

**Order Management System and Vision Based Autonomous Delivery Robot for Restaurants**

Graduation project report

Submitted in partial fulfillment of the requirements for the

**Honors Degree of Bachelor of the Science of Engineering**

in

The Department of Electrical and Telecommunication Engineering

Faculty of Engineering

South Eastern University of Sri Lanka

University Park, Oluvil, #32360

Sri Lanka

by

**A. N. F. M. Begum (SEU/IS/13/EG/002)**

**G. S. U. AHMADH (SEU/IS/13/EG/011)**

**I. I. S. PREMASRI (SEU/IS/14/EG/032)**

Supervised by:

Dr. W.G.C.W. Kumara

Mr. M.F.M. Abdul Cadar

05 July 2019

# ABSTRACT

In the recent years, application of robots in daily life has become very normal. Robot autonomy has been exploited to extend their usage in several daily life. For mobile robots to have an autonomy, a very good and a reliable navigation system is vital. Object avoidance, path planning, obtain maps and other information about their environment are very much required for such an autonomous navigation. The process of determining the location of the robot and generating a map of its environment has been termed in the literature as Simultaneous Localization and Mapping (SLAM). There are various approaches by making use of Light Detection and Ranging (LiDAR), Sound Navigation and Ranging (SONAR) sensors, and depth cameras in order to achieve this purpose. However a vision based approach is more preferable in replacement of these techniques as it enables the paths for adopting to suddenly changing environments. In vision based approaches for the robot navigation, most of the existing systems are working based on a single camera mounted in the front side of the robot to sense the immediate environment in front of it. They use Markers like ARUCO markers and similar other various approaches. Considering their inefficiency, inaccuracy and prone to errors, there are some advanced systems which are deploying an external camera to sense the entire environment and to build the map and thus having the path planned according to the overall status of the environment. Our approach is to build such a system and to implement it on a real application.

This project is consisting of developing such centralized-external camera based navigation system and to deploy it on a restaurant delivery robot. Existing restaurant waiter robots are mostly involving some classical navigation systems such as visible or magnetic line following. Therefore, this was chosen for implementing the system. In addition to that, our project is also including a web based order management system for customers to reserve tables and order foods and thus the navigation system can acquire the delivery table details and navigate the robot to the table.

# 

# TABLE OF CONTENTS

ABSTRACT ii

TABLE OF CONTENTS iii

LIST OF FIGURES v

LIST OF TABLES vi

1. INTRODUCTION 7

1.1 BACKGROUND 7

1.2 OBJECTIVES 7

2. LITERATURE REVIEW 9

2.1 Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments 9

2.1.1 Image Processing 9

2.1.2 Robot Localization 11

2.2 Path planning algorithm for robot navigation 13

2.3 Restaurant delivery robot design and control system 14

2.3.1 Design of the Robot Mechanism 15

2.3.2 The Fixed Path Guided Systems 15

3. PROPOSED METHODOLOGY 16

3.1 Customer order management system – For customers to reserve the table and place the orders. 16

3.2 External vision-based path planning – Planning a path for the robot to navigate to the delivery table based on the current objects in the environment 17

3.2.1 Capturing and Processing the Images of the Room 17

3.2.2 Detection of Objects in the Room and Classifying of Moveable Paths and Obstacles 17

3.2.3 Generating the Path for the Robot to Its Destination 18

3.2.4 Regeneration of Path When a New Obstacle Is Introduced in the Previous Path 18

3.3 Robot’s hardware and control system 19

3.3.1 Defining Requirements of the robot 19

3.3.2 Hardware Design 19

3.3.3 Robot drive mechanism 19

3.3.4 Control System 20

4. PRESENT PROGRESS AND IMPLEMENTATION PLAN 22

4.1 Order management system 22

4.2 Navigation system 24

4.2.1 Image capturing and calibration of camera 24

4.2.2 Object detection and obstacle-path map building 24

4.2.3 Path planning and obstacle avoidance by re-planning of path 26

4.3 Robot Design and weight analysis 27

4.3.1 Designing Robot model 27

4.3.2 Total Weight Estimation 28

4.3.3 Calculation of motor torque 28

4.3.4 Center of Gravity Calculation 29

4.3.5 Hardware Selection 30

REFERENCES 32

# LIST OF FIGURES

Figure 1: Image before Camera calibration [1] 10

Figure 2: Image after Camera calibration [1] 10

Figure 3: Map of the experimental location [1] 11

Figure 4: Schematic diagram of the IBDMS [1] 12

Figure 5: Schematic diagram of the PLDMS [1] 13

Figure 6: Path planning in workspace with many obstacles [3] 14

Figure 7: The first generation of the restaurant service robot [4] 15

Figure 8: System diagram of the overall proposed system 16

Figure 9: Ultrasonic sensor detects immediate obstacle in the front [5] 18

Figure 10: Proposed differential drive system for the robot 20

Figure 11: Flow chart showing the control system for the robot 21

Figure 12: Home page 22

Figure 13: Login page 22

Figure 14: Dish order on web page 23

Figure 15: Orders are listed on the admin interface. 23

Figure 16: Original bird view image on the left and the calibrated and corrected image on the right 24

Figure 17: Image segmentation and clustering. Image on the left shows the identification of objects and moveable path. And the image in the left shows after applying a contour. 25

Figure 18: Image with moving objects on the right and the final processed image after applying segmentation and clustering 25

Figure 19: Binary image of objects and the moveable path is on the left, generated path is in the right 26

Figure 20: Sudden introduction of a new object on the path of the robot is on the left and the regenerated path is on the right 26

Figure 21: 3D model of the proposed robot design 27

Figure 22: Single line diagram shows the weight distribution 29

Figure 23: NEMA 23 Stepper motor and its properties 31

Figure 24: TB6600 Stepper motor controller and its properties 31

Figure 25: Arduino UNO development board 31

Figure 26: NVIDIA Jetson Nano Development platform 31

# LIST OF TABLES

**Table 1: Estimated weight for the components and total weight of the robot design** 28

# 

# 1. INTRODUCTION

## 1.1 BACKGROUND

Path planning and navigation of autonomous robots and vehicles have always been one of the mostly researched areas in robotics and automobile industry. Giving the autonomous robots a power to navigate to their target destination without having a collision nor a mis-navigation is a very essential task. Indoor robots, unlike autonomous vehicles on outdoor environment, their configuration space is not typical and also the obstacles are mostly unpredictable. Number of indoor robots like restaurant service robot, elder care robots and room cleaning robots etc., are increasing to take their places everywhere. Some of them have cameras to see their immediate surroundings and some have range sensors to detect immediate obstacles. But none of them give the robots an insight of what's beyond their line-of-sight and thus their path planning methods are very ineffective and irrespective of other obstacles which cannot be seen or detected with the sensors built-in.

In order to overcome this limitation and thus giving indoor robots a possibility to get their paths planned according to the overall status of their configuration spaces, our approach will make use of cameras mounted on top of the room to get an over-sight of the entire room. With the use of those cameras, the robot will be able to get a map of the entire room with obstacles marked on it. Therefore, it will be able to navigate to its final destination considering the entire configuration space.

