets come together to form an entire functioning system. System twins provide visibility regarding the interaction of assets, and may suggest performance enhancements.

1.3.1.4 Process Twins

Process twins, the macro level of magnification, reveal how systems work together to create an entire production facility. Are those systems all synchronized to operate at peak efficiency, or will delays in one system affect others? Process twins can help determine the precise timing schemes that ultimately influence overall effectiveness.

1.3.2 Methods for developing digital twin

A Digital Twin can be created with different methods. One method is to create a 3D CAD of the robotic arm and use Virtual Reality (VR) equipment to control the arm remotely; however, the software is needed to build a VR scene that resembles the prototype setting and to calibrate the location of the robotic arm in the virtual world. In another method, a digital twin is created to connect the virtual and actual robots so that the articulated robotic arm can be controlled virtually. The design and model for the robotic arm that is used in the Gazebo simulation environment are built using the Robotic Operating System (ROS). The Message Queuing Telemetry Transport (MQTT) platform is used to connect the virtual and actual robotic arms for two-way communication.

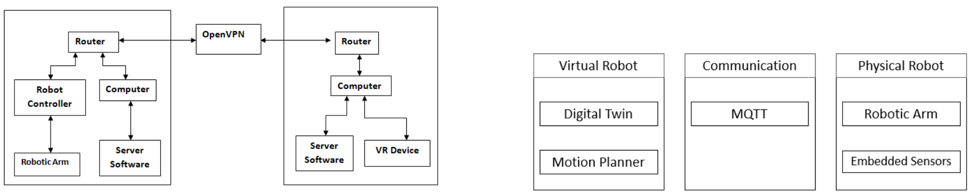


Fig. 1.6: Flowchart of different technique

1.3.3 Advantages of Digital Twin

1.3.3.1 Better RD

The use of digital twins enables more effective research and design of products, with an abundance of data created about likely performance outcomes. That information can lead to insights that help companies make needed product refinements before starting production.

1.3.3.2 Greater efficiency

Even after a new product has gone into production, digital twins can help mirror and monitor production systems, with an eye to achieving and maintaining peak efficiency throughout the entire manufacturing process.

1.3.3.3 Product end-of-life

Digital twins can even help manufacturers decide what to do with products that reach the end of their product life-cycle and need to receive final processing, through recycling or other measures. By using digital twins, they can determine which product materials can be harvested.

1.4 ROBOT OPERATING SYSTEM (ROS)

The Robot operating system (ROS) is free and open source. It consists of vast software libraries and tools by which most robotic applications can be developed quickly. It consists of everything that is required to build a robot from scratch. It has a set of drivers for commonly used hardware so that hardware can be connected without writing code for drivers. It also consists of algorithms and tools for visualization, simulation, and robotic system development. ROS packages can be written in multi-language because ROS has multilingual support like *python, cpp, Java, MATLAB,* etc. [**?**].

ROS has the capability by which individual programs can communicate over a defined API (Application Program Interface) and have a peer-to-peer connection. ROS can run the program on multiple computers by communicating over the network. ROS can be run on a system with small data space because it is lightweight.

As described above, ROS's main properties are listed below –

- Message-passing between processes and package management
- Hardware abstraction
- Low-level device control
- Implementation of commonly-used functionality

ROS has a wide variety of libraries and tools by which programs can be written, developed, and run on a distributed system.

These capabilities can divide mainly into four categories [**?**]–

- Plumbing
  - Process Management

- Inter-Process Communication
  - Device Drivers

- Tools
  - Simulation
  - Visualization
  - Graphical User Interface
  - Data Logging

- Capabilities
  - Control
  - Planning
  - Perception
  - Mapping
  - Manipulation

- Ecosystem
  - Package Organization
  - Software Distribution
  - Documentation
  - Tutorials

1.4.0.1 Master

ROS Master is working as a registration service provider to the other ROS nodes for specific names and identities. All ROS Nodes are registered with ROS Master, and every ROS Topic defined under each Node shares its subscriber and publisher details with the ROS Master. ROS Master is also responsible for location and communication between Node using TCP/IP (Transmission Control Protocol / Internet Protocol) or TCPROS Protocol. The full process node can communicate peer to peer.

1.4.0.2 Node

A ROS Node is just an executable file published or subscribed to a ROS topic and can also be a service provider or can use a service. ROS Node can be written in python or c ++ using ROS client library *rospy* or *roscpp*, respectively.

1.4.0.3 Topic

ROS Nodes are communicated with each other by publishing / subscribing data in the form of ROS Messages to the particular ROS Topic anonymously, which means Nodes are unaware of the nodes they are publishing or subscribing to, separating the generation and consumption of data. These Topics are named buses. Data generated by the Nodes published to a particular topic in the form of ROS Message, and other Nodes that need that data subscribe to that Topic. ROS Message defines the type of ROS Topic. ROS Node keeps publishing the data to a defined ROS Topic, and the Topic's subscription is not established in the message communication until the message type is matched. All transportation of messages will be done through TCP/IP or UDP (User Datagram Protocol). These protocols are named TCPROS and UDPROS, respectively. UDPROS is currently only supported by the *rospy* library.

1.4.0.4 Messages

Nodes transfer or receive the data in the form of ROS Messages. A ROS Message contains data structure, which can be a standard primitive type, arbitrarily nested structure, and arrays.

1.4.0.5 Service

RPC (Remote Procedure Call) is suitable for communication in a distributed system. RPC request/reply transportation of message cannot be done through publish/subscribe model. For that purpose, ROS introduces the ROS Service, defined by a pair of messages, one for requesting the service and one for the reply. A ROS Node offers a service under a string name, and a ROS Node client calls the service by sending the request message. The Client node is waiting for a reply for a defined time duration.

1.4.0.6 Parameter Server

A parameter server stores or retrieves various parameters accessible via network APIs to nodes. It is not that efficient, so it is used for static, non–binary data. It is globally viewable and allows easily see the system's configuration status and change it as needed.

1.4.0.7 Bags

ROS Message data is stored in ROS in a Bag file format. These files are so crucial in ROS for further analysis and development. ROS bag's data can be stored, processed, analyzed, and visualized using vastly available tools like *rosbag*, *rqtbag*, *rostopic*, *Foxglove Studio*, etc. The bag file has a ". bag" extension.

1.4.1 ROS Tools

There are some essential tools listed below –

1.4.1.1 TF Library

Tf library is a ROS tool used to navigate and keep track of the different coordinate systems over time while maintaining the respective relation of the other coordinate systems. No need to store the transformation in a central server because it can be implemented in the distributed system, so that information about all coordinate frames is available to all ROS components [**?**].

1.4.1.2 RVIZ

To 3D visualization of robot model, combined data of sensors and other 3D data rviz is one of the most used tools in ROS. Users can also mark and send their data to the system [**?**].

1.4.1.3 Gazebo

A gazebo is a simulation tool that is easily implemented in ROS packages. It provides realistic simulation using sensors and noise models, precise physics, and advanced 3D graphics. It also provides distributed simulation, dynamic asset loading, and tunable performance [**?**].

1.4.1.4 MoveIt

It's an open-source 3D interactive visualizer integrated with ROS and Gazebo. It is used for motion planning, manipulation, inverse kinematics, control, 3D perception, and collision checking of the robot system [**?**].

1.5 Cloud Storage

A cloud computing provider who maintains and runs data storage as a service provides cloud storage, which is a cloud computing paradigm that saves data on the Internet. It eliminates the need to purchase and manage your own data storage infrastructure because it is offered on demand with just-in-time capacity and pricing. You get flexibility, durability, and "anytime, anywhere" data access as a result. A third-party cloud provider sells cloud storage, which is owned and operated by them and delivered over the Internet using a pay-as-you-go business model. The volume, security, and durability of your data are managed by these cloud storage providers so that it is available to your apps

Fig. 1.7: Respresentation of Cloud Storage

anywhere. Applications can directly use an API to access cloud storage or use conventional storage protocols. In order to gather, manage, protect, and analyse data at scale, several companies offer complimentary services.

1.5.0.1 Advantages of cloud storage

1.5.0.2 Total Ownership Cost

With cloud storage, there is no need to invest money on hardware, storage, or ”someday” scenarios. You may instantly alter performance and retention parameters, add or remove capacity as needed, and only pay for storage that you really use. Even more efficiencies can be achieved by automatically moving less-used data to lower cost levels in compliance with audit-able standards.

1.5.0.3 Time to Deployment

Infrastructure shouldn’t ever hold back development teams from acting when they are ready to. IT can swiftly offer the precise quantity of storage required, exactly when it’s needed, thanks to cloud storage. Instead of having to maintain storage systems, this enables IT to concentrate on resolving complicated application-related issues.

1.5.0.4 Information Management

A huge leverage point for new use cases is created by centralising storage on the cloud. You may carry out effective information management activities, such as automatic data tiering or locking down data in support of compliance needs, by employing cloud storage life cycle management rules.

CHAPTER 2

LITERATURE REVIEW

Chapter 2

LITERATURE REVIEW

There are several classes of robots: robotic aircraft, robotic ships, mobile robots and others. An important application of robots is in industry – for machine tending, welding, painting, assembly and etc. These “industrial robots” can be viewed as consisting of a mechanical portion “the manipulator” controlled by a microprocessor.

2.1 CURRENT STATUS OF RESEARCH

Marius Matulis a amp; Carlo Harvey [?] The development of a digital twin with physical and virtual equivalencies is described in this manuscript. It creates an open-source 3D printed robot arm that is similar to current robot arms in use in the manufacturing industry, The physical element of this digital doppelganger is controlled by Marlin software, which has enough functionality to construct a command centre and interface with the virtual training remedy for the environment To both build and test, we used Unity ml-agents. The digital twin of the robot arm39;s virtual depiction, but also interact with TensorFlow for simulated learning training paradigm. This hardware and software combo was used to reinforce learning is used to simulate train a robot arm to execute a task.

Niki Kousi et al. [?] The implementation of a decision-making framework aimed at enabling combined task scheduling and resource motion planning in hybrid production systems was explored in this research [21]. The suggested architecture is open in the sense that it is based on the open-source ROS framework, which allows for interfacing with a variety of robot kinds as well as different robot manufacturers. As a result, incorporating new sorts of resources into the decision-making process is simple. At the same time, the unified digital world modeling method supports the scheduling of operations for the assembly of various components by allowing the interpretation of diverse assembly sequences.

Yang Yi et al. [?] Furthermore, the virtual space working mechanism was thoroughly discussed, and the virtual space layer's constitution and implementation process for two functional subsystems, including the assembly process planning and simulation verification subsystem and the assembly prediction analysis and control management subsystem, were demonstrated. Finally, it offered a reduced satellite assembly case study, which included the creation of a DT-AAS experimental testbed for the development of smart assembly processes. To give a reference for smart assembly engineering methods, the implementation methodology and application process for this scenario are described in full. On this foundation, the projected DT-assembly AAS's performance was addressed, including the benefits and remaining problems, in order to establish a solid foundation for attaining smart assembly.

C. J. Liang et al. [?] The beginning development of the online robot Digital Twin system for human-robotisation in construction and digital fabrication was presented in this study. The virtual robot module, the real robot module, and the communication module are all part of the system. It used ROS Gazebo to create the virtual robot module, which is essentially a digital twin of the actual robot and connected it to the physical robot module via MQTT Bridge in the communication module. The robotic arm39;s joint angles are swapped and coordinated between two robots. It also used the MATLAB package in the virtual robot module to plan and operate the robotic arm, then sent the command to the real robot module for execution.

X. Wang et al. [?] The goal of this research is to develop a real-time, process-level, immersive digital twin system for collaborative human-robot construction work. The system has a number of benefits. First, human workers may view construction site conditions and collaborate with the robot remotely, protecting them from any hazards on the job. Second, the communication network allows the human worker and the robot to communicate in near real-time about task plans and status updates. Third, it enables the human worker to improvise high-level building plans based on the geometry of the construction site as it exists. Finally, the robot creates its motion plan and does practical building work on-site, reducing human workload dramatically.

