Automated Guided Robotic Armed Vehicle (AGRAV)

Project Report submitted in the partial fulfilment

of

B.Tech Integrated
In
Mechanical Engineering

by

A007 - Sahil Bhuta
A018 - Paurash Deboo
A028 - Ryan Fernandes

Under the supervision of

Professor Giridhar Chavan

(Assistant Professor, Mechanical Department, MPSTME)

SVKM's NMIMS University

(Deemed-to-be University)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING (MPSTME)

Vile Parle (W), Mumbai-56

(2023-24)

CERTIFICATE

This is to certify that the project entitled **"Automated Guided Robotic Armed Vehicle (AGRAV)"**, has been done by Sahil Bhuta, Ryan Fernandes and Paurash Deboo under my guidance and supervision & has been submitted in partial fulfilment of the degree of BTI in Mechanical Engineering of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

Project mentor (name and Signature) | Examiner (name and Signature)

(Intenal Guide)

Date:

Place: Mumbai

(HoD) (name and Signature)

ABSTRACT

This paper presents a prototype of a model which consists of a robotic arm on top of an automated guided vehicle. The problem statement kept in mind was “How to increase productivity in industries while reducing repetitive work and human involvement in pick and place applications.” The main problem in industries and warehouses where humans operate to pick and transport objects is that the working time is limited to human working hours. Robotic arms play a crucial role in automation as they assist in tasks like picking objects and move them from one place to another. However robotic arms are primarily fixed at a single location and cannot transport objects from one place to another. A serial robotic arm is one where the joints are arranged in series so that each joint is connected sequentially to the previous one to allow precise and controlled movement, making them useful for various applications, like manufacturing, assembly, and automation processes. Thus, placing a serial robotic arm on top of an AGV increases the mobility of the arm and it can facilitate the transport of objects. Hence the topic of this project; Automated Guided Robotic Armed Vehicle (AGRAV). The design of the model will be done on Fusion 360 software after taking into consideration the calculations of torque acting on the various links of the robotic arm along with the factor of safety of the model. This will be followed by a finite element analysis of the model based on the stresses and deformations occurring in the model. The model uses an Arduino microcontroller and sensors which will enable line-controlled motion. In addition, a Radio Frequency Identification (RFID) System is used with an RF Reader on the robotic arm and RF tag on the objects to be handled for the arm to detect which objects to be handled at a time. This paper also explores the possibility of a scaled-up version of this model which can be implemented in industries.

Keywords-

Serial Robotic Arm, Automated Guided Vehicle, Material Handling, Warehouses, RFID system

Table of Contents

List of the figures-

List of tables-

Chapter 1

Introduction

1.1 Background of the project topic

Robotic arms are mechanical devices with multiple joints and end-effectors designed to mimic the movements of a human arm. They are widely used in industrial automation for tasks such as assembly, welding, painting, and packaging. These arms are controlled by computers and can perform repetitive tasks with high precision and speed, enhancing efficiency and reducing human labor in manufacturing processes.

Automated Guided Vehicles (AGVs) are mobile robots used for material handling and transportation within industrial environments. They navigate through predefined routes using various guidance systems such as magnetic tape, line detection, lasers, or vision-based systems.

Both robotic arms and AGVs play crucial roles in modern manufacturing and logistics, contributing to increased productivity, safety, and flexibility in industrial operations. Thus, our report examines and explains how we can integrate a robotic arm on top of an AGV to further increase productivity in industries.

1.2 Motivation and scope of the report

The motivation behind our project and report is to increase efficiency and productivity in industries. We are currently studying a subject called Industrial Engineering where we have learnt different methods to examine productivity in given industries. We researched about the process of material handling in industries where pick and place applications are required, and found out some limitations. Thus, our report covers these issues and how our model can help overcome them.

1.3 Problem statement

“How to increase productivity in industries while reducing repetitive work and human involvement in pick and place applications.”

Chapter 2

Literature survey

2.1 Introduction to overall topic

Robots are nowadays being used in a wide variety of industrial applications. The main purpose of robotic equipment is to increase the accuracy, speed and reliability of production. It should be able to provide a considerably higher efficiency than humans. Robotic arms have various applications in different sectors some of which include-

1] Packing and Logistics- In the food and beverage, pharmaceutical, and consumer goods industries, robotic arms are employed for packaging products into boxes, cartons, or containers. They can also palletize goods for storage and transportation, increasing throughput and reducing manual labor.

2] Medical Field- Robotic arms play a crucial role in minimally invasive surgery, where they enable surgeons to perform precise movements with small incisions, leading to faster recovery times and reduced risk for patients.

3] Agriculture- Robotic arms can automate the process of planting seeds or transplanting seedlings into the soil with precision and consistency. This reduces labor costs and ensures proper spacing and depth for optimal growth.

4] Electronics- Robotic arms are utilized in the assembly of electronic components such as circuit boards and small devices. Their high precision and dexterity make them ideal for delicate tasks like soldering, placement of components, and quality inspection.

5] Automotive Industry- In car manufacturing, robotic arms are used for tasks like welding car bodies, installing components, painting, and handling materials along the production line. They enhance efficiency, quality, and safety in the assembly process.

6] Material Removal- Robotic arms equipped with tools like grinders, routers, or water jets are used for material removal processes such as deburring, trimming, and cutting in industries like aerospace, metal fabrication, and woodworking.

7] Manufacturing- Robotic arms are extensively used in manufacturing processes for tasks such as welding, painting, assembly, and material handling. They can operate with high precision and repeatability, improving product quality and consistency while reducing production time and costs.

However, the most common and majorly used application of a robotic arm is as a pick and place device in industries and warehouses[1].

2.2 Literature Review

Robots are used in a wide variety of industrial applications. The main purpose of robotic equipment is to increase the accuracy, speed and reliability of production. It should be able to provide a considerably higher efficiency than humans. There are two types of robotic arms based on their structure: Serial Robotic Arm and Parallel Robotic Arm. A Serial Robotic Arm has multiple links connected by joints in a chain-like structure. These joints allow the arm to rotate or move in specific directions allowing precise manipulation of objects within its workspace. These are used in industrial automation, assembly lines, pick-and-place tasks and other such applications. A parallel robotic arm on the other hand has multiple links connected to a common base by separate joints at one end of each link. It has fewer degrees of freedom as compared to serial robotic arms and is mainly used in high speed and force requirement applications like machining, 3D printing, packaging, etc. For material handling in warehouses, a serial robotic arm has to be used because in this case the arm has to pick objects from shelves, i.e. precise vertical motion is needed, and serial robotic arms have more ease of vertical motion as compared to parallel robotic arms. In addition, it is not so practical to fix parallel robotic arms on AGVs because of their complex structure.

Fig 1: Parallel Robotic Arm

Fig 2: Serial Robotic Arm

In big manufacturing plants some processes are repetitive which cannot be done at a high speed by humans. Thus, these processes can be automated by the use of robotic arm. Robotic arms also play a key role in increasing the efficiency and productivity of a manufacturing process. A pick and place robotic arm is a device used to grab objects (especially heavy ones) and move them to another place.

For example, consider a manufacturing process that has different conveyors to carry out different operations. The robotic arm carries the workpiece from one conveyor to another at room temperature in order to reduce human efforts and efficient material handling[2].

Thus, the picking and placing application of a robotic arm reduces human effort and increases productivity. The arm should be re-programmable according to the applications it will be used for.

The use of automation in every field works, is not something alien to industries in today's world. Material handling is one of the most important factors affecting productivity and by improving its efficiency, total productivity will increase. It's in warehouses and transportation where business turns over, where minutes are made and money is saved around the clock. Warehousing, along with inventory activities, accounts for about 50 percent of the overall logistic costs. This includes mainly transporting the raw materials in a warehouse to the required vehicle (mainly trucks) which takes it to the workshops, transporting the product, in the process of being made in a workshop, again to such vehicles that carry them to other workshops if necessary and also in inventories to sort out the final products into their respective selling trucks, vans, etc[3]. For these purposes the most commonly used method is human labor which is a method being used since ancient times wherein at least 2 or 3 workers are needed to transport the material at various stages. Moving bulky or heavy items from one place to another is mainly done by using equipment like forklifts, conveyors and cranes. Human involvement not only costs a lot due to the high cost of labor but also, increases the time taken for material handling and indirectly for the total production by a lot. In addition, fewer production rates could be caused as there are chances of accidents occurring as well due to the carelessness and distractions in the human mind which could lead to injury of humans as well as broken materials[4].

A real-life implementation by Amazon in their warehouses-

Amazon has been serving as the largest online retailer in the last decades. The most important reason for this is that it has been a front runner in the transition to automated warehousing systems by making continuous improvements by leveraging the developing technologies, resulting in the successful operation of one of the most complex supply chains. The same day delivery and the next day delivery options offered by Amazon takes the competition to a higher level, and is quite effective in terms of attracting customers. The main objective of Amazon is to deliver a wide range of products to the customers with a reasonable prices and fast deliveries. To achieve these goals, a large scale of products at the right quantity must be kept in inventory and distributed at low costs. In Amazon products where the profit margin is as low as 1.7%, warehouse and distributions costs stand for 13.4% of the total cost. Thus, a slight improvement on warehousing and distribution system can make a big difference on total profit. For this reason, it is essential for Amazon to maintain its profitability by using automation technologies to facilitate warehousing operations and to complete them at a lower cost. Amazon has made great strides in the development of automation in its warehouses with the robotic company Kiva Systems which it purchased in 2012. Although initially selling robots to different companies, Kiva

System started to develop robots for only Amazon after some time. By the help of robots developed in Kiva, most of the material handling activities such as order picking, packaging, etc., have been automated. These robots (see Figure 3), 40 cm and 145 kg can carry approximately 300 kg of load. Since 2016, 45,000 robots have been serving in Amazon’s 20 logistic centers. All these implications result in 20% saving of warehousing costs. Amazon is expected to save $22 million from each automated warehouse. Addition to all benefits on performance, these robots also offer the most environmentally friendly solution to the warehouse operations. They can work with low energy and no need for light, resulting in 30% saving in total energy consumption in a warehouse[5].

Fig 3: Kiva Automated Robot

The main objectives of an AGRAV include: To enable working 24/7; to decrease the overall and long-term operational cost; to reduce the possibilities of work accidents; to reach vertically stored shelves with ease and finally to integrate a pick and place system that can transport the packages by itself.

Due do these demands, the in-house logistics sector is the focal point of physical automation. Also, the rapid advances in the robotic sector, computer vision, artificial intelligence, and edge computing capabilities lead to the formation of machines that can potentially think, see, hear and move at a faster rate and higher efficiency than human handlers[4].

Chapter 3

Methodology and Implementation

3.1 Block diagram

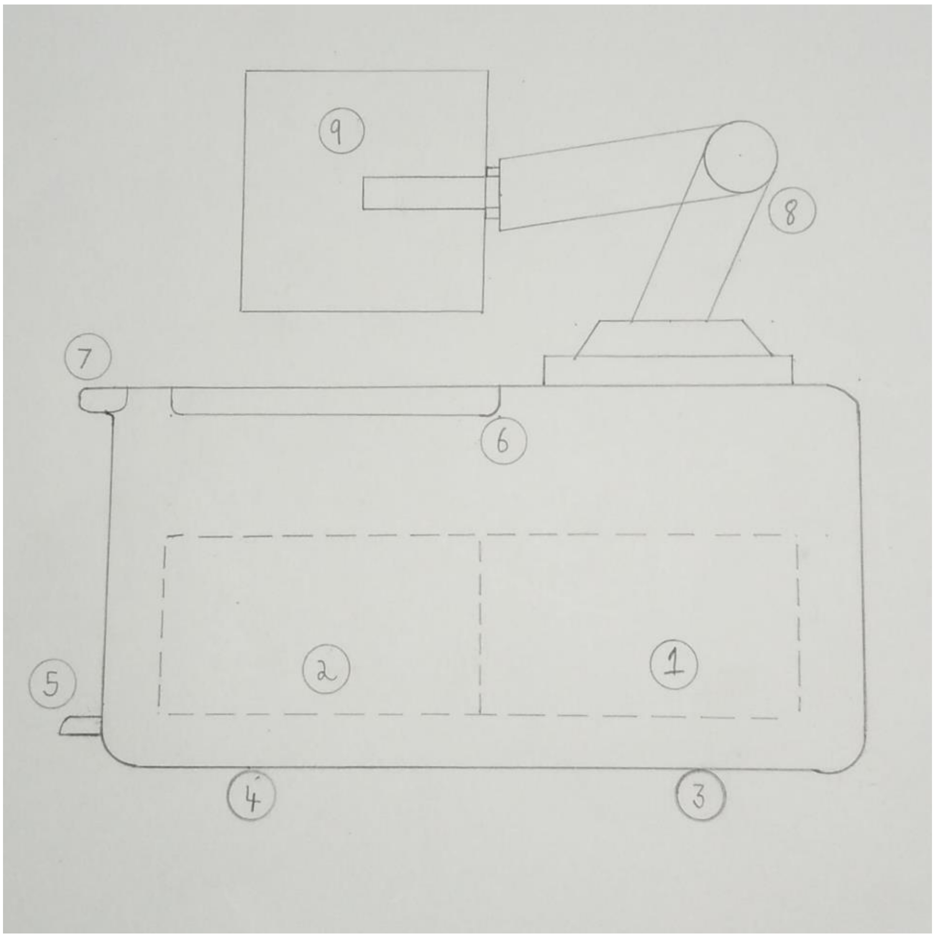


Fig 4: Block Diagram of Model

1) Arduino Mega
2) Battery
3) Rear wheels and bo motor
4) Front wheels and bo motor
5) Ir Sensor
6) Provision for accommodation of material
7) Ultrasonic sensor
8) Robotic arm
9) Material to be shifted

3.2 Hardware description

CAD Model-

Autodesk Fusion 360 was used to make the robotic arm as well as the body of the AGV after which they were assembles together. Holes of suitable diameters were added to the model for making attachments at various places and for fitting motors in it.

