Collision Avoidance

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

MECHANICAL ENGINEERING

Submitted by:
Tirth Gadhvi - 1403038

Guided by:
Ms. Jaina Mehta - Guide
Dr. Harshal Oza - Co.Guide

MAY 2018

CANDIDATE'S DECLARATION

I hereby declare that the project entitled "Collision Avoidance" submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Mechanical Engineering completed under the supervision of Ms.Jaina Mehta (Lecturer) and Dr.Harshal Oza (Assistant Professor) Ahmedabad University is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere. I certify that whenever I have used materials (data, images , theoretical analysis, and text) from other sources, we have given fully credit to them in the text of the report and giving their details in the references.

Tirth Gadhvi

CERTIFICATE by Project Guide

It is certified that the above statement made by the student is correct to the best of our knowledge.

Dr. Harshal Oza
Assistant Professor

Ms. Jaina Mehta
Lecturer

Acknowledgement

I wish to thank Ms.Jaina Mehta and Dr.Harshal Oza for their kind support and valuable guidance.

It is their help and support, due to which I was able to complete the design and technical report.

Without their support this project would not have been possible.

Tirth Gadhvi

Abstract

In the following project, Collision Avoidance theory has been developed along with the tools and test-bed to test the theory. The concept of this project is derived from simple (Automatic Storage and Retrieval Systems) AS/RS systems used in any industry. A robot has been designed and developed as an Automatic Ground Vehicle (AGV) to test the anti-collision algorithm and its fidelity. The procedure for this project was initially proving a theory in MATLAB. Following that designing a setup with Automatic Ground Vehicle which is a robot for delivering a book. The purpose of this robot is to avoid static and dynamic obstacles and deliver luggage to a specified point in the testbed. AGV is controlled by motor driver attached to the motors. The motor driver gets commands through the Arduino Uno connected to NodeMcu. NodeMcu is connected to central PC which does simulation in A* algorithm. The feedback in the complete system explained above is through live tracking done through a camera.

The Collision Avoidance strategy dexterously uses MATLAB toolbox for live tracking the robot reducing the data loss which occurred while scanning the tags. Also, the power will also be saved because instead of batteries used on each scanner we will be using a single power source on the camera. Product delivery will be faster and without any error as camera tracking will reduce latency and increase the accuracy of the path. Hence, project can be used in AS/RS industries with ease and lesser skill set. Collision Avoidance strategy will be useful in industries and will play a major role in improving the efficiency of the system.

Contents

List of Figures

List of Tables

1 Introduction

Amazon and Alibaba use small, autonomous robots to find and transport items. These exceedingly versatile robots explore utilizing fast sensors and are equipped for scanning barcode and RFID tags. They can even transfer data to each other on the off chance that they sense that an item is misplaced, consequently re-enrolling where the tag is. This gives the adaptability of the robot to deliver them to wherever require be.

AS/RS exchange information through the wireless system which contains scanning of standardized identification and RFID labels. Information losses happen in the perusing of data. On the off chance that breakdown of the robot entire framework should be stopped as the framework keeps running on way arranging of every single robot. skill and expertise are expected to repair robot in the event of breakdown or maintenance of robot.

Live tracking of robot is done through camera. With help of tracking, we can find and expel the Robot from warehousing framework and as well as guide the robot. The robot has a basic circuit clarified which can be investigated effectively.

2 Aim and Objective of the Work

My aim is to devise a collision avoidance technique alongside manufacturing a robot which delivers a book from point A point to point B without impacting static and dynamic objects. The AS/ RS system developed will have ability to navigate in environments with static objects as well as dynamic objects.

3 Literature Review

This section covers my study in for project on collision avoidance. Through study of all research papers containing why the method was adopted and why any of the method was dropped. Initially my project was to make a Quadcopter which had ability to adapt to its environment and perform manoeuvres. The Control loop used by Raffelo d' Andrea as shown(Fig. 1) was to be used[8].

Fundamental idea behind using the below loop was to improve accuracy and reduce the error during flight. The FMA is a distributed framework, both at a theoretical and physical level: it comprises of autonomously running computational procedures that are discretionary circulated among at least one physical processing stages connected together by either an Ethernet network or concentrated industrial wireless channels (Fig. 1). Essentially, the physical computing framework ranges from a Linux or Windows PC to two to at least six arranged personal computers.

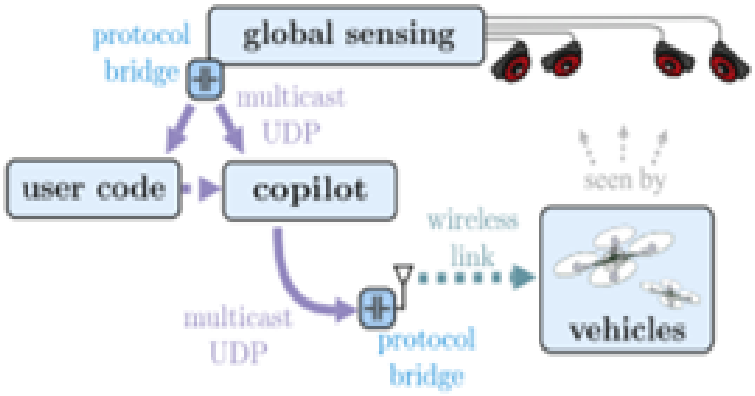

FIGURE 1: Various Flight Parameters

The above mentioned research was done before deciding the topic of my project. As the above project required longer time, more personnel and knowledge. The project then was scaled down from three-degree Quadcoper to three-degree Automatic Ground Vehicle(AGV).

My first task was to locate my AGV in 2D space. The research by Guang-yao Jin and Xiao-yi Lu which mentioned to track the robot in real time with help of RFID tags was used[1]. This paper was chosen because it solved the problem of using three RFID tags instead of multiple Tags for locating a single object. Earlier the tracking was done with help of a large number of RFID tags and data taken from them regarding the location of the target. This method involved high inaccuracy and errors as the output was result of a large data set.

As a solution to above problem the paper proposed to track the indoor robot by triangulation method and using 3 RFID readers as mentioned in - (Figure 2)[1].

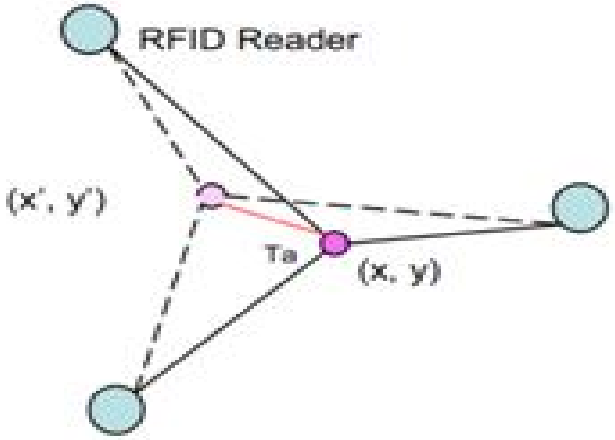

FIGURE 2: The estimated error

In application of these techniques there are huge errors were found as spectrum of RFID was limited. RFID had to be glued to each possible obstacle leaving loopholes for moving obstacles. So, it was decided to implement MATLAB OpenCV Toolbox object detection and collision avoidance.

Motor requires three loop motor control. One of three loops is needed torque control. For controlling the torque, we need current data.In paper by Modular Motor Driver with Torque Control for Gripping Mechanism by Dickson Neoh Tze How, the mechanism was as follows:PC was used to program Arduino and evaluate the data from the Arduino Uno. PC was connected to Arduino UNO. The purpose of using Arduino UNO was to test the current sensing mechanism on the structure using motor driver (L298 Board). The motor driver will collect data from the encoder and send to

Arduino Uno.[2].

Hence, the mechanism was simple to use and fabricate. The diagram of the whole setup is as shown below. However, this couldn't be implemented to use shown mechanism as there were lot of inaccuracies and errors. We needed accurate values from the Arduino and Output was showing Error.

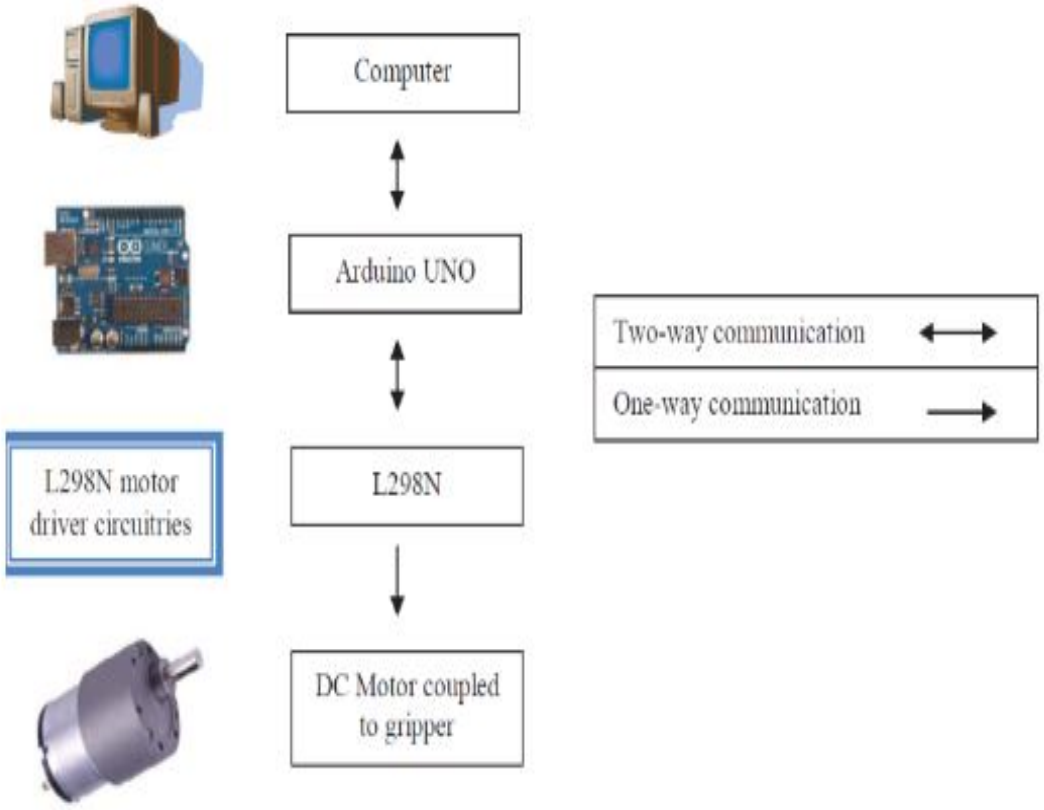

FIGURE 3: Connection between components

Indoor Positioning System(IPS):