Havard V et al. [?] Architecture for real-time co-simulation between a digital twin and a virtual reality environment is proposed in this article. This real-time co-simulation ensures that the actual system behaves realistically. The suggested system is built on a client-server architecture, and real-time machine-to-machine communication is accomplished through the usage of ZMQ sockets. The architecture was then applied to a specific use case including a manual station with a robotic arm UR10 added to assist the operator with his assembly operation. The co-simulation provides for realistic robotic arm behavior, and the combination with virtual reality enables for digital prototyping, operator safety assessment, and workstation architecture and ergonomics research.

Roger A. Light [?] Mosquito delivers MQTT server and client solutions that are consistent with industry standards. MQTT is a publish/subscribe protocol with little network overhead that can be implemented on low-power devices like microcontrollers that could be utilised in Internet of Things sensors. As such, Mosquitto is intended for use in all situations where there is a need for lightweight messaging, particularly on constrained devices with limited resources.

Azad M. Madni et al. [?] This paper presents an overall vision and rationale for incorporating digital twin technology into MBSE. The paper discusses the benefits of integrating digital twins with system simulation and the Internet of Things (IoT) in support of MBSE and provides specific examples of the use and benefits of digital twin technology in different industries. It concludes with a recommendation to make digital twin technology an integral part of MBSE methodology and experimentation testbeds.

Matthew Q. Marshall amp; Cameron Redovian [?] A robotic system39;s design decision (beginning of life stage) is aided by an experimental digital twin. The goal of this device is to automate a material-feeding system. A six-axis manipulator is mounted on a mobile platform in the robot. The material-placement routine is believed to take place in an unstructured environment due to variations in the dimensions of the material-feed system and positioning mistakes of the mobile base. Working there necessitates the use of exteroceptive sensors, which in this case take the shape of computer vision. This subsystem's data is utilized to match the digital twin's geometry to the physical world. Simulated trials can lead to crucial design decisions due to the tight connection between physical and virtual embodiments. In this scenario, two motion-planning systems are compared, and it is found that the costs of installing the dynamic one in the lab for testing are justified by its ease of use and reliability because simulation-based control makes use of all available data.

Park et al. [?] developed new mid-sized humanoid robot hardware. The study focused on the use of an integrated application of CAD/CAM/CAE and rapid prototyping (RP) for the rapid development of the robot's outer body parts. In the study, most parts are designed three-dimensionally with 3D CAD software which enable effective connection with CAE analyses, the basis of which is laid in kinematic simulation and structural analysis. The study applied integrated analyses successfully on the humanoid robot prototype Bonobo.

Vukobratovic et al. [?] discussed kinds of robot driving systems and described CAD systems for industrial robots. The study avoided giving specific constructive solutions and discussed the impact of the actuators on the robot design along with explaining the principles of advanced robot design.

Rufang Li. [?] presented 3-D static and dynamic contract/impact analysis of gear drives. She analyzes the stress distribution of gear system under dynamic loading conditions and simulates the stress of gears under conditions of initial speed and a sudden load being applied.

Cheung [?] developed a new type of element consistent with this specific finite element theory. In his study, he showed the deformations of the spatial motions of a gear pair and found some errors induced by manufacturing and assembly processes. All new findings reported on the simulation studies.

2.2 INTERFERENCE DRAWN FROM THE LITERATURE REVIEW

There are most essential inferences are written below based on the literature review -

- Robot manipulator is a combination of joints and links in which power is transmitted through Servo motors and the transmission system with the help of the controller.
- Digital twin helps to create a hybrid environment where humans and machines can interact with each other.
- Robots can be controlled remotely to perform work.
- MQTT platform is used to develop digital twin.
- Feedback from physical robots plays an important role.
- Latest development in industry 4.0.
- Development of digital twin can help to manufacture reliable and application-oriented products.
- Digital twin can reduce the maintenance and operating cost and eliminate the risk of accidents.
- Remote operation of device can be done.

.

CHAPTER 2

PROBLEM FORMULATION

Chapter 3

PROBLEM FORMULATION

Articulated robots are the most common types of industrial robots. The articulated arm design combines an extensive range of rotational motion and linear reach with the advantages of precision movement. It is ideal for multiple industrial operations such as welding, assembly, material handling, pick-and-place operations, etc.Industrial articulated robots are expensive yet cost-effective investment since they can be redeployed should product lines change, by reprogramming and replacing the end-effector.

3.1 PROBLEM STATEMENT

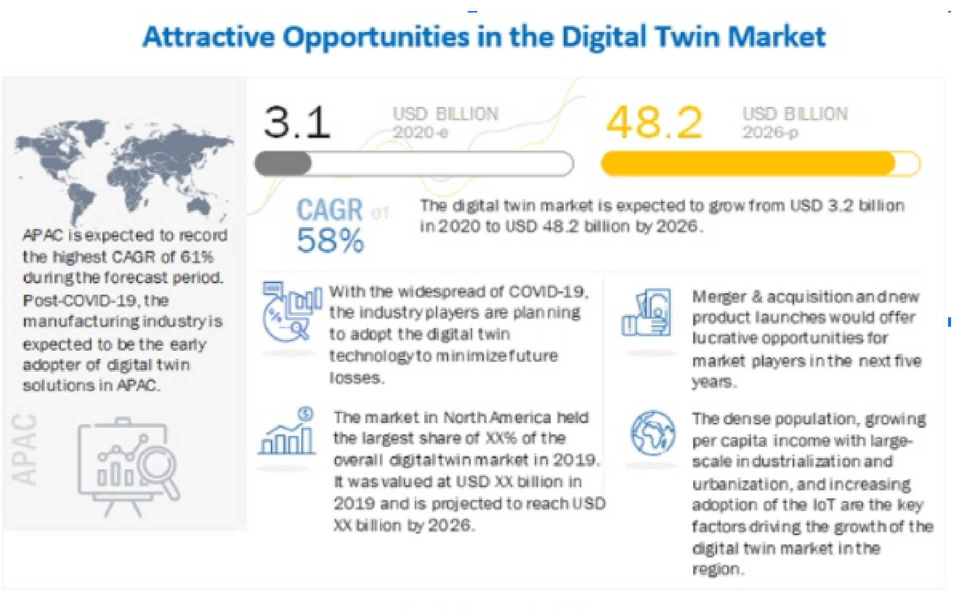


Fig. 3.1: IoT Based Controlling System.

The global articulated robot market reached a value of US$ 7.91 Billion in 2020. The emerging trend of factory automation, specifically Industrial IoT, is one of the key factors driving the growth of the market. Other factors including, growing adoption of industrial robots, significant growth in the electronics manufacturing, automotive industry, and product innovations, such as the introduction of soft and vacuum grippers for handling fragile products, are projected to drive the market further. The global articulated robot market is expected to grow at a compound annual growth rate of around 9% during 2021-2026. Therefore, there is a need of developing tools for design, development, programming, simulation, installation and predictive health monitoring of articulated robot manipulators. Digital twin consists of three parts viz. physical product, digital product and the linkage between the physical and digital products. It enables real time communication between physical and virtual models. Data can be read from the physical product via digital twin, can be analyzed and simulations can be executed. Vice versa actuation commands, control signals and other data can be sent to the physical twin with the help of a digital twin. Thus, this technology can be used to realistically simulate a robotic system for various purposes such as performance testing, installation, learning its operation, predicting health condition etc. leading to cost cutting and improved industrial safety.

3.2 Thesis Title

The title for the thesis work is **"Development of IoT based control and condition monitoring of articulated robotic arm."**

3.3 OBJECTIVES OF WORK

The Objective of the thesis work is as follow:

- To develop an open source technological framework for remotely controlling an articulated robotic arm, which is a major component in smart manufacturing.
- To create a functional prototype of bi-directional remote communication for the robotic arm.
- To operate the physical robotic arm remotely through a digital model of the arm.
- To acquire acceleration and temperature data of the physical robotic arm in real time through cloud.

CHAPTER 3

METHODOLOGY

Chapter 4

METHODOLOGY

This chapter deals with the methodology that was adopted to develop IoT based control and condition monitoring system of an articulated arm. Figure 4.1 shows the different steps adopted in the methodology.

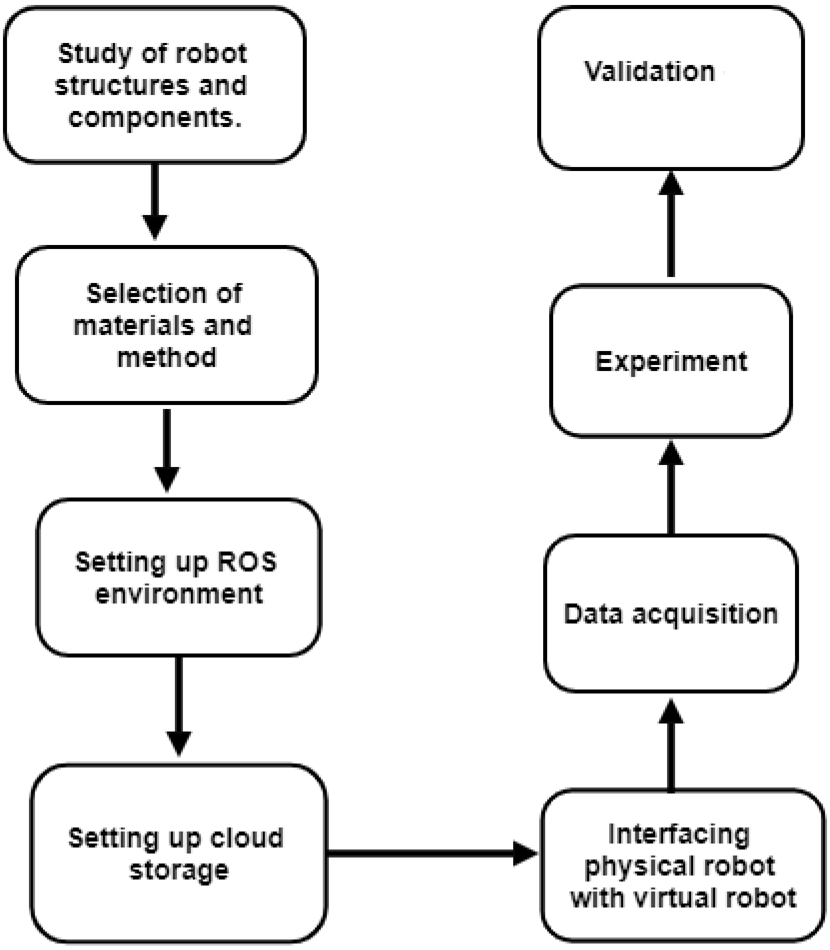


Fig. 4.1: Methodology Flow Chart.

4.1 MATERIALS AND METHODS

To develop an IoT based system Wlkata mirobot, an educational 6-axis articulated robotic arm along with raspberry pi- mini computer ,DHT-11 temperature sensor ,ADXL-345 accelerometer and AWS cloud storage account is used.

4.1.1 Walkata MI ROBOT

WLKATA Mi-robot is a platform for showcasing emerging robotic technologies, Figure 4.2. It is small, safe, and educational robot, and it has six axes, which is a required standard specification for the majority of industrial robotic applications. It has the following features:

- Lightweight and pre-assembled device: The outer shell is made of ABS engineering plastic and weighs only 1.5kg. It's a robotic arm that fits in your hand.
- Methods of Intelligent Control: Mirobot can be controlled via PC, mobile phone, Mirobot Bluetooth Controller, or APP.
- Highly Accurate: Its repeated positioning accuracy is 0.2mm, making it ideal for education and light-duty tasks.
- Multi-functional Box (Extension Module): Extend communication interface by supporting WIFI, Bluetooth, and RS485 protocols.

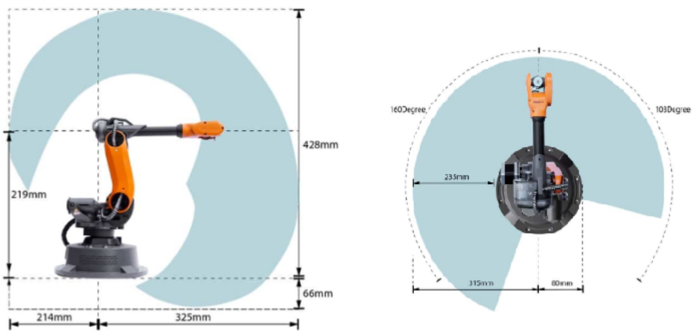


Fig. 4.2: Wlkata Mirobot.