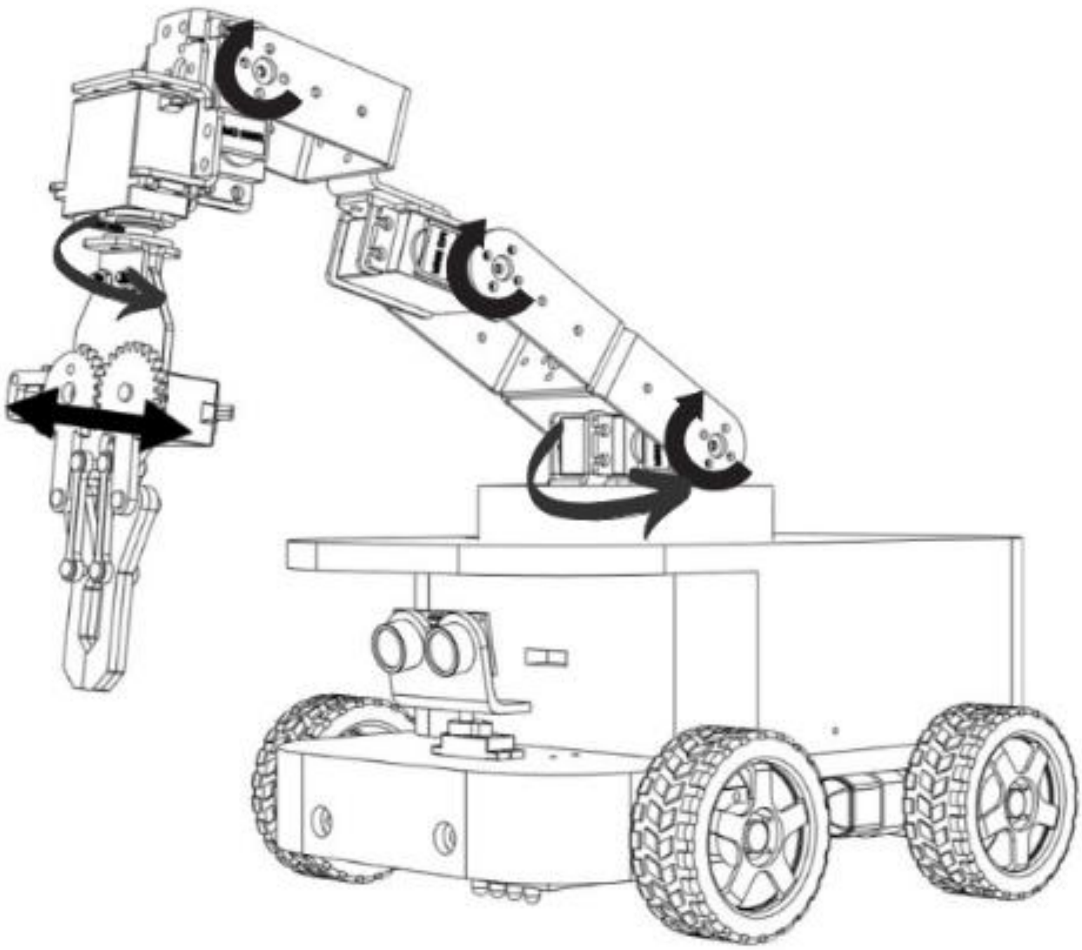
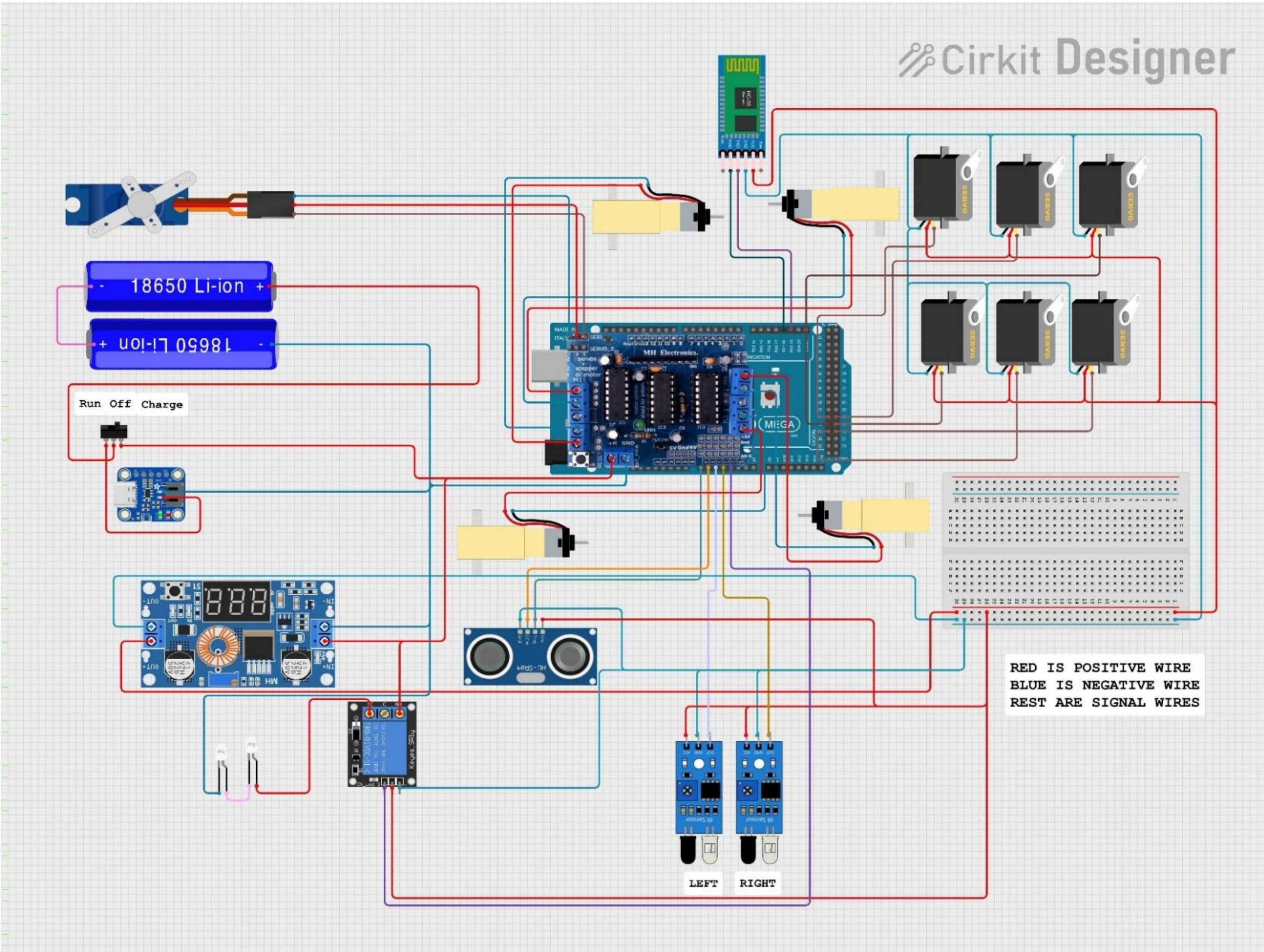


Fig 4: AGRAV with 6-axis Rotation of Robotic Arm

Materials used-

A light and strong material is suitable to fabricate the structure of the arm. This is essential for the servo motor to pull up and turn the arm accordingly. Some materials that could be considered to were Aluminum, Perspex, Plastic Polymer and Carbon Fiber. Considering the aspects of availability, overall cost and flexibility, among the four materials considered, aluminum was found to be the best material for this job. An Aluminum Alloy was specifically used for the Robotic Arm and Polylactic Acid (PLA) was used for the body of the AGV as it was 3D Printed.

Schematic of Electrical Connections:-



Reach of the Robotic Arm-

The arm has a horizontal reach of 15cm from the outer most edge of the AGRAV and it can pick an object at a height of 25cm to 30 cm from the ground depending upon the distance between the AGRAV and the shelf (from which object will be picked)[6].

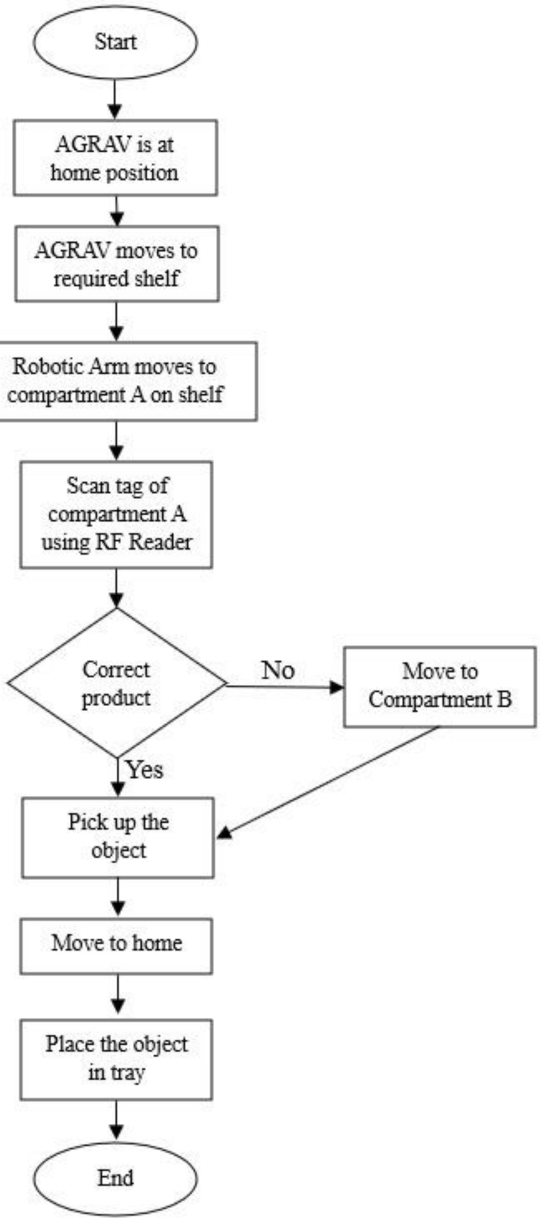
Components and their Functions-

COMPONENT	FUNCTION
Arduino Uno rev3	Microcontroller board
Arduino screw terminal	Wire Connection
Arduino Motor Shield	Motor control
400 Points Half Size Solderless Breadboard	Prototyping circuits
2 pcs 18650 2600mAh Lithium-Ion Battery	Power storage
2 x 18650 Battery Holder Case	Battery storage
18650 Polymer Lithium-ion Charger Type C to 4S 16.8V 2A Booster Module	Battery charger booster
Jumper wires	Connections, bridging
PCA9685 Servo Motor Driver	Converts 6 signal pins into 2
Ultrasonic Distance Sensor Module - HC-SR04	Distance measurement
Rocker Button Mini switch DPDT 3 Position ON-Off-ON	Toggle switch
Mounted Transparent Case Holder Bracket for Ultrasonic HC-SR04	Mounting Bracket
Servo Motor Micro SG90-360 Degree Continuous Rotation	Precision positioning of Ultrasonic Sensor
MG996R Digital Metal Gear High Torque Servo Motor (360 Degree Rotation) 6pcs	Precision positioning
Aluminium Alloy Based 6DOF Robotic Arm Brackets and 3d printing material	Structural Support, high strength
Robotic Gripper	Object manipulation
DC BO Motor Dual Shaft Smart Car Robot Gear Motor Pack of (4)	Drive mechanism.
80mm-A Mecanum Wheel Compatible with 6.7mm Coupling (Pack of 4)-Black	Omnidirectional Mobility
Infrared Obstacle Avoidance IR Sensor Module	Detects obstacles
1 watt led with heat sink 2pcs	Cooling LEDs