Indoor positioning systems (IPS) empower finding the situation of articles or individuals inside structures. Since GPS is erroneous in inside spaces in light of the fact that there is no visual contact with the GPS satellites, an IPS must utilize other situating strategies. These include, for instance, the Wi-Fi or Bluetooth Low Energy (BLE), yet additionally arrangements such as RFID. The first and most vital advance in the usage of indoor situating frameworks is the choice of the indoor situating strategy and hardware.

IPS consists of parts:

3.1 *Software*

Trilateration is an upgraded version of triangulation. Data from a single receiver pinpoints a position to a large area of the test-bed. Adding data from a second receiver narrows the position down to the region where the two spheres of receiver's data overlap. Adding data from a third receiver provides a relatively accurate position, and all the GPS units require three receivers for an accurate placement. Data from a fourth receiver — or more than four receivers—enhances precision and determines accurate elevation or, in the case of aircraft, altitude. Receivers routinely track four to seven receiver or even more simultaneously and use trilateration to analyze the information.

In straightforward terms, trilateration is a mathematical strategy in which the location of a point in space is calculated utilizing the distances from such a point to a progression of known geometrical substances, e.g. a circle.

In a 3D space, if a point of interest lies on the surface of three intersecting circles, at that point, knowing the centre of the circles and their radii, it is possible to limit the probable location of the point. Beginning with the general case, circles focused at (x1, y1), (x2, y2), (x3, y3) is appeared in Figure (4) Simplified Trilateration in 2D:

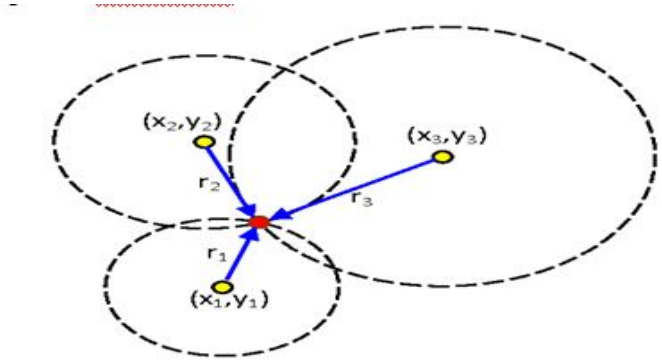

FIGURE 4: Trilateration Method

$$(X - X_1)^2 + (Y - Y_1)^2 = (r_1)^2$$

$$(X - X_2)^2 + (Y - Y_2)^2 = (r_2)^2$$

$$(X - X_3)^2 + (Y - Y_3)^2 = (r_3)^2$$

As seen above we have three equations for single point. Hence, from three equations two unknowns can be found out from solving these equations.

3.2 *IPS Hardware Options*

3.2.1 *Radio Frequency Identidfication Devices (RFID)*

A basic RFID framework has tags attached to all data that should be followed. Fabricated using a small tag-chip, these labels are otherwise called incorporated circuit (IC) and are associated with a receiving antenna that can be incorporated with different kinds of labels, for example, clothing hang labels, names and security labels – yet in addition modern resource labels. Basically, every label chip contains memory that is stored in it and contains the Basic Product Code (EPC) of the product and other data enabling it to be followed and identified by the UHF RFID scanner or reader all over.

Then again, the UHF RFID reader is a network associated gadget which can either be fixed or mobile – and features an antenna apparatus that sends the power and gets motions as orders to the labels (Fig - 5). It is an access point for RFID tagged product with the goal that the labels information is made accessible to each application.

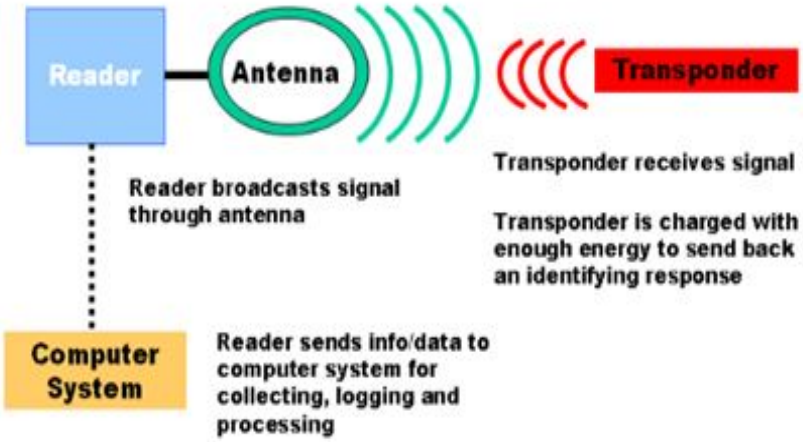

FIGURE 5: Basic working of RFID

3.3 *ZigBee*

The Zigbee standard defines a distance (10 meters) and low rate wireless personal area network of 204 GHz frequency. A basic Zigbee node is small and has low complexity and cost. It consists of a micro-controller and a multi-channel two - way radio on a single circuit board. Zigbee is designed for applications that require low power consumption and low data throughput, and requires a button battery life of 2 years (Fig - 6).

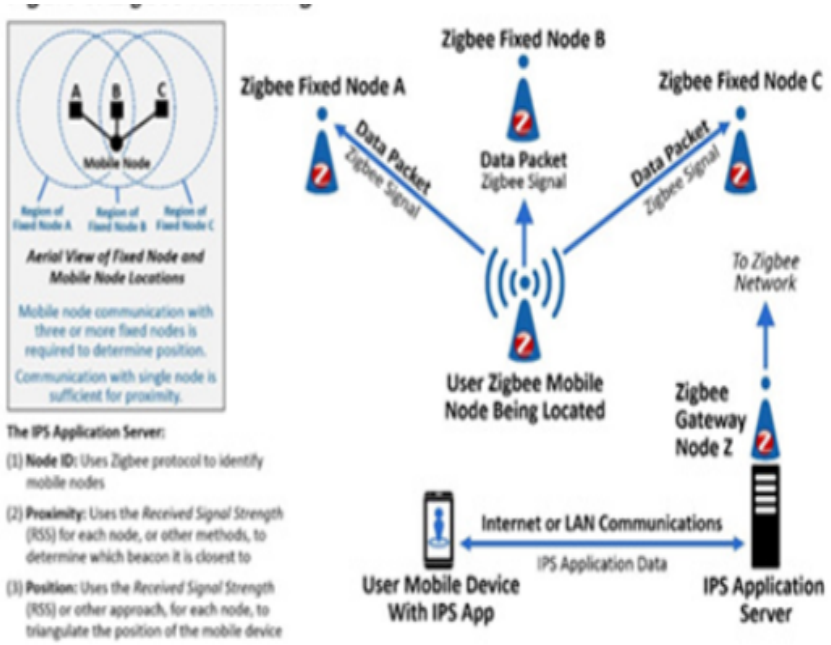

FIGURE 6: Zigbee Network.

3.4 *i-Beacons*

Beacons are placed at several positions in the building. They send signals to mobile devices via Bluetooth. In this manner, it is possible to determine their position continuously and transmit it to the indoor navigation system. An app installed on the smart-phone interprets these signals (Fig - 7).