Its significant specifications and maximum and minimum joint angles are listed in

Table 4.1: **Specification of Wlkata Mirobot**

Properties	Description
Axle Number	6+1
Maximum payload	400 gram
Type of six joint	High accuracy stepping motor + reducer
Repeated positioning accuracy	0.2 mm
working environment	0-60 $^\circ c$
Joint1	-100° + 100°
Joint2	-60° + 90°
Joint3	-180° + 50°
Joint4	-180° + 180°
Joint5	-180° + 40°
Joint6	-180° + 180°

4.1.2 Raspberry pi

The Raspberry Pi is a low-cost, credit-card-sized computer that connects to a computer monitor or TV and operates with a standard keyboard and mouse. It is a capable little device that allows to experiment with computing and learn to program in languages such as Scratch and Python. The Raspberry Pi can communicate with the outside world and has been used in a wide range of digital maker projects, including music machines and parent detectors, as well as weather stations and tweeting birdhouses with infrared cameras.

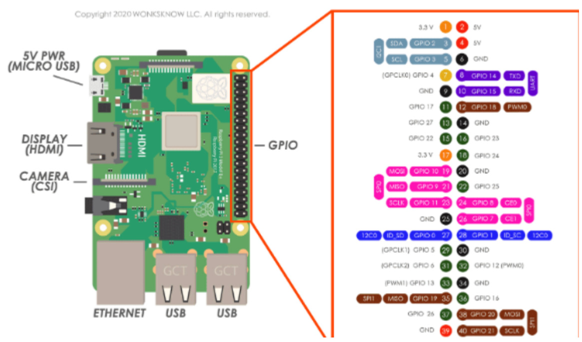


Fig. 4.3: raspberry pi.

The Raspberry Pi 4 board has 40 pins.It has four power pins, two of which are 5V pins and the other two are 3.3V pins.It can use the 5V power pins to run low power applications because they are directly connected to the Raspberry Pi's power input.

There are eight ground pins that are all linked together; any of these ground pins can be used for grounding the component. Rest 28 pins are general purpose input output(GPIO) pins labelled from GPIO 0 to GPIO 27. The GPIO pins, as their full name implies, can be programmed to be either output or input pins. Output pin values can be set and even read input pin values. The GPIO pins can be digitally programmed to be turned on and off. Any GPIO pin's output is 3.3v and can be used to control output components such as an LED or a motor. These are the most common types of Raspberry Pi 4 pins. Some of these pins also serve two purposes. For example, pin 3 or GPIO 2 can also function as an I2C pin.

4.1.2.1 Software

Raspbian, a Debian-based (32-bit) software is used. It promotes Python and Scratch as the primary programming languages, with support for numerous others. The stock firmware is freely available but closed source.

4.1.3 Temperature Sensor

DHT-11 sensor is selected to gather temperature from the base link of motor. It comprises of a thermistor for measuring temperature and a capacitive humidity detecting device. The humidity detecting capacitor consists of two electrodes separated by a substrate that may store moisture as a dielectric. The capacitance value changes as the humidity levels fluctuate. The IC calculates, interprets, and converts the modified resistance values into digital form.

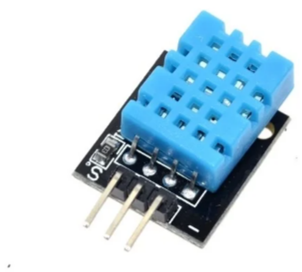


Fig. 4.4: DHT 11.

This sensor employs a negative temperature coefficient thermistor to measure temperature, which results in a drop in resistance value as temperature rises. This sensor is often built of semiconductor ceramics or polymers in order to obtain higher resistance values even for the slightest change in temperature. The specifications are summarised in Table 4.2

Table 4.2: Specification of DHT-11

Properties	Description
PCB size	22.0mm ×20.5mm ×1.6mm
Working voltage	3 or 5.5VDC
Operating voltage	3 or 5.5 VDC
Measurement Range	20-95 percent RH; 0-50°c
Resolution	8bit(temperature), 8bit(humidity)
compatible interfaces	2.543 -pin interface and 4-pin Grove

Electrical properties are summarised in Table 2 below:

Table 4.3: Electrical Properties of DHT-11

Parameter	Min.	Typical	Max.	Unit
Working voltage	3	5	5.5	VDC
Working Current	0.5	-	2.5	mA
Sampling Interval	1	-	-	s

4.1.4 Accelerometer

The sensor contains pins that may be used for I2C or SPI digital connection and three measurement axes, X, Y, and Z. The sensitivity level may be adjusted to +-2g, +-4g, +-8g, or +-16g. The greater range is useful for high speed tracking, while the lower range provides more detail for sluggish motions. The ADXL345 is Analog Devices' newest and best product. Analog Devices is renowned for producing MEMS devices of the highest calibre. A low-power, 3-axis MEMS accelerometer module with I2C and SPI interfaces is called the ADXL345. These modules' Adafruit Breakout boards have 3.3v voltage regulation and level shifting built in, making it straightforward to connect them to 5v microcontrollers like the Arduino. The ADXL345 provides 4 sensitivity settings from +/- 2G to +/- 16G.

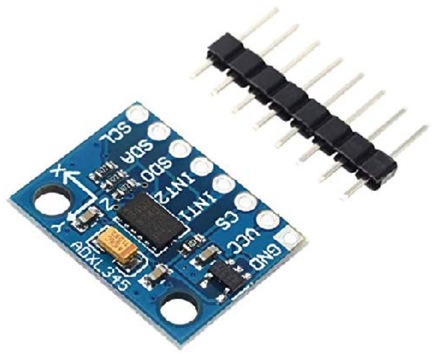


Fig. 4.5: ADXL 345.

Electromechanical principles are used by accelerometer to measure acceleration forces. These forces, can be dynamic or static depending on how the accelerometer is moving or vibrating.By measuring the static acceleration brought on by gravity, it is possible to calculate the tilt of the apparatus with respect to the earth. By detecting the dynamic acceleration, it can be checked how the object is moving.

Properties of ADXL 345 are summarised in Table below:

Table 4.4: Properties of ADXL 345

Properties	Description
Interface	I2C
Measurement Range	±16g
Resolution	The resolution of 10-bit fixed increases with the increase of g range up to 13 bits at +/-16g.
Operating voltage (VDC)	5
Operating current	23uA
Operating Temperature°	-40 to 85

4.2 DEVELOPMENT OF IoT BASED SYSTEM

The system was developed using RViz, MoveIt, and Gazebo software that run on a ROS environment. RViz is visualization software that shows the visualization of a robot after taking the joint positions from MoveIt software. The simulation of the virtual robot is superimposed in a gazebo program, a physics simulator that builds a world and simulates the robot, and tells the robot's actual behavior that is expected. The flow chart in Figure 4.6 can be referred to for a better understanding.

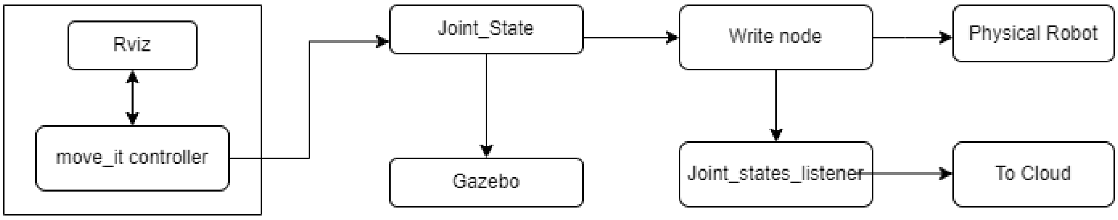


Fig. 4.6: flowchart of IoT based controlling system

4.2.0.1 Environment Setup

To install the ROS Noetic run the following commands in the terminal in the given order.

The following command enables computer to accept software from packages.ros.org

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.
d/ros-latest.list'
```

The following command setup the key

```
sudo apt install curl # if you haven't already installed curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

The following make sure that Debian package index is up-to-date

```
sudo apt update
```

This command enables us to install the ROS Noetic

```
sudo apt install ros-noetic-desktop-full
```

This command enable us to source this script in every bash terminal in which Ros is used.

```
source /opt/ros/noetic/setup.bash
```

4.2.0.2 Creating Catkin workspace

To create the Catkin workspace run the following command

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
```

To check the path of ROS run the following command

```
$ echo $ROS_PACKAGE_PATH
```

Run the following command to clone the WlkataMirobot Ros package

