

**DEVELOPMENT OF AUTONOMOUS MOBILE ROBOT FOR PESTICIDE  
APPLICATION AND IRRIGATION**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**

**BIJAY RAI**

**MZU REGN. NO. 1904980**

**PH.D. REGN. NO. MZU/Ph.D./1732 OF 08.08.2019**



**DEPARTMENT OF ELECTRICAL ENGINEERING  
SCHOOL OF ENGINEERING AND TECHNOLOGY**

**FEBRUARY 2023**

**DEVELOPMENT OF AUTONOMOUS MOBILE ROBOT FOR PESTICIDE  
APPLICATION AND IRRIGATION**

BY

**BIJAY RAI**

Department of Electrical Engineering

Name of Supervisor: Dr. Amarendra Matsa (Mizoram University, Mizoram)

Name of Co-Supervisor: Dr. Asim Datta (Tezpur University, Assam)

Submitted

In partial fulfillment of the requirement of the Degree of Doctor of Philosophy in  
Electrical Engineering of Mizoram University, Aizawl



Dr. Amarendra Matsa  
Associate Professor  
Department of Electrical  
Engineering  
Mizoram University  
Mizoram, Aizawl, Tanhril-796004

**Department of Electrical Engineering**

School of Engineering and Technology

**MIZORAM UNIVERSITY**

(A Central University)

Tanhril, Aizawl – 796 004, Mizoram

Phones: +91-389-2330654/642(O)

FAX: +91-389-2330834

Email: mzut224@mzu.edu.in

---

### **CERTIFICATE**

This is to certify that the thesis entitled “Development of autonomous mobile robot for pesticide application and irrigation” submitted by Bijay Rai, Ph.D. Registration No. MZU/Ph.D./1732 OF 08.08.2019, for the degree of Doctor of Philosophy in Electrical Engineering, of Mizoram University: Aizawl, India, embodies the record of original investigations carried out by her under my supervision. He has been duly registered, and the thesis presented is worthy of being considered for the award of Ph.D. degree. This research work has not been submitted for any degree of any other university.

Dr. Aamrendra Matsa

(Supervisor)



**Dr. Asim Datta**  
**Associate Professor**  
Mobile: +91 8131880802  
Email: [asimdatta2012@gmail.com](mailto:asimdatta2012@gmail.com)  
[asim@tezu.ernet.in](mailto:asim@tezu.ernet.in)

**Department of Electrical Engineering**  
**TEZPUR UNIVERSITY**  
(A Central University Established by Parliament Act)  
NAPAAM, TEZPUR-784 028::ASSAM

## CERTIFICATE

This is to certify that the thesis entitled “Development of autonomous mobile robot for pesticide application and irrigation” submitted by Bijay Rai, Ph.D. Registration No. MZU/Ph.D./1732 OF 08.08.2019, for the degree of Doctor of Philosophy in Electrical Engineering, of Mizoram University: Aizawl, India, embodies the record of original investigations carried out by her under my supervision. He has been duly registered, and the thesis presented is worthy of being considered for the award of Ph.D. degree. This research work has not been submitted for any degree of any other university.

Dr. Asim Datta

(Joint Supervisor)

Date: 7<sup>th</sup> February 2023

**DECLARATION**

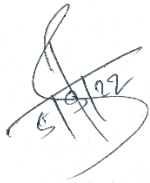
**MIZORAM UNIVERSITY**

**February 2023**

I, Bijay Rai, hereby declare that the subject matter of this thesis is the record of work done by me, that the contents of this thesis did not form basis of the award of any previous degree to me or to the best of my knowledge to anybody else, and that the thesis has not been submitted by me for any research degree in any other University/Institute.

This is being submitted to the Mizoram University for the degree of *Doctor of Philosophy* in Electrical Engineering.

Date: 7<sup>th</sup> February 2023



-----  
(Candidate)



Dr. Prof. Amarendra Matsa, Supervisor

-----  
(Dr. Subir Datta, Head of Department)

## ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to my supervisor, **Dr. Amarendra Matsa**, Associate Professor, Department of Electrical Engineering, School of science, for his support and deep insights. I would also like to thank **Dr. Asim Datta**, Professor, Tezpur University, Assam for his colossal support, endless ideas and high spirits encouraged me during the entire period of the study. I am very grateful to both my guides for introducing me to the recent industry trends in electrical engineering and steering me towards the right direction until this thesis was completed. I would like to express my gratitude to the other faculty members of the department for their positive criticism and valuable suggestions, which directed me into the right direction.

I would like to express my gratitude to Mr. Sajal Kanti Das of Mathematics Department, for his constant help, support and kind co-operation at the time of coursework completion. I would like to thank scholar friends **Mr. Rajesh, Mr. Arup Das, Mr. F. Vanlalhriatpuia (Sheta), Mr. VLMS Dawngliana Sailo (MS)** from Electrical Engineering Department, MZU for always giving me time, suggesting and helping out in and around MZU. I would like to take the opportunity to thank my fellow mates and juniors, who helped me in various ways.

Lastly, I would like to thank **Lord Vasudeva** for giving the strength, time, and support to be able to complete my research work.

Date: 7<sup>th</sup> February, 2023

-----  
(BIJAY RAI)

Department of Electrical Engineering Mizoram University Aizawl, Mizoram

## **DEDICATION**

**Dedicated to my father, my wife and child “Brownie”.**

## CONTENTS

Title of the Thesis	i
Inner Cover	ii
Supervisor Certificate	iii
Declaration	iv
Acknowledgement	v
Contents	vi
List of Figures	vii
List of Tables	viii
Abbreviations	ix

<b>CHAPTER 1: Introduction</b>		<b>Page No</b>
1.1	Introduction	2
1.2	Literature Survey	3
1.3	Problem Statement	11
1.4	Objectives	12
1.5	Phases of Research Work	12
1.6	Thesis Organization	12
<b>CHAPTER 2: Study on Pure Pursuit Algorithm for differential robot and its applicability in agricultural robot</b>		
2.1	Introduction	15
2.2	Theory of Pure Pursuit Algorithm	15
2.3	Mathematical Model for 2-Wheel Differential Mobile Robot	20
2.4	Forward Kinematics of Mobile Robot	21
2.5	Wheel Odometry for Robot Localization	22
2.6	Various Platforms for Ground Mobile Robots	24
2.7	Regulations of Drone Operations	27
2.8	Path Tracking Controllers	29
2.9	PPA applied on Autonomous NISSAN Passenger Car	30
2.10	Comparative Case Study on various Path Tracking Algorithm	31
	2.10.1 Pure Pursuit Controller	32
	2.10.2 The LQR Controller	33
	2.10.3 Model Predictive Controller	34
2.11	Experiment for Comparison of Path Tracking Controllers	37
2.12	Observations and Analysis for Comparison of Path Tracking Controllers	38
2.13	Conclusion	40
<b>CHAPTER 3: Study on LIDAR sensor and Obstacle Avoidance Algorithms for Mobile Agricultural Robot</b>		
3.1	Introduction	43
3.2	Sensor Technologies for Agricultural Robots	43
3.3	LIDAR Sensors for Agricultural Robots	45
3.4	RTK-GPS Technology for Agricultural Robots	50



3.5	Comparative Study on Sensor Technology for Localization of Agricultural Robot	52
3.6	Sensor Fusion for Mobile Robots	57
	3.6.1 Sensor Refinement	58
	3.6.2 Calibration and Time Alignment	58
	3.6.3 Extraction of Features	59
	3.6.4 Object Recognition and Filtering Methods	59
3.7	Extended Kalman Filter	60
3.8	Obstacle Avoidance Algorithms and Techniques	62
	3.8.1 Potential Field Algorithm	62
	3.8.2 Virtual Force Field Algorithm	62
	3.8.3 Vector Field Histogram (VFH) Algorithm	64
	3.8.4 Bubble Band Technique	69
	3.8.5 BUG algorithm	69
3.9	Localization and Mapping Strategies of Agricultural Robots	72
	3.9.1 Mono SLAM	73
	3.9.2 Binocular SLAM	74
	3.9.3 LIDAR SLAM	76
3.10	Experiment for Comparison of Obstacle Avoidance Algorithms	77
3.11	Observations and Analysis for Comparison of different Obstacle Avoidance Algorithms	80
3.12	Conclusion	82
<b>CHAPTER 4: Hybrid Algorithm using PPA-VFH-Bug Algorithms for Autonomous Navigation of Agricultural Robot</b>		
4.1	Introduction	85
4.2	Unique Challenges of Obstacle Avoidance Algorithms for Agricultural Robots	85
4.3	PPA-VFH Hybrid Algorithm	86
4.4	Methodology for PPA-VFH Hybrid Algorithm	85
4.5	Results and Observations for PPA-VFH Algorithm	92
4.6	Specific Observations	94
4.7	Path Planning Algorithms for Circumnavigation of Obstacles	94
4.8	Efficiency Issues of Autonomous Robot and Precision Farming	97
4.9	PPA-VFH-BUG Algorithm	99
4.10	Methodology for PPA-VFH-BUG Algorithm	99
4.12	Observations and Results for PPA-VFH-BUG Algorithm	103
4.13	Conclusion for Hybrid Algorithm	106
<b>CHAPTER 5: Conclusion and Future Scope</b>		
5.1	Conclusion	109
5.2	Future Scope	110
	Appendices	111
	References	120
	Biodata of candidate	128