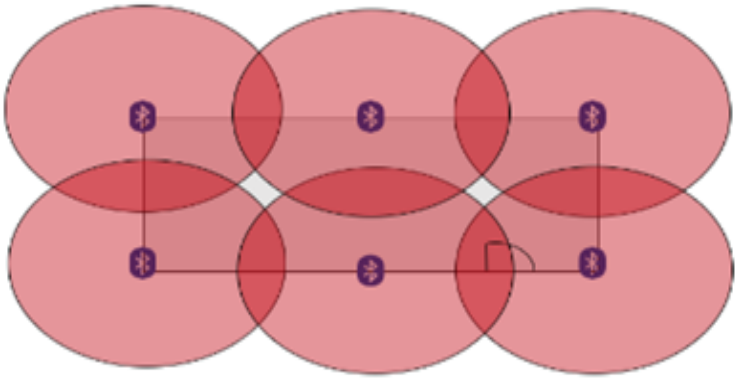

FIGURE 7: I-Beacons

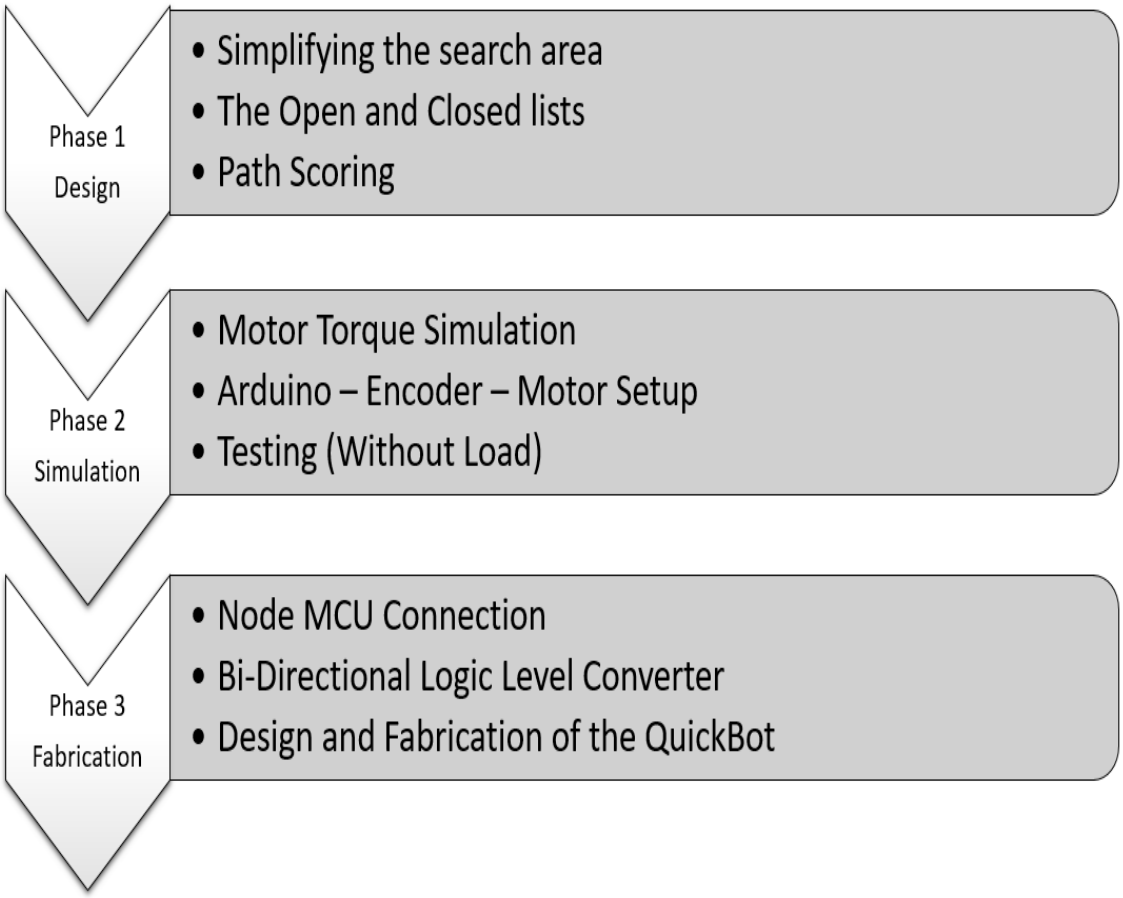

FIGURE 8: Flow Chart of project

4 Trajectory Planning - Cart Location (Phase 1)

After locating the robot in 2D plane, the next task is to give a trajectory to the robot in the work-space. For this purpose, the A* algorithm is used. The concept of the A* algorithm is explained below.

4.1 *Simplifying the search area*

Consider that the accompanying framework design in the figure Fig-9 is the library of School of Engineering and Applied Science. What is done here is isolated the entire territory of the library into squares of equivalent sizes to make a matrix. Separated like this inquiry region is disentangled and can be viewed as a two-dimensional exhibit. The aggregate proving ground is of 121 square units. The main idea behind applying this thought is that this information can, without much of a stretch, be changed over into pixels while coding.

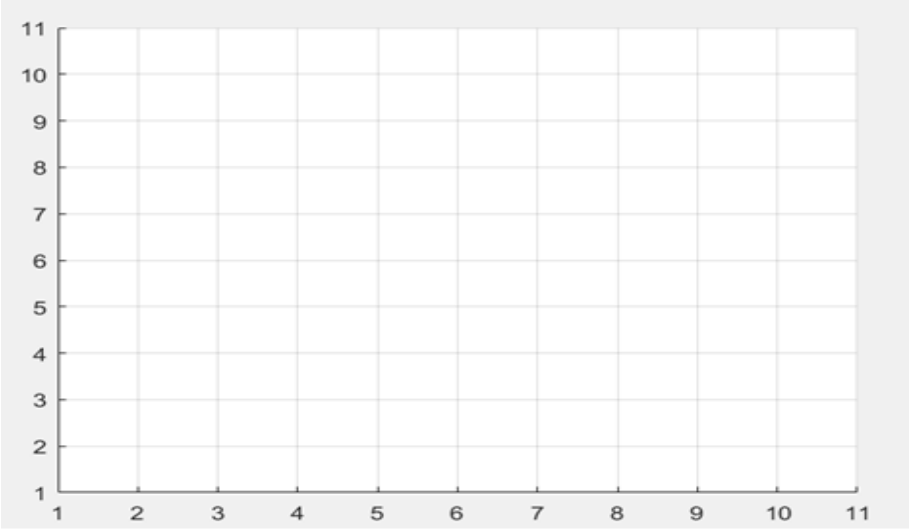

FIGURE 9: Trajectory Planning

4.2 *The Open and Closed Lists*

For using the memory in a more efficient manner the accompanying two records are made. One to write down all the squares that are being considered to find the shortest path (called the open list). The other to write down the square that it does not have to consider again (closed list).The BOT starts by including its present position (we'll call this the beginning position indicated by "A") in the closed list. At that point, it adds every walkable tile adjoining its present position to the open list(Fig - 10). Here's a case of what this would look like if the BOT was out in the open (green representing the open list).

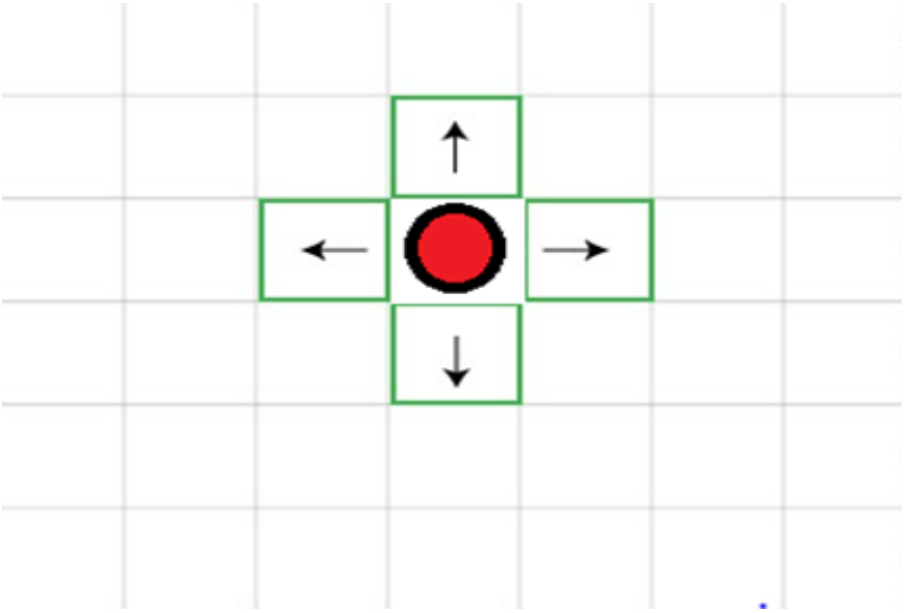

FIGURE 10: Open List

4.3 *Path Scoring*

A * Algorithm works on two basic principles - Heuristics (Estimate) and Cost function.

Each square is given a score G(N) + H(N) where:

F(N) = G(N) + H(N)
G(N) : Development Cost

G(N) is the development cost, in various squares, from the beginning to the present square. Keeping in mind the end goal to compute G(N), we have to take the G(N) of its source (the square where we originated from) and add 1 to it (Fig. 11). Along these lines, the G(N) of each square will speak to the aggregate cost of the produced way from point A until the square. For instance, this graph indicates two ways to two distinct objectives, with the G(N) score of each square recorded on the square[3].