```
git clone:https://github.com/wlkata/RosForMirobot-master.git
```

The Wlkata ROS package comprises the RViz node, gazebo node, and `move_group` node. Two additional nodes, `write_node`, and `Joint_States_listener`, are created to establish bi-directional communication. RViz publishes messages on the subject `joint_state`, which is subscribed by the gazebo node. The joint position given in RViz is subscribed by the gazebo node, which runs the simulation and shows the position of the virtual robot. The `write_node reads` the joint position message using a callback function and sends a command to the physical robot to move to a particular position `Joint_states_listener` node is a python script that subscribes to the `write_node` and publishes the data on the cloud using MQTT protocol. Figure 4 shows the flow chart of the process.

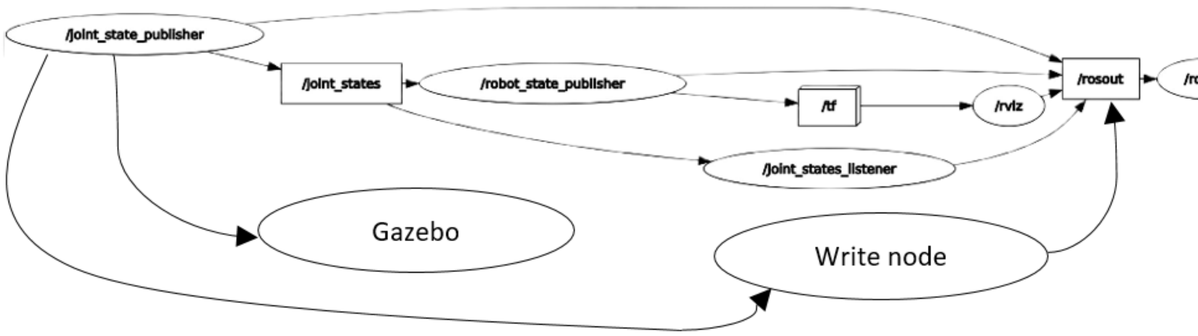


Fig. 4.7: ROS graph representing flowchart

A flowchart is shown in Figure 4.8 to create `joint_states_listener` node that listen the joint angles of the robot, the complete python script for creating the node is attached in the appendix.

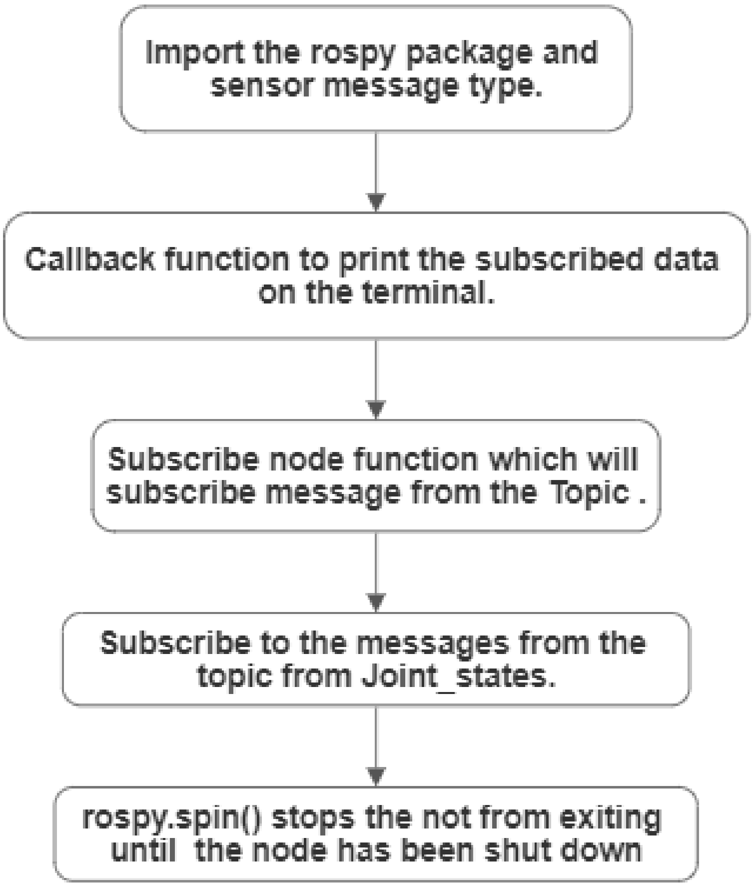


Fig. 4.8: flowchart to create listener node

Refer the below flowchart to create the `write_node`.This node subscribes to the `"/joint_states"` topic and calls the function `"angle_write_callback"` to deal with the messages of the `"/joint_states"` topic. The `"angle_write_callback"` function reads the angle values of the joints of the robot arm, translates the angle values into G code strings, sends the strings to the serial port. Complete code is attached in the appendix.

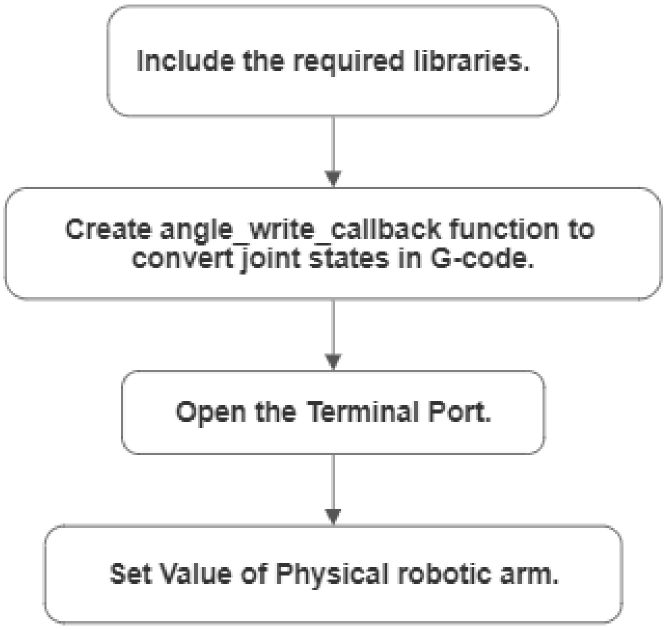
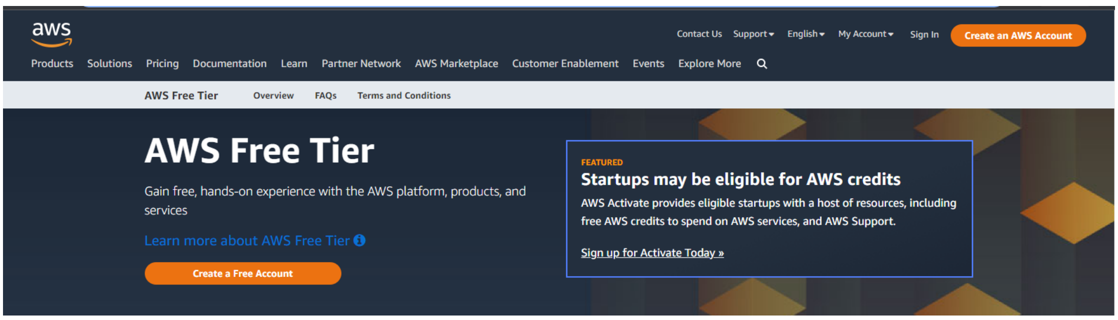


Fig. 4.9: flowchart to create listener node

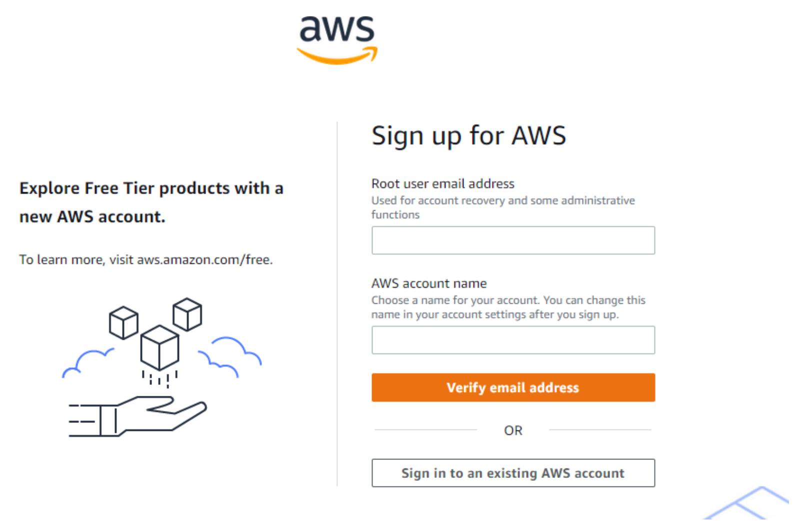
4.2.1 Setting-up of Cloud Storage

To create a cloud database a cloud storage provider is needed for experimentation. Amazon web service (AWS) is used to setup the free storage account.

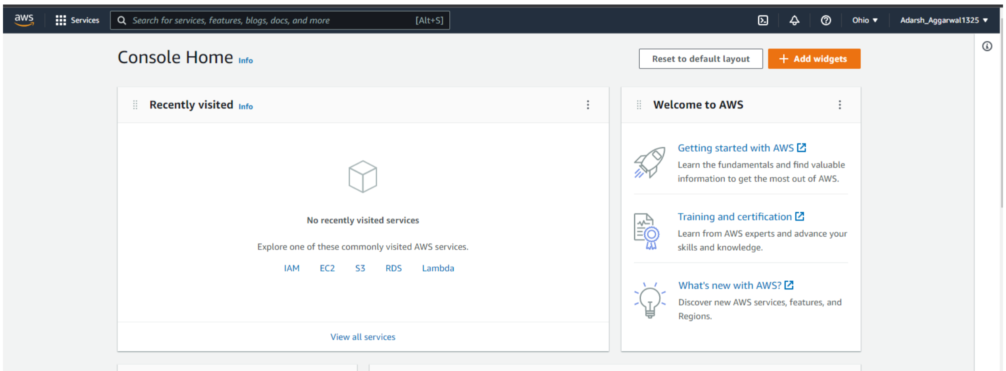
- visit $https://aws.amazon.com/$ to create an aws account



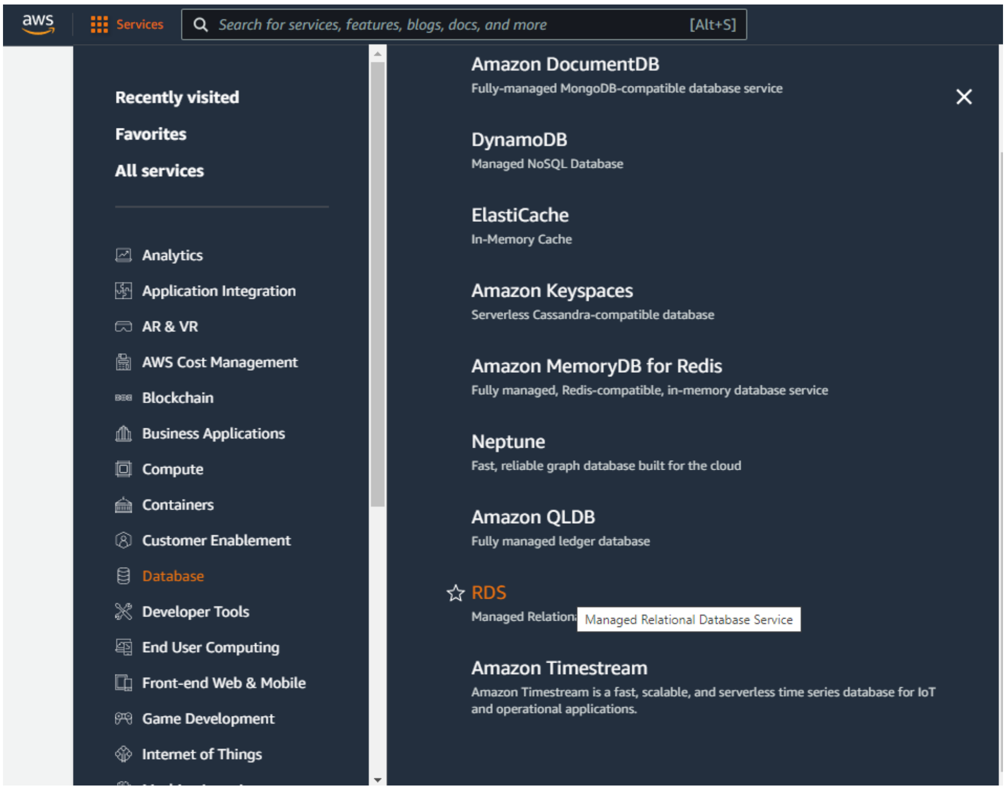
- click on create free account which will direct to aws sign up console as shown in figure below.



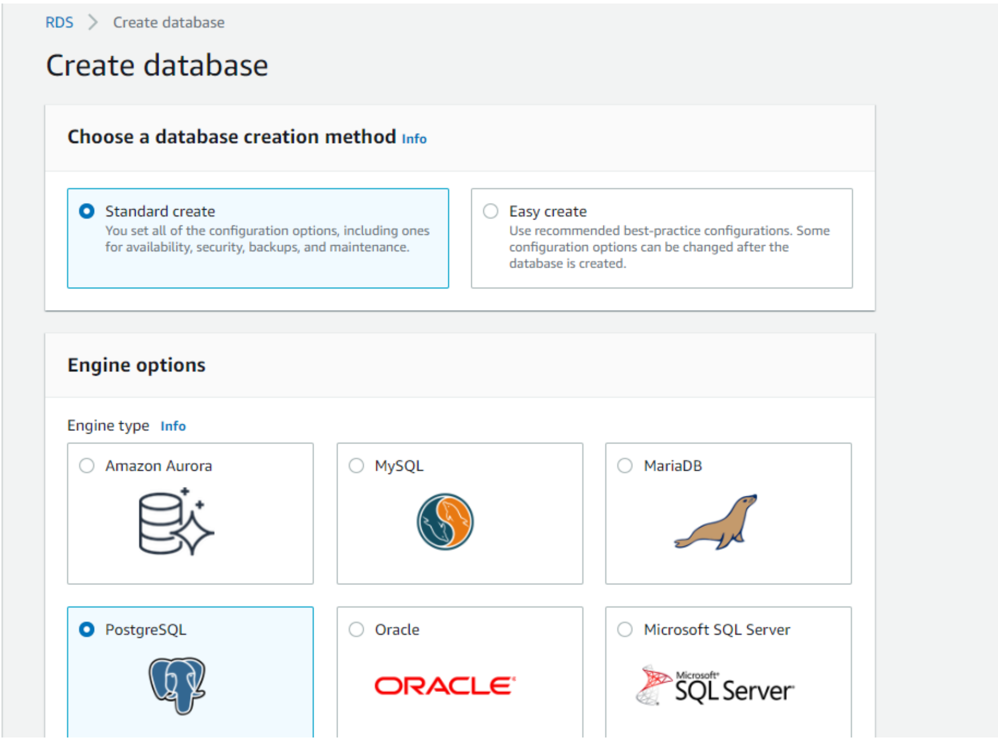
- create free account by providing asked information.
- once an account is created sign in the account which open the dashboard as seen below



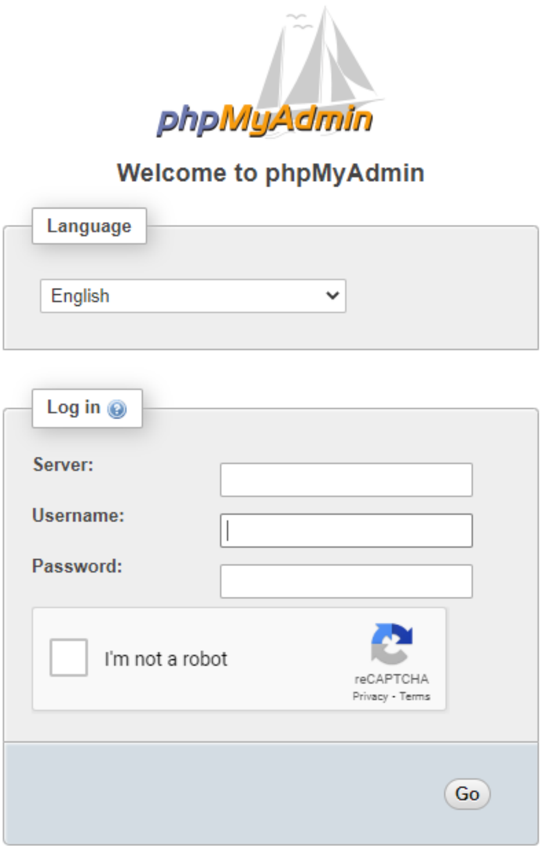
- choose Relation database service from the service tab as shown below.



- once RDS is selected a new page will open,select create database and choose the specification of your database accordingly.



- to access the database visit the link https://www.phpmyadmin.co/index.php following page will open.



- fill the asked details and database is accessible

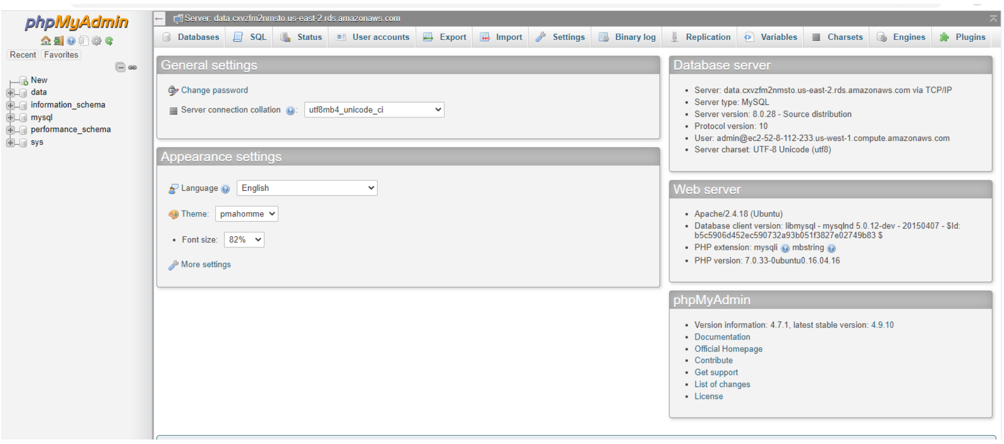


Fig. 4.10: Actual cloud database ui

4.3 DATA ACQUISITION

The physical condition of the robot is monitored using the installed temperature and accelerometer sensors. Data collected from the sensors is sent to a free tier version of the Aws cloud database for future analysis. The complete flowchart of the process can be seen in figure.

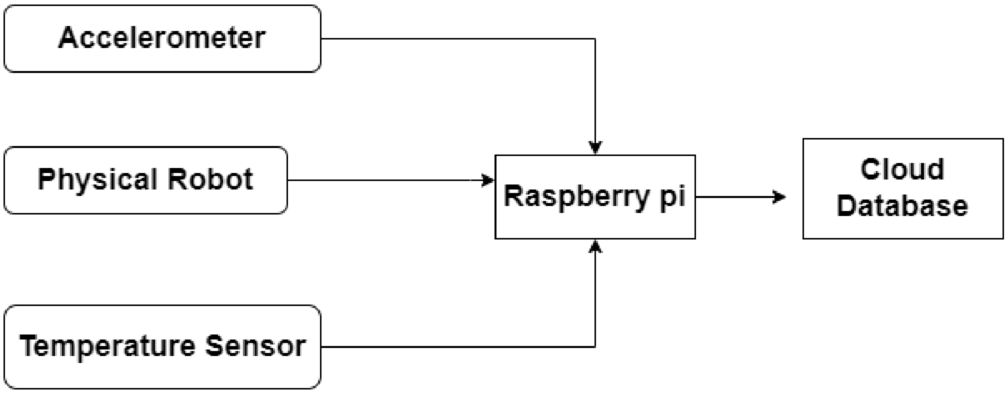


Fig. 4.11: Flowchart of data acquisition

4.3.1 Interfacing Sensors with Raspberry pi

To get the data from sensors they need to be connected to raspberry pi pin. The vcc pin is connected to power supply pin. Ground pin is connected to GND pin and output pin is connected to GPIO pins.