## 1.2 OBJECTIVES

The purpose of this project to compensate for the issues faced in a real time indoor navigation system by providing a solution that is irrespective of the environment and the type of robot by making this system to be able to consider the varying factors in the environment such as moving objects like human and even displacements of objects like furniture in the room.

While the system itself would be having its own hardware including camera and a computer for processing, we planned this project to have the deliverables as a complete application used in a real environment. Such navigation system is very much applicable for service robots. Therefore, our project will bind this system over a real application and thus a working model will be demonstrated. A restaurant delivery robot will be built and its navigation will be handled by the aforementioned navigation system of our project to ease its navigation to the delivery table. In addition to automating the delivery process, it is planned to build an order management system to receive orders from the customers and to reserve tables for them. This system will be integrated to the navigation system to get the target destination for the robots.

As a summary, our deliverables are as follows,

i. Customer order management system – For customers to reserve table and place orders,

ii. External vision-based path planning – Planning a path for the robot to navigate to the delivery table by capturing the environment, and

iii. Robot’s hardware and control system – Building the robot and making a control system for the robot for driving it along the generated path by avoiding any sudden obstacles.

# 2. LITERATURE REVIEW

## 2.1 Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments

Many recent studies have focused on the problem of mobile robot self-localization. This paper discusses the concept under two main parts which are image processing technique to detect the robot in real time and the distance measurement system to localize the robot.

### 2.1.1 Image Processing

Image processing technique uses following photography methods in order to identify robot,

1. Camera Calibration,
2. Image Segmentation,
3. Morphological Image Processing, and
4. Connected-Components Labeling.

1. Camera Calibration:

Camera calibration is the process of calculating the extrinsic and intrinsic parameters of a camera. Once we calibrate it the 3-D information can be retrieved from 2-D image. World points are transformed to camera coordinates using the extrinsic parameters. The intrinsic parameters have the polynomial mapping coefficients of projection function. The camera coordinates are mapped into the image plan using the intrinsic parameters.

The transformation from world points to camera point is given by the following equation and the component R and T together will form the extrinsic parameters as .

(1)

where,

: Camera points,

: World points,

*R*: Rotation, and

*T*: Translation.

In order to obtain the accuracy of the robot location, the distortions should be removed. Using camera calibration, we can correct radial and tangential distortions in the images as given below.

|  |  |
| --- | --- |
| Figure : Image before Camera calibration [1] | Figure : Image after Camera calibration [1] |

2. Image Segmentation:

Image segmentation allows to isolate objects by converting the grayscale image into binary image. For this, a threshold value is used in the image’s pixel and each pixel is compared with it.

*if f1(x, y)> t* (2)

*Fbinary (x, y) = 255*

else

*Fbinary (x, y) =0*

*Ffg (x, y) = |fbg (x, y) – fcurrent (x, y) |* (3)

where,

*t*: threshold value, and

*f‑1(x, y)*: The luminance of the pixel.

3. Morphological Image Processing:

It helps to extract image components that are useful in the representation and description of region shape using morphological image processing operations such as dilation and erosion. They deal with discontinuous edges and noise in the images.

4. Connected-Components Labeling:

It is an algorithm which is used to identify the connected-components that share similar pixel intensity. It helps for pattern recognition.

### 2.1.2 Robot Localization

After the procedures of the image processing, the coordinates in the image domain can be received. Consider the robot’s environment with three webcams as shown in Figure 3. To find out the coordinates (*xi*, *yi*), the distance between the wall and webcam *wi*and the distance between wall and robot’s center *di* have to be calculated.

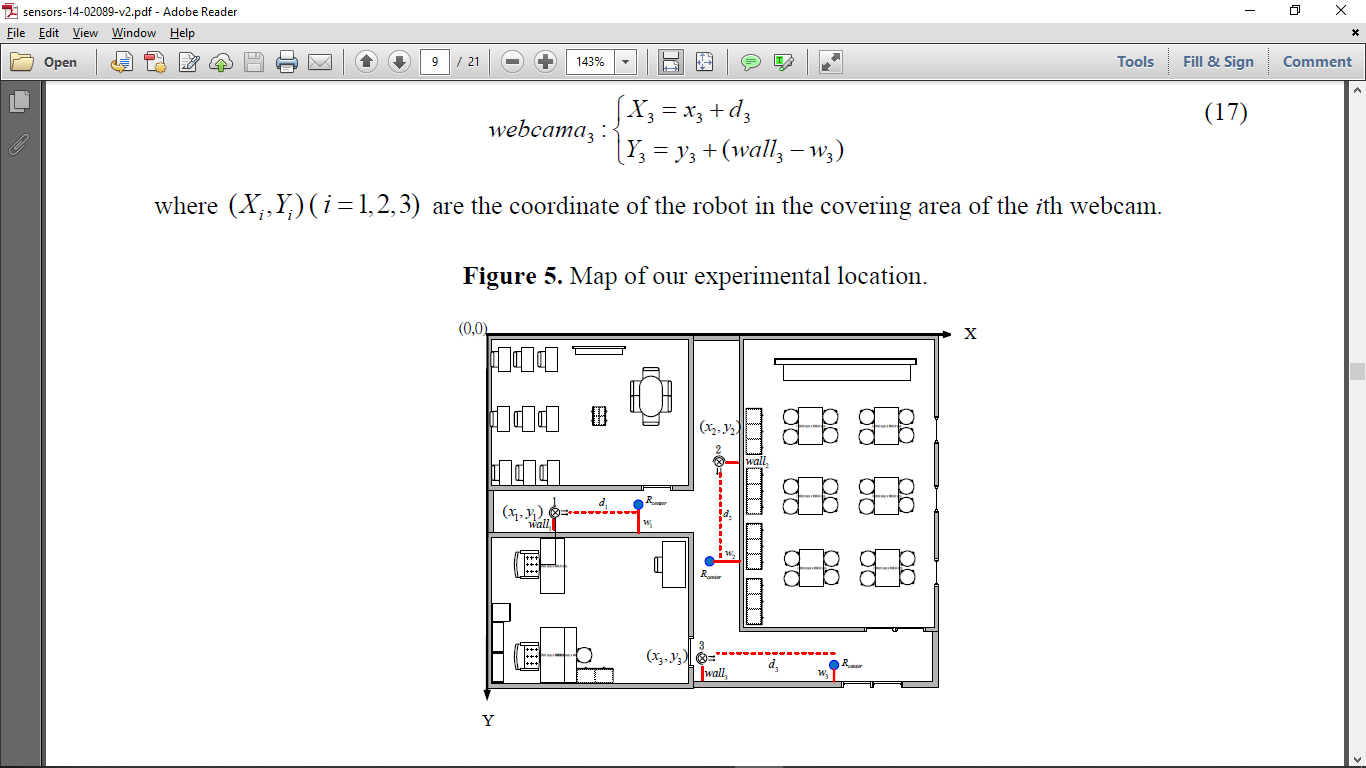


Figure : Map of the experimental location [1]

In order to do this, a single-webcam distance measurement technique for indoor robot localization is proposed in the paper which are Image-Based Distance Measurement System (IBDMS) and Parallel Lines Distance Measurement System (PLDMS). Here, only one webcam is needed for estimating the distance.