5V 1 Channel Relay Module	Control high voltage
RFID System	Scanning and Identification

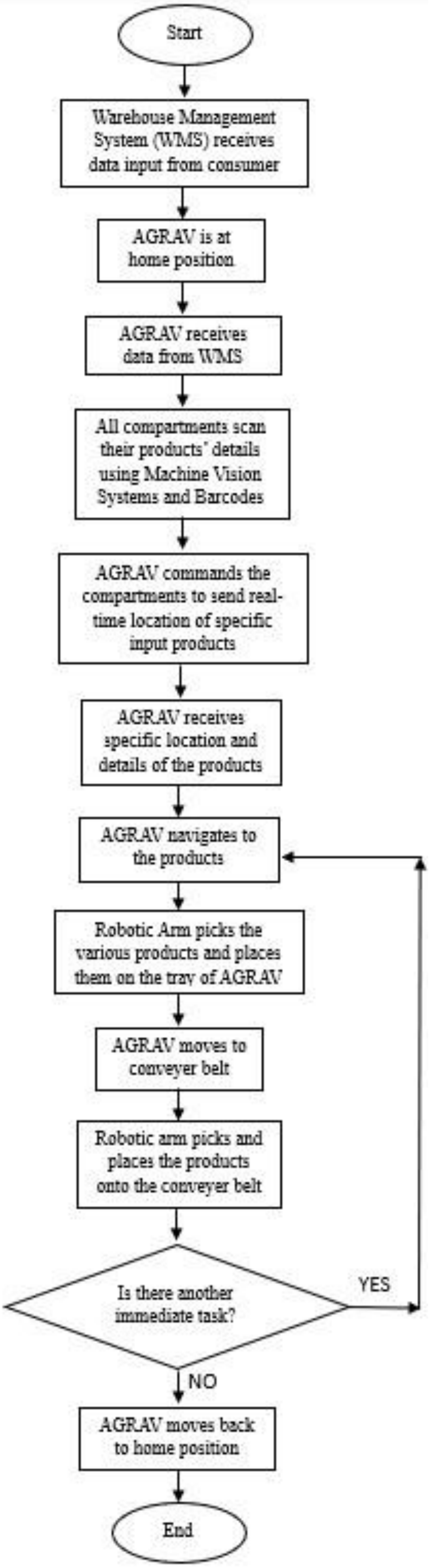
Table 1. Components used in the model and their respective functions[7]

3.3 Software description-flowchart



Flowchart of small-scale demonstration of the model

Note: 2 compartments are considered in each shelf



Flowchart of Working of Futuristic Model

Chapter 4

Results and Analysis

4.1 Torque Calculations of Robotic Arm-

Calculations were done to find the torque requirements of each motor used in the robotic arm and the factor of safety. Equilibrium conditions were used for the same, starting from the gripper carrying a load of 20 grams and ending with the H-Link at the base. A circular cross-section was assumed for these calculations and the weight of the motors are assumed to act at the center of each link, considering it along with the actual weight of the link.

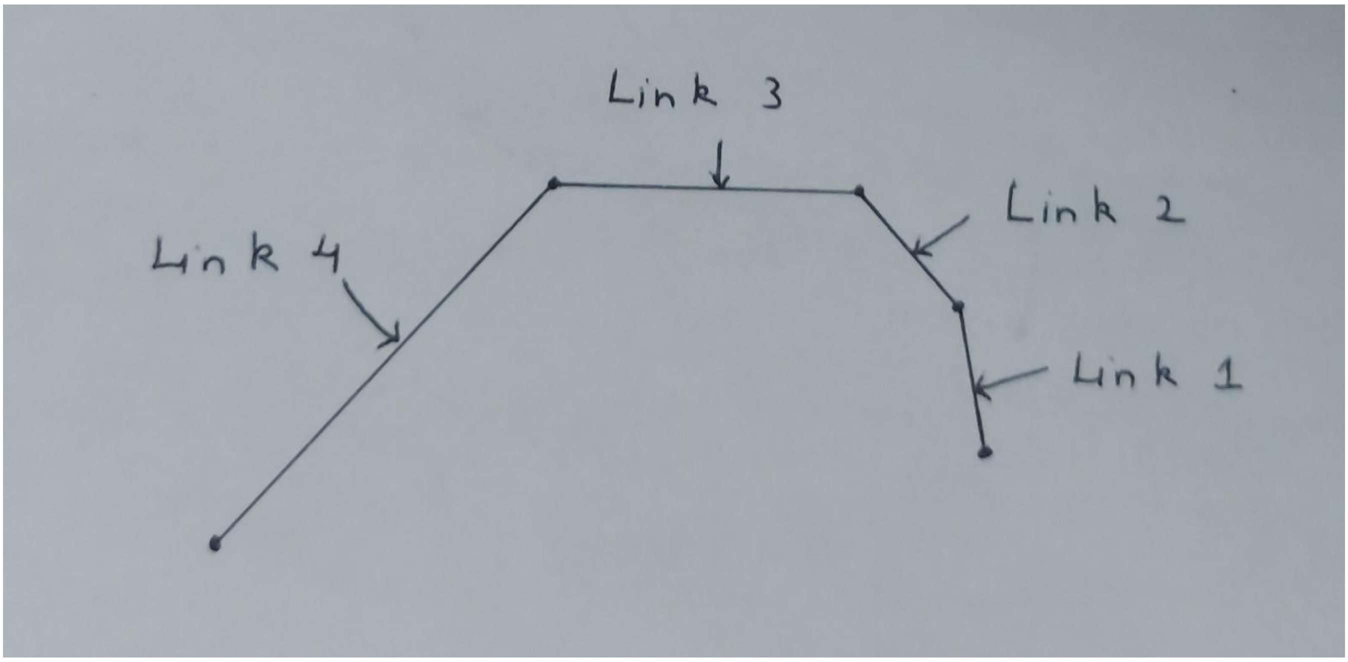


Fig 5: Robotic Arm

1st Link:

Weight to be carried = 20 g = 10.196 N

Weight of gripper = 68 g

Weight of motor = 55 g

Total weight of gripper = 123 g =1.205 N

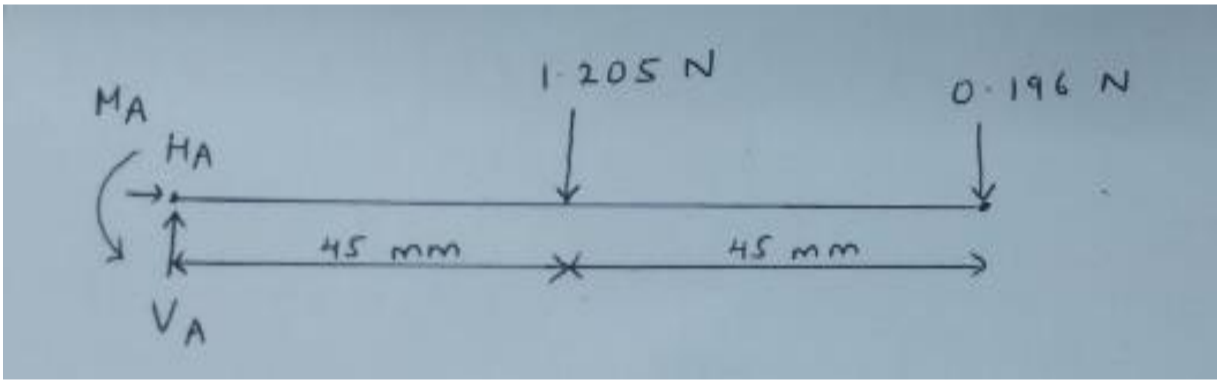


Fig 6: Link 1

$\sum F_x = 0$

$\therefore H_A = 0$

$\sum F_y = 0$

$\therefore V_A - 1.205 - 0.196 = 0$

$V_A = 1.401\ N$

$\sum M = 0$

$M_A - (1.205*45) - (0.196*90) = 0$

$\therefore M_A = 71.865\ N.mm = 0.071865\ N.m$

2^{nd} Link:

Weight of the motor = 55 g

Weight of the holder = 14 g

There are 2 motors and 2 holders so total weight = 2*(55+14) = 138 g = 1.352 N

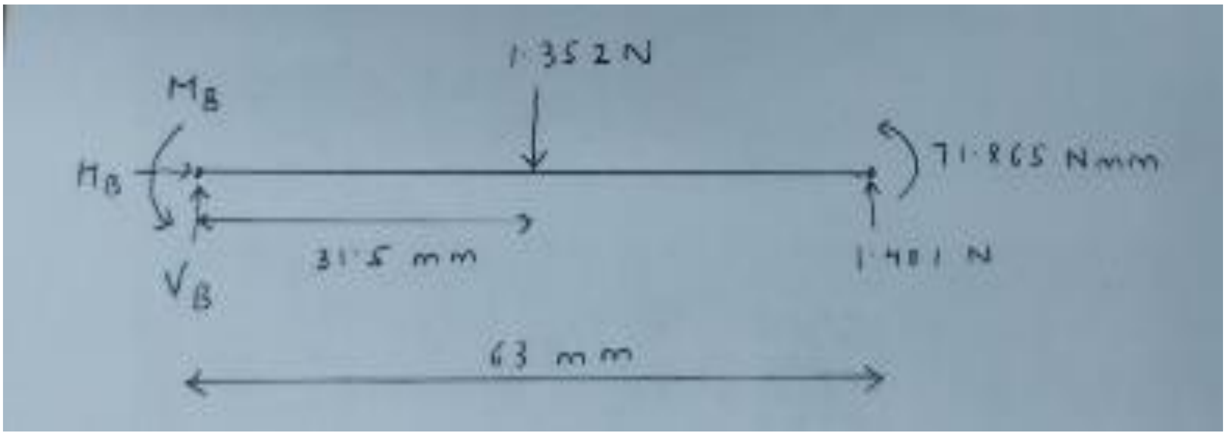


Fig 7: Link 2

$\sum F_x = 0$

$\therefore H_B = 0$

$\sum F_y = 0$

$\therefore V_B -1.352 + 0.196 = 0$

V_B = -0.049 N or 0.049 N (↓)

$\sum M = 0$

$M_B - (1.352*31.5) + (1.401*63) + 71.685 = 0$

$\therefore M_B = -117.54$ N.mm = 0.11754 N.m (↻)

3rd Link:

Weight of the motor = 55 g

Weight of the holder = 14 g

Weight of L bracket = 6 g

Weight of U bracket = 21 g

Total weight = 96 g = 0.941 N

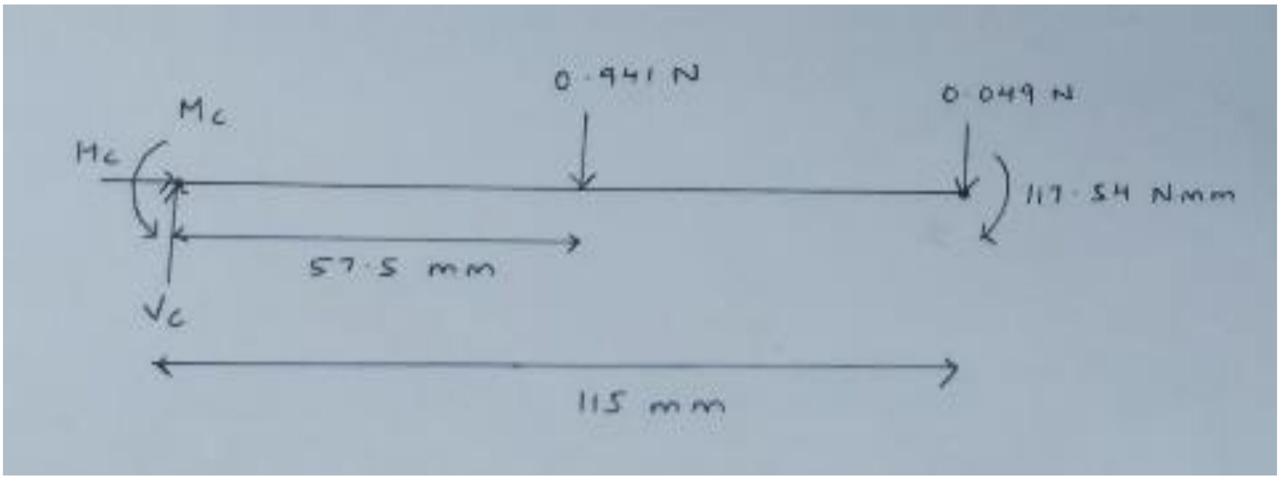


Fig 8: Link 3

$\sum F_x = 0$

$\therefore H_C = 0$

$\sum F_y = 0$

$\therefore V_C - 0.941 - 0.049 = 0$

$V_C = 0.99$ N

$\sum M = 0$

$M_C - (0.941*57.5) - (0.049*115) - 117.54 = 0$

$\therefore M_C = 171.283$ N.mm = 0.171283 N.m

4^{th} Link:

Weight of U bracket = 21 g

There are 2 U brackets so total weight = 42 g = 0.412 N

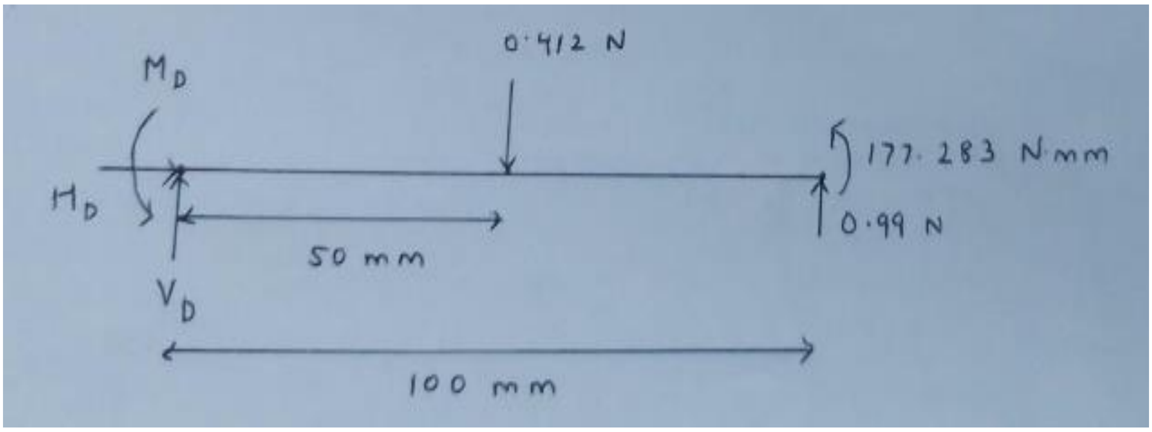


Fig 9: Link 4

$\sum F_x = 0$

$\therefore H_D = 0$

$\therefore V_D - 0.412 + 0.99 = 0$

$V_D = -0.578$ N or 0.578 N (↓)

$\sum M = 0$

$\sum F_y = 0$

$M_D - (0.412*50) + (0.99*100) + 177.283 = 0$

$\therefore M_D = -255.683$ N.mm = 0.255683 N.m (↻)

Factor of Safety Calculations-

UTS for Aluminum alloy = 310 MPa

∴ Allowable shear = 0.18*310 MPa = 55.8 MPa

$$\text{Induced Shear} = \text{Torque} * \frac{\pi}{16 * (d)^3}$$

$$= 0.255683 * \frac{\pi}{16 * (0.053)^3}$$

= 8746 kPa

= 8.746 MPa

Factor of safety = $\frac{\text{Allowable Shear}}{\text{Induced Shear}} = \frac{55.8}{8.746}$ = 6.38

4.2 Finite Element Analysis-

Finite Element Analysis was carried out on Ansys to find out the total deformation and equivalent stress induced in each link of the robotic arm and the body of the automated guided vehicle[8].