	List of Publications	129
	Particulars of the candidate	135

## LIST OF FIGURES

Figure No.	Figure Captions	Page No.
Figure 1.1	SAE Levels of Autonomy(Level:0-2)[20]	9
Figure 1.2	SAE Levels of Autonomy(Level:3-5)[20]	10
Figure 2.1	Geometrical representation of PPA algorithm	16
Figure 2.2	Flowchart for PPA algorithm	17
Figure 2.3	Trajectory with large lookahead distance	19
Figure 2.4	Trajectory with small lookahead distance	19
Figure 2.5	Differential Drive kinematics	20
Figure 2.6	Forward kinematics of the differential robot.	22
Figure 2.7	2-Wheel drive different robot prototype (SWAGBOT)	22
Figure 2.8	Wheel odometer sensor	23
Figure 2.9	Wheeled robot	25
Figure 2.10	Crawler robot	26
Figure 2.11	Legged robot	26
Figure 2.12	Stages of navigation process	29
Figure 2.13	Geometrical depiction for 2-wheel robot	32
Figure 2.14	Result of tracking in 8-shaped path.	35
Figure 2.15	Result of tracking in circle shaped path	36
Figure 2.16	Result of tracking in circle shaped path	36
Figure 2.17	Robot trajectory for tracking analysis performed in MATLAB simulation	37
Figure 2.18	Block diagram for LQR/MPC controller	37
Figure 2.19	Performance chart at different velocities of robot	39
Figure 2.20	Performance chart at different lookahead distance	39
Figure 3.1(a)	LIDAR Sensor 3D LiDAR sensors MRS1000	45
Figure 3.1(b)	Sparkfun ZF9P RTK-GPS	45
Figure 3.1(c)	SR04 Ultrasonic sensor	45
Figure 3.1(d)	Frequency Modulated QT 50 RT Radar Sensor	45
Figure 3.2(a)	LIDAR Scanner YD X2 placed with objects around the environment.	46
Figure 3.2(b)	Scanned data and point cloud plotted in MATLAB software	46
Figure 3.3(a)	Mobile robot in field with LIDAR scanner,	46
Figure 3.3(b)	Obstacle perception of robot	48
Figure 3.4	Trajectory comparison of RTCM	51
Figure 3.5	RTCM propagation system	52
Figure 3.6	Block diagram of sensor fusion stages	58

Figure 3.7	EKF sensor fusion model for autonomous vehicle	61
Figure 3.8	EKF probability distribution	61
Figure 3.9	Robot in obstacle course	66
Figure 3.10(a)	Polar histogram in polar form	67
Figure 3.10(b)	Polar histogram (filtered) represented w.r.t robot orientation	67
Figure 3.11	Flowchart for VFH Algorithm	69
Figure 3.12	Bubble band strategy	70
Figure 3.13(a)	BUG1 path strategy	71
Figure 3.13(b)	BUG2 path strategy	71
Figure 3.14	Evolution of the probability density varying depth signified by object sets [102].	74
Figure 3.15	Stereo Visual SLAM processing pipeline	75
Figure 3.16	Block scheme for lidar Architecture	78
Figure 3.17(a)	Field obstacle with start-end points	80
Figure 3.17(b)	Plot of search algorithms	80
Figure 3.18	Performance indicators for Dijkstra vs RRT vs VFH at 900m <sup>2</sup> area.	81
Figure 3.19	Performance indicators for Dijkstra vs RRT vs VFH at 3600m <sup>2</sup> area.	81
Figure 4.1	Real field image scenario.	89
Figure 4.2	Field image with grid occupancy map	89
Figure 4.3	Flowchart of overall operation	90
Figure 4.4.	Performance plot under varying velocity of robot	91
Figure 4.5	Performance plot under varying lookahead distance	91
Figure 4.6(a)	Shortest reach strategy	98
Figure 4.6(b)	Bug strategy	98
Figure 4.7(a)	U shaped trap local minima	99
Figure 4.7(b)	Avoiding local minima	99
Figure 4.8	PPA-VFH-BUG Algorithm	100
Figure 4.9	Robot control strategy with large obstacles.	102
Figure 4.10	Illustration of MATLAB Simulation	102
Figure 4.11	Graph plot for results of circumnavigation against obstacles	103
Figure 4.12	Pilot scheme diagram for PPA-VFH-BUG system	106

## LIST OF TABLES

<b>Table No.</b>	<b>Table Captions</b>	<b>Page No.</b>
Table 2.1	Comparison of characteristics of various mobile robots	25
Table 2.2	Prototype models of several mobile robots	27
Table 2.3	Feature comparison of different tracking controller	30
Table 2.4	Deviations caused to Nissan Leaf prototype (Széchenyi Istv'an University)	31
Table 2.5	Error measure of different techniques.	35
Table 2.6	Performance chart at different velocities of robot	38
Table 3.1	Data format for YD X2 sensor	47
Table 3.2	Data byte description for YD X2 sensor	47
Table 3.3	Comparison of RTK-GPS sensor with other localizing sensors.	54
Table 3.4	Characteristics of obstacle sensor	55
Table 3.5	Suitability and Application of obstacle sensor	56
Table 3.6	Comparison of various avoidance algorithm	72
Table 3.7	Speed comparison of CPUs	79
Table 3.8	Performance comparison for different obstacle avoidance algorithms	80
Table 4.1	Performance data with lookahead distance	92
Table 4.2	Performance data with minimum turning radius.	94
Table 4.3	Comparison of different search algorithms	97
Table 4.4	Characteristics of robot control with different $Ang_{Thres}$ values.	104
Table 4.5	Performance readings of robot control with different LIDAR Range(cm) values.	104
Table 4.6	Sample of real time data reading taken in MATLAB program simulation	105

## ABBREVIATIONS

LIDAR	Light detection and ranging
VFH	Vector field histogram
VFF	Vector Field Force
PPA	Pure Pursuit algorithm
PRM	Probabilistic road map
OA	Obstacle avoidance
GPS	Global Positioning System
GNSS	Global navigation satellite system
RTK-GPS	Real-time kinematic Global Positioning System
DGPS	Differential Global Positioning System
SLAM	Simultaneous localization and mapping
VSLAM	Visual Simultaneous localization and mapping
EKF	Extended Kalman Filter
WMRs	Wheel mobile robots
MPC	Model predictive controller
LQR	Linear Quadratic Regulator
DFS	Depth first searching algorithm
MAV	Micro Aerial Vehicle
BFS	Breadth first searching algorithm
RRT	Rapidly exploring random tree
2D	2-Dimension
3D	3-Dimension
MIPS	Million instruction per second
CPU	Central processing unit
IOT	Internet of things

**Abstract:**

The deployment of autonomous robots has increased in the agricultural industry to assist human labour operations and increase production yields. Autonomous robot can work in difficult conditions of weather for very long hours. Autonomous mobile robot can be an ideal vehicle for transportation and delivery of water and spraying of pesticides as it is one of the most labour-intensive work at agricultural field. Most autonomous robot has been studied with the similar approach to autonomous passenger car like Tesla, Waymo. However, this approach is creating problem is acceptance of robots by famers. Involvement of AI, machine learning and cloud computing are simply too expensive and sophisticated technology for farmers to operate. Such systems require rigorous training of machine and data collection in suitable environment which is mostly difficult to be executed in agricultural land.

Drone technologies have been popular but has security, safety issues and well as inefficiency of delivery. The drones used are planes or multiple rotor helicopters. Delivery from air to ground is done at high rate without capability to spray at precise location of plants. Wind turbulence added by flying machine decrease the precision of operations. Drones are facing new issues of security as they can harm safety of people and infiltrate security of sensitive places, for example many north-eastern states have very less area permissible to fly drones due to presence of close international borders.

Wheeled robots can deliver at very precise locations of plants at slower rate, they are also energy efficient machines compared to drones. 2-wheel differential drive robot is a good choice for agricultural purpose due to its maneuverability and simpler mathematical kinetic model. Farm mobile robots are slow operating machines, hence their kinematic model is enough for implementation of prototypes. The robot can orient and move to any positions based on control of its two drive wheels, supportive dummy castor wheels are necessary to running 2-wheel differential drive stable. Such type of drive is simpler and cheaper to design in hardware compared to other legged and crawler robots.