1. Calculation of *di* using IBDMS:

The idea behind the IBDMS is derived from the triangular relationship. An image is captured by combining the known dimensional rectangle at first. Then the proportion relationship between real and image dimension of the rectangular can be found. Hence, the distance between wall and webcam (*di*) is calculated as in the Equation (4).

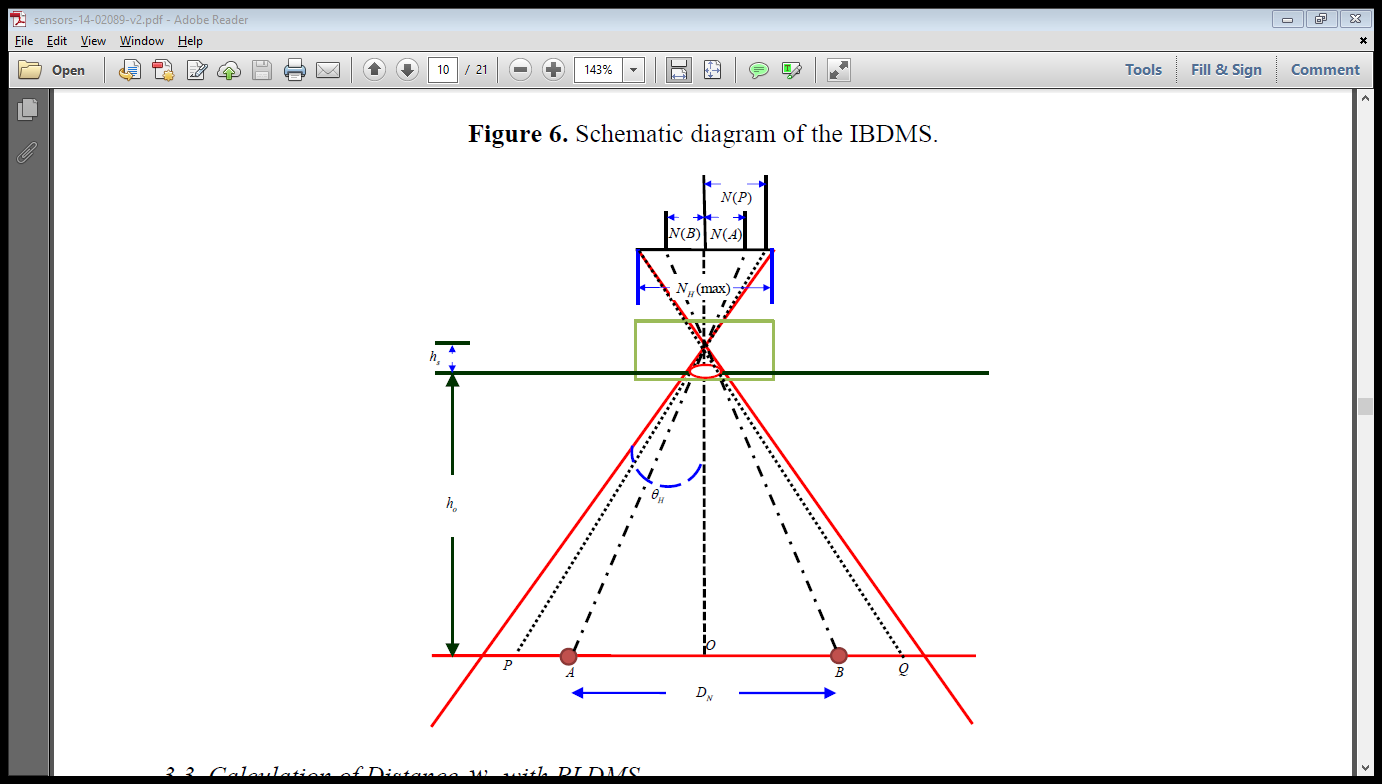


Figure : Schematic diagram of the IBDMS [1]

(4)

where,

*ho*: distance *di* (*i* = 1,2,3),

*hs*: constant parameter of the webcam,

*A, B*: given location points,

*N(A), N(B)*: pixel values in the captured image,

*NH max*: maximum pixel width values in the captured image,

*DN*: width between points *A* and *B*, and

*θH*: horizontal view angle.

2. Calculation of *wi* using PLDMS:

As in the Figure 5, there is a linear relationship between actual distance and image distance which similar to the concept of IBDMS. Therefore, the actual distance from the wall to the robot can be calculated from a simple formula as in Equation (5).

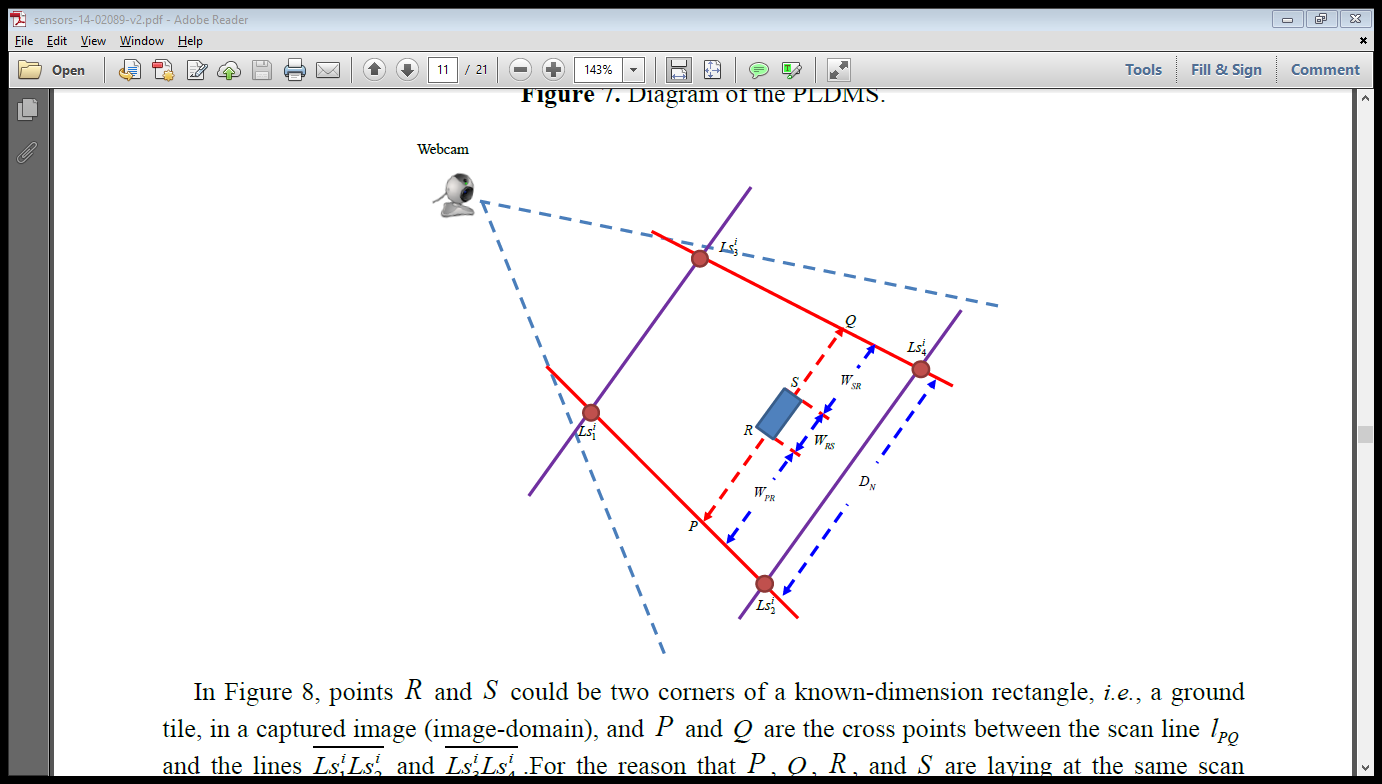


Figure : Schematic diagram of the PLDMS [1]

(5)

where,

*R, S*: corners of a known-dimension rectangle,

*WRS*: width between *R* and *S*,

*DN*: width between points *A* and *B*,

*NH(RS)*: number of pixels between *R* and *S*, and

*NH(PQ)*: number of pixels between *P* and *Q*.