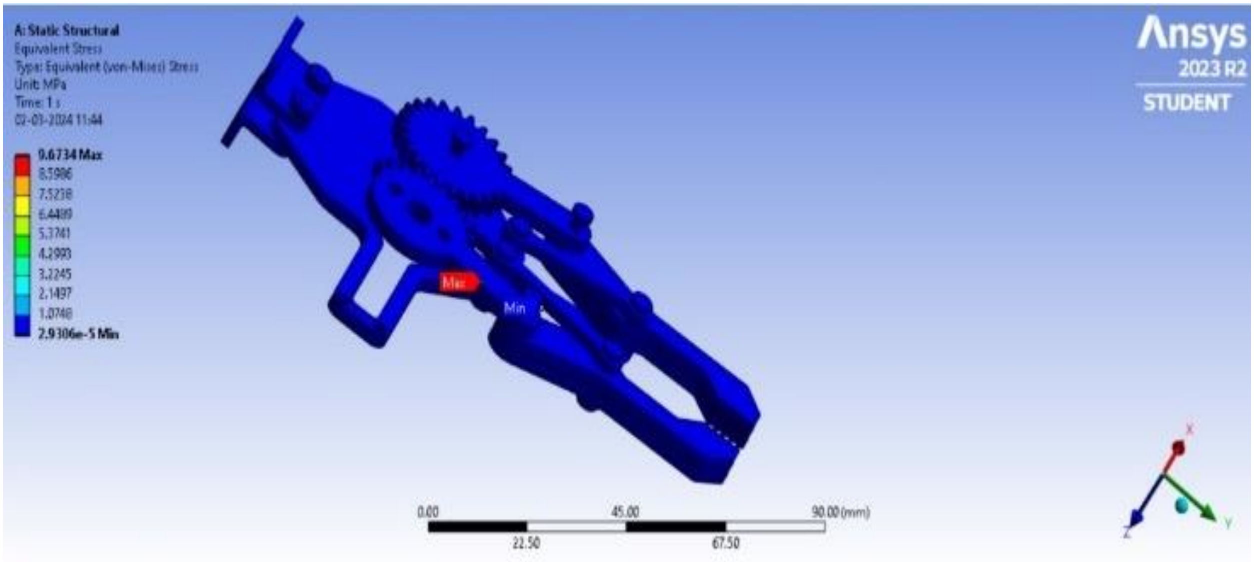


Fig 10: Link 1 Equivalent Stress

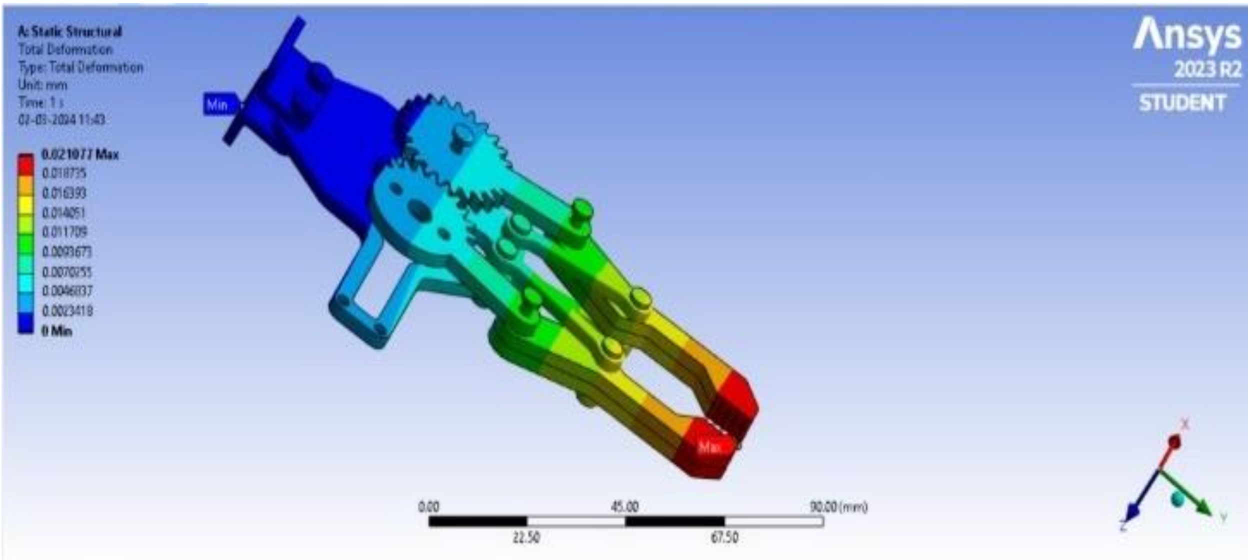


Fig 11: Link 1 Total Deformation

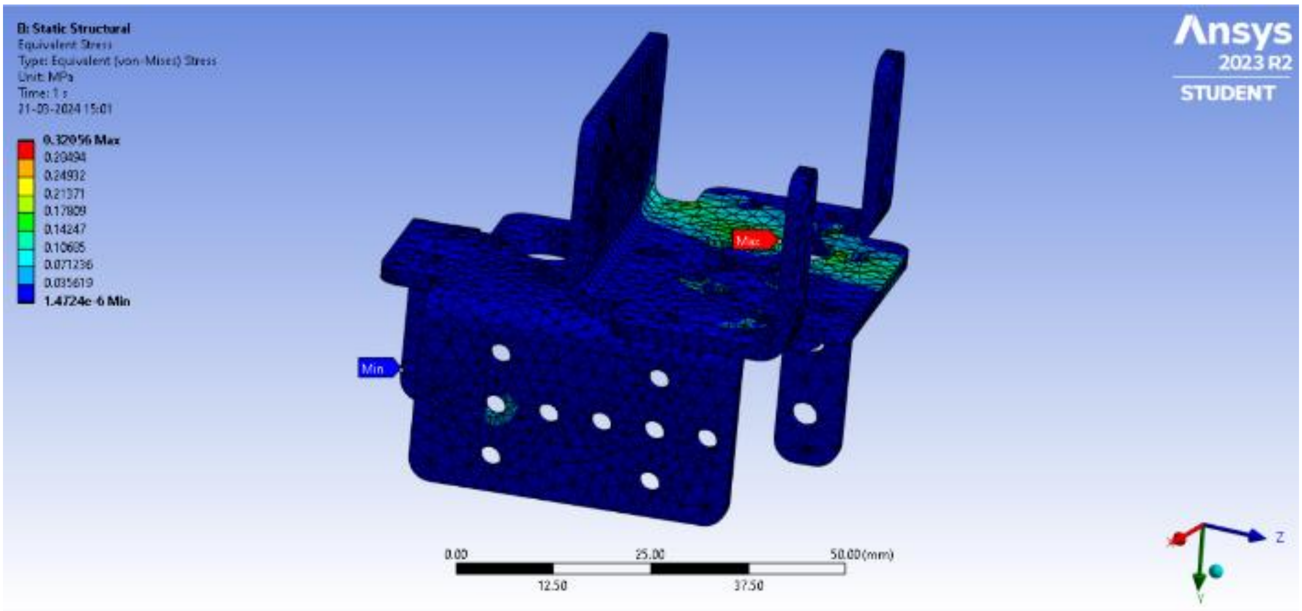


Fig 12: Link 2 Equivalent Stress

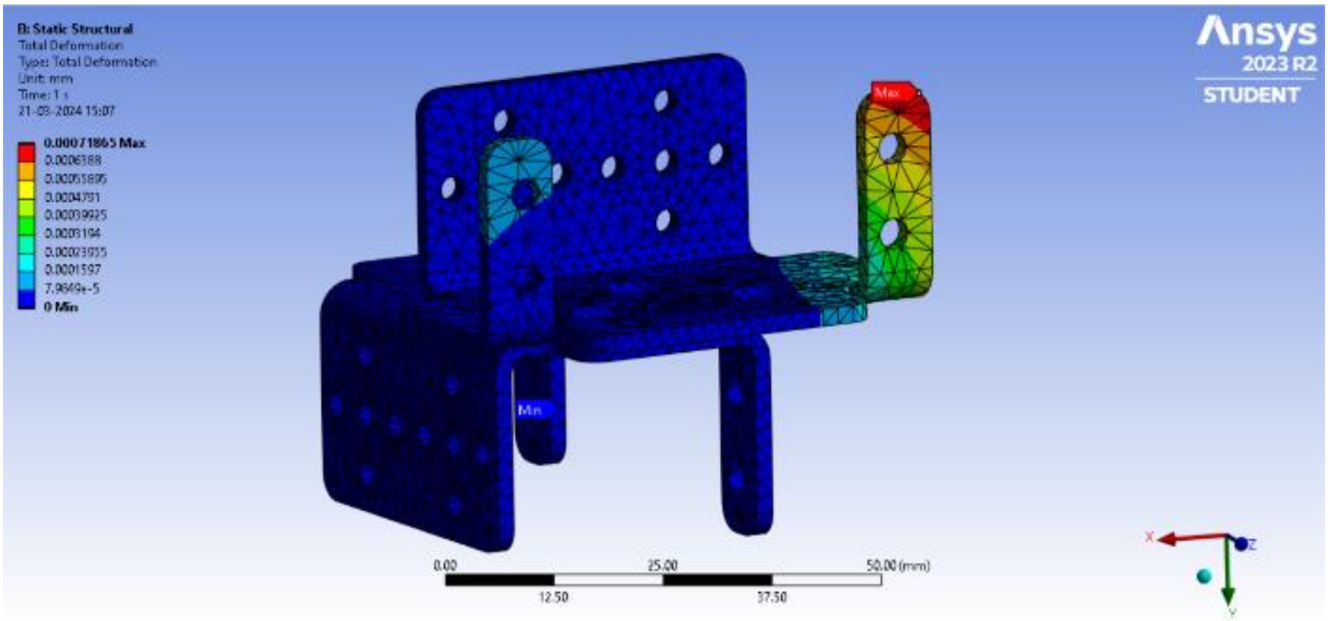


Fig 13: Link 2 Total Deformation

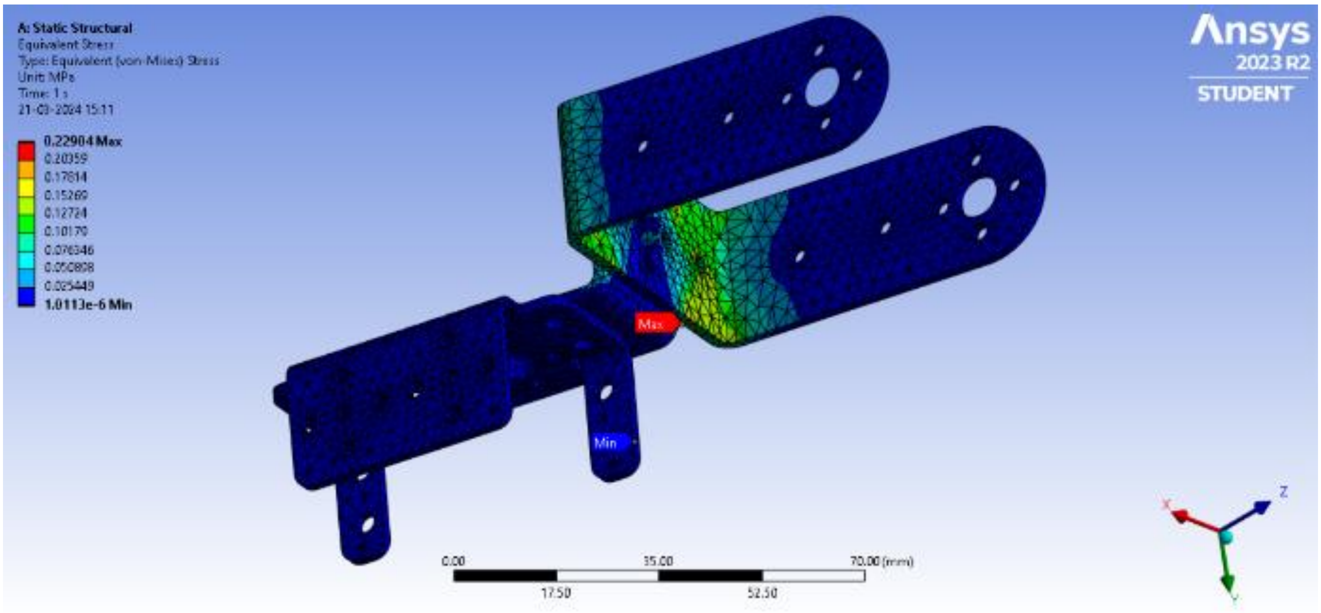


Fig 14: Link 3 Equivalent Stress

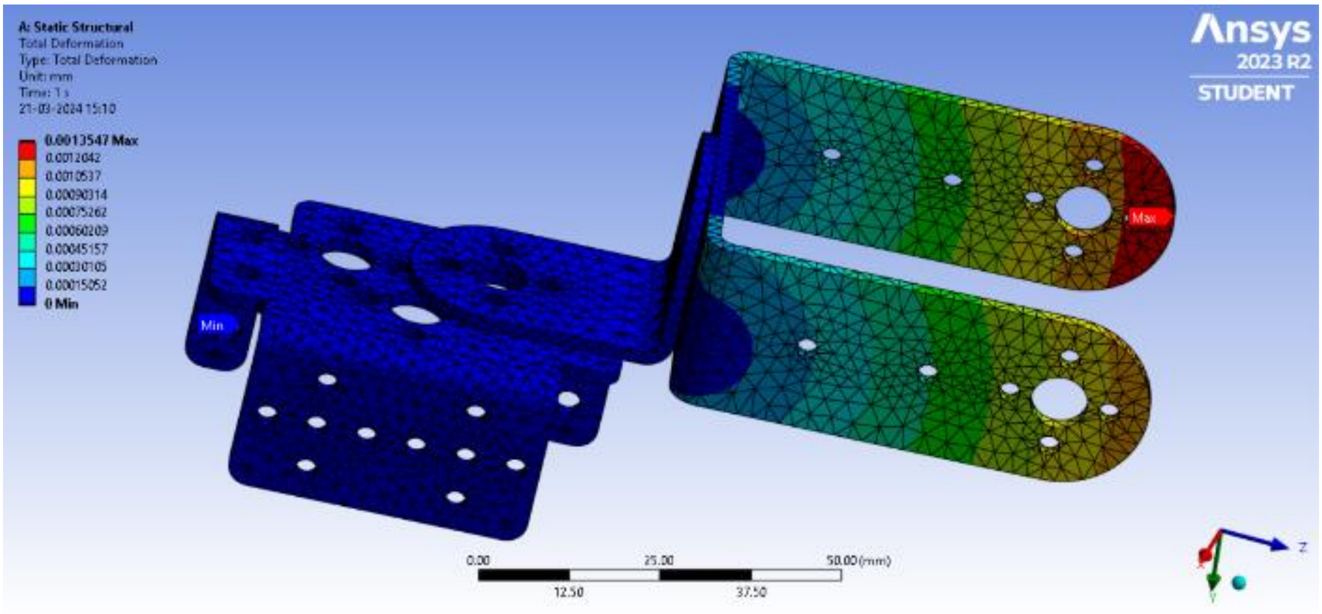


Fig 15: Link 3 Total Deformation

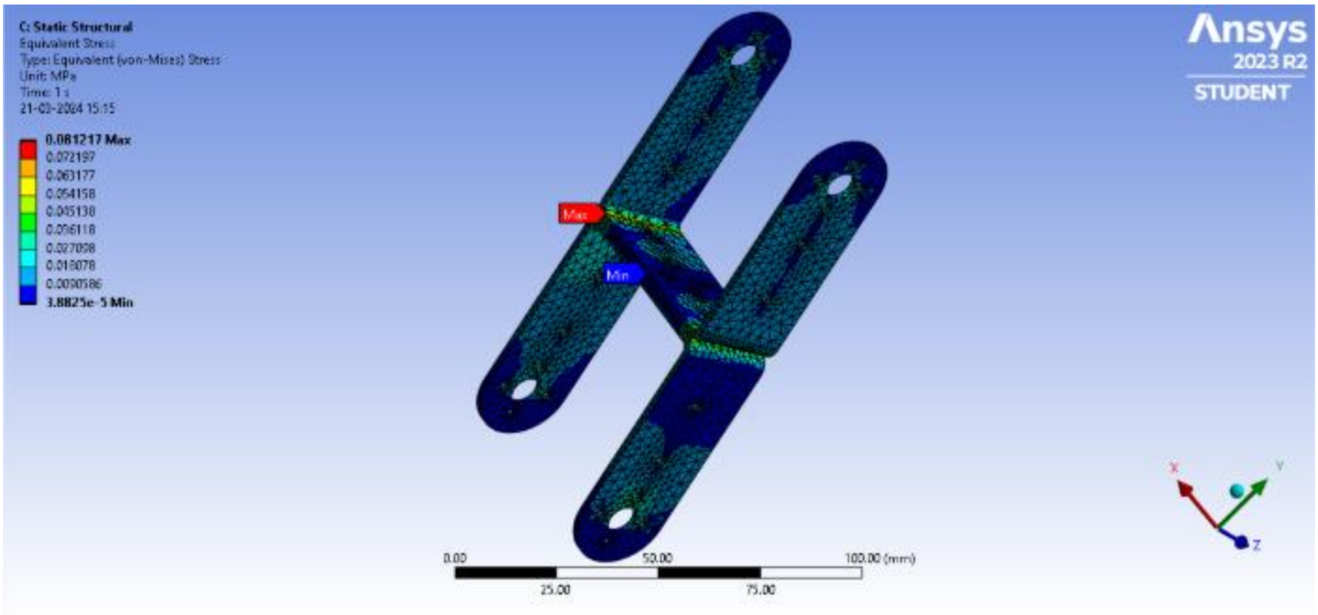


Fig 16: Link 4 Equivalent Stress

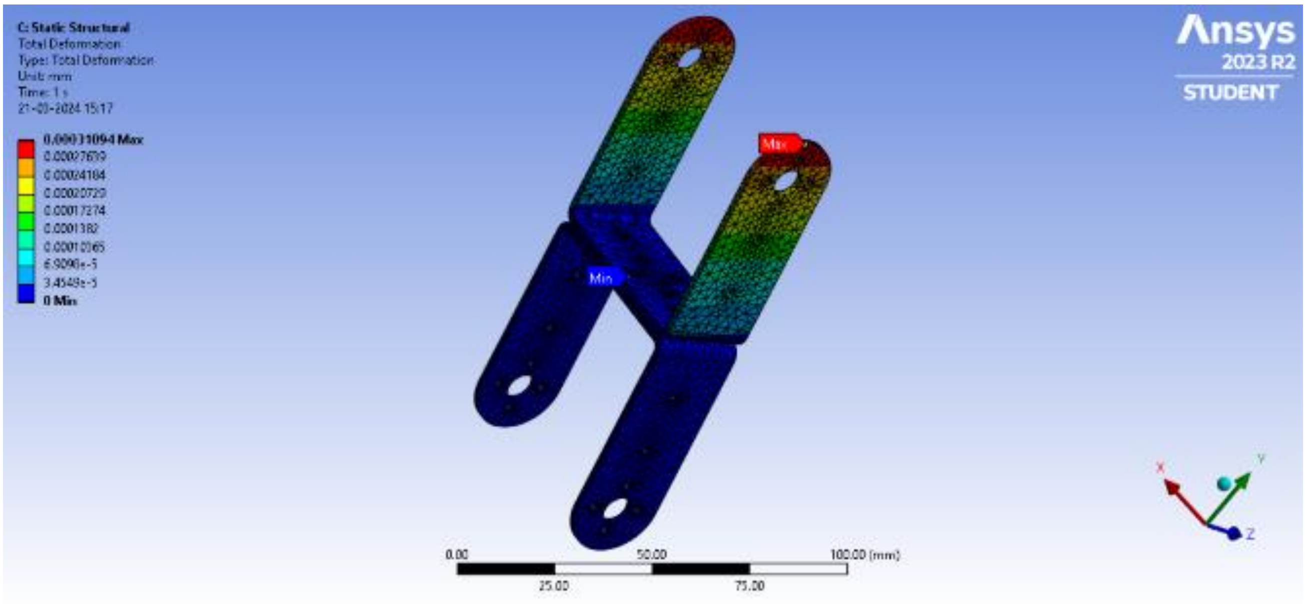


Fig 17: Link 4 Total Deformation

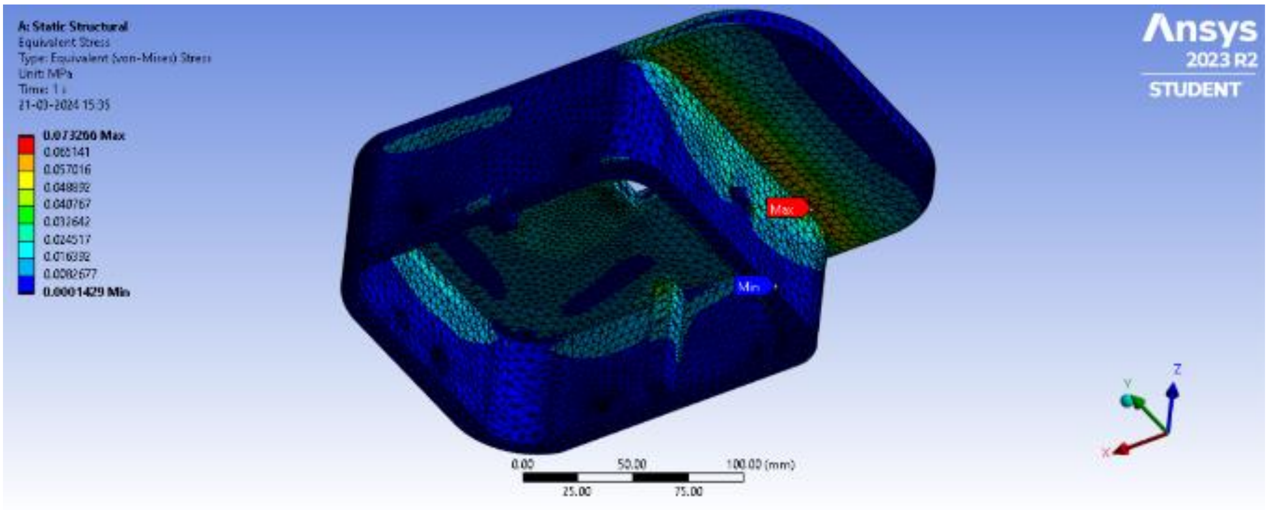


Fig 18: Car Body Equivalent Stress

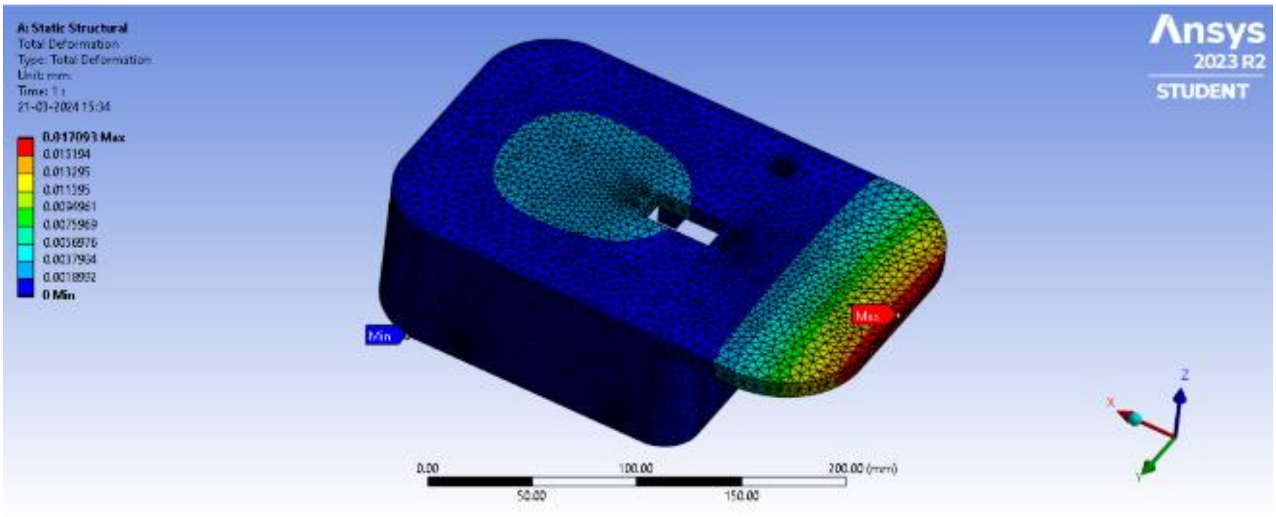


Fig 19: Car Body Total Deformation[9]

Chapter 5

Advantages, Limitations and Applications

5.1 Advantages-

1] Efficiency: AGRAV combines the functionalities of an Automated Guided Vehicle (AGV) with a robotic arm, allowing it to autonomously navigate the warehouse, pick up items from shelves, and transport them to the designated location. This eliminates the need for manual labor in these tasks, resulting in faster and more efficient operations.

2] Increased Throughput: By automating the picking and placing process, AGRAV can handle a higher volume of products compared to traditional manual methods. This increased throughput means that warehouses can process more orders in a shorter amount of time, leading to improved customer satisfaction and potentially increased revenue.

3] Accuracy: The robotic arm integrated into AGRAV is equipped with advanced sensors and precision control systems, ensuring accurate picking and placing of products. This reduces the risk of errors such as misplaced items or damaged goods, enhancing overall inventory accuracy and reducing costly mistakes.

4] 24/7 Operations: Unlike human workers, AGRAV can operate continuously without the need for breaks, resulting in round-the-clock warehouse operations. This constant workflow ensures that orders can be processed quickly and efficiently, even during non-business hours, allowing for faster order fulfillment and delivery times.

5] Safety: With advanced collision detection and avoidance systems, AGRAV ensures safe operation within the warehouse environment. It can detect obstacles in its path and navigate around them, minimizing the risk of accidents or injuries to both personnel and equipment. Additionally, by reducing the reliance on manual labor for heavy lifting tasks, AGRAV helps mitigate the risk of workplace injuries.

6] Cost Savings: While the initial investment in AGRAV technology may be significant, the long-term cost savings can be substantial. By reducing labor costs, improving operational efficiency, and minimizing errors, AGRAV helps lower overall operating expenses and improve the bottom line for warehouse operations.

5.2 Disadvantages-

1] Initial Cost: The initial investment required for implementing AGRAV systems can be substantial. This includes the cost of purchasing the AGV units, robotic arms, sensors, software, and infrastructure modifications needed to integrate them into existing warehouse setups. For some businesses, especially smaller ones, the upfront cost may pose a barrier to adoption.
2] Complexity of Implementation: Deploying AGRAV systems involves significant planning, customization, and integration with existing warehouse infrastructure and software systems. This complexity can lead to longer implementation timelines and require specialized expertise, both during the initial setup phase and for ongoing maintenance and support.
3] Limited Flexibility: While AGRAV systems offer scalability in terms of adding more units to meet increased demand, they may have limited flexibility in handling diverse types of products or tasks. The robotic arms are typically designed for specific applications and may not be easily reconfigured for different tasks or product variations without significant reprogramming or hardware modifications.

5.3 Applications-

1] Warehousing and Distribution Centers: AGRAV systems are particularly well-suited for automating tasks in warehousing and distribution centers. They can efficiently pick and place items from shelves, transport them to packing stations, and load them onto conveyors or delivery trucks. This application significantly speeds up order fulfillment processes and improves inventory management.
2] Manufacturing Facilities: In manufacturing plants, AGRAV units can be used to transport raw materials, work-in-progress components, and finished products between different production areas. By automating material handling tasks, AGRAV systems optimize workflow efficiency, reduce cycle times, and enhance overall manufacturing productivity.