4.3.1.1 connection of ADXL 345

To connect the ADXL 345 following connections were made.

- GND pin of the Accelerometer is connected to Physical Pin 6 (GND) on the Raspberry Pi.
- VCC pin of the Accelerometer is connected to Physical Pin 1 (3v3) on the Raspberry Pi.
- SDA pin of the Accelerometer is connected to Physical Pin 3 (SDA) on the Raspberry Pi.
- SCL pin of the Accelerometer to is connected Physical Pin 5 (SCL) on the Raspberry Pi.

Refer to the below image for connections.

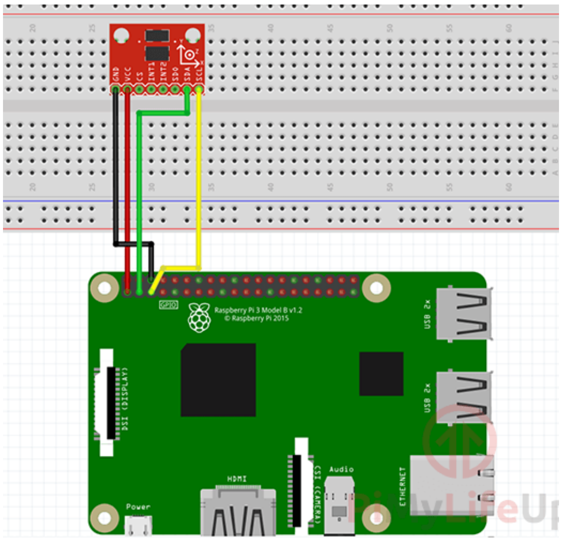


Fig. 4.12: Connection of ADXL 345

4.3.1.2 Connection of DHT-11

To connect the DHT-11 make the following connections.

- GND pin of the DHT-11 is connected to Physical Pin 6 (GND) on the Raspberry Pi.
- VCC pin of the DHT-11 is connected to Physical Pin 0 (3v3) on the Raspberry Pi.
- SDA pin of the DHT-11 is connected to Physical Pin 4 (Gpio) on the Raspberry Pi.

Refer to the below image for connections.

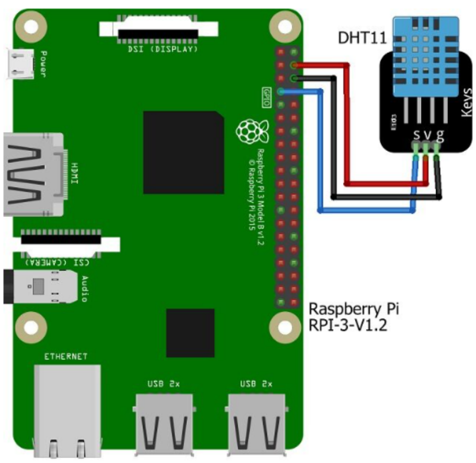


Fig. 4.13: Connection of DHT-11

4.4 connecting Database using Python

the objective of experiment is to upload sensor data in the database, the database is connected using a python script. To connect the database we need following arguments.

- Username:The username that you use to work with MySQL Server. The default username for the MySQL database is a root.
- Password: Password is given by the user at the time of installing the MySQL server. If you are using root then you won't need the password.
- Host name : The server name or Ip address on which MySQL is running. if you are running on localhost, then you can use localhost or its IP 127.0.0.0
- Database name : The name of the database to which you want to connect and perform the operations.

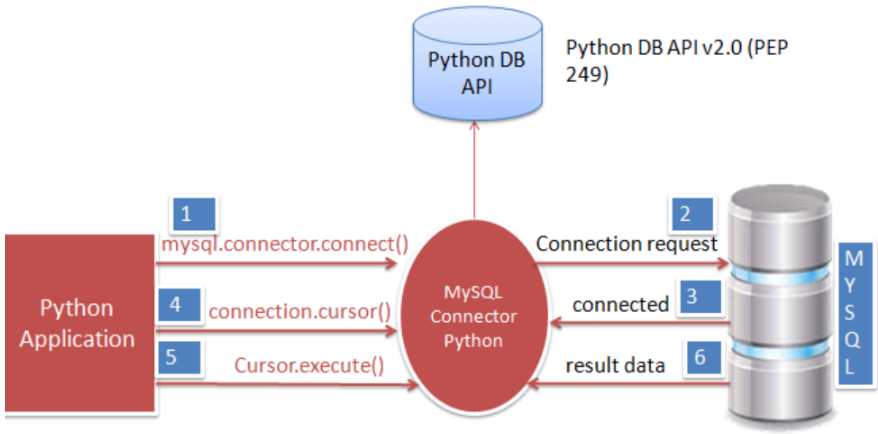


Fig. 4.14: Flowchart of mysql module

MySQL connector module will be used for python and database connectivity. To install the connector run the command pip install mysql-connector-python To use this command use import mysql.connector statement to communicate with the MySQL database.The complete code is attached in the appendix.

To gather data from sensor and uploading it to cloud storage a python script was created whose flowchart is shown in below figure and complete code is attached in the appendix section.

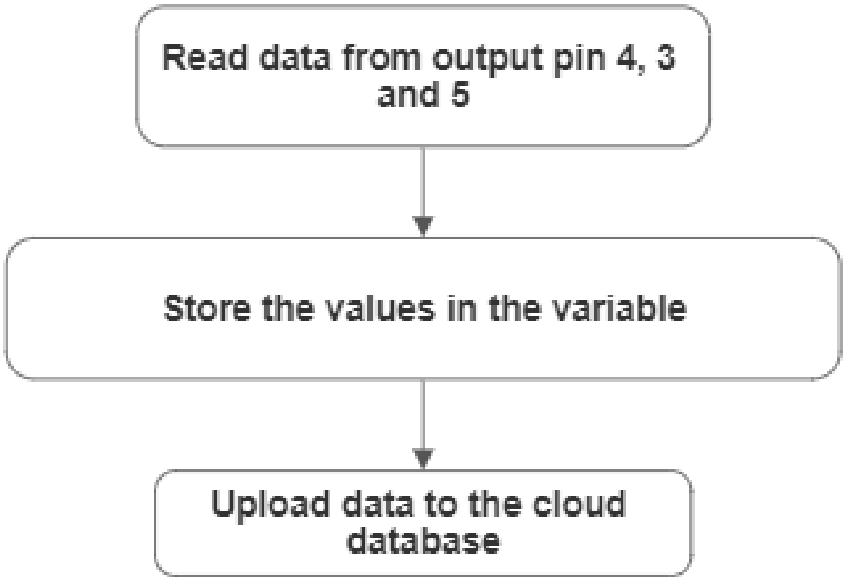


Fig. 4.15: Flowchart of mysql module

4.5 EXPERIMENTATION

This experiment aims to validate and analyze the connectivity between physical and virtual robots. The IoT based controlling system is tested, and data is accumulated using sensors. The experiment was performed in two stages. In the first stage, the values of the joint angles were given as inputs using the Joint State controller node in the ROS to set the position of the virtual robotic manipulator.A sample of the input values for the joint angles is given in Table 4.1. The same input was transmitted to the physical robot `write_node`, resulting in the physical movement of the actual robotic arm. The actual position attained by the robot was fetched in terms of its joint angles values from the robot controller using a script developed in WLKATA Python SDK.

Table 4.5: Experiment 1

Target Position	J1	J2	J3	J4	J5	J6
Position 1	60	0	0	0	0	0
Position 2	60	11	-42	0	0	0
Position 3	-41	11	-94	0	-48	0
Position 4	65	18	-50	20	-20	20

The complete setup of the experiment is shown in figure 4.16.

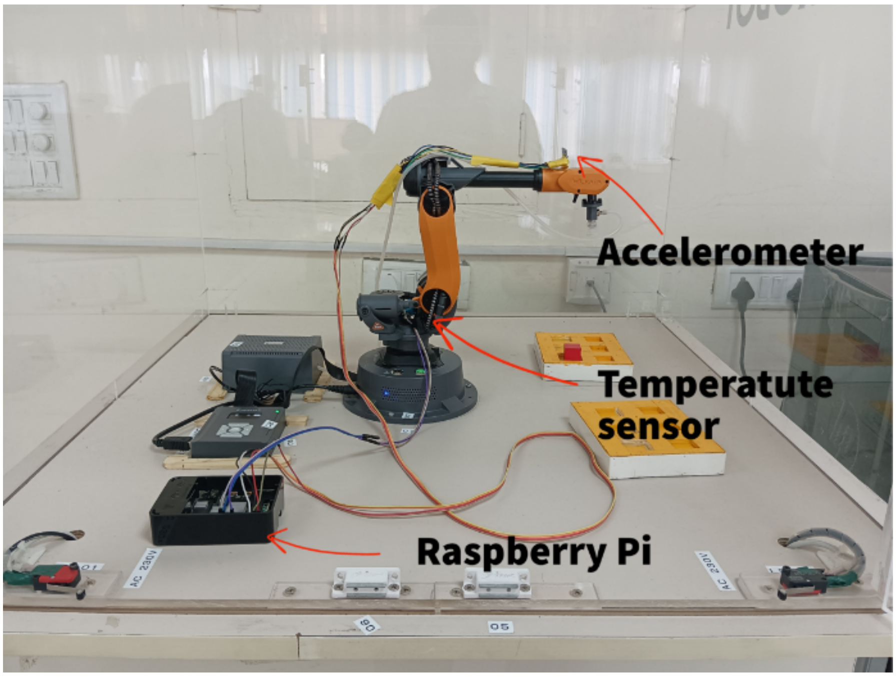


Fig. 4.16: Complete flowchart of the experiment