The same concept is also used to calculate the width *WPR* and *WSQ*. Then the distance between wall and robot will be calculated by adding above three distance easily.

With the use of the external cameras, the robot will be able to get a map of the entire room with obstacles marked on it. Therefore, it will be able to navigate to its final destination considering the entire configuration space.

## 2.2 Path planning algorithm for robot navigation

The path planning of mobile robot in an unknown environment is the core technology in mobile robot research, which attracts a large number of researchers all over the world. It refers to find a path from the start point to the goal point in the work environment with unknown obstacles, and assures the safe, non-collisions and avoidance of all obstacles as mobile walking. According to the degree of integrity of the environmental information, robot path planning can be divided into local path planning and global path planning [2]

There are considerable researches on the first field, such as space-based partition of unity method, projection method, configuration space method and so on. In local planning, environment information is unknown, or is only partly known, and robot sensor system can only detect local environmental information it perceived. The most representative study in this field is artificial potential field and probabilistic roadmap. Artificial potential field is simple and innovative, which has been widely used in recent years, but is can easily lead to a partial die point, which makes the robots unable to go to the goal. Probabilistic roadmap method in complicated environments is difficult. [3]

As an in improved and a better consideration for the path planning, Rapidly Exploring Random Trees and its variations are being proposed by many researches.

The RRT approach to path planning introduces a technique of determining a planning path by selecting random points within a known environment and moving towards that point an incremental distance from the nearest node of an expanding tree. The movement from existing traversed points to random points in the environment will make a path that looks like a tree, and will cover most of the free space in the environment. A planned path will be found when a branch in the tree comes close the goal positions. [3]



Figure : Path planning in workspace with many obstacles [3]

## 2.3 Restaurant delivery robot design and control system

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. For any mobile device, the ability to navigate in its environment is important. Avoiding dangerous situations such as collisions and unsafe conditions comes first, but if the robot has a purpose that relates to specific places in the robot environment, it must find those places. This section describes about robot building, Fixed Path Guided Systems and Laser positioning systems which are used to localize and navigate mobile robots.

### 2.3.1 Design of the Robot Mechanism

The design selected for the mobile robot should be able to fulfil the requirements of the task have to be archived. Here the design for a restaurant service robot is discussed. For this design, Differential drive system is used and main body of the service robot is consisted with five layers,

1. First layer (Bottom layer) - batteries and reflective type infrared sensors for obstacle avoidance,
2. Second layer - Sensors signal level converters, Voltage converters, Voltage and current meters and Motor Drivers,
3. Third layer - PC based robot controller, I/O card and USB to RS232 converter.
4. Fourth layer - The dish transmission system, and
5. Fifth layer (Top layer) - Touch screen, Wireless network module, camera, Speaker and Laser positioning system.



Figure : The first generation of the restaurant service robot [4]

### 2.3.2 The Fixed Path Guided Systems

The fixed path guided systems considered as old guided methods regarded as a low-cost guided method for robot navigation. Five infrared sensors have placed on the bottom facing the ground which can sense the location of the path and follows it. Obstacle Avoidance is done by eight reflective infrared sensors placed around the robot.

# 3. PROPOSED METHODOLOGY

For the achievement of the objectives the overall system is being proposed to be as illustrated in the Figure 8 with the system diagram of the overall proposed system. A User interface is at the end-user side which will be used by the customer for ordering food and to reserve their tables. The data will be sent to the main server and from there the management will receive the reserve and prepare the food and mark it ready to deliver once it is ready. Then, the delivery table number will be notified to the navigation system for its use for planning the path for the robot and depending on the logic explained further below the robot will be navigated to the target destination.

Figure : System diagram of the overall proposed system

Each components of the project are further three main parts of the project is further described below.

## 3.1 Customer order management system – For customers to reserve the table and place the orders.

A web-based order and reserve management system will be built with the following tools,

* In the back-end:
  + PHP – Scripting Language
  + MySQL – Database
  + Apache – Server
* In the front-end:
  + HTML – Markup language
  + CSS - Styles
  + JavaScript – Browser level processing

Management staff will have the available foods updated on the database and for security purposes, admin interface will have a login interface that blocks any unauthorized access to the back-end. The customer on their mobile device, will access the restaurant’s portal and reserve their table and order their foods.

When an order is placed, staff gets notified in their portal and once the food is ready, they mark it ready to deliver. Based on reservations and food orders, the database is automatically updated by marking the reserved tables as not-available and reducing the available food count.

As a summary, the following will be available on the system,

* User interface for the order to view available food and tables and thus making reservations,
* Management staffs, updating availability of food and tables and receiving new reservation notification
* Staffs, updating the system when the food is ready to be delivered.

## 3.2 External vision-based path planning – Planning a path for the robot to navigate to the delivery table based on the current objects in the environment

### 3.2.1 Capturing and Processing the Images of the Room

A fish eye lens camera will be mounted on top of the room to cover a wide area. Images captured from the camera will be having distortions throughout the images. In order to get a fixed and a proper image a calibration is needed. A standard method of calibration is to use a known size of chess board to get the camera intrinsic and extrinsic parameters and thus correcting the image distortions. From the calibration process a matrix is generated and is used for further operations.

### 3.2.2 Detection of Objects in the Room and Classifying of Moveable Paths and Obstacles

Once undistorted image of the room is achieved, then the objects in the room need to be identified in order to get a view of moveable path to allow the path planning algorithm to generate its paths in such a way it doesn’t go through any obstacles.

There are two kinds of objects in the room,

1. Fixed objects like furniture and pillars, and
2. Dynamic objects like human.

These objects will be identified and thus an occupancy grid map will be generated. Based on which the path planning will be performed.

### 3.2.3 Generating the Path for the Robot to Its Destination

Once the objects and moveable area in the environment are identified, now the path needs to be generated. For this purpose, the starting and destination nodes need to be defined. From the order management system, the delivery table number will be received and having its coordinates as the destination node and the robots resting point as the starting node, the map will be updated.

Using this map, navigation system will use Extended Rapidly Exploring Random Tree (RRT\*) algorithm for generating a path. This algorithm allows us to work with dynamically varying environment where path generation need to be re-done according to the moving obstacles.