3] E- Commerce Fulfillment Centers: With the rapid growth of e-commerce, AGRAV systems play a crucial role in streamlining order fulfillment operations in fulfillment centers. They can navigate through aisles of shelves, accurately pick items for customer orders, and transport them to packing stations for shipment. This application helps e-commerce companies meet customer demands for fast and reliable order delivery.
4] Pharmaceutical and Healthcare: In pharmaceutical manufacturing and distribution facilities, AGRAV systems help automate the handling of medical supplies, medications, and healthcare products. They can

ensure precise inventory management, accurate order picking, and secure transportation of sensitive pharmaceuticals, contributing to patient safety and regulatory compliance.

5] Cold Storage and Logistics: In cold storage warehouses and logistics facilities, AGRAV systems can operate in low-temperature environments while efficiently handling frozen or refrigerated products. They can transport perishable goods between storage chambers, loading docks, and delivery vehicles, ensuring proper temperature control and preserving product quality.

6] Retail Operations: In retail environments, AGRAV systems can automate inventory replenishment, shelf stocking, and order fulfillment processes. They can efficiently handle tasks such as restocking shelves, organizing merchandise, and preparing orders for in-store pickup or delivery. This application improves operational efficiency and enhances the customer shopping experience.

Chapter 6

Conclusion and Future Scope

There can be significant advantage of using this system of AGRAVs in warehouses in the future as the maximum possible productivity will be achieved by implementing it. For practical warehousing processes a scaled-up version of this model can be made wherein there will be a proportionate increase in dimensions of the Robotic Arm and the body of the AGV. By connecting tray-like-features to the design of the AGV, which is much bigger in size and hence has more space compared to our model, multiple objects can be transported by this system. The manufacturing of the AGV starts by cutting the sheet metal plates according to the required dimensions on a circular saw. The AGV is then assembled by welding and using mechanical fasteners like rivets or nut and bolts. The whole production line is automated. The scaled-up version will use a Programmable Logic Controller (PLC) which helps manage the control systems. The PLC will receive inputs from the communication and navigation system allowing it to make real time decisions to control the motors and other components. The cost of the product depends on three factors- the number of models being purchased, the payload capacity requirement of the model and the dimensions of the robotic arm and AGV to be manufactured. The cost of an AGRAV will be around 20 to 40 lakhs INR, considering the cost of the Industrial Control Systems (Automation Systems), which is justified over a long-term period as compared to having human labor. Many advancements are also being made about completely smart warehouses involving features like charging paths. Thus, the AGRAV system can potentially change the entire market of warehouses[10].

References

[1] D. R. Swerdlow et al., "Robotic Arm–Assisted Sonography: Review of Technical Developments and Potential Clinical Applications," American Journal of Roentgenology, vol. 208, pp. 733–738, 2017. [Online]. Available: www.ajronline.org

[2] K. Ghadge, S. More, P. Gaikwad, and S. Chillal, “Robotic Arm for Pick and Place Application,” *International Journal of Mechanical Engineering and Technology (IJMET*, vol. 9, no. 1, pp. 125–133, 2018, [Online]. Available: http://www.iaeme.com/IJMET/index.asp125http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=9&IType=1http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=9&IType=1

[3] M. Alim and S. Erhan Kesen, “Smart Warehouses in Industry 4.0,” in *LOGISTICS 4.0*, T. Paksoy, C. Gonul Kochan, and S. Samar Ali, Eds., CRC Press Taylor & Francis Group, 2020, pp. 186–201.

[4] P. Ponnusamy, C. Maheswari, E. B. Priyanka, S. Thangavel, and P. Parameswari, “Development of unmanned guided vehicle for material handling automation for industry 4.0,” 2018. [Online]. Available: https://www.researchgate.net/publication/330846976

[5] M. Tamara *et al.*, “System Design Warehouse Management AGV for Packages Sorting in Supporting Industry E-Commerce,” European Alliance for Innovation n.o., Jul. 2023. doi: 10.4108/eai.5-10-2022.2329535.

[6] W. Earnest, L. Fina, K. Chau, M. Quaranta, G. Araujo, and H. Erhan Sevil, “Design and Feasibility Study of Cost-Efficient Robotic Arm for Object Pick-and-Place Application.”

[7] V. Jaiganesh, J. Dhileep Kumar, and J. Girijadevi, “Automated guided vehicle with robotic logistics system,” in *Procedia Engineering*, Elsevier Ltd, 2014, pp. 2011–2021. doi: 10.1016/j.proeng.2014.12.444.