In this research study an attempt to design an autonomous navigation system for a 2-wheel differential drive agricultural robot is carried out to design a system based on

fast and simple geometrical algorithms using high advanced LIDAR sensors. The tracking algorithms popularly used are PPA, LQR and MPC controllers. These algorithms are compared to each other to find their suitability to my study. MPC and LQR are possible solutions for fast accurate tracking control but are only suited for ideally modelled conditions. Disturbances caused by nonlinear factors while driving the robot (like slippage or wheel lock) could make the system unpredictable and unstable. Model based algorithms also take considerable time for its operations challenging to be used in 8-16 bit micro controllers. PPA however has been favored as more suitable method due to its simplicity of computation, however it does need proper tuning of its lookahead distance based on size of robot and the farm rows.

There are various kinds of sensors and methods discussed for localization and mapping. LIDAR has been popularly used for simultaneous localization and mapping, the method is interesting however is very heavy in computation, It requires continuous 2D or 3D modelling of working space increasing load of computation and amount of memory required. To remove errors caused by faulty sensors and non-linearities of data sensed, such method also required data from several many sensors and fusion of those multiple data using Extended Kalman filter. All such intelligence added to the system increases the overall operations to the robot computer. To design a simple navigation system workable in a structured farm field a well-tuned obstacle avoidance algorithm should be sufficient.

For the operation of obstacle avoidance very popular algorithms like RRT, Dijkstra's and AVFH algorithms have been compared. Dijkstra and RRT are branching search algorithm where the path to avoid the obstacle is based on the knowledge of map and its obstacle. A global perception of the map needs to be continuously updated to such algorithms for overcoming obstacles. Based on the knowledge and map available, these algorithms will compute a path to make a new path to avoid the obstacle and reach the next waypoints in the fastest shortest route. Planning is a crucial step in such algorithms where it takes considerable time based on the size of the area handles and type of parameters set to these algorithms for tuning sensitivity. Sensitive search algorithms in a large map take considerable time before being able to mitigate the obstacle. The constant demand for knowledge of the map of surrounding also adds to

computational load. VFH on the other hand works like a human driver where his knowledge of the object face become more important than knowing the whole map. VFH is inspired for potential force strategy where the obstacle in proximity exerts repulsive certainty force. The VFH algorithm converts the cartesian map of surrounding and obstacles to a polar form having sectors. It then helps robot find the closest angle or sector to given target angle, that which is having least certainty value. The robot can take the angle to avoid obstacle. Therefore, VFH has been proposed to be used with PPA for critical decision for obstacle handling.

The pesticide spraying and irrigation requires peculiar attention even after obstacle avoidance. Obstacle are measured and understood only from the front face by a robot travelling towards it. Once obstacle detected, apart from steering away from it going to next waypoint, the proper application of water and pesticides on plants at the rear side of obstacle is very important. For efficiency in application of pesticides and water, a strategy to circumnavigate the obstacle is carried out using BUG algorithm. This strategy overcomes the problem of neglected spraying operation on plants at rear areas of obstacles. This negligence happens in most search plan algorithm which try to reach next waypoint using the shortest route possible.

Using a combined low computation hybrid PPA-VFH-BUG algorithm, the simulation was successfully carried in MATLAB program environment with acceptable results of performance. Simulation was carried out for low-speed mobile robots with travel speeds up to 2.5 km/hr.



## **CHAPTER-1**

### **Introduction**

## 1.1 Introduction

Despite being an agricultural giant, India's agricultural water management is inefficient. 62% of the water comes from precipitation, whereas 37% comes from irrigation [1]. In rural areas, around 85 percent of water is often lost. Younger generations are migrating away from agriculture and toward urban occupations, resulting in labour shortages for farmers. Consequently, autonomous navigation robots will play a crucial role in agriculture. With the proliferation of mobile robot applications, researchers have concentrated on robot navigation research. Obstacle avoidance increases the challenge of mobile robot navigation. Numerous navigation systems based on remote controls, radio frequency identification (RFID) systems, and wireless network devices have been created for agriculture; nevertheless, they are continuously improved. Simple operations, such as the application of pesticides and irrigation, need straightforward navigation. India's harsh climate has worsened the challenges of agricultural labour; extreme noon heat waves make it impossible for farmers to work in the field. Farmers adopt technology in a methodical, incremental approach. Advanced technologies based on machine learning and data science will be prohibitively expensive and challenging for farmers to run. The capacity of Internet connections in rural locations is inadequate for high-speed applications. Various kinds of sensor like optical, laser, odometry and electromagnetic types of sensors are used in autonomous robots. A proper study on such sensors and their capability needs to be investigated for agro-robots. This research focuses on the best low cost computation algorithm for tracking, obstacle avoidance and strategies to improve precision farming. Some of the control system techniques used include nonlinear proportional-integral-derivative (PID), fuzzy logic system, adaptive, machine learning, and model predictive control (MPC). Map-making, localization, simultaneous localization and mapping (SLAM) controllers, obstacle avoidance, row followers of plants, and route planning algorithms are all necessary for autonomous navigation in an free environment [1][2]. These strategies have their merits as well as significant demerits discussed in this study. This study is designed for the robot to move out of such pitfalls. This work focuses on examining an adaptive VFH algorithm to solve issues of obstacle avoidance. PPA algorithm The VFH and PPA hybrid systems based on a geometric

algorithm and utilizing modern LIDAR sensors has been tried to be sufficient technology for developing farmers like in India. Due to problem of oddly shaped obstacles that can create gaps in application of pesticides and water, the possibility of BUG algorithm is tested to improve the efficiency on application of pesticides. It can help in proper circumnavigation of obstacles when they are detected.

## **1.2 Literature survey**

Certain heuristic approaches for identifying obstructions and regulating direction have been implemented using intelligent techniques such as fuzzy logic control technique [3], neural network technique, and others. Due of imprecise results, fuzzy logic control methods are not always applicable. Regular rule modifications and intensive testing and validation are necessary to enhance precision. For a suitable and optimal output, the neural network approach requires a vast quantity of training and data. Controlling heuristic algorithms also requires the assistance of a trained operator [4]. A single-camera vision and ultrasonic sensor-based interior navigation system for mobile robots is described. Some agricultural robots employ computer vision in combination with other sensors to improve GPS data for autonomous navigation, but these methods are vulnerable to ambient illumination, which is a significant disadvantage in an outdoor setting. A gaze-controlled architecture features a fuzzy-integrated and condition-clarified preference for robot navigation, allowing the fuzzy degree sets to be dynamically activated when the state next to the robot changes [5]. A quadrupole potential field (QPF) technique is offered [6] to prevent collisions in the planned trajectory

Farmers are gradually relying on technology to deal with a number of important problems that the agriculture industry is now grappling with, such as the increasing global food scarcity and the declining agricultural labor force. Agricultural robots mechanize labor-intensive, dull, and slow-moving farming tasks so that farmers may focus on improving crop yields, boosting farm productivity, and reducing labour costs in general [7]. Agricultural robots provide precision agriculture, which more efficiently distributes resources, and reduces resource use significantly. Currently, weed elimination, agriculture harvesting, and picking are all done using advanced

robotic equipment. Numerous vehicle mobility systems, mostly for agricultural use, have also been developed, including center-articulated drives, four-wheel drive, legged robots, omnidirectional drive robot, and chain track wheels. When developing the mobility, different soil and topographical conditions, open-air or greenhouse cultivation, as well as the requirements for maneuverability, are all taken into consideration. Sensor with machine vision methods as image segmentation, sensor fusion, hough transformation, machine learning is used to produce perception in order to collect environmental characteristics. Models that can conduct perception at a reduced processing cost must be used by small mobile robots [8]. The mobile robot senses its environment and moves on to the next waypoint.