Once this path is generated, it is transmitted to the robot and thus the robot follows the path. A built-in range sensor in the front of the robot will keep looking for any sudden obstacles come to collide on it. To avoid collision, the robot will stop moving if there is an object on its path. Once there is an obstacle on its path and it can’t move forward, the robot will allow a waiting period to check whether the object is moving away from the path. If not, the robot will need a new path and it requests the system for a new path.

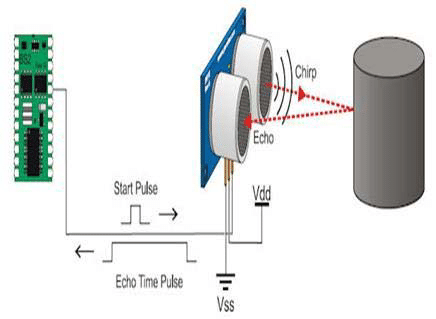


Figure : Ultrasonic sensor detects immediate obstacle in the front [5]

### 3.2.4 Regeneration of Path When a New Obstacle Is Introduced in the Previous Path

A regeneration of path will be done by simply changing the start point to the current position of the robot and generating the path again based on current parameters. And the newly generated path will be transmitted to the robot.

When the request for a path generation is received by the system it goes through all the previous processes of detecting obstacles and generating the binary image of obstacles and moveable area.

## 3.3 Robot’s hardware and control system

Building the robot and making a control system for the robot for driving it along the generated path by avoiding any sudden obstacles.

### 3.3.1 Defining Requirements of the robot

Before designing any robotic system, the first procedure is to identify its requirements. The following are a set of hardware requirements needs to be defined for this robot build.

* The robot should be able to move while carrying food to any table avoiding obstacles.
* The maximum payload robot should be able to carry.
* Speed that robot should be able to travel.
* Ground clearance of the robot.
* Robot height.
* Robot should be of low cost.

### 3.3.2 Hardware Design

The design is chosen such that it should be able to hold electronics components such as microcontrollers, sensors, battery and hold food we need to deliver. Based on the considerations mentioned in section 3.3.1, a model for the robot will be designed and the robot will be built. It will have a tray for carrying food and have spaces for accommodating necessary electronics and control elements.

### 3.3.3 Robot drive mechanism

A mobile robot needs locomotion mechanisms that enable it to move unbounded throughout its environment. One of the cost-effective solutions for mobile robot navigation is differential drive systems. The differential drive robot (Figure 10) consists of two wheels mounted on a common axis controlled by two separate motors and for balance and smooth riding, two caster wheels will be employed in the front and in the back.

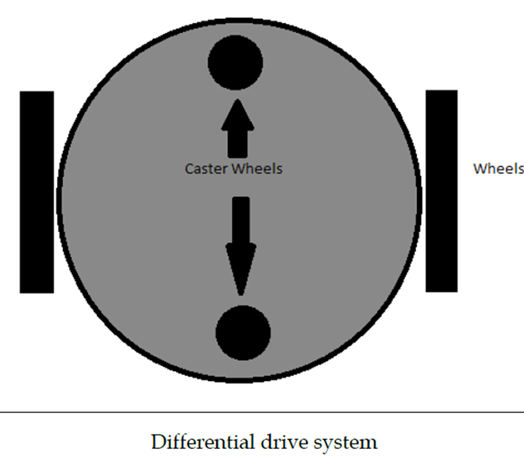


Figure : Proposed differential drive system for the robot

### 3.3.4 Control System

Robot needs to receive the coordinates from the navigation system continuously for the path and to navigate to each point. For this purpose, a wireless communication is needed. Once the coordinates for a point is received the motor speeds will be calculated differentially and the robot will move to the point.

Once it reaches the point, it will send a message to the navigation system and ask for the next coordinate and so on and so forth this will keep continuing until the desired delivery location is achieved.

The flow chart in Figure 11 portrays the logic flow that will power the robot’s control system.



Figure : Flow chart showing the control system for the robot

# 4. PRESENT PROGRESS AND IMPLEMENTATION PLAN

## 4.1 Order management system

The webpage on the handheld device is designed to give user interaction with the user as shown in Figure 12. It has two main interfaces: User Interface and Admin Interface. Login page is integrated to block unauthorized access to the admin interface (Figure 13).