[8] R. Mourya, A. Shelke, S. Satpute, S. Kakade, and M. Botre, “Design and Implementation of Pick and Place Robotic Arm,” 2015. [Online]. Available: www.paperpublications.org

[9] B. Raghu, B. Rajesh, C. Swathi, D. Rishikesh, G. Abdul Kadir, and A. Professor, “MODELLING AND ANALYSIS OF EXCAVATOR ARM,” 2020. [Online]. Available: www.ijcrt.org

[10] Seprényi, K., Tamás, P., & Cservenák, Á. (2022). Trends in Warehousing and Material Handling Solutions : Past. Advanced Logistic Systems - Theory and Practice, 16(2), 71–81. https://doi.org/10.32971/als.2022.015

Appendix A: Soft Codes

Car Code:-

```
// Green WIRE RX  ON HC05 TO TX PIN NO-1 ON ARDUINO
// Yellow WIRE TX ON HC05 TO RX PIN NO-0 ON ARDUINO
// A4 is led
#include <AFMotor.h>
#include <Servo.h>    //add Servo Motor library
#include <NewPing.h>  //add Ultrasonic sensor library
#define TRIG_PIN A0  // Pin A0 on the Motor Drive Shield soldered to the ultrasonic sensor
#define ECHO_PIN A1  // Pin A1 on the Motor Drive Shield soldered to the ultrasonic sensor
int leftir = A2;       //A2 PIN
int rightir = A3;      //A3 PIN
#define horn_Buzz 2    //WHITE WIRE
AF_DCMotor motor1(1);  // M1 on Motor Shield
AF_DCMotor motor2(2);  // M2
AF_DCMotor motor3(3);  // M3
AF_DCMotor motor4(4);  // M4
Servo myservo;  // create servo object to control a servo
char bluetoothInput = 'S';  // Input from Bluetooth HC-05
int speedCar = 100;         // 50 - 255.
int speedCar1 = 255; // Speed for Line Follower only
int speed_Coeff = 3;
boolean lightFront = false;
boolean lightBack = false;
boolean horn = false;
void setup() {
  Serial1.begin(9600);  // Connect to Bluetooth HC-05
  pinMode(leftir, INPUT);
  pinMode(rightir, INPUT);
  pinMode(horn_Buzz, OUTPUT);
  motor1.setSpeed(speedCar);  // Initialise Motor Speed
  motor2.setSpeed(speedCar);
  motor3.setSpeed(speedCar);
  motor4.setSpeed(speedCar);
  myservo.attach(10);  // attaches the servo on pin 10 (SERVO_1 on the Motor Drive Shield to the servo
object
  myservo.write(90); // tells the servo to position at 90-degrees ie. facing forward.
  pinMode(horn_Buzz, OUTPUT);
  digitalWrite(horn_Buzz, LOW);
  ;
  stopCar();  // Initialise Car Position
}
void loop() {
  //LineFollower();

  bluetoothInput = Serial1.read();  // Read Bluetooth HC-05
   switch (bluetoothInput) {
    if (horn) {
      digitalWrite(horn_Buzz, HIGH);
      ;
```

```
    }
    if (!horn) {
      digitalWrite(horn_Buzz, LOW);
      ;
    }

    case '0': speedCar = 100; break;
    case '1': speedCar = 115; break;
    case '2': speedCar = 130; break;
    case '3': speedCar = 145; break;
    case '4': speedCar = 160; break;
    case '5': speedCar = 175; break;
    case '6': speedCar = 190; break;
    case '7': speedCar = 205; break;
    case '8': speedCar = 220; break;
    case '9': speedCar = 235; break;
    case 'q': speedCar = 255; break;
    case 'F': goForward(); break;
    case 'B': goBackward(); break;
    case 'L': turnLeft(); break;
    case 'R': turnRight(); break;
    case 'S': stopCar(); break;
    case 'W': lightFront = true; break;
    case 'w': lightFront = false; break;
    case 'U': lightBack = true; break;
    case 'u': lightBack = false; break;
    case 'V': horn = true; break;
    case 'v': horn = false; break;
    case 'X': LineFollower(); break;
    case 'x': stopCar(); break;
  }  // end switch
}  // end loop()
void goForward() {
  motor1.setSpeed(speedCar);
  motor2.setSpeed(speedCar);
  motor3.setSpeed(speedCar);
  motor4.setSpeed(speedCar);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
void goBackward() {
  motor1.setSpeed(speedCar);
  motor2.setSpeed(speedCar);
  motor3.setSpeed(speedCar);
  motor4.setSpeed(speedCar);
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
```

```
}
void turnRight() {
  motor1.setSpeed(speedCar);
  motor2.setSpeed(speedCar);
  motor3.setSpeed(speedCar);
  motor4.setSpeed(speedCar);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
}
void turnLeft() {
  motor1.setSpeed(speedCar);
  motor2.setSpeed(speedCar);
  motor3.setSpeed(speedCar);
  motor4.setSpeed(speedCar);
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
}
void stopCar() {
  motor1.setSpeed(speedCar);
  motor2.setSpeed(speedCar);
  motor3.setSpeed(speedCar);
  motor4.setSpeed(speedCar);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
void LineFollower() {
    while (Serial1.read() !='x'){
  //BLACK COLOUR = HIGH - 1
  //WHITE COLOUR = LOW - 0
  int lefteye = analogRead(leftir);
  int righteye = analogRead(rightir);
  Serial1.print("leftValue:");  //print the values in the monitor
  Serial1.print(lefteye);
  Serial1.print("......rightValue:");
  Serial1.println(righteye);
  if ((righteye >= 50) && (lefteye >= 50)) {
    goForward1();
  }
  else if ((righteye < 50) && (lefteye > 50)) {
    turnLeft1();
  }
  else if ((righteye > 50) && (lefteye < 50)) {
    turnRight1();
  }
else if ((righteye < 50) && (lefteye < 50)) {
```

```
    turnRight1();
  }

  //else {
    //stopCar1();
    ;  //if all else fails, stop the motors
    //delay(500);
  //}
  //digitalWrite(horn_Buzz, HIGH);
  delay(20);
  // digitalWrite(buzzer, LOW);

  }
}
void goForward1() {
  motor1.setSpeed(speedCar1);
  motor2.setSpeed(speedCar1);
  motor3.setSpeed(speedCar1);
  motor4.setSpeed(speedCar1);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(15);
  stopCar();
}
void goBackward1() {
  motor1.setSpeed(speedCar1);
  motor2.setSpeed(speedCar1);
  motor3.setSpeed(speedCar1);
  motor4.setSpeed(speedCar1);
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
}
void turnRight1() {
  motor1.setSpeed(speedCar1);
  motor2.setSpeed(speedCar1);
  motor3.setSpeed(speedCar1);
  motor4.setSpeed(speedCar1);
  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  delay(30);
  stopCar();
}
void turnLeft1() {
  motor1.setSpeed(speedCar1);
  motor2.setSpeed(speedCar1);
```

```
  motor3.setSpeed(speedCar1);
  motor4.setSpeed(speedCar1);
  motor1.run(BACKWARD);
  motor2.run(BACKWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(30);
  stopCar();
}
void stopCar1() {
  motor1.setSpeed(speedCar1);
  motor2.setSpeed(speedCar1);
  motor3.setSpeed(speedCar1);
  motor4.setSpeed(speedCar1);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
```

Robotic Arm Code:-

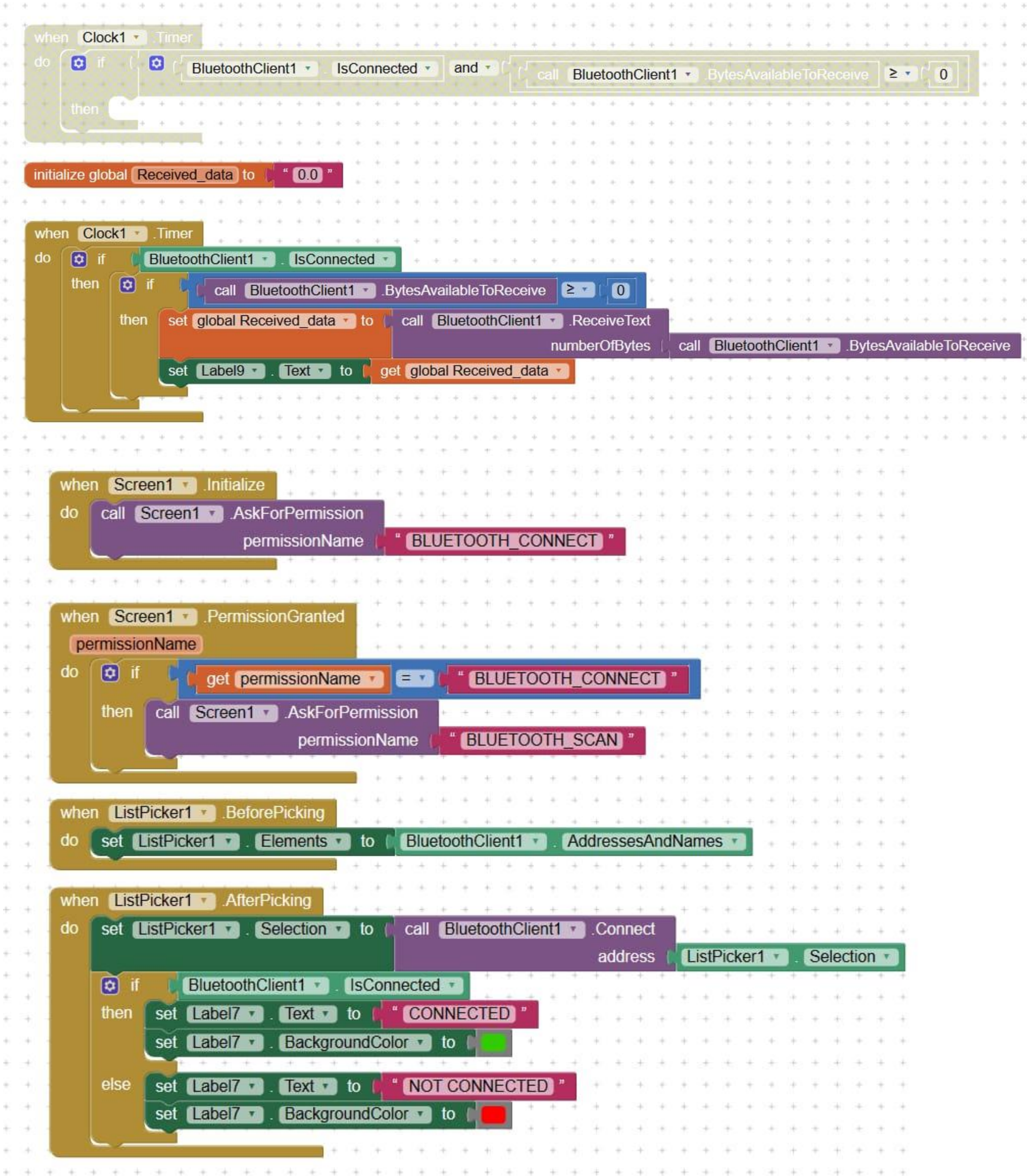
```
#include <Servo.h>

Servo motor_1;
Servo motor_2;
Servo motor_3;
Servo motor_4;
Servo motor_5;

#define enA 5  //Enable1 L298 Pin enA
#define in1 A3 //Motor1  L298 Pin in1
#define in2 A2 //Motor1  L298 Pin in1
#define in3 A1 //Motor2  L298 Pin in1
#define in4 A0 //Motor2  L298 Pin in1
#define enB 5  //Enable2 L298 Pin enB
```

```
int servo1 = 90;
int servo2 = 90;
int servo3 = 90;
int servo4 = 90;
int servo5 = 90;

int bt_data;
int Speed = 130;

void setup(){

Serial1.begin(9600); // initialize serial communication at 9600 bits per second:

motor_1.attach(44);
motor_2.attach(45);
motor_3.attach(46);
motor_4.attach(47);
motor_5.attach(48);

motor_1.write(servo1);
motor_2.write(servo2);
motor_3.write(servo3);
motor_4.write(servo4);
motor_5.write(servo5);

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

delay(1000);
}

void loop(){
//if some date is sent, reads it and saves in state

if(Serial1.available() > 0){
bt_data = Serial1.read();
Serial1.println(bt_data);
if(bt_data > 20){Speed = bt_data;}
}

analogWrite(enA, Speed); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
analogWrite(enB, Speed); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed


    if(bt_data == 1){forword(); }  // if the bt_data is '1' the DC motor will go forward
else if(bt_data == 2){backword();}  // if the bt_data is '2' the motor will Reverse
else if(bt_data == 3){turnLeft();}  // if the bt_data is '3' the motor will turn left
```