In the second stage, the robot was programmed to perform a pick and place task at different speeds to determine how speed impacts the temperature and acceleration of the robot placed at the base motor and upper link, respectively. Proper cooling time was given to the robot between the experiments to acquire temperature readings.A python script was created to perform a program where three different speeds were kept that were 300 mm/sec, 1500 mm/sec and 3000mm/sec The sensors' values are pushed to the cloud database hosted on a free Amazon cloud service, 'aws.'

The complete flowchart of the experiment is shown in figure 4.17. The flowchart explains how experiment is performed. The Joint values given to the virtual robot is superimposed on physical robot. The actual Joint states of robot along with acceleration and temperature is also published to the cloud storage for condition monitoring and validation.

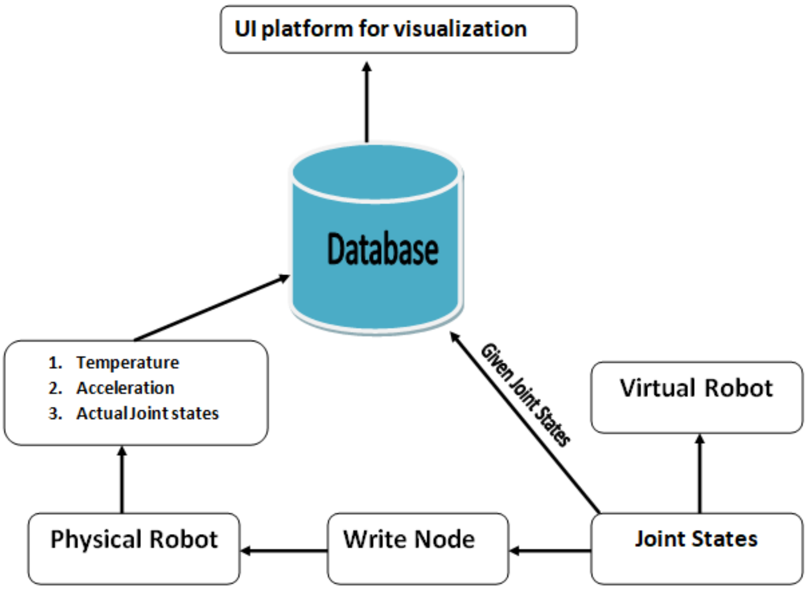


Fig. 4.17: Complete Flowchart of the Process

CHAPTER 4

RESULT AND CONCLUSION

Chapter 5

RESULTS AND DISCUSSIONS

Several experiments were performed in which different joint positions were given to the robot, starting from the robot's home position. Results of four iterations are shown below the whose coordinates are given in table 4.1 .

5.1 IoT BASED ROBOT CONTROL

The physical robot and its virtual position along with the virtual controller are shown in Figure 5.1.

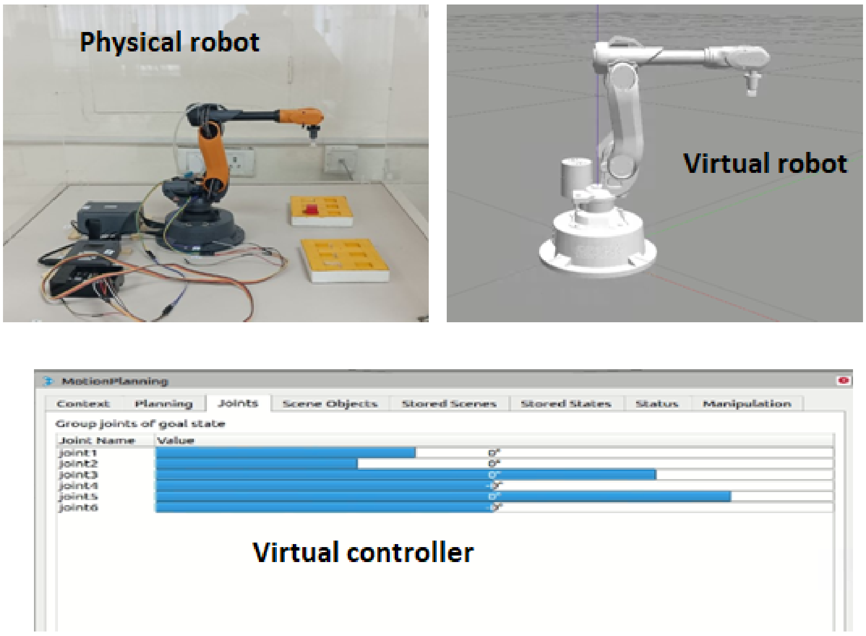


Fig. 5.1: Home position of the robot

When the joint position is changed via the virtual controller, the virtual robot moves to the given position along with the physical robot. Figure 5.2 shows the position of Virtual and physical robot after performing First iteration. The position angles were given from home position. The actual joint positions are determined using a python script and compared with the given robotic position from the virtual controller, and the absolute error was calculated.

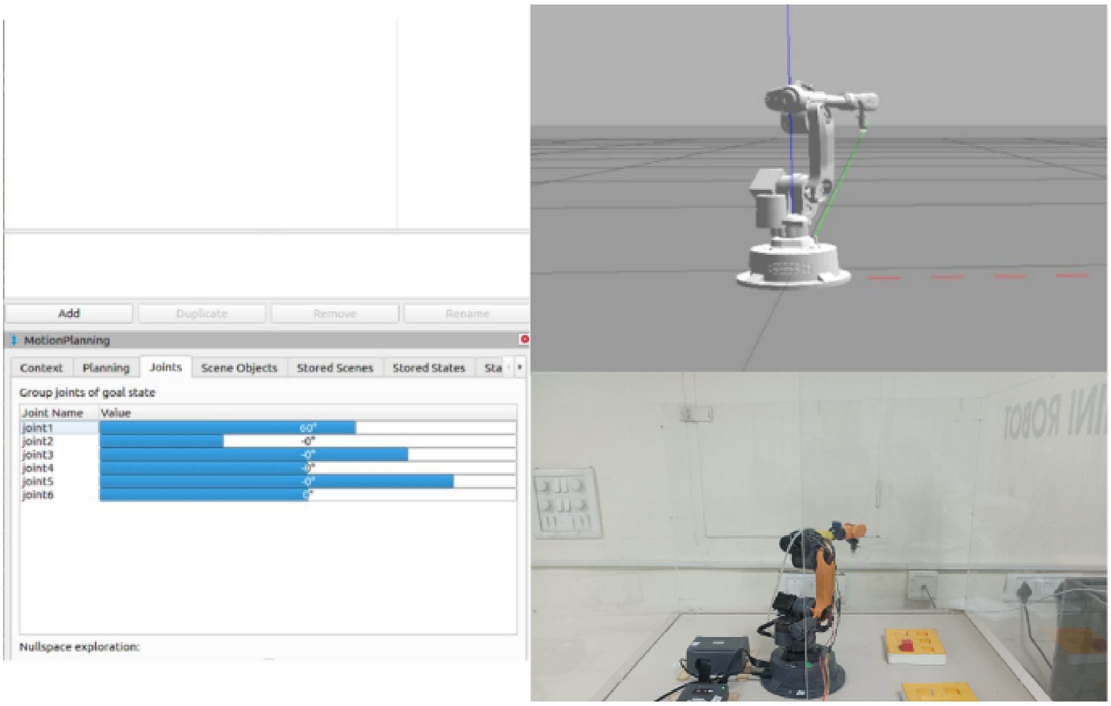


Fig. 5.2: Iteration 1

The input angle values of first iteration from the virtual controller and the actual joint angles of the Physical robot received from the python script are shown in Table 5.1. It is observed that the difference between input angle and actual angle for joint one is 0.002 degree or 0.003%.

Table 5.1: Iteration 1

joint number	Given angle	actual angle	percentage error
1	60	59.998	0.003
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0

Figure 5.3 shows the position of Virtual and physical robot after performing second iteration. The position angles were again given from home position.

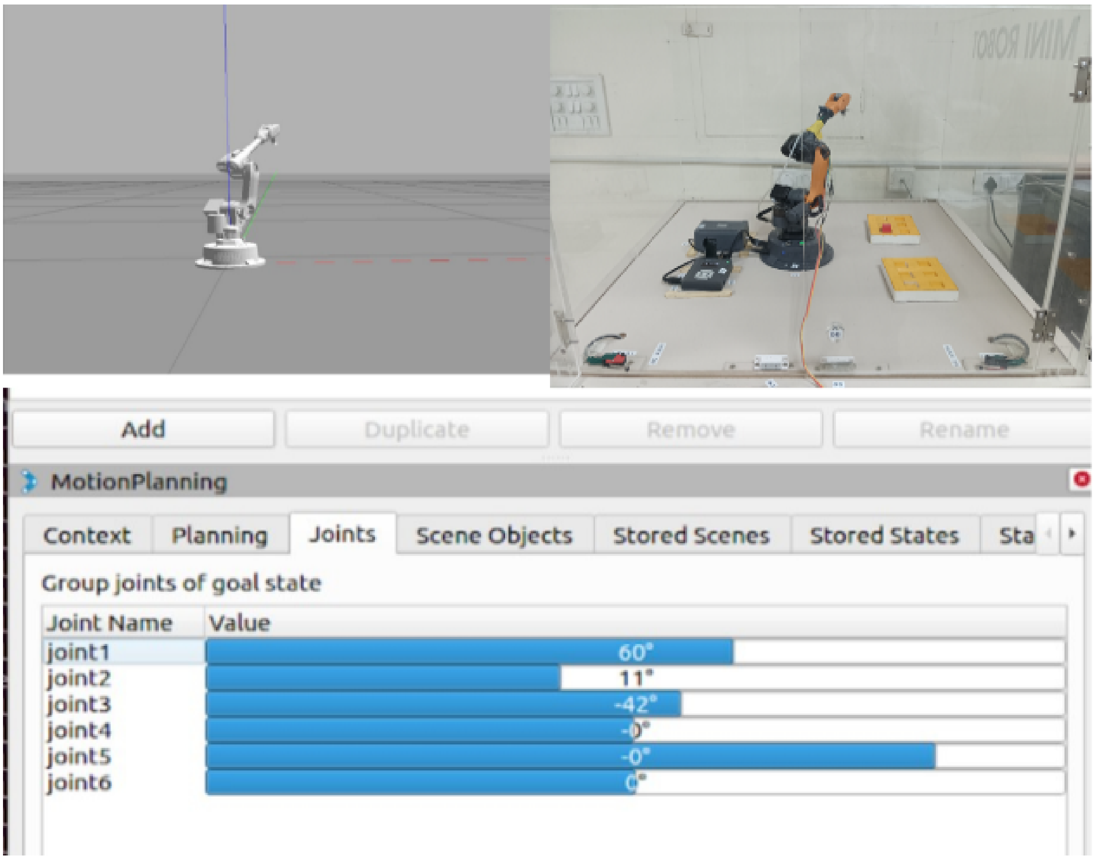


Fig. 5.3: Iteration 2

The input angle values of first iteration from the virtual controller and the actual joint angles of the Physical robot received from the python script are shown in Table 5.2. From the it can visualized that percentage error of joint one remains which is lowest when compared with joint two and three. Joint 3 has highest percentage error among three with value of 0.076 percent.

Table 5.2: Iteration 2

joint number	Given angle	actual angle	percentage error
1	60	59.998	0.003
2	11	11.002	0.0181
3	-42	-41.968	0.076
4	0	0	0
5	0	0	0
6	0	0	0

Figure 5.4 shows the position of Virtual and physical robot after performing third iteration. The position angles were again given from home position.

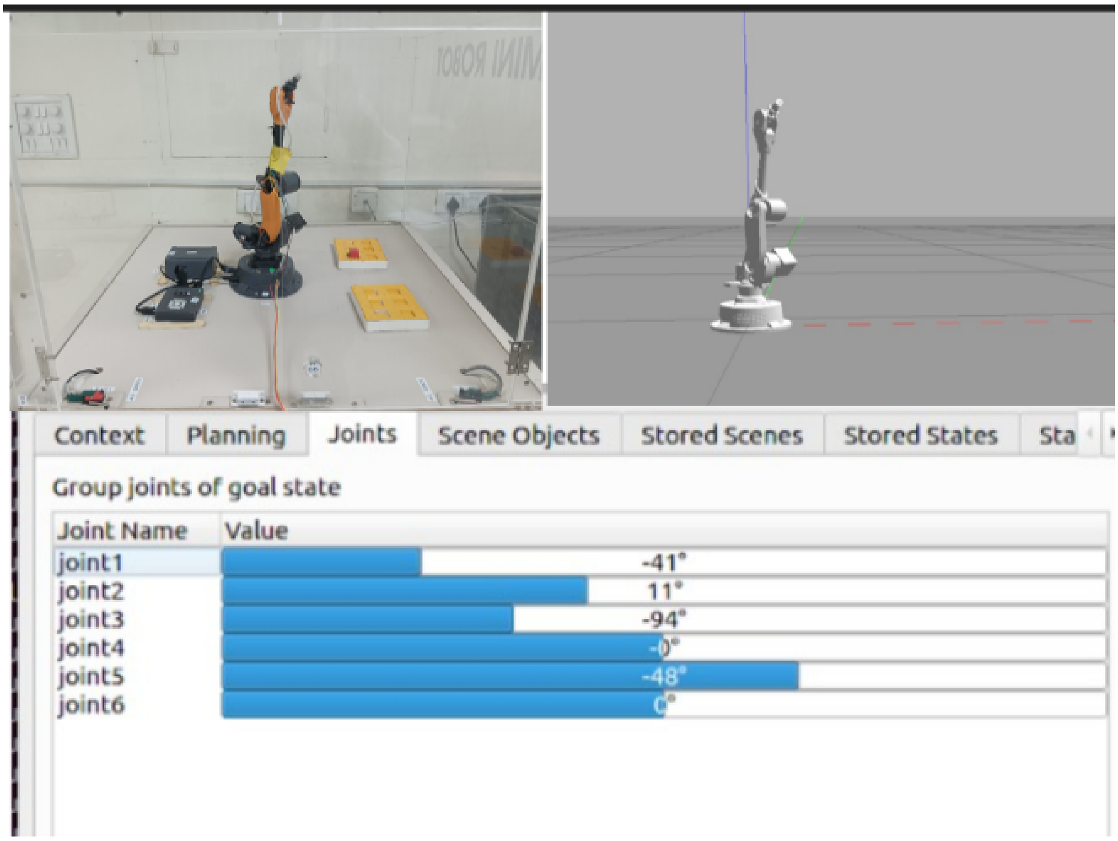


Fig. 5.4: Iteration 3