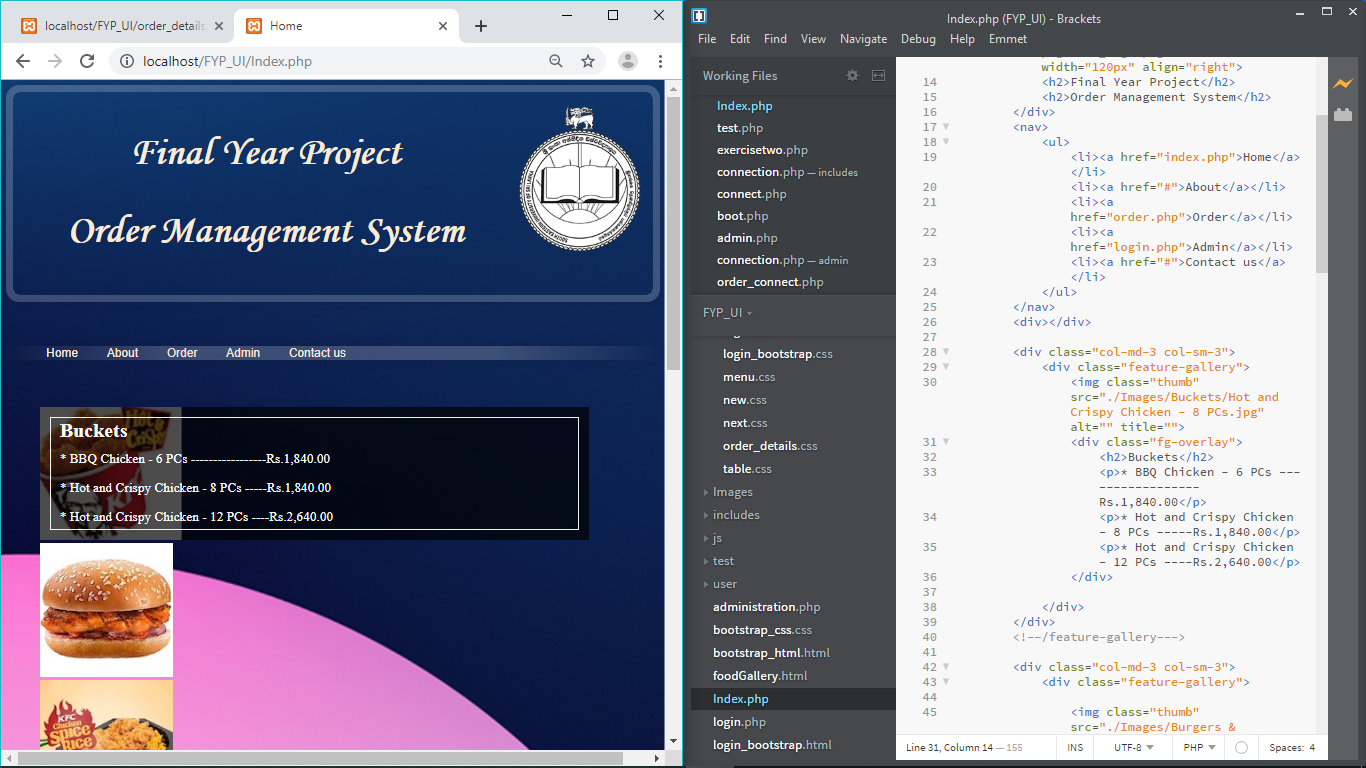


Figure : Home page

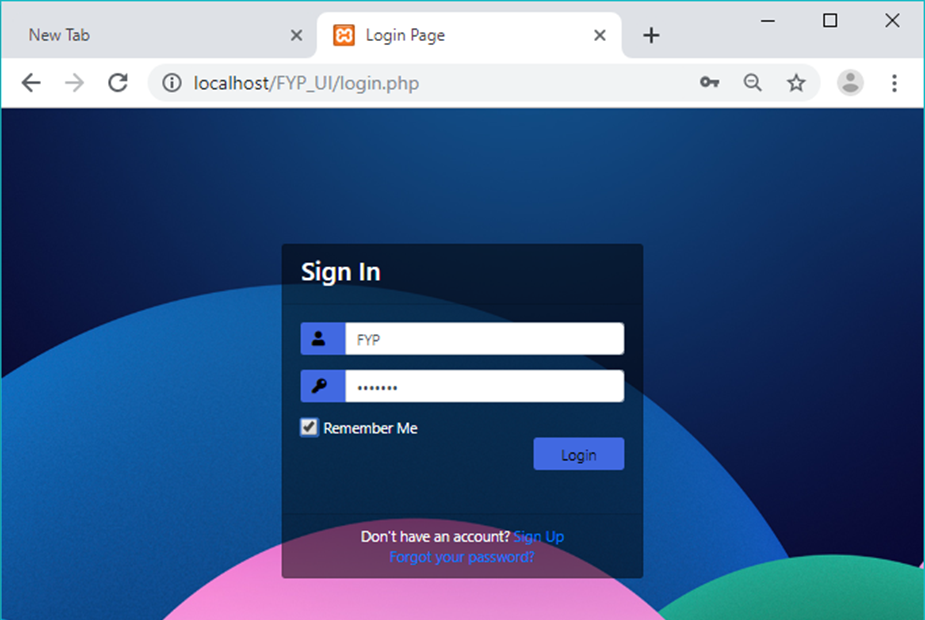


Figure : Login page

User will be asked to select the food items, enter the table number and give other payment details on the web page. Likewise, the admin will update the order details on the web page and food availability on the database. Since the food items that are displayed for ordering is directly taken from the database, customer can order only the available dish in the restaurant.

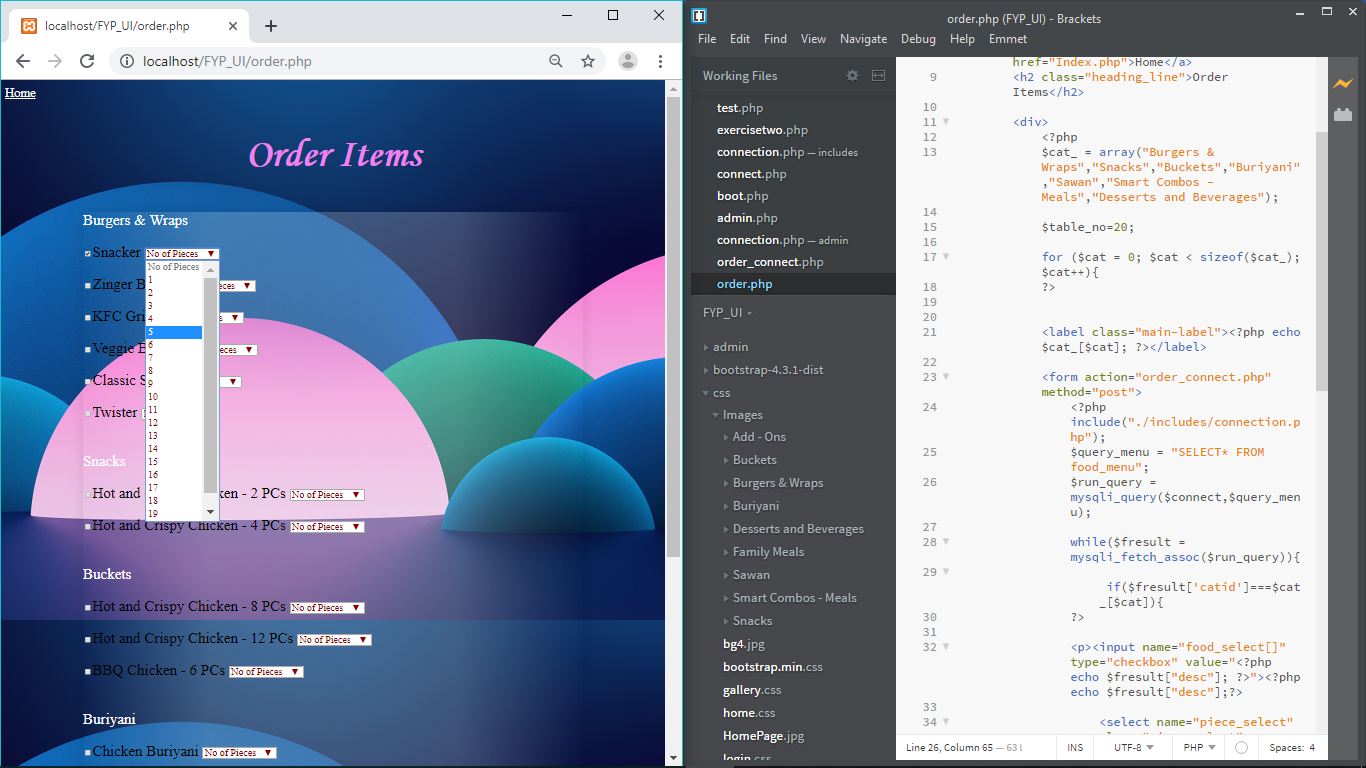


Figure : Dish order on web page

Orders are listed in admin interface like in the Figure 15. Main purpose for having this listed item is to notify the table number to the robot and showing the food list to the chef. In addition, it can be used to inform whether the payment was made or not to the billing system.