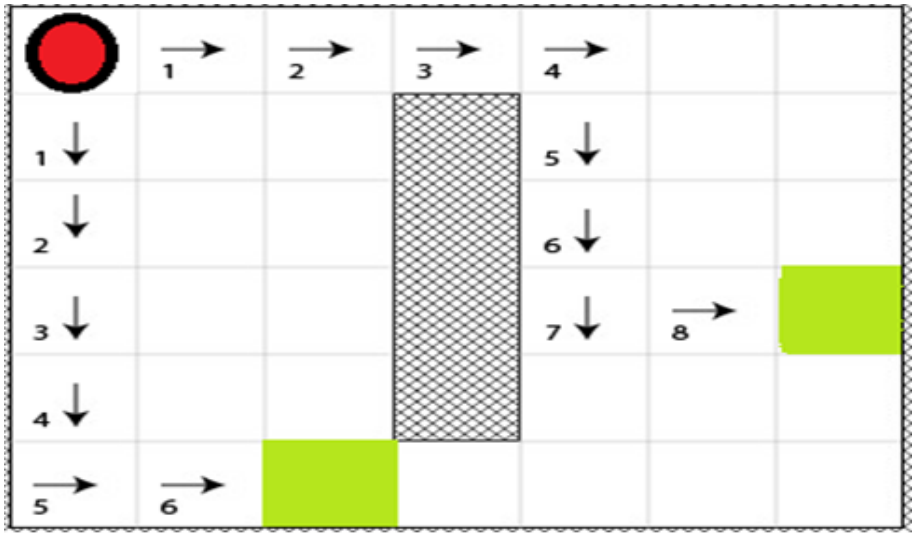

FIGURE 11: G(N)

H(N)

The cost H(N) is evaluated development cost (in various squares) from the present square to the goal point. Development cost (in various squares) from the present square to the goal point. The nearer the assessed development cost is to the real cost, the more precise the last path will be. We will utilize the "Manhattan Distance" that just checks the quantity of even and vertical squares staying to achieve point B. For instance, Figure - 12 shows an outline that shows utilizing “the Block removal" to gauge H (appeared in dark) from different begins and goals.

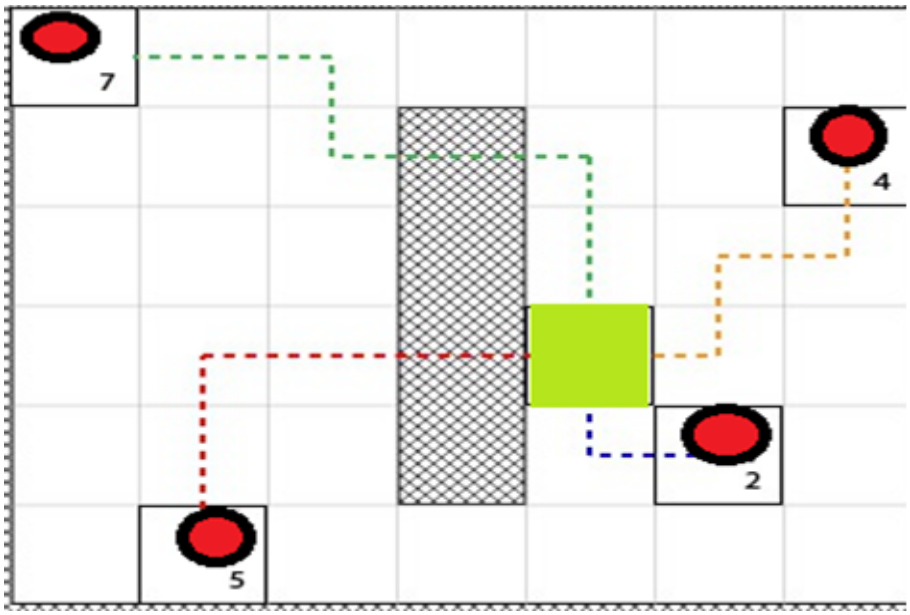

FIGURE 12: Trajectory Planning

Algorithm output after is as shown Fig - 13.

Horizontal as well as vertical walls are drawn in the below image as obstacles.

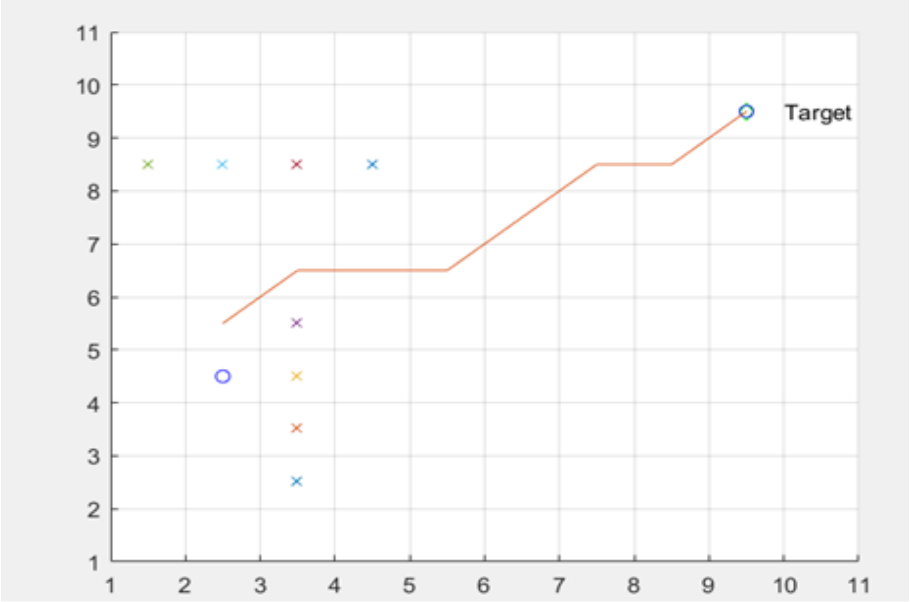

FIGURE 13: Corner Path

The problem with above path is there are many corners (Discontinuous Regions).Robot needs a smooth profile to follow and work smoothly.Hence, a MATLAB function called smooth() was used.The function interpolates around the corners and replaces sharp corners with smooth curves.Smooth profile like below are generated which are easy to follow for the robot.

FIGURE 14: Curve Path

As seen in above Fig 14 a curve path was available but if the path passes from obstacle tile which is (3,6). Hence, for robot to pass through that obstacle is not possible. For that purpose, a code to fit the path through simple three points is implemented. By giving three points to A* algorithm and giving radius to the curve the obstacle can be avoided (Fig - 15).

The curve was able to avoid corners and circumvent obstacles. But, in some cases as seen below the optimal path was not as per the defined criteria and would pass through the obstacle tile in the area.

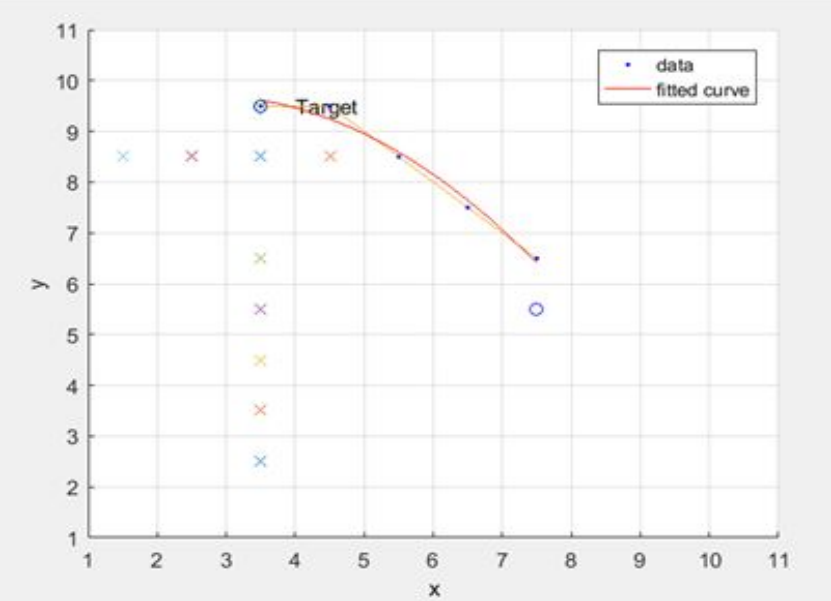

FIGURE 15: Schematic Diagram

Cubic spline data interpolation method is used to fit the curve between three points. s = spline(x,y,xq) function in MATLAB fits a curve between the points provided by us with N Degree curve. Spline function works on a logic where the arrays of X coordinate and Y coordinate must have same value if both of the coordinates have different size matrix error occurs.

Also, curve needs two different points to pass through them. If two same co-ordinates are calculated by the algorithm the curve spline would not fit the curve through them. A for() loop is added to the code which has each variable X_val and Y_val added a point value of 0.1 after each execution. Also, a condition is applied in the code which stops to plot the path in if the same co – ordinates are plotted by the algorithm.

5 Current Sensing - Cart Control (Phase 2)

The motor requires current based torque control. Current based torque control is needed to control torque in the motor. Which indirectly controls the direction and speed of robot. Voltage is proportional to speed, and torque is proportional to the current. The maximum current it could take is rated current and the corresponding torque can be found out from speed torque curve (as the speed from the voltage (rpm=k*v)) where k is the speed constant of the motor). Current sensing is done through ACS-712 chip and the circuit diagram for experiment is as in Fig - 16.[5].