Due to a probable lack of farmers in near future, research into the creation of autonomous mobile robots for agricultural application is crucial. The work that robots do on farms is repetitive and tedious. They might involve planting, weeding, and harvesting. The usage of mobile robots in agricultural settings may lead to cheaper manufacture costs and less manual labor. However, because agricultural landscapes are so vast and largely unstructured (unknown), robot navigation presents challenges including how to change tracks and targets that are challenging to find and approach. Crop production has a tremendous economic impact on South Asian country and India, where it generates millions rupees of annual revenue. A promising substitute for traditional farming operations like pesticides and fertilizer application is an autonomous navigation robot that can maneuver through intra-row gaps [9]. Only a small number of researchers have, to the authors' knowledge, published their work on agricultural autonomous mobile robot navigation. A mobile robot with 2 wheels that can steer was modelled and commanded to move around farmland in. In order to achieve both the robot's lateral deviation from the target course and the direction (orientation) of the vehicle with regard to a reference position. [10] also explored the outside navigation of a wheeled mobile robot; the procedure involved four steps: map construction, laser sensor scanning and data processing, vehicle robot localization, and robot movement control. One of the hardest issues in guiding mobile robots in an uncharted and unstructured area is managing the uncertainty and changing situations. The artificial potential field, the edge detection, the obstacle boundary follower

[11][12][13], Goal-oriented recursive track planning [14], vector field histogram (VFH) [6], the dynamic window method [15], and the fuzzy logic as well as artificial intelligence based reactive techniques [16] are just a few of the approaches that have been proposed for robot navigation. The model-based approaches create a path for mobile robot to track toward the destination point by using a map or model of the surroundings. [17] proposed an online method for defining an obstruction-free path for an outdoor mobile manipulator. A map of the terrain was created using aerial image views, and a path clear of obstacles was set forth for the robot. The combination of Pure pursuit algorithm with set waypoint coordinates which were to be reached consecutively until last point is reached. However, the PPA lookahead distance and speed setting for a real scenario field setting would be challenging for efficiently moving the mobile robot. This is due to the fact that it is frequently challenging or even impossible to create a precise model of a dynamic external environment. Different parameter settings are required for different type of farming field and robot design. Sensor-based techniques [18] create control orders for the movements of the mobile robot using data from various sensors as sonar, a laser rangefinder, or a visual camera. The fundamental benefit of sensor-based methods is that a robot can move safely through an unpredictable environment by responding to potential hazards. The potential field approach has issues with oscillations in tight routes, oscillations in the presence of several barriers, oscillations in local minima, accidental stopping between tightly packed obstacles, and local minima [11].

WMRs are the most often utilized robot among the several highly developed, mature robots. The body, the wheels, the wheels supporting drive system, and the wheel driving method make up the majority of the standard WMRs chassis [19]. The chassis may be classified as two-wheel, three-wheel, and four-wheel constructions depending on how many wheels are present. Commonly, these structures are employed. One of the most popular types of constructions is the four-wheel chassis. The mobile platform of mobile robots in the field of agricultural engineering uses a vehicle frame with such a structural configuration. Sometimes it is important to take into account installing a buffer suspension device [19] on the site, the one of the standard Agricultural WMRs frame, in order to provide a steady driving ability owing to the complexity of the

terrain structure. The driving mode of nearly all four-wheeled chassis uses 2 wheel differential driving and front-wheel synchronous steering to ensure the stability of the motion plane and streamline the operating process. Two wheeled differential steering has the benefits of being easy, affordable, and quicker at avoiding obstacles. Additionally, three-wheeled omnidirectional driving is another typical driving technique in the creation of WMRs. Nevertheless, the body frame must be kinematically represented during the navigation process regardless of the type of control mode the chassis is in it. Mobile robots ability to navigate accurately is largely dependent on their workplace environment. Scientists have partially resolved the issues pertaining to robot navigation in recent decades through a significant amount of scientific study, analysis, and demonstration, whether in the outside world or the inside environment. Global Navigation Satellite System, inertial navigation, laser , electromagnetic navigation ,beacon navigation radio navigation , visual navigation [20] and other techniques are now used to navigate robots. In the meanwhile, techniques for fusing different navigation technologies are also widely employed. To decrease the **localization** error and increase navigational precision, these methods make use of the complementing concept of multiple kinds of sensors. The GNSS has been used extensively outside and is backed by established technologies, the most popular that is the global positioning system [21]. At the moment, GPS is the entry level is civilian. The current range accuracy is between 2.93 m and 29.3 m, thanks to improvements in atomic clock precision. But in most situations, its precision falls short of the requirements. Some enhanced strategies have been carried out in order to achieve localising precision that may be measured in centimeters or even more. Additionally, there may be significant localization mistakes due to the surroundings, which includes bad weather, blockage from barriers, etc. For WMR to successfully complete all the many jobs of the correct navigation operation in the challenging environment, it must rely on every component of the robots, and its navigation control structure is displayed. Precise location determination will be more challenging due to irrevocable variables such more obstructions and quicker signal strength attenuation. The four components of the WMRs' navigation structure are path localization, planning, mapping, and obstacle avoidance control. The robots must first assess their position and posture in relation to the map's characteristics after obtaining the

navigational challenge. If the previous map is recognized, the robots utilize the sensors to sense their surroundings. After analyzing and processing the data from the sensors, the vehicles extract points, lines, and other attributes in order to establish their location and posture. If it is unknown, the robots must first estimate the posture by extracting, processing, and integrating environmental data from the sensors in order to construct a local map and update the global map in real time. The robots must obtain their location on the global map in real time, collect the characteristics of the environment in accordance with the map, and then correlate the maps. At this point, the robots will integrate the odometry and other sensors to estimate their posture. The robots must determine their navigation route after finishing the map building. Three issues must be resolved in route planning, including determining the beginning point and the goal position, and obstacle handling. The beginning location and the target position must be differentiated in practical applications. In terms of route planning, the beginning location and the objective location must be obtained via both the global and local path plans before proceeding. The accuracy of the map construction is a requirement for finishing the routing because both the initial position  $(x_0, y_0, z_0)$  and goal location  $(x, y, z)$  of the native path planning both depend on the existing map environment. The local path definition approach and the global path planning are nearly equivalent in complexity. The accuracy of the global planning may be ensured as long as the local path planning is completed. [22] Obstacle avoidance is yet another issue that has to be resolved in path design. To complete the obstacle avoidance control in this situation, sensors like ultrasonic sensors are needed to identify the obstructions. The posture is then assessed and changed in real-time to lead the robots around the barriers. Robots must always perform localization during the navigation process, and it is a crucial component of path planning. Relative localization and absolute localization [23] are the two main processes in the localization of the WMRs. The wheels will slip during moving if the relative localization is employed alone. The measuring system based on external distance, such as GPS, is referred to as absolute localizing. In real-world applications, the WMRs first estimate their locations and postures using the odometer to acquire their relative position coordinates  $(X, Y, Z)$ , and then they use an external measurement device to get their global positions  $(X, Y, Z)$ . The localization of the robots is crucial, whether they are working indoors, or outside, and accurate

localization is a must for the timely completion of all tasks. Real-time motion control, which primarily focuses on regulating wheel rotation movement, is another need for WMRs. The wheels can come into touch with flat surfaces and uneven terrain in different locations, and unstable slippage might happen. This can lead to mistakes building up and erroneous navigation. Special considerations must be made when using robots in agricultural settings in addition to the mentioned job routines. The first issue is the selection and calibration of the sensors. The agricultural environment is not regular like the structured of the urban environment. As a result, there are stricter standards for the precision of robot sensors. In order to calibrate the sensors and ensure that reliable environmental data can be recorded in implementations, it is necessary for the sensors to have a suitably greater sensitivity and reduced latency. It can only offer security for further navigation inside this way. How to achieve obstacle avoidance is the next phase. Obstacle avoidance is a more challenging challenge to solve in an agricultural setting than it is in an urban one because crops have varied morphologies at different growth stages. This behaviour is crucial for a variety of navigation systems, including autonomous vehicles, unmanned mobile robots, both individual and for collaborative activities [24][25][26], and for automated road safety. Many autonomous vehicle systems achieve navigation by fusing a local obstacle avoidance module with a global path planner module. The obstacle avoidance method chooses an acceptable direction of motion based on recent sensor data, whereas the global path planner chooses a good path based on the map of the given field-environment. Local obstacle avoidance is carried out to guarantee that real-time limitations are met. For the robot to move safely at any kind of obstacle. The obstacle can never be a part of the map if it can dynamic and random. Accurate presentation of map is costly process that needs surveying and sensor data collection, this cost can be difficult achievement for the farmers. The obstacle avoidance algorithm can be sufficient provided sensors data is collected and processed fast at a low-cost hardware.

Robots not only need to avoid the relatively permanent navigational obstacles, but they also need to stay clear of the fields. Thirdly, safety concerns are also crucial. The security of the crops and mobile robots in an agricultural setting requires ongoing care. The operator must make sure the robots can be controlled while travelling. In order to



protect the robots from harming the crop if they go uncontrolled, they should be able to quickly halt all of their operations and switch to hibernation mode. Safety concerns must come first, especially when using huge robots outside. There is a scarcity of publicly available studies on the regulation of mobile robot technology since commercial farms have little familiarity with autonomous equipment, and several autonomous agricultural technologies are still in the model or beta test drive stages and are thus secret. However, the usage of autonomous equipment is expanding in other industries (e.g., driverless automobiles, robotic milking, autonomous mining equipment), which might teach us some lessons on how to regulate intelligent farming equipment. In order to be clear about the comparison's emphasis, GPS-guided devices used in commercial agriculture should be distinguished as autonomous crop equipment. Modern GPS guidance systems may be seen as a step toward autonomy, although most current autosteer technology still requires a human driver. Different levels of autonomy has been explained in Figure 1.1 and 1.2 [27].