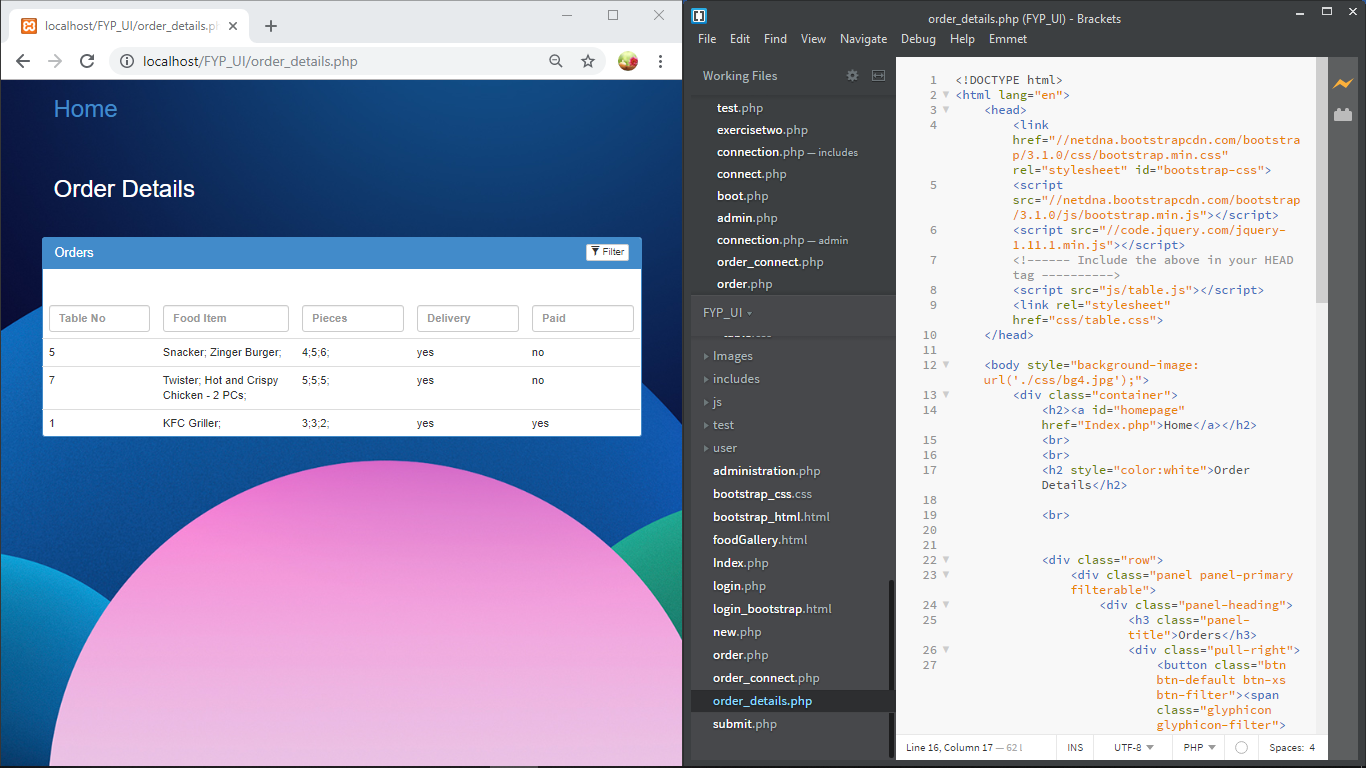


Figure : Orders are listed on the admin interface.

## 4.2 Navigation system

For the centralized navigation system, initial experiments are done for the purpose choosing,

* The right camera for capturing most of the area of interest
* Well working algorithm that suits the environment conditions for object detection and map building
* Path planning algorithm that fits the circumstances where the environment changes and the conditions are not specific.

### 4.2.1 Image capturing and calibration of camera

After experimenting with various types of cameras, a camera with a wide-angle lens was chosen to cover most area in one shot in order to avoid further processing of images from different many cameras.

After mounting the camera on the top center of the faculty cafeteria, images were taken and it was observed to be distorted. For fixing them, a chess board having cells with known dimensions was used. From the images of the chess board of various orientations, matrix was generated which was later applied over all the images taken from the camera to have an undistorted image.

|  |  |
| --- | --- |
| WhatsApp Image 2019-07-26 at 21 | 2 |

Figure : Original bird view image on the left and the calibrated and corrected image on the right

### 4.2.2 Object detection and obstacle-path map building

Firstly, the captured image on which there were no presence of any dynamic objects was run through image segmentation and clustering and then a cost map was generated. Later, based on a threshold it was turned into a binary image marking objects as black and the moveable area as white.

But as this image was having inner white space between parts of objects which may be assumed by our path planning algorithms as a moveable path area, these spaces needed to be blacked out. For this purpose, detected objects were bounded into a contour.

|  |  |
| --- | --- |
| C:\Users\Umair Ahmadh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3.jpg | C:\Users\Umair Ahmadh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2.jpg |

Figure : Image segmentation and clustering. Image on the left shows the identification of objects and moveable path. And the image in the left shows after applying a contour.

And as in a real environment, there will be moving objects. In order to give the robot a path, all objects in the environment at the time of generating the path need to be considered. To do so, the reference background image was used for identifying moving objects on top of it. The same process which was used in the above background reference generation was used here on top of this reference for this purpose.

|  |  |
| --- | --- |
| C:\Users\Umair Ahmadh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.5.jpg | C:\Users\Umair Ahmadh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3.jpg |

Figure : Image with moving objects on the right and the final processed image after applying segmentation and clustering

### 4.2.3 Path planning and obstacle avoidance by re-planning of path

As for experimenting the path planning with the proposed algorithm – RRT\*, the starting and target destination coordinates were manually fed into the algorithm and the paths were generated. For this purpose, the binary map of objects and moveable path that was generated earlier was utilized.

|  |  |
| --- | --- |
|  |  |

Figure : Binary image of objects and the moveable path is on the left, generated path is in the right

As for experimenting the sudden crossover of an objects in front of the robot image with moving objects on the cafeteria was used in the flow. And rerunning the path planning algorithm on it resulted a non-colliding path for the robot.

|  |  |
| --- | --- |
| 8.111 |  |

Figure : Sudden introduction of a new object on the path of the robot is on the left and the regenerated path is on the right

## 4.3 Robot Design and weight analysis

* Motor torque estimation
* Materials and component selection
* Building the robot

### 4.3.1 Designing Robot model

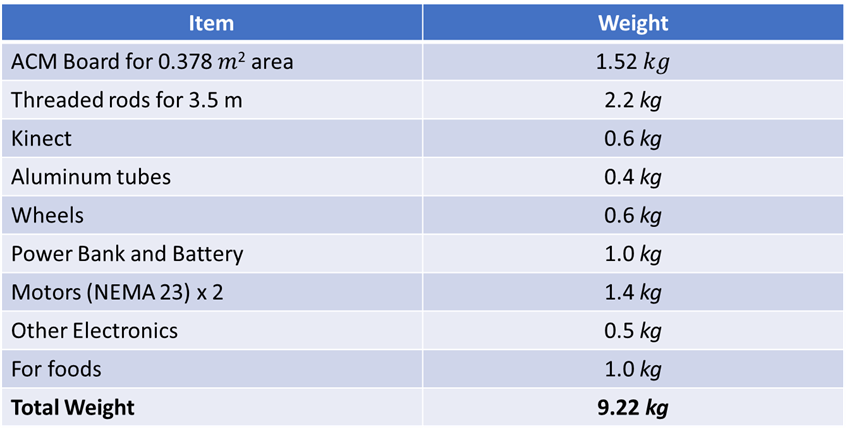
* A 3D model for the Robot design was built considering the requirements mentioned in 3.3.1.
* Materials for the build were chosen.



Figure : 3D model of the proposed robot design

### 4.3.2 Total Weight Estimation

**Table 1: Estimated weight for the components and total weight of the robot design**



### 4.3.3 Calculation of motor torque

In our design we have 4 wheels and 2 motors. For calculation, we assume that the coefficient of friction is 0.6 and radius of wheel is 5 cm. The weight acting on the four wheels can be written as,

(6)

Where,

*N1*: weight acting on each caster wheel.