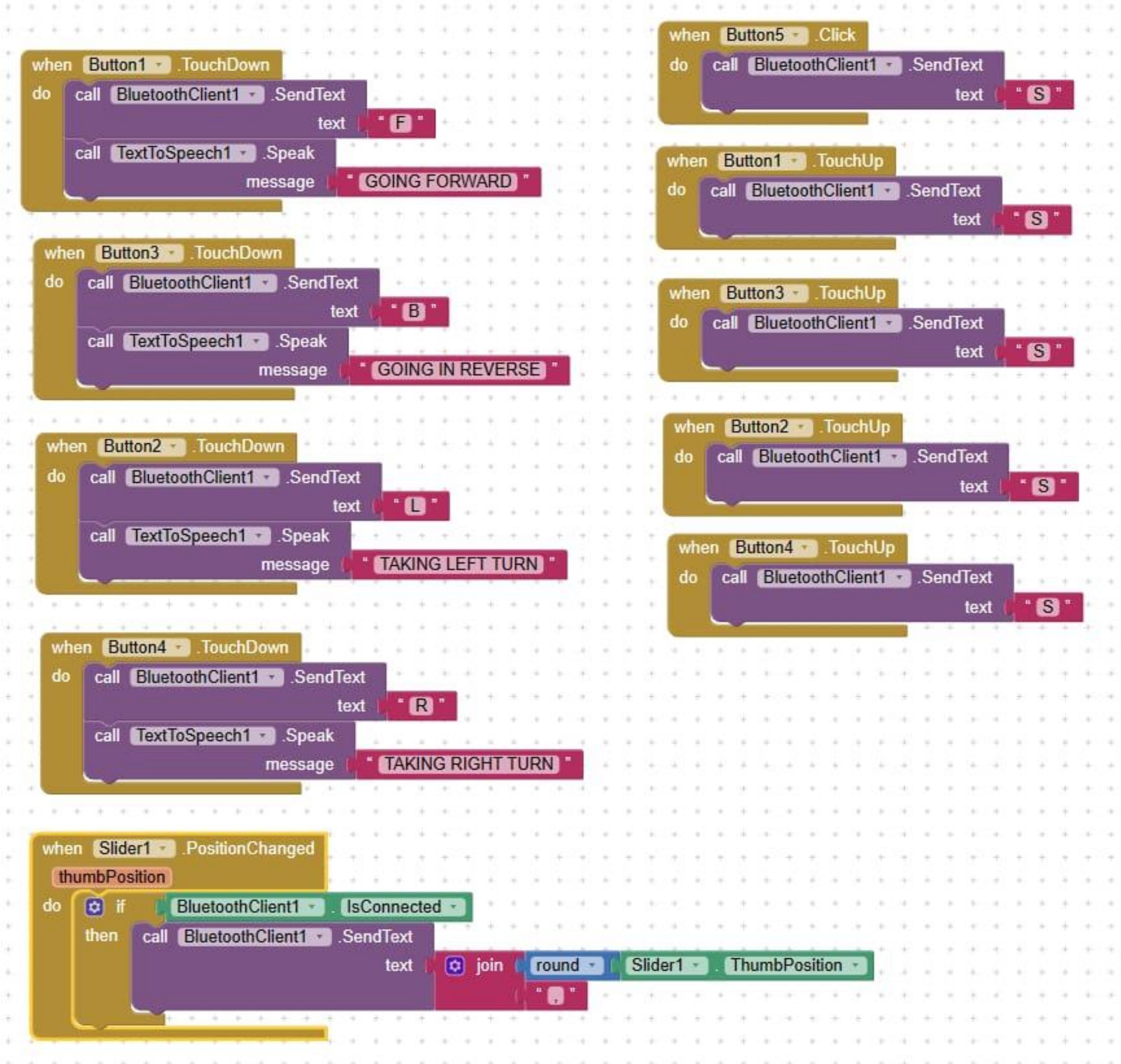
```
else if(bt_data == 4){turnRight();} // if the bt_data is '4' the motor will turn right
else if(bt_data == 5){Stop(); }     // if the bt_data '5' the motor will Stop

else if(bt_data == 6){turnLeft();  delay(400);  bt_data = 5;}
else if(bt_data == 7){turnRight(); delay(400);  bt_data = 5;}

else if (bt_data == 8){
if(servo1<180){servo1 = servo1+1;}
motor_1.write(servo1);
}
else if (bt_data == 9){
if(servo1>0){servo1 = servo1-1;}
motor_1.write(servo1);
}

else if (bt_data == 10){
if(servo2>0){servo2 = servo2-1;}
motor_2.write(servo2);
}
else if (bt_data == 11){
if(servo2<180){servo2 = servo2+1;}
motor_2.write(servo2);

}

else if(bt_data == 12){
if(servo3>0){servo3 = servo3-1;}
motor_3.write(servo3);
}
else if (bt_data == 13){
if(servo3<180){servo3 = servo3+1;}
motor_3.write(servo3);
}

else if (bt_data == 14){
if(servo4<180){servo4 = servo4+1;}
motor_4.write(servo4);
}
else if(bt_data == 15){
if(servo4>0){servo4 = servo4-1;}
motor_4.write(servo4);
}

else if (bt_data == 16){
if(servo5>90){servo5 = servo5-1;}
motor_5.write(servo5);
}
else if (bt_data == 17){
if(servo5<150){servo5 = servo5+1;}
motor_5.write(servo5);
}
```

```
delay(70);
}


void forword(){  //forword
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void backword(){ //backword
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void turnRight(){ //turnRight
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void turnLeft(){ //turnLeft
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void Stop(){ //stop
digitalWrite(in1, LOW); //Right Motor forword Pin
digitalWrite(in2, LOW); //Right Motor backword Pin
digitalWrite(in3, LOW); //Left Motor backword Pin
digitalWrite(in4, LOW); //Left Motor forword Pin
}
```

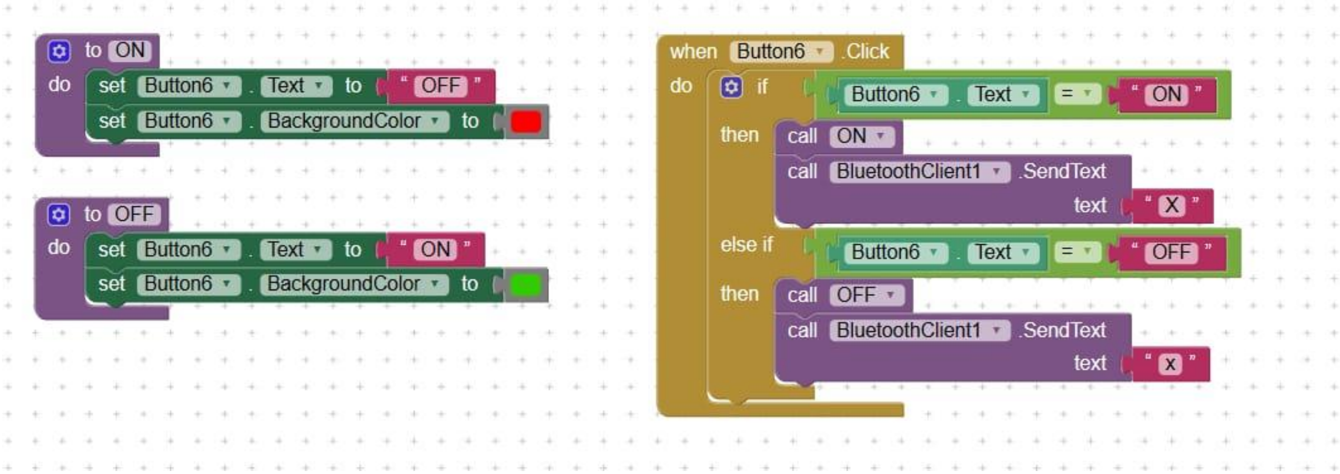
Application Code-Blocks:-



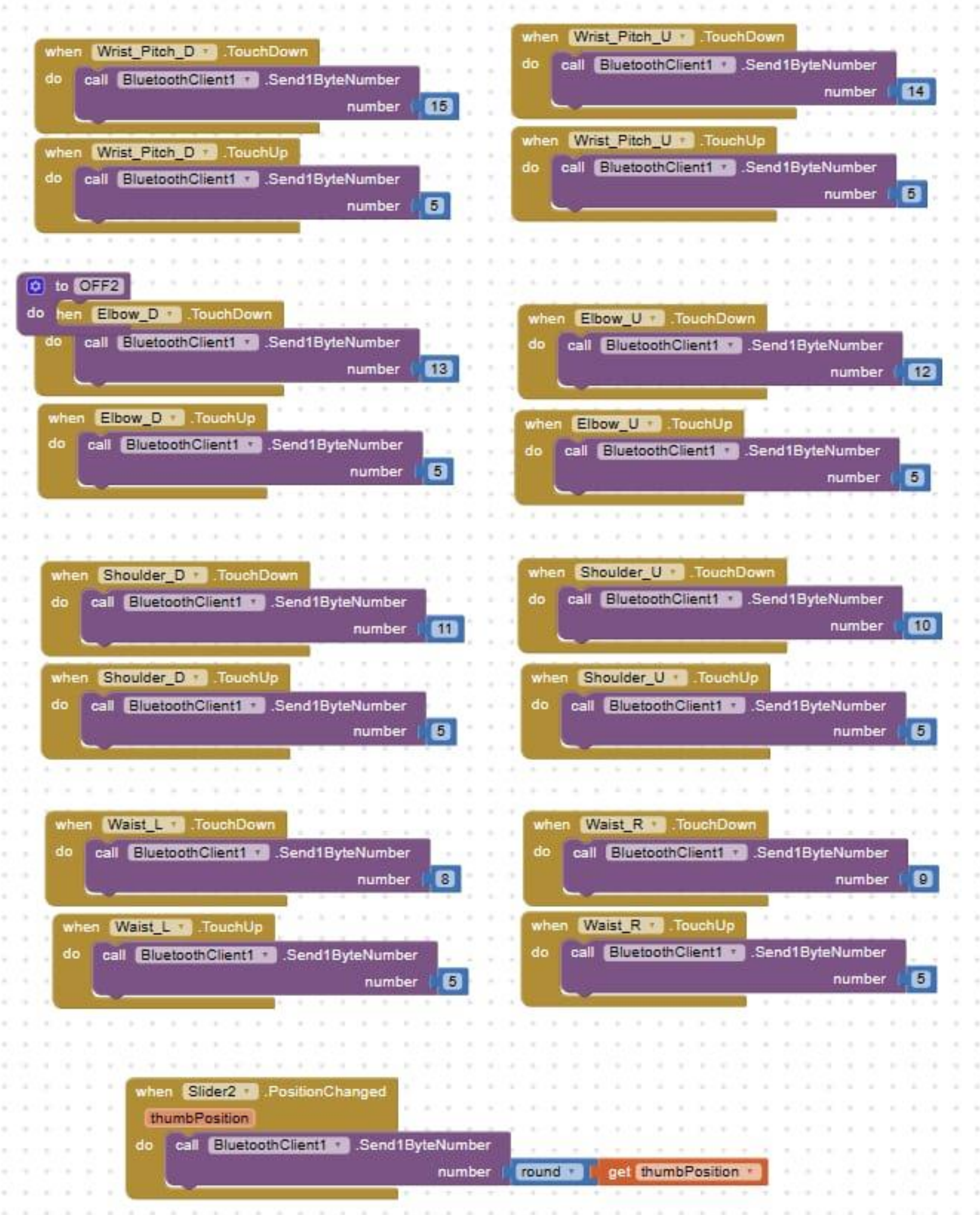
Bluetooth



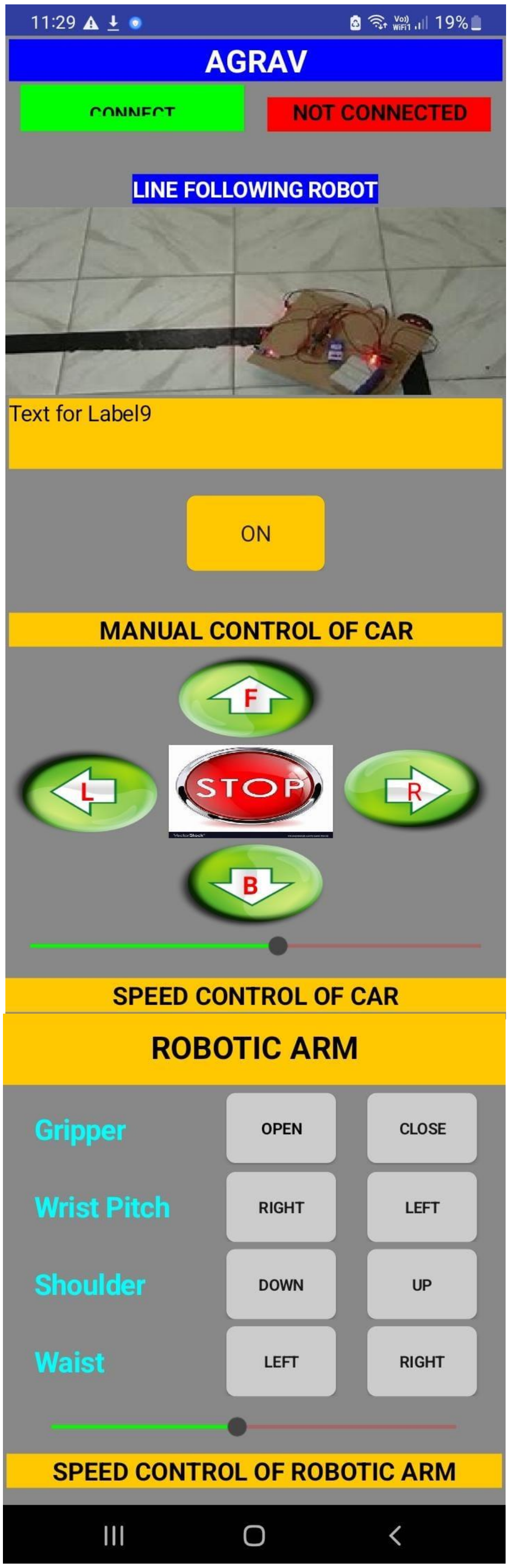
Button Control (Forward, Backward, Left, Right) & Slider Control (Speed Control of Car)



Line Following



Robotic Arm Controller with Speed Control Slider



Application Interface