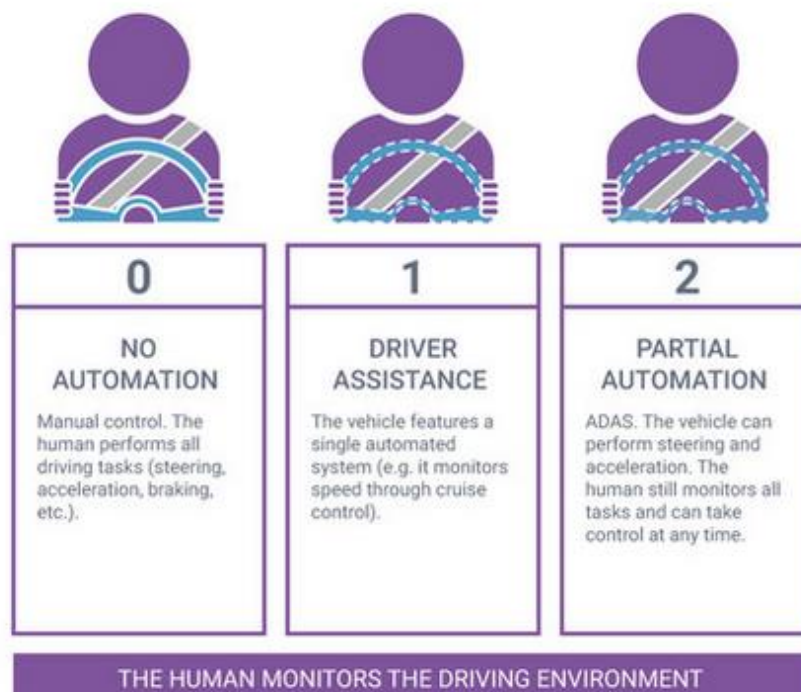


Figure 1.1 SAE Levels of Autonomy (Level:0-2)

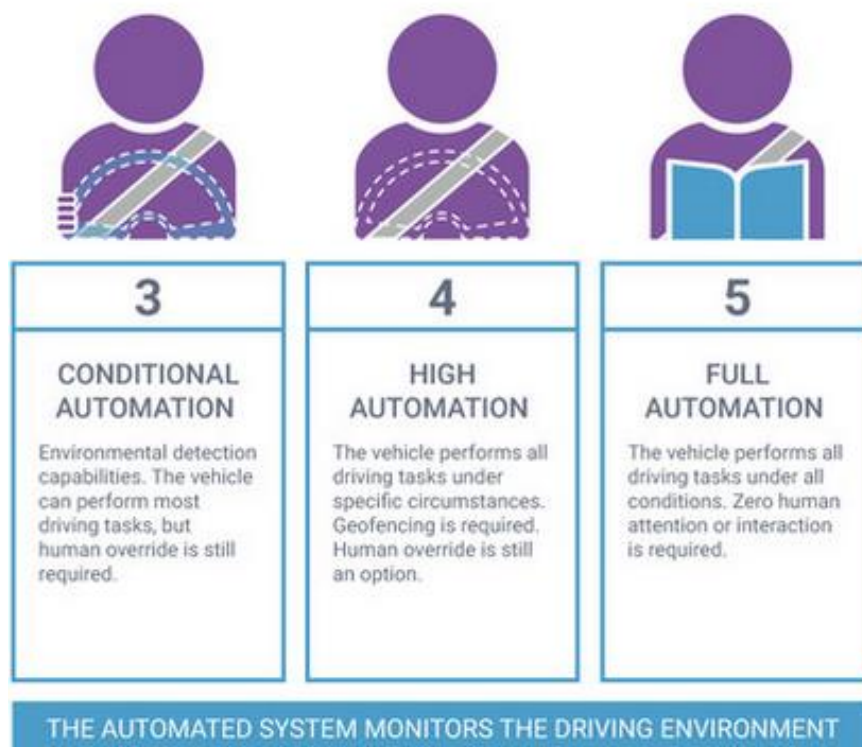


Figure 1.2 SAE Levels of Autonomy (Level:3-5)

Mobile, autonomous agricultural equipment is referred to by a variety of names. In this research, the word "robot" solely refers to machines having strong AI decision-making power in accordance with the reasons in [27]. On the other hand, a piece of technology that has autonomy of operation with a preset course or itinerary is referred to as autonomous equipment (or machines). The primary subjects of this research are levels 2 and 3 on the driving automation level scale (SAE, 2018).

Agriculture robots have not been explored more, for platform based on simpler mathematical approaches for autonomous vehicle. Most research studies on agriculture robot have been approached similar to passenger vehicle technologies which is different its purpose and environment. Conventional (data science oriented) technology requires large database, expert programmer/operator and advanced calibration systems. This can defer farmers from adopting such technology. Complex algorithm for localization and mapping required 3D cloud point computation and suffers from inconsistent results under various conditions. Like AI or Machine

Learning algorithms, these methods also adds more complexity and cost to the system. Navigation system must be robust and predictable at the expense that some preconditioning of farming field maybe acceptable. Though drones have become a popular vehicle for farming applications, several nations have come with restrictions of flying drones, they are limited in range of operation and its remote control, drones flying also require pre declaration of area of operation agreement with authorities in some countries. Indian govt has online digital-sky airspace map for flyability of drones. Northeast states like Mizoram and Sikkim have very less flyable zones due to international borders. Usability of mobile wheeled robots will face lesser regulations and adoption challenges than legged and flying machines. There is also a setback in autonomous robots for agriculture as no proper protocol and regulations has been adopted by govt to practice and manufacture them. Safety and privacy concerns for humans are concerning matters in flying drone operations and AI based machines. Cases of failures and accidents should be accountable.

### **1.3 Problem statement**

For acceptance of autonomous mobile robots in agriculture, basic standard fulfilling, simplistic method of navigation is necessary for the robot such that the system is low cost design and operable by farmers. Modern autonomous vehicles controller based on AI/Machine learning and SLAM technology can be costly and unpredictable. Data processing and learning oriented autonomous vehicle technologies require very long distances training. Human level driving accuracy required billion kms of driving and training. Such practices are difficult for farming operations. Passenger vehicles ply across roads with different types of landmarks, road signs. Farm robots have lesser obstacles and traffic protocols to bother, but each farm areas have unique type of landmark, objects, and texture. Internet based IOT and Cloud computing system required significantly high bandwidth connectivity which has not yet been solved in developing nations. A semi-automatic agricultural mobile robot (SAE level 2-3) would be appropriate technology to starting practising technologies in developing South Asian countries. Vehicle should be based on smart yet predictable navigation system. Navigation system based on hybrid algorithm motivated from classical methods can be attempted.

## 1.4 Objectives

- To make low computation, low data handling navigation algorithm capable of detecting and avoiding dynamic obstacles based on hybrid algorithm.
- To mitigate inefficiency problem after obstacle avoidance on pesticides application caused by dynamic obstacles.
- To test the navigation algorithm in MATLAB software for results and validation of algorithm.

## 1.5 Phases of Research Work

- Phase 1: The study was carried out for robotic platform and algorithms for finding a low computation, capable of detecting and avoiding dynamic obstacles possible in agriculture.
- Phase2: Development of a hybrid algorithm suitable for autonomous navigation algorithm was carried out. Algorithm was built keeping in mind the energy efficiency of mobile vehicle and optimal performance for precision farming (w.r.t implementation in pesticide applying operation and irrigation).

## 1.6 Thesis Organization:

**Chapter 1:** discusses the overall introduction, the literature review of the research. The scope of the study is discussed, the phases and chapter of the research is explained in brief.

**Chapter 2:** This section makes comparative study on different navigation algorithm and is compared with Pure Pursuit Algorithm. The suitability of differential robot is understood for its applicability in agricultural robot. Different kinds of tracking algorithms are discussed. A simulation experiment is conducted to compare the tracking algorithms.

**Chapter 3:** This part of the research deals with study on various types of sensors available for obstacle detection, they are studied and compared with LIDAR based technology. Different kinds of localization methods, types of mapping and obstacle

avoidance algorithms for mobile agricultural robot are reviewed. A simulation experiment is conducted discussed to compare the obstacle avoidance algorithms.

**Chapter 4:** This chapter focuses on the hybrid algorithm developed during the research. The algorithm is simulated in MATLAB software, and results are analyzed and concluded.