*N2*: on each motor wheel.

The maximum torque is required when the robot starts moving. It should also overcome friction. To calculate the maximum torque (*T*),

(7)

Where,

*μ* is the coefficient of friction.

*N* is the average weight acting on each wheel.

*r* is the radius of wheels.

*T* is the torque.

Assuming that the weight of the robot is equally distributed on all the four wheels,

(8)

Motor torque required,

(9)

### 4.3.4 Center of Gravity Calculation

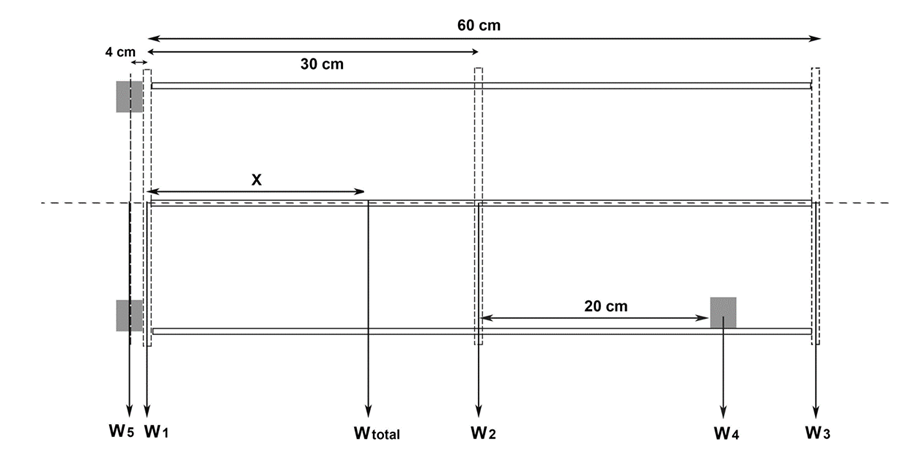


Figure : Single line diagram shows the weight distribution

Considering the moments around center of the bottom plate in clockwise direction,

*W1*= (0.8 kg + 0.95 kg + 1 kg + 0.5 kg) x

= 3.25 kg x

= 31.85 N

*W2* = (0.5 kg + 2.2 kg + 0.4 kg + 0.136 kg + 0.05 kg) x

= 3.286 kg x

= 32.20 N

*W3* = (0.5 kg + 0.2 kg + 1.0 kg) x

= 1.7 kg

= 16.66 N

*W4* = (0.6 kg + 0.15 kg) x

= 0.75 kg x

= 7.35 N

*W5* = (1.4 kg + 0.6 kg) x

= 2.0 kg x

= 19.6 N

*W1* x 0 + *W2* x 0.3 m + *W3* x 0.6 m + *W4* x 0.5 m - *W5* x 0.04m = *Wtotal* x *X* (10)

9.66 Nm + 9.996 Nm + 3.675 Nm – 0.784 Nm = 107.653 N x *X*

= (11)

*X* = 0.209 m

where,

*W1*: Steel Bottom plate with frame, Power bank, Battery and Other Electronics,

*W2*: ACM Board, Treaded rods, Aluminum tubes Jetson Nano and Arduino,

*W3*: ACM Board, Aluminum L profile and food,

*W4*: Kinect and holding plate, and

*W5*: Motors and wheels.

According to the calculation the Center of Gravity of the robot design is 0.209 m above the bottom plate. It is 9.1cm below the center.

### 4.3.5 Hardware Selection

Based on above calculations, the following hardware parts were chosen to be employed in the robot build.

* Drive Motor – NEMA 23 Stepper Motor

Holding torque - 1.0 Nm (Required )

Step Angle - 1.8 0

Current - 1.5A

Figure 23: NEMA 23 Stepper motor and its properties

* Motor Controller – TB6600 Stepper Motor Driver

****

Output Current - up to 4.0A

Micro Step - 1, 2/A, 2/B, 4, 8, 16 & 32

Figure : TB6600 Stepper motor controller and its properties

* Controller board – Arduino Uno

****

Digital I/O Pins - 14

Operating Voltage - 5V

Figure : Arduino UNO development board

* Single Board Computer – Nvidia Jetson Nano

****

Quad-core Arm A57 processor @ 1.43

4GB System Memory

4 x USB 3.0 ports, 1x USB 2.0 Micro-B port for power

Figure : NVIDIA Jetson Nano Development platform

# REFERENCES

|  |  |
| --- | --- |
| [1] | I.-H. &. C. M.-C. &. W. W.-J. &. S. S.-F. &. L. T.-W. Li, "Mobile Robot Self-Localization System Using Single Webcam Distance Measurement Technology in Indoor Environments," p. 21, 2008. |
| [2] | X. Z. Cai, H. g. He and Chen hong, "Some issues for mobile robot navigation under unknown environments [J]," in *Control and Decision. Vol.8*, 2002. |
| [3] | Liu Chang-an, Chang Jin-gang, Li Guo-dong and Liu Chu, "Mobile Robot Path Planning Based on an Improved," *International Conference on Automation and Logistics,* pp. 2375-2379, 2008. |
| [4] | T. Jyh-Hwa and L. Kuo, "The development of the restaurant service mobile robot with a laser positioning system," *IEEE - 27th Chinese Control Conference,* pp. 662-666, 2008. |
| [5] | S. B. Anila S, "INTELLIGENT SYSTEM FOR AUTOMATIC RAILWAY GATE CONTROLLING AND OBSTACLE DETECTION," 25 07 2019. [Online]. Available: https://www.researchgate.net/publication/318960459\_INTELLIGENT\_SYSTEM\_FOR\_AUTOMATIC\_RAILWAY\_GATE\_CONTROLLING\_AND\_OBSTACLE\_DETECTION. |
| [6] | H. M. L. Devin Connell, "Extended rapidly exploring random tree–based dynamic path planning and replanning for mobile robots," *International Journal of Advanced Robotic Systems,* 2018. |
| [7] | A. K. Z. H. Iram Noreen, "A Comparison of RRT, RRT\* and RRT\*-Smart Path Planning Algorithms," *International Journal of Computer Science and Network Security,* 2016. |
| [8] | C. J.-g. L. G.-d. L. C.-y. Liu Chang-an, "Mobile Robot Path Planning Based on an Improved," in *International Conference on Automation and Logistics*, Qingdao, China. |
| [9] | Zbigniew, "Mapping Camera Coordinates to a 2D Floor Plan," 15 May 2019. [Online]. Available: https://zbigatron.com/mapping-camera-coordinates-to-a-2d-floor-plan/. |
| [10] | R. H. A. M. G. I. &. C. A. Abiyev, "Robot pathfinding using vision based obstacle detection," *3rd IEEE International Conference on Cybernetics (CYBCONF) ,* 2017. |
| [11] | L. Joseph, Learning Robotics using Python, Packt Publishing, 2015. |
| [12] | R. Siegwart and I. Nourbakhsh, Introduction to Autonomous Mobile Robots, Cambridge: MIT Press, 2016. |
| [13] | S. K. L. Tzou Jyh-Hwa, "The development of the restaurant service mobile robot with a Laser positioning system.," p. 5, 2008. |

# 