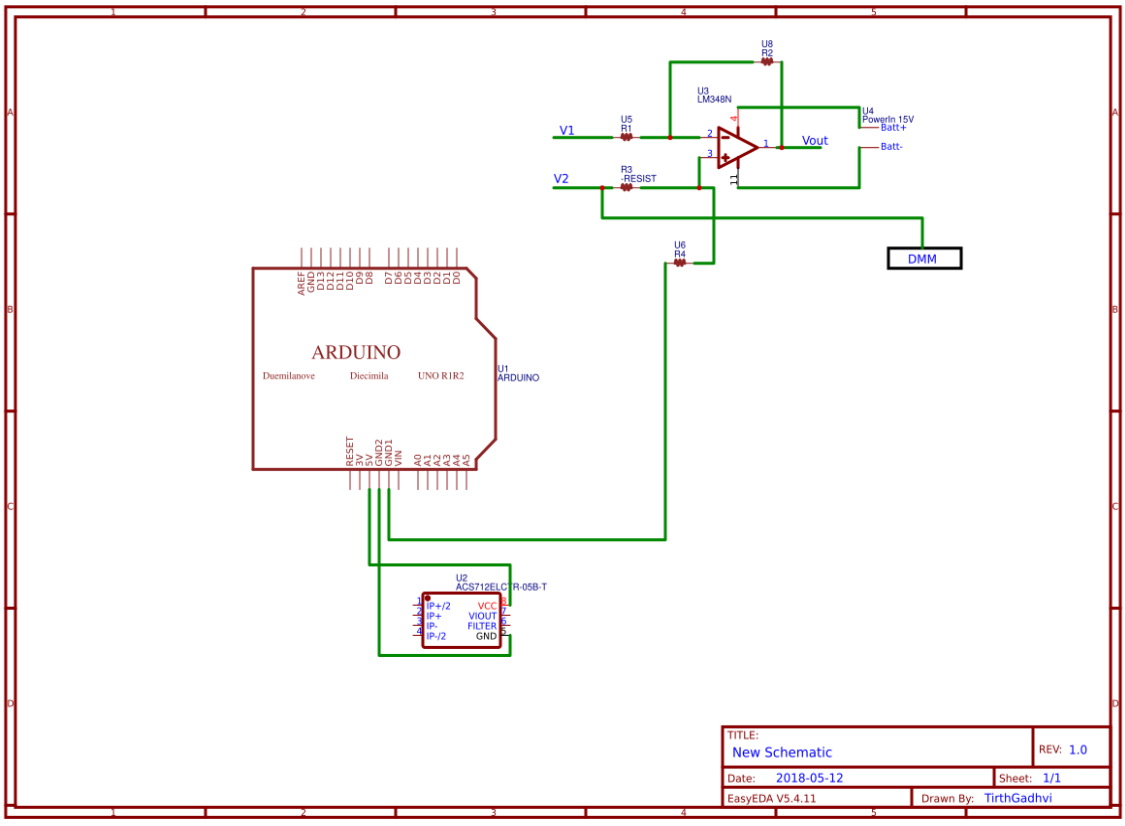

FIGURE 16: Circuit Diagram

In table 1 shown is the Data of output from the current sensor voltage and output voltage.

TABLE 1: Output Data

	Current	Sensor Voltage	Output Voltage	Voltage W/O Offset	Error	
1	0	2.475	1.366	0.975	0.391	
2	0.5	2.546	1.287	1.046	0.241	
3	1	2.614	1.496	1.114	0.382	
4	1.5	2.681	1.455	1.181	0.274	
5	2	2.747	1.528	1.247	0.281	
6	2.5	2.812	1.53	1.312	0.218	
7	3	2.881	1.62	1.381	0.239	
8	3.5	2.951	1.76	1.451	0.309	
9	4	2.016	1.81	1.516	0.294	
10	4.5	2.073	1.853	1.573	0.28	
					0.2909	Constant Error
					0.05445814	SD

As seen in above Table 1 the error of 0.3 can be avoided by giving a feedback to the system. As shown in Table 1 the current sensing was cumbersome and full of errors so it was decided to try another approach which included to utilize L298 micro-controller for Torque control. The chip has a separate pin for current sensing hence we needn't bother with a circuit for current sensing mechanism. The other favourable position of utilizing L298 motor driver is that it has an H- bridge connection. As a result of this sort of association, voltage can be in either direction. Hence, the motor can run forwards or backward as the current will flow Bidirectional.

5.1 *Motor Torque Simulation Setup*

Before fabricating the actual Robot, it is necessary to calculate the torque and current relationship for three loop motor control. The setup shown in Fig - 17 is created to recreate this present reality Torque expected to precisely control the motor.

FIGURE 17: Wooden Setup

As seen in Fig - 17, the setup comprises a wooden frame. The spring is associated with the wooden column. This spring is connected to the motor through a wire. A certain measure of force is required to pull the spring. The measure of spring that has been broadened is referred in sub-section 5.2.

The angle(θ) which will be obtained from the positioning detecting setup of Arduino-Encoder-Motor will give us an expansion of spring. The amount of force which is required for this setup is already set from the L298 motor driver circuit. The force and angle(θ) will give the torque required by the motor. Current for this torque can be estimated through DMM or seen in the Power supply.

5.2 *Arduino – Encoder – Motor Setup*

Arduino and encoder are associated as appeared beneath circuit picture. The motivation behind this circuit is to precisely measure the angle(θ) value of revolution of the shaft of the motor. As found in the circuit the angle(θ) value changes between 0 – 360° and Angular Velocity is steady at 3 rad/s2. The setup beneath keeps running on 5V. When the voltage is not supplied the angle(θ) remains constant and Angular Velocity is zero. PWM is supplied to the encoder through the Arduino(Fig - 18).

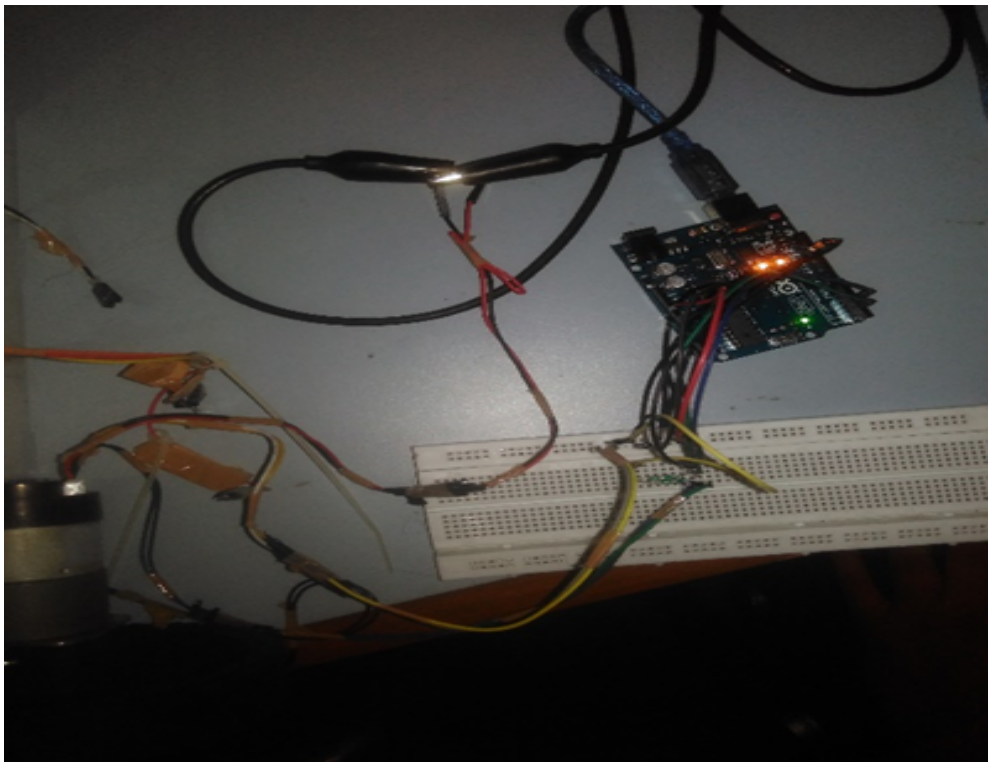

FIGURE 18: Arduino – Encoder – Motor Setup

5.3 *Testing (Without Load)*

SMPS was used to control voltage and current applied to motor. angle(θ) and angular velocity(ω) are varied for each value with respect to the current and voltage. Corresponding Values are as shown in Table - 2,3,4,5,6 for each current and voltage without any load(Spring). Below shown is serial monitor output in Arduino IDE.

TABLE 2: For 0 Volt:

	ω	θ
1	0	0.00
2	0	0.00
3	0	0.00
4	0	0.00
5	0	0.00

TABLE 3: For 5 V – 0.07A:

	ω	θ
1	3	180.60
2	3	191.10
3	3	201.60
4	3	243.45
5	3	253.95

TABLE 4: For 7 V – 0.07A:

	ω	θ
1	4	172.05
2	4	187.20
3	4	202.65
4	5	248.85
5	4	294.90

TABLE 5: For 10 V – 0.09A:

	ω	θ
1	7	311.10
2	7	332.85
3	7	80.55
4	7	123.75
5	6	145.35

TABLE 6: Detailed Data:

	Voltage	Current	Theta	Angular Acceleration
1	5	0.07	0	3
2	6	0.07	0	4
3	7	0.07	0	4
4	8	0.08	0	5
5	9	0.08	0	6
6	10	0.09	0	7

6 Node – MCU - Cart Data transmission (Phase 3)

NodeMCU(Chip) is a low-cost Wi-Fi Module Chip[6]. This chip is utilized to transfer data from one source to another wireless. The Chip is used to exchange data from Master PC to Arduino set on the robot. The data will be forwarded from Master PC through WI-FI to the Chip. The chip will fetch data from the string sent through WI-FI. In case of more than one robots, the data will be automatically taken by respective chip by identification. Data contains simulated (X, Y, θ)(Fig - 19). We accomplished this part of our project in three parts.

Task 1: Transferring Data from Master PC to chip through Arduino as shown below connection.