The input angle values of first iteration from the virtual controller and the actual joint angles of the Physical robot received from the python script are shown in Table 5.3. It can observed that percentage error of joint one is almost same with difference of 0.001 whereas percentage error of joint two got approximately twice of previous one i.e

0.0364 and error of joint three got reduced from 0.076 to 0.011. joint five has highest percentage error with value of 0.067 during iteration 3.

Table 5.3: Iteration 3

joint number	Given angle	actual angle	percentage error
1	-41	-40.988	0.029
2	11	10.996	0.0364
3	-94	-94.010	0.011
4	0	0	0
5	-48	-47.968	0.067
6	0	0	0

The below figure shows the position of physical and virtual robot after performing iteration four.The position angles were again given from home position.

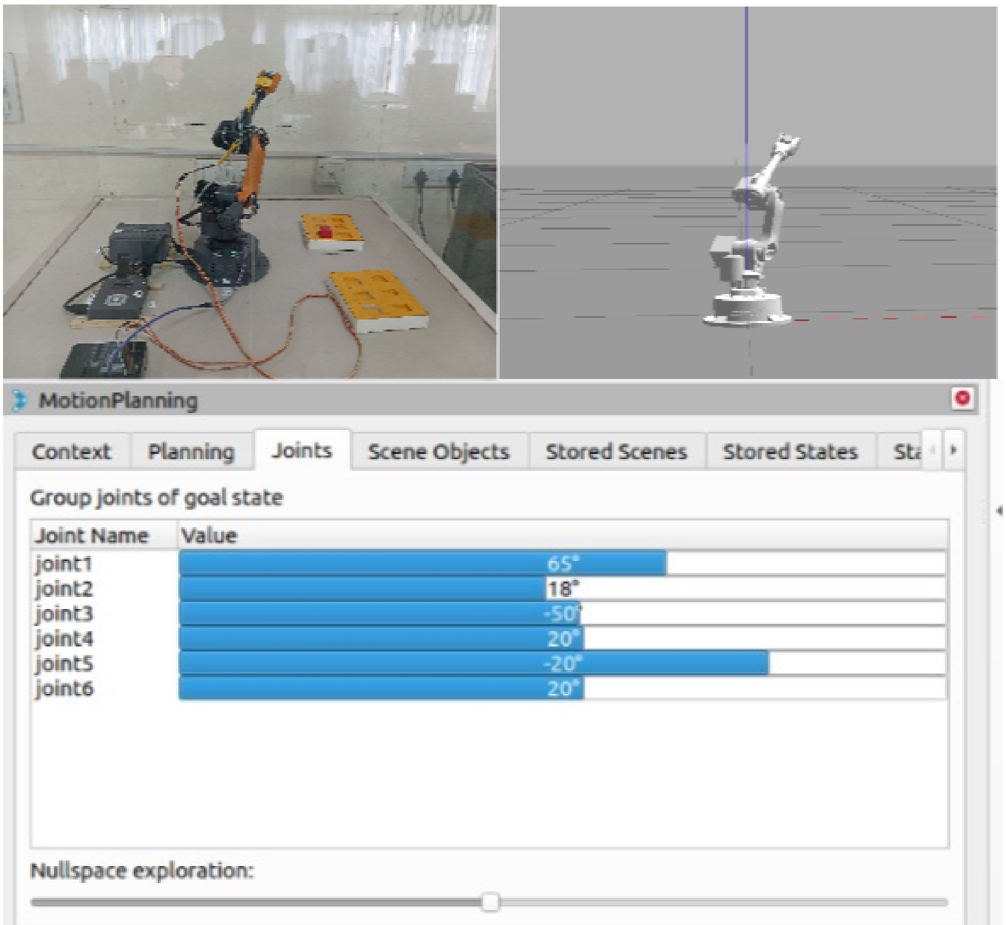


Fig. 5.5: Iteration 4

The input angle values of first iteration from the virtual controller and the actual joint angles of the Physical robot received from the python script are shown in Table 5.1. It can visualized that percentage error of joint one, joint two, joint three and joint five is same with value of 0.029, 0.0364, 0.011 and 0.067 respectively, whereas joint four and joint six has zero percentage error.

Table 5.4: Iteration 4

joint number	Given angle	actual angle	percentage error
1	65	65.011	0.029
2	18	17.968	0.0364
3	-50	-49.986	0.011
4	20	20	0
5	-20	-20.003	0.067
6	20	20.001	0

Figure 5.6 shows the graph of absolute percentage error during all four iterations. From the graph it can be observed that the maximum percentage error is of 0.075% and the average percentage error of 0.013%.

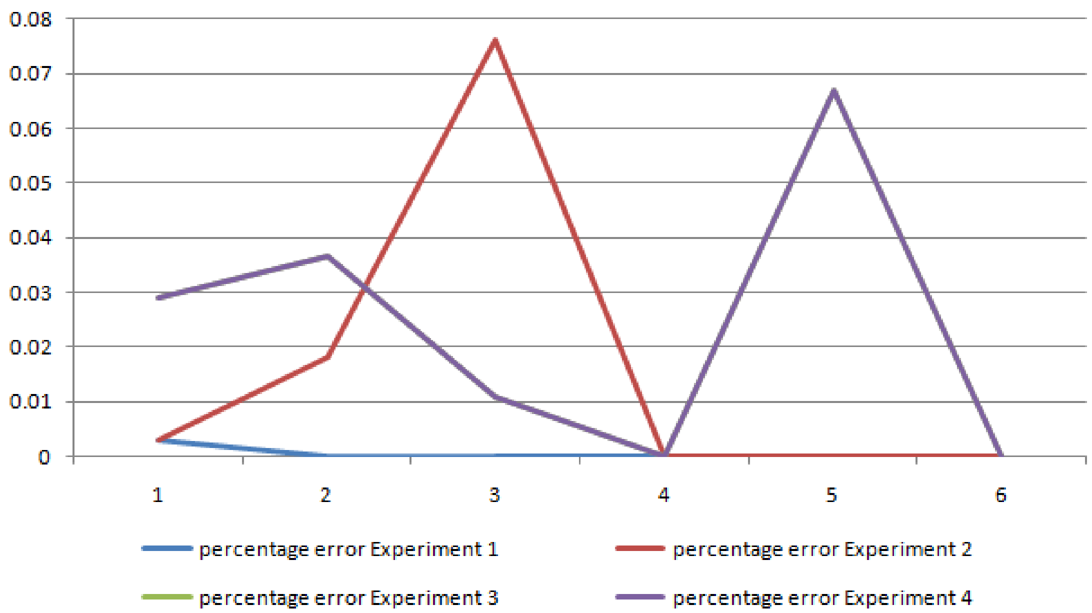


Fig. 5.6: Graph of absolute percentage error

The graph below shows the trend of absolute percentage error of different joints. From the graph, it is observed that the Joint 3 has the highest absolute percentage error of 0.076.

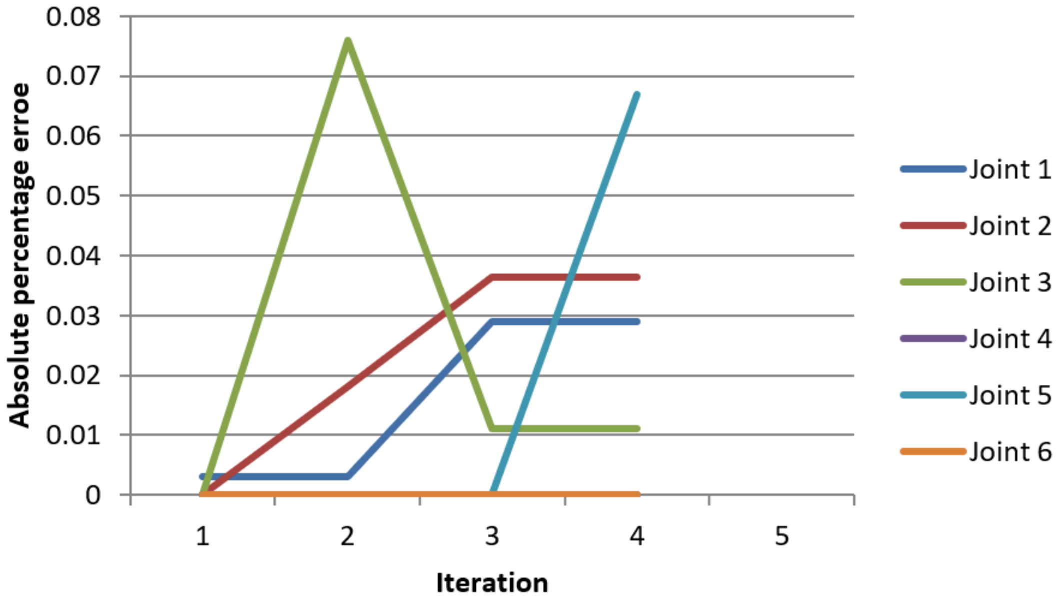


Fig. 5.7: Absolute percentage error

From the absolute errors it can be concluded that since the stepper motors are used in robot which converts a pulsing electrical current,into precise one-step movements of gear-like toothed component around a central shaft which are not highly accurate as compared to optical encoders. The error can be further reduced to zero by replacing stepper motors with encoder motors.

5.2 Condition Monitoring

In the second stage of experiments, the speed of the robot was kept at 10%, 50%, and 100% of the full speed. The values of acceleration at different speed in x,y, and z coordinates of the upper links and the temperature of the base motor at different speed is shown in tables below.

Table 5.5 shows the acceleration of X, Y, and Z coordinates at ten percent speed i.e 300mm/min of the robot. acceleration in X-coordinate is zero throughout the movement.The highest acceleration in Y-coordinate is 987.2912 mm/s^2 whereas maximum acceleration in Z axis is 169.9908 mm/s^2. From the Figure 5.8 it can be visualized that robot has constant acceleration.

Table 5.5: Acceleration at 10 percent speed

acc_x	acc_y	acc_z
0	983.4496	169.9908
0	987.2912	169.9908
0	983.4496	169.9908
0	983.4496	169.9908
0	983.4496	169.9908
0	979.608	166.1492
0	987.2912	162.3076
0	991.1328	166.1492
0	987.2912	166.1492
0	983.4496	166.1492

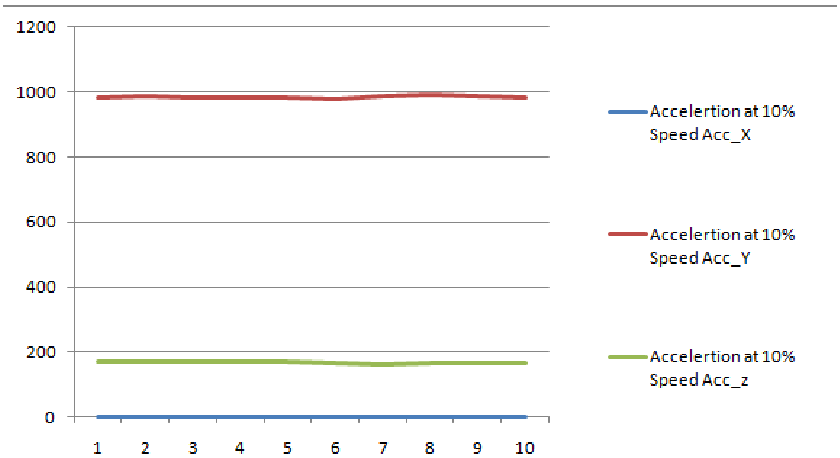


Fig. 5.8: acceleration at 10 percent speed

Table 5.6 shows the acceleration of X, Y, and Z coordinate at fifty percent i.e speed 1500mm/min of the robot. It can be seen that acceleration of Y and Z becomes almost ten times when compared to acceleration at ten percent speed of the robot. The highest acceleration in Y coordinate is 9713.101 mm/s^2 whereas maximum acceleration in Z axis is 1665.91 mm/s^2. From the Figure 5.9 it can be visualized that robot has constant acceleration.