**Chapter 5:** Future scope and conclusion of the research is discussed.

## **CHAPTER-2**

### **Study on Pure Pursuit Algorithm for Differential Robot and its applicability in Agricultural Robot**

## **2.1 Introduction**

The pure pursuit was an early method used for autonomous tracking on robots. It is based on preplanned coordinates called waypoints to be covered by robot one after another until last waypoint is reached. It is a geometrical approach where a vehicle assumes a lookahead point in trajectory path. The steering angle required to reach the path is calculated based on a curvature created between the lookahead point and current position of the robot. The theory behind the PPA algorithm has been discussed in the section. Wheel based robots has been compared with other technologies of farm robots. The study on differential wheel drive comparison with other farm robot technologies of robot will be discussed. Different types of basic tracking system for autonomous robots have been discussed. The comparison among PPA LQR and MPC model robot has been carried out with an experimental simulation.

## **2.2 Theory of Pure Pursuit Algorithm**

The pure-pursuit method then first appears in the field of robotics in 1985 . [28] developed a system for estimating the maneuvering required to keep the vehicle on the road. This was done by maintaining the road's center of gravity in the onboard camera's view. [29] suggests the pure-pursuit approach to follow explicit paths based on this idea. Pure pursuit refers to the idea that the vehicle is pursuing a point on the path some distance in the future. The work of Coulter [30] discusses a few pure-pursuit algorithm implementation problems. Since then, indoor [31] and outdoor [32] navigation have both utilized the pure-pursuit technique for specific path monitoring. Additionally, pure pursuit algorithm has some suggested enhancements. [29] proposes a virtual objective point at the lookahead distance to address the issue of the vehicle being far off the path. In [30], an intrinsic correction is added to the fundamental pure-pursuit tracker to lessen systematic path tracking errors caused by differences between desired and actual steering angles. A term proportional controller to the heading inaccuracy between the vehicle and the path was added by [33]. Based on screw theory, vector pursuit uses the goal point's orientation in addition to its location [34]. In conclusion, pure pursuit is a common strategy continuously examined by numerous authors. Except for previous work on human following with a rotary sonar and a 2D

laser scanner [35], monitoring of implicit pathways hasn't gotten much attention since the concept was first suggested for mobile robots [28]. Additionally, no previous research was investigated the method's potential for combining several implicit and explicit path types. Pure pursuit tracking is used in pure chasing waypoints. To advance the robot from its present location to a look-ahead point in front of the robot, it computes the necessary angular velocity instruction. The look-ahead point on the route is then moved by the algorithm depending on the robot's current location until it reaches the end of the path, the end waypoint. This may be compared to a robot that is always pursuing a spot in front of it. The look-ahead point's placement is determined by the Lookahead Distance setting. The Pure pursuit object serves as a tracking algorithm for path-following algorithm that can work autonomously. A given set of waypoints is specific to your controller. PPA algorithm can have extended possibility to specify the desired linear and maximum angular velocities. The specifications of the vehicle dimension are used to establish these characteristics setting. The Lookahead distance defining how far along the route to track toward, is the most significant attribute.

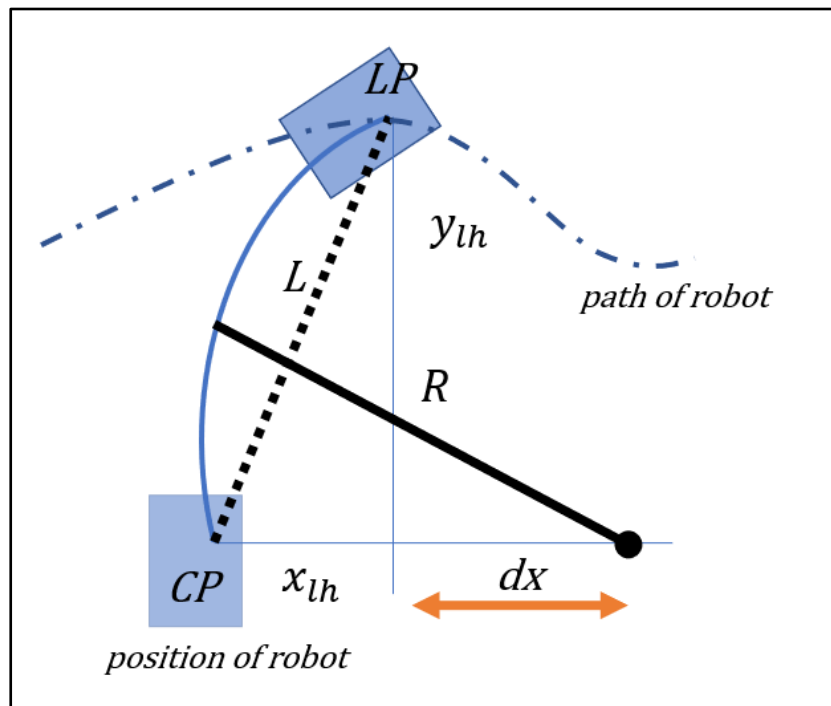


Figure 2.1 Geometrical representation of PPA algorithm



The system is a goal-seeking algorithm known as the pure pursuit approach developed [29]. The control algorithm calculated the proper turning radius to provide the desired look-ahead coordinate based on a preset look-ahead distance. **LP** is the lookahead point in the reference to be followed by the robot. **CP** is the present position of robot. **x,y** is calculated position of **LP** with respect to the position of the robot. At a distance, of radius **R** is calculated such that it passes through the point **CP** and **LP**. The circumference of the circle with radius R find the desired steering angle to move towards point LP on the path. The connection between **x, y, R, L**, and dx was calculated using the nomenclature as in figure 1. Following equation (2.1),(2.2) and (2.3) can be derived from the geometrical representation

$$x_{lh} + dx = R \quad (2.1)$$

$$x_{lh}^2 + y_{lh}^2 = L^2 \quad (2.2)$$

$$R^2 = dx^2 + y_{lh}^2 \quad (2.3)$$

By solving the equation (2.1),(2.2) and (2.3)the following expression are reduced to  $R = (L^2/2x)$ ,corresponding to the curvature for path desirable for travel, the desired the curvature becomes steering angle given by

$$\gamma = (2x/L^2) \quad (2.4)$$

Since x was the displacement error and  $2/L^2$  was the gain, the pure-pursuit method was a proportional control algorithm. The look-ahead distance L was raised to lower the controller's gain, and the look-ahead distance was reduced to enhance the gain. Flowchart of operation is explained in Figure 2.2. The pictures given below can be understood as effects of changing lookahead distance of the PPA algorithm. Closer distance produce unnecessary oscillations as seen in Figure 2.3 while very large lookahead distance can make the robot travel very far at large curvature at turns explained in Figure 2.2. Hence it is necessary to tune the lookahead distance properly such that it does mobile vehicle has optimum maneuverability. The distance depends on the type of path and robot degree of freedom.