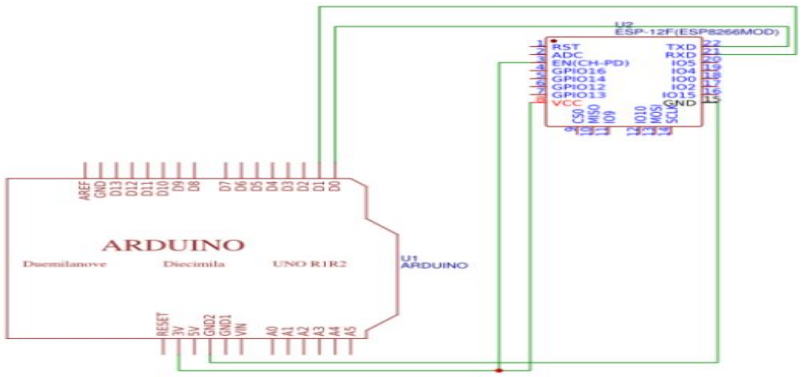

FIGURE 19: Circuit Diagram

Task 2: Transferring Data from Master Chip to robot chips remotely. Data transfer through Wi-Fi is shown in the video uploaded.

Task 3: Orientation (θ) and coordinates (X, Y) sent again from robot to Master PC for calculation of next factors and in addition feedback.

6.1 *Node MCU Connections*

The connection was established following the above steps. The coordinates are calculated from the MATLAB which are transferred from the MATLAB to arduino through 5V to 3.3V voltage level converter[7] .

6.2 *BI Directional Logic Level Converter*

Bi Directional Logic Level Converter converts voltage 5V to 3.3V (High to low) and 3.3V to 5V(Low to High) with I2C connections and other collector type gates. Logic Level Converter needs to be connected from both the voltages high (5V) and low (3.V) . As seen in Fig -20 a secure connection is established from Arduino - Logic Level Converter - NodeMcu From one PC to another. Our next task is to send same data from Central PC to three different robots via same connection Arduino - Logic Level Converter - NodeMcu Arduino(Robot) - Motor Driver (Fig - 20).

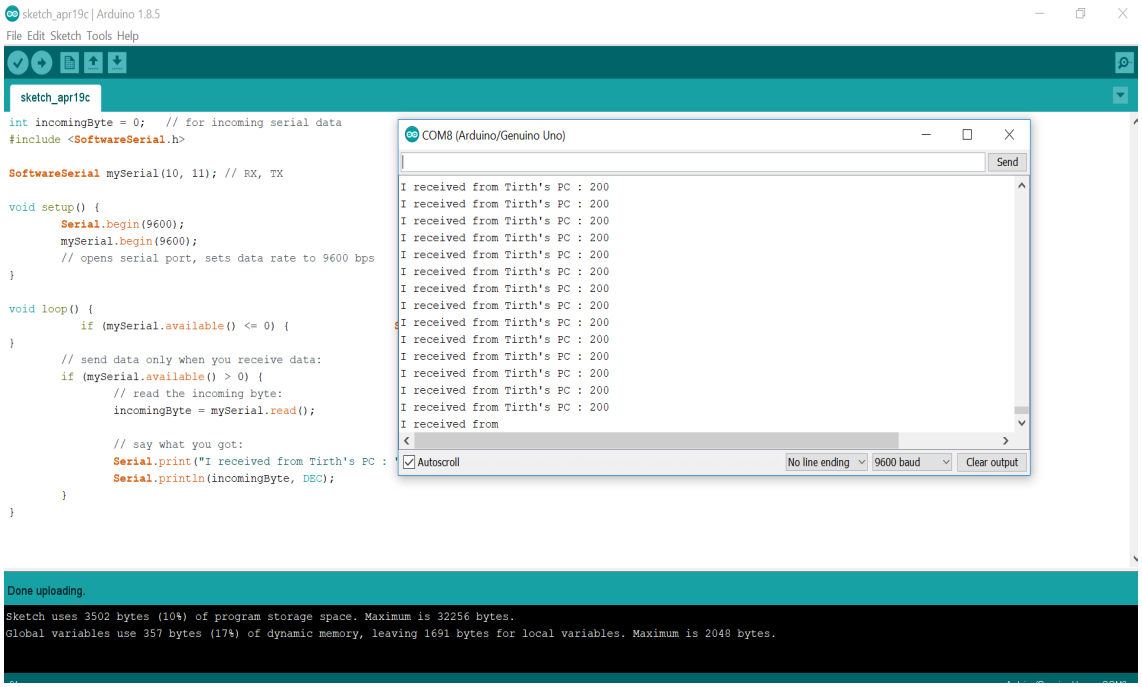

FIGURE 20: Output in slave PC from master PC

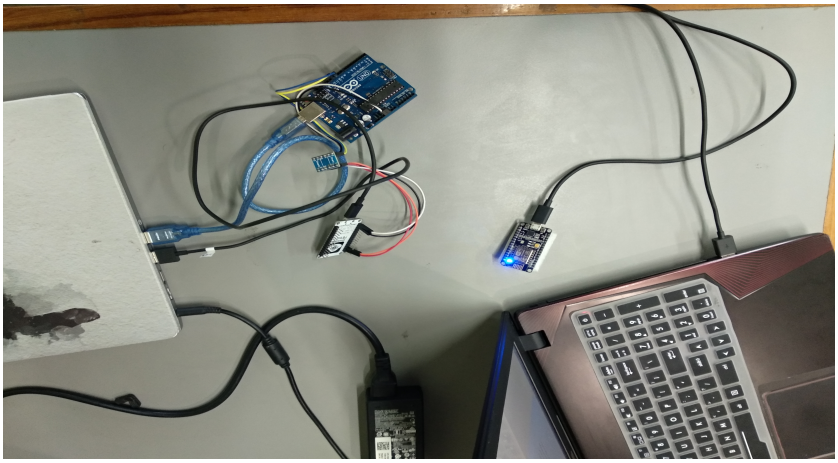

FIGURE 21: Circuit Connections in actual

6.3 *Design and Fabrication of the QuickBot*

As per Quickbot design initially, two pieces from the Acrylic sheet of 6mm were made. The Upper and Lower piece of the sheet were made in the Laser Cutter. Following that Drill holes were made in to sheet on both sides for fixing the motor clamps.Holes were of diameter of 5 mm (Fig - 22). Clamps were fixed on both sides which are to be fixed with the motor.

Drilling for the front wheel of robot was done of 1 cm which will be used for passing the shaft of the motor through the acrylic sheet. Clamps were mounted on the both sides of the motor. 5 mm holes around the larger hole of 5mm was done to pass the motor screws through the Acrylic sheet.

FIGURE 22: Robot with clamps

3D printer was used for making the coupler of motor shaft with the front wheel of the robot. Polylactic material is used for making the coupler. Wheel along with the shaft of the motor is joined by this coupler.

Motor was mounted on the front wheel with the coupler and simultaneously with the two rear wheels with the clamps. Wheels were mounted on the motor on the rear side of the Quickbot.

FIGURE 23: Robot with motors

Now, for fixing the Arduino Mega and Motor Driver holes of 3mm and 6mm were done in the acrylic sheet.With help of these holes Arduino Mega and Driver IC were fixed on the robot.

Hole of 1cm was drilled at a distance of 6 cm from bottom of the acrylic sheet and 1.5cm from the side. Same holes were drilled on identical acrylic sheet for fixing this sheet as a roof for the robot as well as mounting the Nodemcu.

Now, power each of the component - Nodemcu, Arduino Mega, Motor Driver is given through 12V battery placed between them.For conversion of power from 12 V battery Nodemcu, Motor Driver, and Arduino Mega a separate circuit has to be made with the diagram as Fig - 25.