Table 5.6: acceleration at 50 percent speed

acc_x	acc_y	acc_z
0	9637.806	1665.91
0	9675.454	1665.91
0	9637.806	1665.91
0	9637.806	1665.91
0	9637.806	1665.91
0	9600.158	1628.262
0	9675.454	1590.614
0	9713.101	1628.262
0	9675.454	1628.262
0	9637.806	1628.262

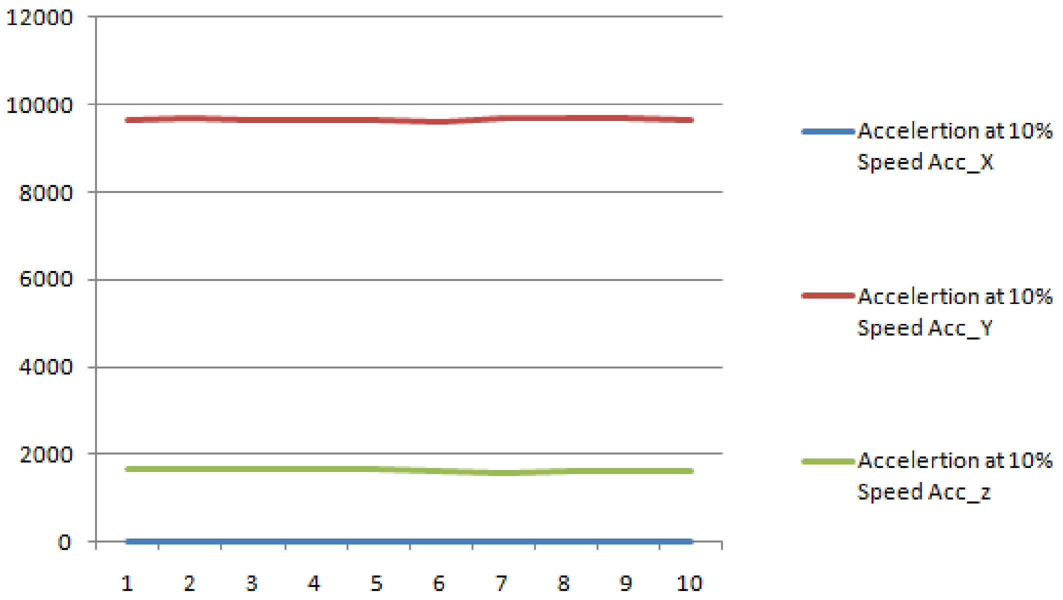


Fig. 5.9: acceleration at 50 percent speed

Table 5.7 show the values of acceleration in X, Y, Z coordinate whose graph is also plotted in Figure 5.10. It can be visualized that there is jerky motion when robot is operated at full speed i.e 3000mm/min moreover acceleration in X axis also recorded.

Table 5.7: Acceleration at 100 percent speed

acc_x	acc_y	acc_z
24.598	98	16.17
-24.598	98	16.17
-23.814	97.608	16.954
-24.206	98.392	15.778
-24.206	98	16.954
-23.422	98	16.954
-24.206	98	16.562
-23.03	98	16.954
-28.812	105.35	17.64
175.322	18.424	-8.428

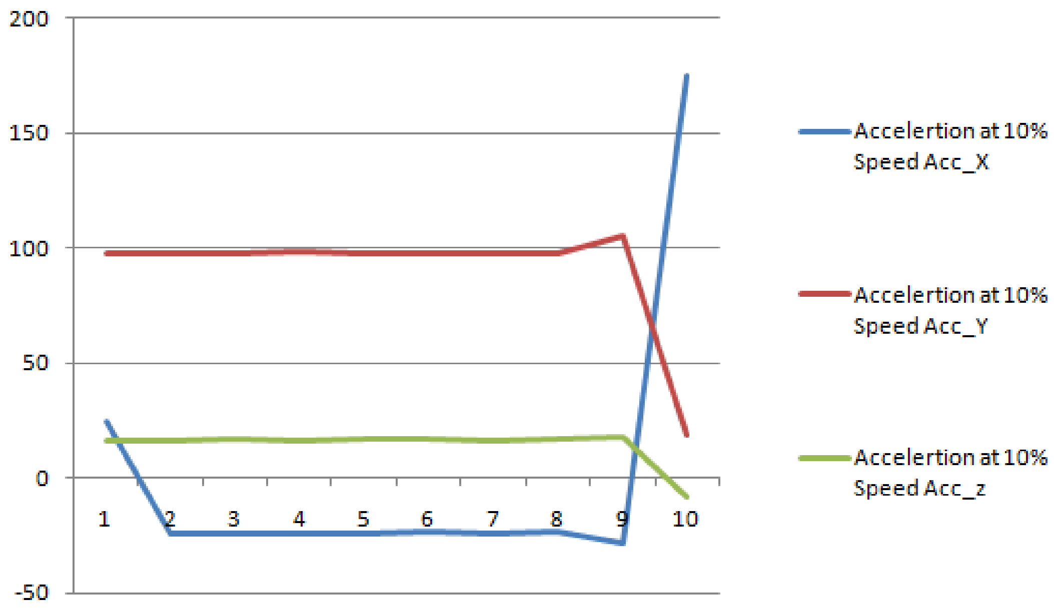


Fig. 5.10: acceleration at 100 percent speed

Table 5.8 shows the temperature of the base motor at ten percent , fifty percent and hundred percent speed it can be observed that the minimum temperature is 31° which is also the ambient temperature and the maximum temperature achieved is 34.1° when operated at full speed.

Table 5.8: Temperature at Different speed percentage

At Speed 10	At Speed 50	At Speed 100
31	31	31
31	31	31.5
31.5	31.5	32
32	32	32.5
32	32.5	32.5
32	32.5	32.7
32.5	33	33.2
32.5	33.5	33.7
33	33.5	34
33	33.5	34.1

From the Figure 5.11 it can be seen that there is not much difference between the temperature during at the difference speed as the payload is same throughout the experiment and proper cooling time was given to the robot.

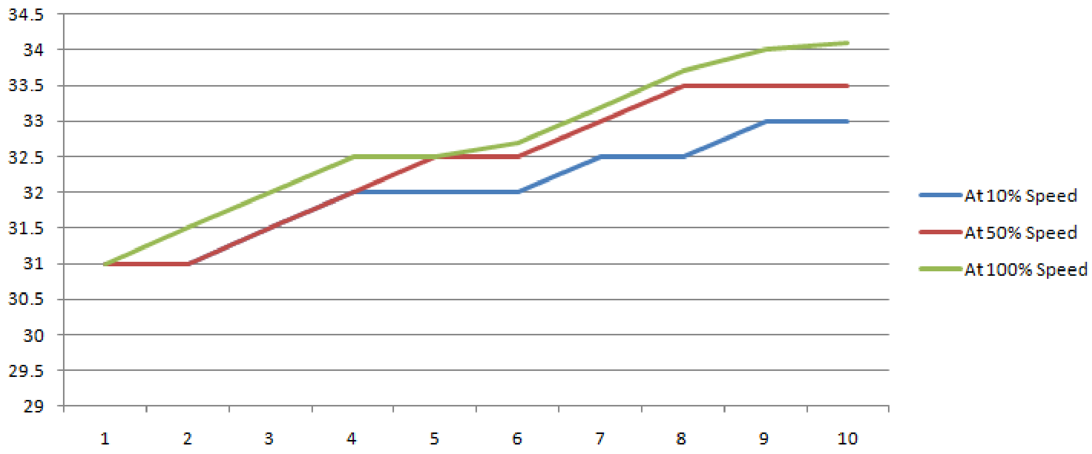


Fig. 5.11: Temperature values at a different speed

5.3 Conclusion

An open source IoT based control and condition monitoring system is developed and an attempt has been made to build a partial digital twin of an articulated robotic arm using software ROS, Gazebo, RViz, and commercially available economic hardware viz. Raspberry Pi controller and sensors for acceleration(ADXL 345) and temperature(DHT-11) measurement. The virtual model of the robot was used to operate the physical robot remotely. The input joint angles were given through the virtual controller and which were superimposed on the physical robot. After performing the experiment it can be confirmed that a robot can be successfully operated remotely. however there was some difference between the input joint values and actual robot values. Actual value of robot was determined using a python script to compare with input joint angles.The maximum percentage error in the virtual and physical robot's position is 0.08%, which is negligible. The critical health conditioning parameters such as acceleration in the upper link and temperature in the base motor were captured in real-time for different speeds and uploaded on a cloud service.

5.4 FUTURE SCOPE

Digital twin technology has genuinely amazing possibilities. The collaborative and predictive value of digital twins cannot be understated as the idea and technology advance. Machine learning algorithms can be implemented on the data so that it can be differentiated between good and bad data. This segregated data can be used to do FE- analysis which can tell the health of robot and can help in preventive maintenance.