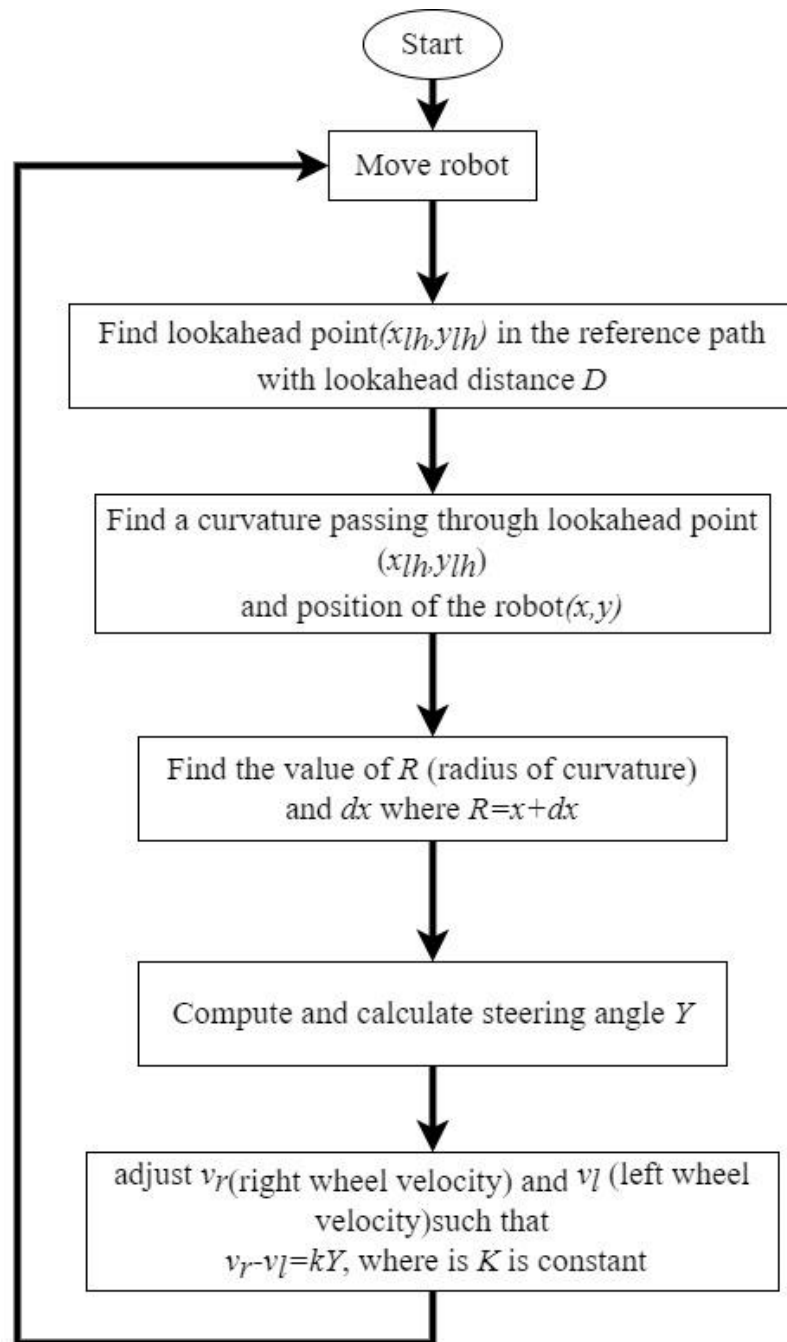


Figure 2.2 Flowchart for PPA algorithm

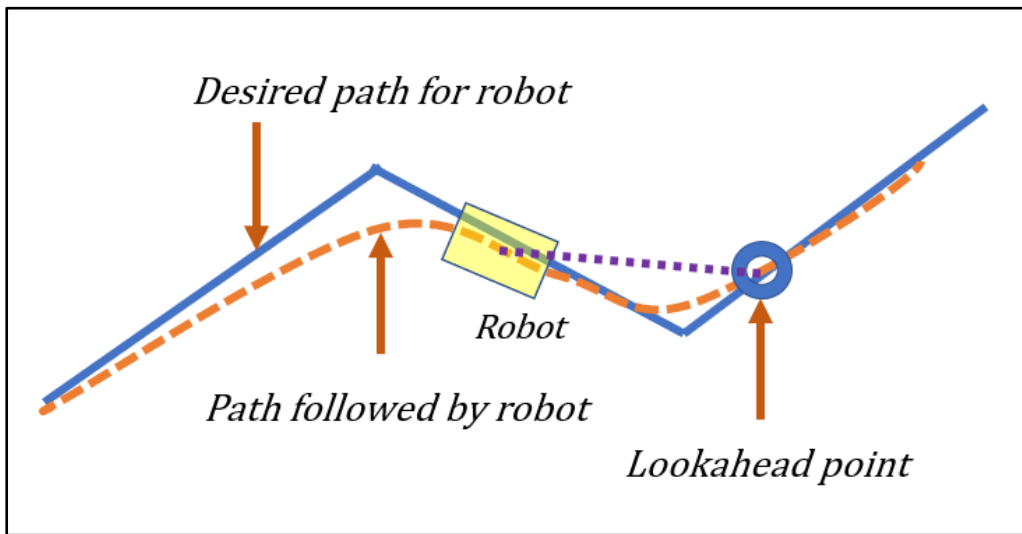


Figure 2.3 Trajectory with large lookahead distance

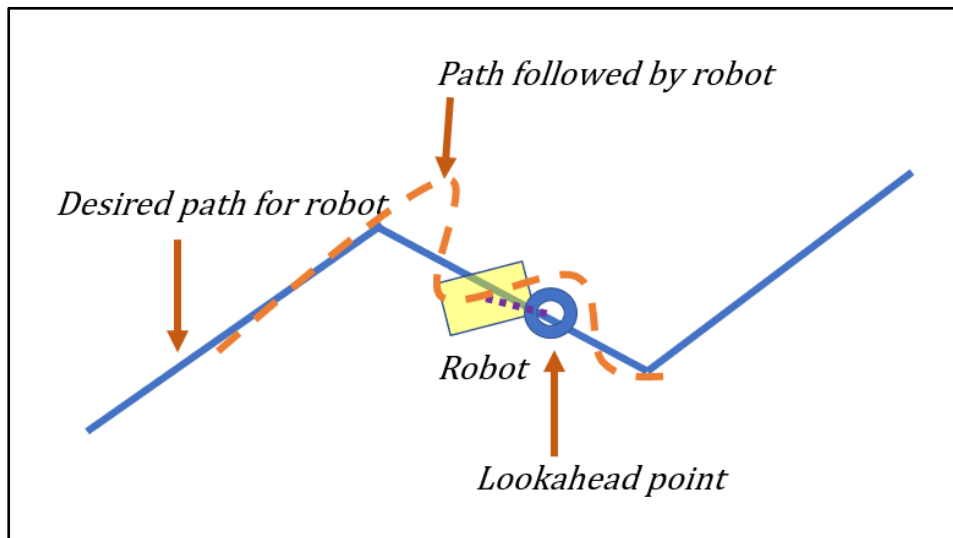


Figure 2.4 Trajectory with small lookahead distance

### 2.3 Mathematical Model for 2-Wheel Differential Mobile Robot

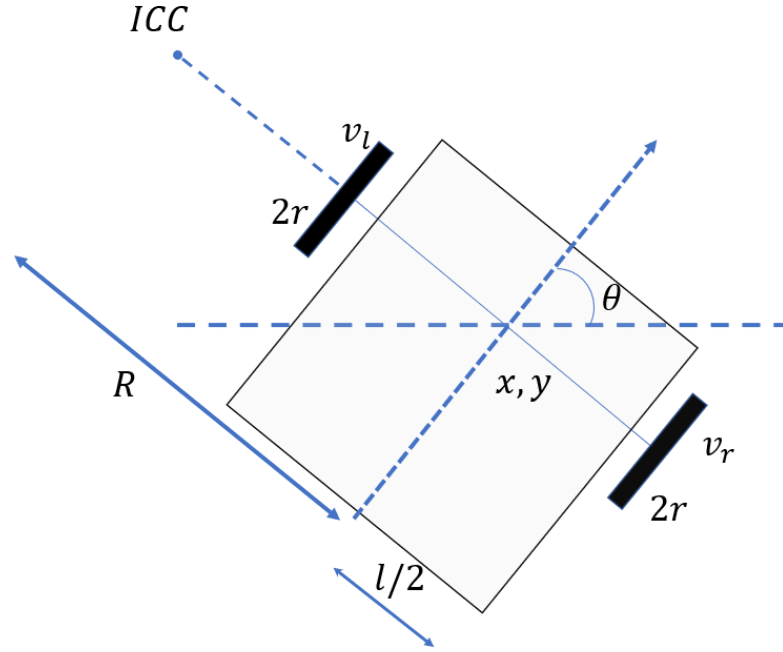


Figure 2.5 Differential Drive kinematics

The robot's trajectory can be changed by adjusting the speed of its left and right wheels. Formulation of the following equations (2.5),(2.6) by both wheels is derived based on rotation about the ICOC Instantaneous Centroid of Curvature:

$$\omega = (R + l/2) = v_{rw} \quad (2.5)$$

$$\omega = (R + l/2) = v_{lw} \quad (2.6)$$

where  $v_{rw}, v_{lw}$ , are the right and left wheel velocities along the ground, and  $R$  is the distance between the ICC to the midpoint between the wheels.  $Ln$  is the wheelbase length. We can compute  $R$  and  $\omega$  at any moment by using the following formulas in equation (2.7) and (2.8):

$$R = \frac{(v_{rw} + v_{lw})}{2(v_{rw} - v_{lw})} \quad (2.7)$$

$$\omega = \frac{(v_{rw} - v_{lw})}{Ln} \quad (2.8)$$

## 2.4 Forward Kinematics of Mobile Robot

Assume that the robot in figure 1 is at a position  $(x, y)$  and moving in a direction that forms an angle with the X axis. We suppose that the wheel axle's middle serves as the robot's centre of gravity. The control parameters  $v_{lw}$  and  $v_{rw}$  can be changed to cause the robot to move to various positions and orientations. Wheel velocities along the ground are  $v_{lw}$  and  $v_{rw}$ . We may determine the location of the ICC using equation (2.9) and the velocities  $v_{lw}$  and  $v_{rw}$ .

$$ICOC = [x - R \sin(\theta), y + R \cos(\theta)] \quad (2.9)$$

After a time change of  $\delta t$ , the change value of position and orientation is given in equation (2.10):