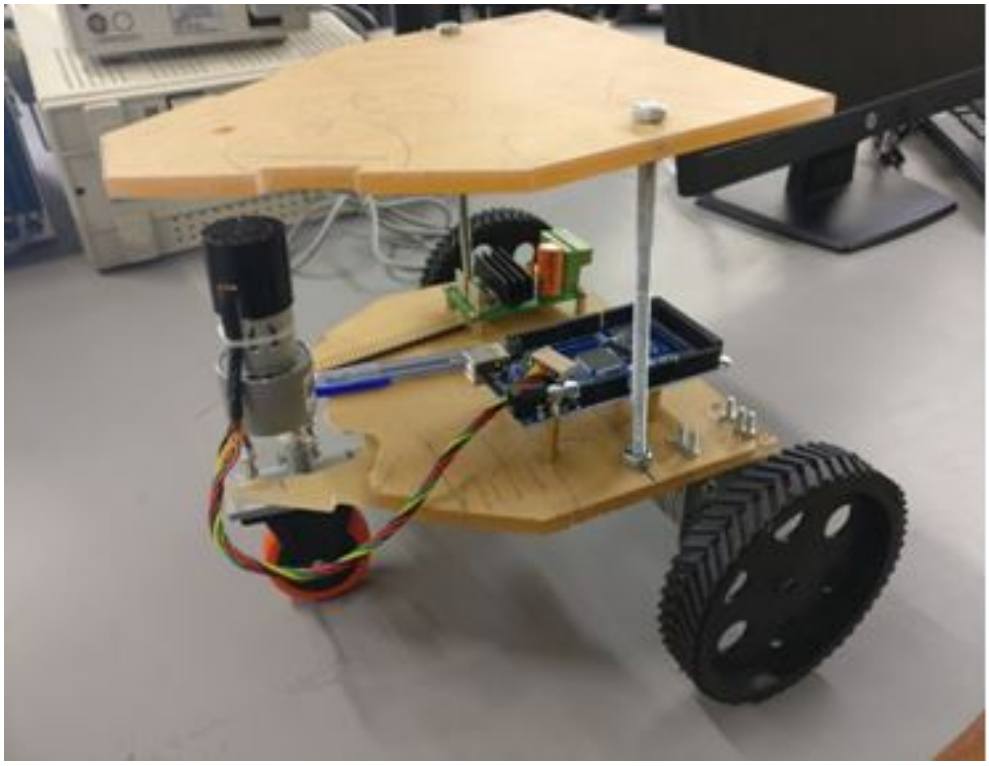

FIGURE 24: Robot with motor driver, Arduino , battery

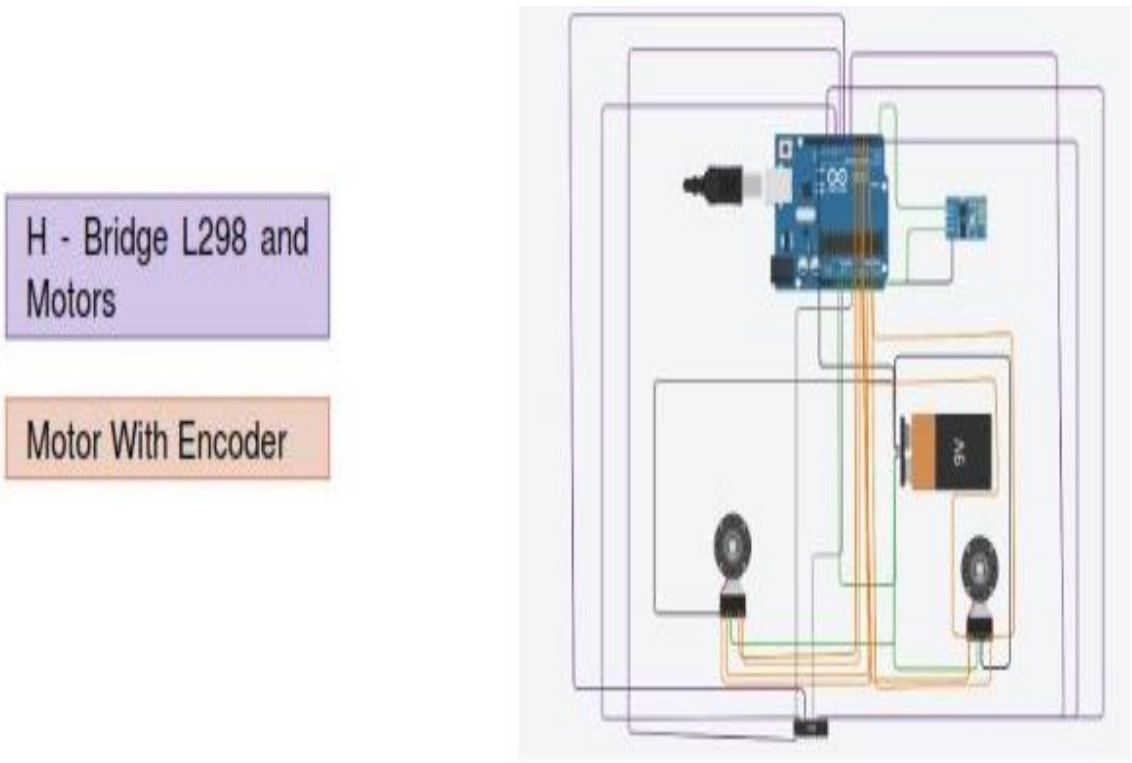

FIGURE 25: Schematic Diagram

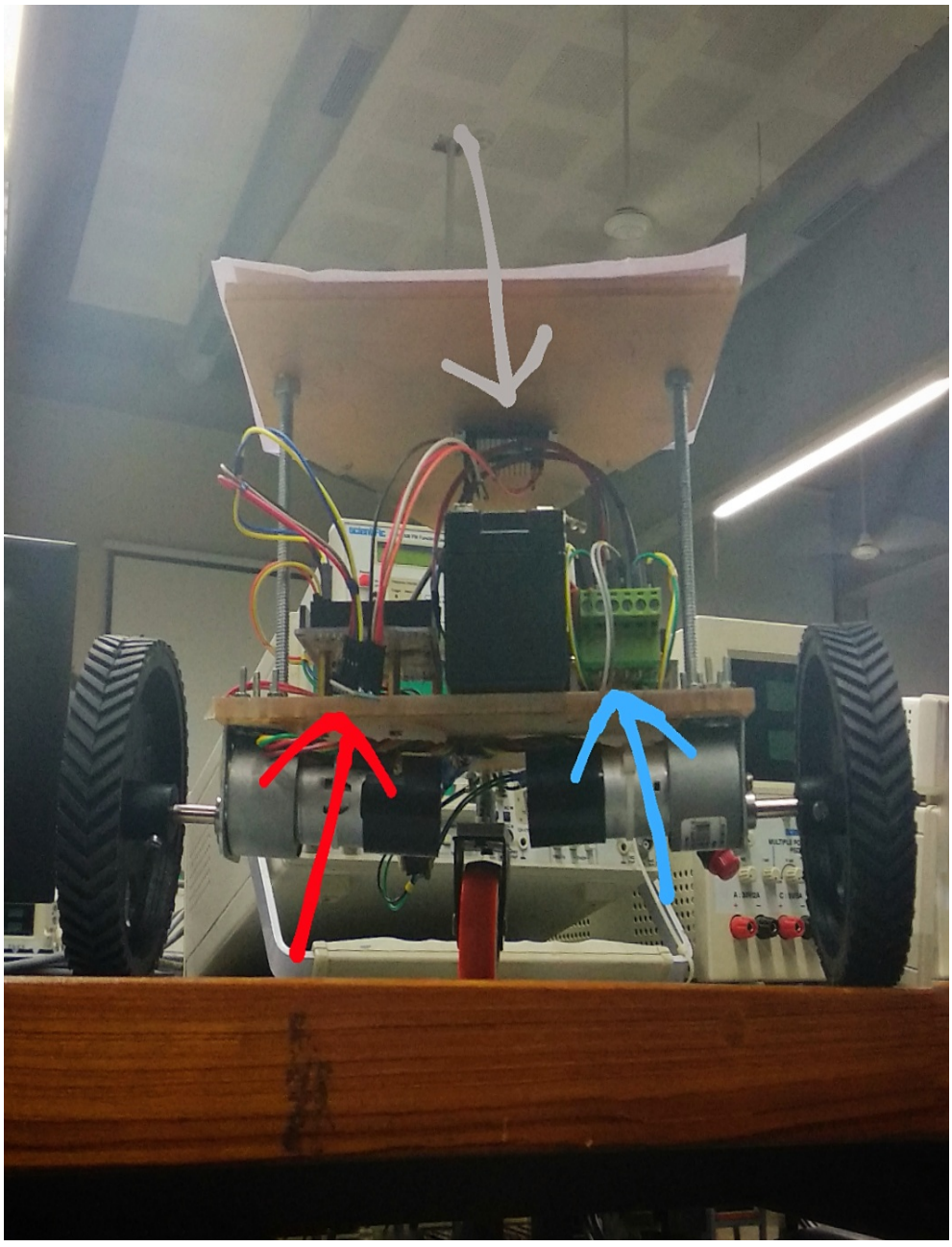

FIGURE 26: Robot with connections

7 Results and Discussion

Current sensing in beginning was done through the hall sensor circuit (ACS 712). But, the sensor was not able to provide exact value to Arduino. The current value in milliampere were very ambiguous and erroneous. The value of current varied from zero to different values which is not suitable for the torque simulation. ACS-712 has very low accuracy and fidelity. Hence, it was decided to make torque simulation setup with L - 298.

Motor torque simulation setup comprised of spring mechanism , motor with encoder and Arduino Uno for current sensing. But the data received from L298 pin was incorrect. The current value in power supply was different from output of L298 current sensing pins. So, the method of current sensing was changed from Motor Torque simulation setup to Motor Driver with Current sensing.

Current values displayed in arduino received from current sensor had a little error in milliampere and we were able to remove error.

Co-ordinates and inclination of the robot will be sent to the robot from the central PC with the help of NodeMcu as explained above.At first the data transmission had to be done through WIFI through server.But due to complexity of the setup and inaccuracy of the data transmission NodeMcu data communication was selected. In NodeMcu Data transmission also we were facing errors such as loss of data , garbage values , random output from the NodeMcu. Zigbee data connection if had been used data tansmission would be fast and transmission losses would be lesser.

8 Conclusion and scope for future work

Data Transmission: NodeMcu transfers data through the Wifi Hotspot at minimum speed of 1 Mbps. Although this speed of data transfer is high and avoids latency data losses occur and Zigbee/Xbee is a better option for efficient use.

Current Sensing: ACS-712 and L298 with spring mechanism failed to detect current with accuracy and unwavering values. Motor with current sensing mechanism or Motor driver with current sensing can be used to save time as well as manpower.

Algorithms - Navigation: Algorithms such as D*, Heuristic and greedy search can be used to navigate an obstacle path through the test area however, each algorithm has a drawback. A* algorithm can overcome these drawbacks but at the same time is cumbersome and convoluted.

Robot Design and fabrication: During fabrication of robot's coupler through shaft and pins, wobbling occurred between the shaft and pins. So, as a solution we designed and fabricated a coupler in 3D printer.

Circuitry: Circuit was planned and made on PCB after designing the robot. we faced problem fitting the PCB in the lower section of the robot.

8.1 *Real Time Tracking*

The camera will be utilized for real-time tracking of the robot in the library. As observed in beneath picture real-time tracking of the robot will be done through Computer Vision. The Barcode pattern will be set on every robot which will locate the robot in the library. As shown, the corners are located real time.

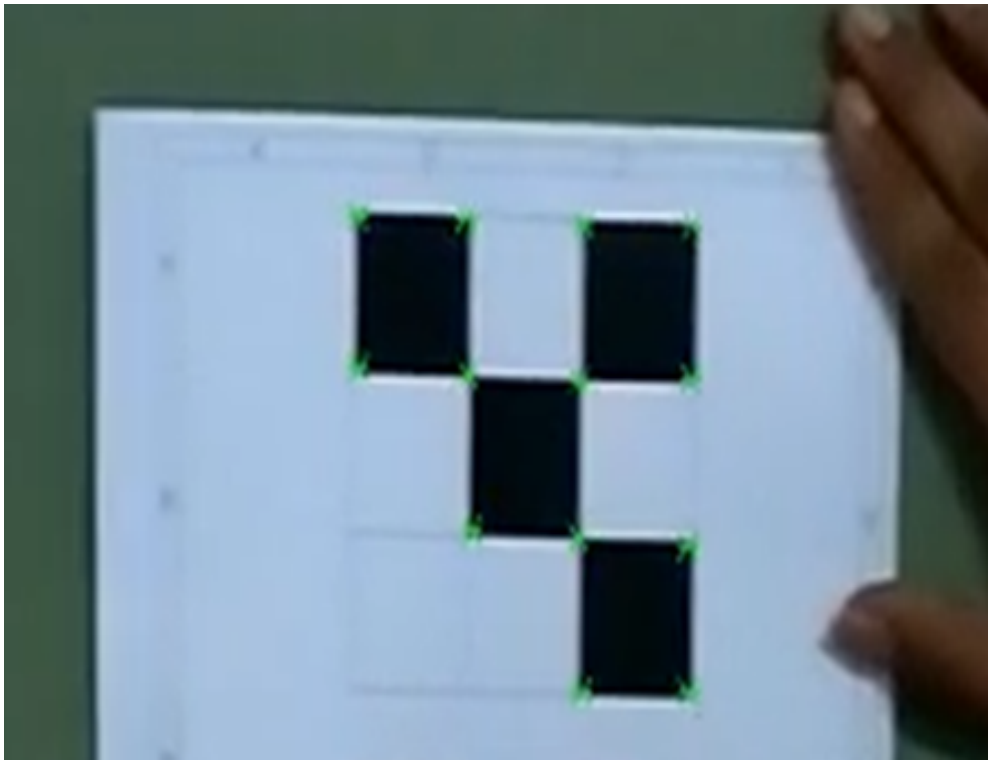

FIGURE 27: Real Time Tracking

8.2 *Tracking of Obstacle using MATLAB OpenCv Toolbox*

The live tracking of the robot is done by the camera which gives constant feedback to the central PC for obstacle avoidance. Initially the obstacle needs to be tracked then around the tracked object(charger) squared region as shown in the figure - 28.A potential field shall be created which will repel the robot resulting the robot to move away from the obstacle. As seen the white coloured charger is tracked.

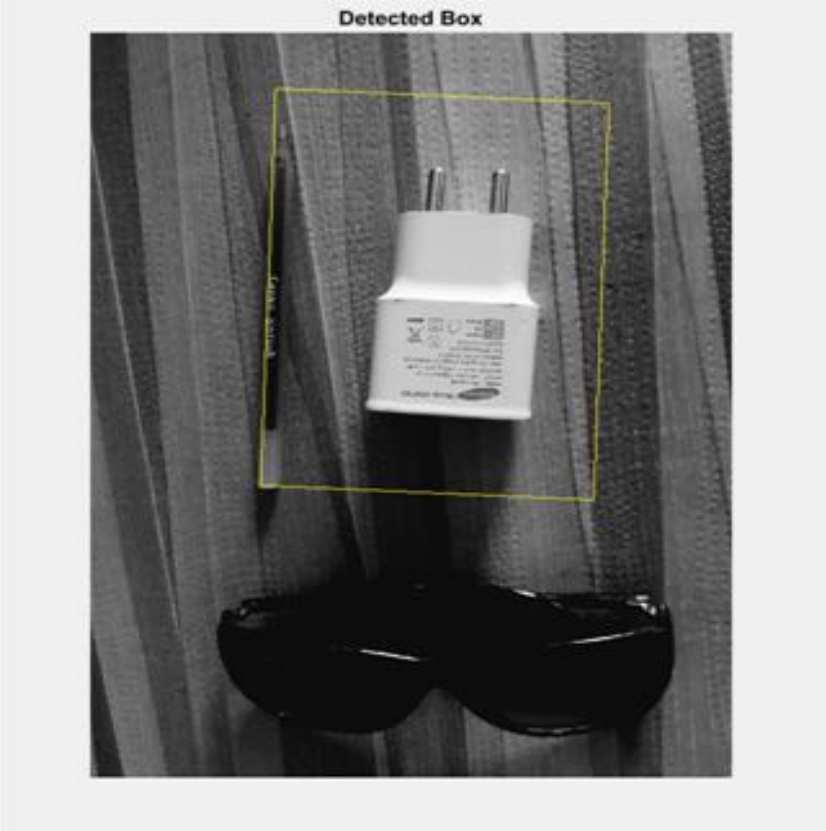

FIGURE 28: Detected Box

In real time tracking we will replace the white coloured charger with a blank white paper hence MATLAB shall read the pixels white – (Charger in image) as obstacle after implementing the code. Before tracking charger, we need to save the pixel and features of charger in to the MATLAB file as identification for code and extracting 100 features from the image as shown in Fig - 29.

FIGURE 29: 100 Strongest feature points from Box Image

The features extracted from the image – number above will be used as shown below to differentiate pixels(charger here) from its environments. Below seen is an example how the object(charger) is tracked from in both images and the lines show matching points from the viewpoint of the MATLAB toolbox (Fig - 30).

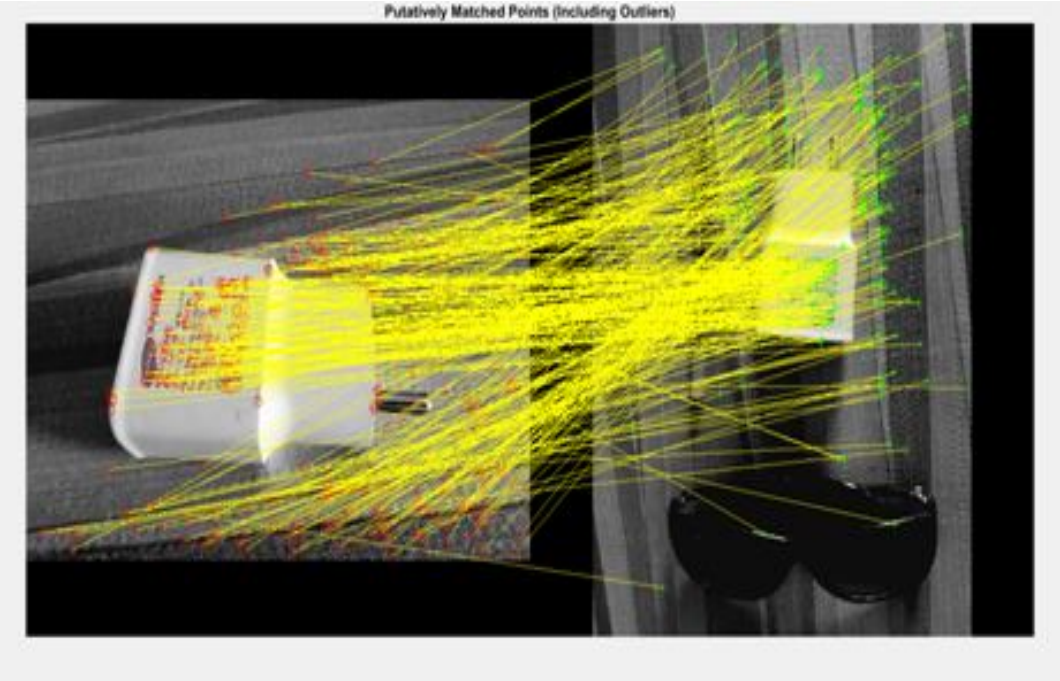

FIGURE 30: Matched Points

9 References

[1] Jin, Guang-yao and Lu, Xiao-yi and Park, Myong-Soon, 1993. Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on

[2] *Modular Motor Driver with Torque Control for Gripping Mechanism*. Dickson Neoh Tze and Baharuddin, Mohd Zafri and Mohideen, Syed Sulaiman Kaja and Sahari, Khairul Salleh Mohamed and Anuar, Adzly 322(10),Procedia Engineering, 2012.

[3] Introduction to A*
http://theory.stanford.edu/ amitp/GameProgramming
/AStarComparison.html

[4] Review of stability analysis of non-linear control systems Boro, Banajit

[5] Current Sensors
http://www.kohshin-ele.com/en/products/cs$_h$*allsensor.html*

[6] SPI Transactions between ESP8266 and Arduino UNO
https://deeplyembedded.org/esp8266-spi

[7] Control of a nonholomic mobile robot: Backstepping kinematics into dynamics. Fierro, Rafael and Lewis, Frank L,Journal of robotic systems, 1997.

[8] A simple learning strategy for high-speed quadrocopter multi-flips Lupashin, Sergei and Schöllig, Angela and Sherback, Michael and

D’Andrea, Raffaello, 1993. Robotics and Automation (ICRA), 2010 IEEE International Conference