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICOC_x \\ y - ICOC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICOC_x \\ ICOC_y \\ \omega\delta t \end{bmatrix} \quad (2.10)$$

equation (2.11),(2.12),(2.13) represents the instantaneous values of position and speed

$$x(t) = \int_0^t [V_{avg}(t) \cos[\theta(t)]] dt \quad (2.11)$$

$$y(t) = \int_0^t [V_{avg}(t) \sin[\theta(t)]] dt \quad (2.12)$$

$$\theta(t) = \int_0^t \omega(t) dt \quad (2.13)$$

The inverse dynamics of capable finding the position of the robot and the orientation are written in equation (2.14),(2.15),(2.16)

$$x(t) = \int_0^t [v_{rw}(t) + v_{lw}(t)] \cos[\theta(t)] dt \quad (2.14)$$

$$y(t) = \int_0^t [v_{rw}(t) + v_{lw}(t)] \sin[\theta(t)] dt \quad (2.15)$$

$$\theta(t) = \int_0^t [v_{rw}(t) - v_{lw}(t)] dt \quad (2.16)$$

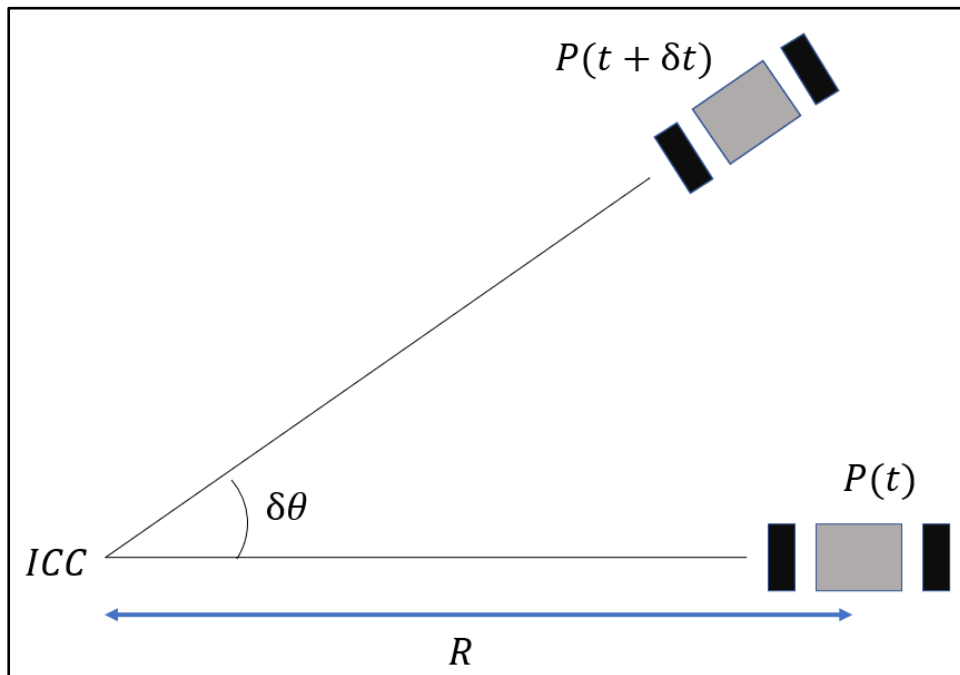


Figure 2.6 Forward kinematics of the differential robot.



Figure 2.7 2-Wheel drive differential robot prototype (SWAGBOT)

### 2.5 Wheel Odometry for Robot Localization

A robot must be aware of its location to perform well in search and retrieval tasks. It may not seem like a difficult effort to know where you are, but as you will see, it may

be difficult. People can locate themselves utilizing their sharp camera system and awareness of their environment. It is based on the stereoscopic viewing and map modelling abilities which are inherent in human brains. However, simple robots do not naturally possess these sophisticated abilities, particularly the kind you will be developing for this class. Odometry is one tool available to robots to help them locate themselves in their surroundings. It is the use of motion sensors to calculate the change in the robot's location in relation to a predetermined point. For instance, if a robot is moving straight forward and knows the diameter of its wheels, it may calculate the distance travelled by counting the number of wheel spins. Drive wheels on robots frequently have shaft encoders connected that produce a predetermined number of pulses each revolution. This pulse count allows the processor to calculate the distance travel led. The sensor data has some drawbacks despite being a widely used position sensor for mobile robots. Since the measurement is continuous, any sensing inaccuracy will grow over time. Robots may occasionally require the use of additional sensors to pinpoint their position in order to avoid having too many errors accumulate.

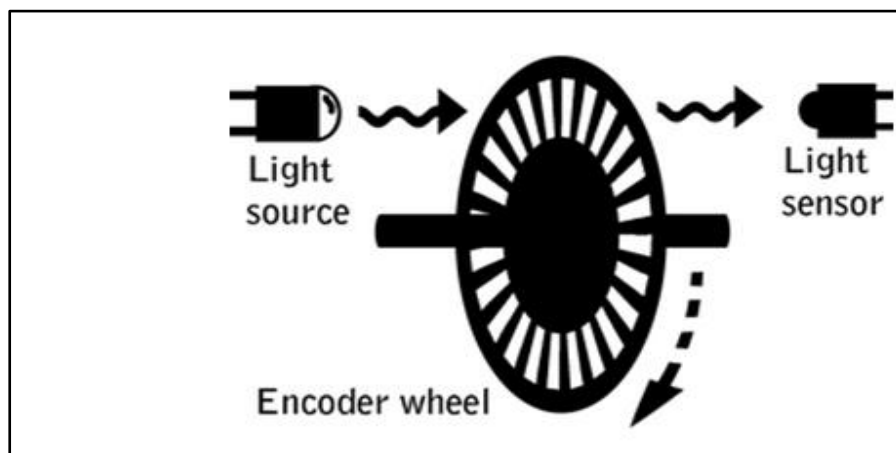


Figure 2.8 Wheel odometer sensor

The change in position of a wheel is given by the number of ticks counted by the wheel encoder at a time  $\Delta t$ . So,  $\Delta Tick$  is the number of ticks read by the robot's computer, the distance  $D$  travelled of the wheel is given as eqn. (2.17)

$$D = 2\pi R \Delta Tick / N \quad (2.17)$$

where  $N$  is the number of slots the encoder has in one revolution. The velocity of the wheel is given by eqn. (2.18):

$$v = D/\Delta t \quad (2.18)$$

## 2.6 Various Platforms for Ground Mobile Robots

Although there are many other platforms for agricultural machinery, farm tractors and harvesters are the most popular and well-known machines. While tractors handle most of the farm work utilizing various implements for tillage, weeding, sowing, fertilizing, and irrigation, combines are employed for harvesting duties. Recently, certain airboats and rice transplanting machines have also been employed in rice paddies. In other unique situations, utility vehicles are used in transportation.

The transporter system is an AV's and mobile platforms' primary component. The four types of mobile systems currently in use are wheel, half crawler, full-crawler. Each of these transporter systems has pros and cons and was created with a particular application and level of performance in mind. Before choosing the best transporter system, the engineers of an agricultural AV must take account of the manageableness of mobile robot in various farming conditions into account. Wheel-based transporters are the most used form of transporting technology. This kind of transporter offers good steering and vehicle speed and is inexpensive, lightweight, easy to design and produce, and repairable. The wheels' tremendous pressure on the earth, however, increases the likelihood of compaction. Because the wheels' contact area is less than that of crawler machines, there is less friction, but this can also lead to more slippage and less efficiency. The vehicle wheels may be built of a variety of materials and help the car to maneuver with accuracy. The crawlers' bigger connecting surfaces improve friction, lessen the likelihood of slipping, prevent long-term soil compaction, and slow down maneuverability. This form of transporter system has more moving parts than the wheel robots, which increases the cost, breakage risk, and weight. Other drawbacks include a shorter system lifetime and less agility, steering, and velocity.



Table 2.1 Comparison of characteristics of various mobile robots

	Wheel	Crawler	Foot
Complexity of designing and development	Low	High	High
Reparation	Easy	Hard (easy to break)	Hard
Life time	Long	Short	Short
Cost	Low	High	High
Material type	Various	Only hard material Ex. Metal	Various
Weight	Light	Heavy	Light
Ground pressure	High	Low	High
Soil compaction	High	Less	Low
Power efficiency	Low	High	Medium
Friction	Low	High	Low
Maneuverability	Good	Bad	Limited
Maneuver environment	Dry and solid	All surfaces	Limited
Traction on slop, slippery and wet ground	Not good	Good	High
Drive over various obstacles	Hard	Easy	Medium
Traction system	Not optimized	Optimized	Not optimized
Steering	Good	Poor	Poor
Spinning	Good	Better than Wheel	Medium
Turning	Big place	Small place	Poor
Speed	High	Low	Low
Precision	Good	Bad	Low

There are various kinds of ground robot in used in agriculture, some of popular ones are mentioned in the Table 2.2. The most cost effective and popular robots has been wheeled robot.



Figure 2.9 Wheeled robot [source:<https://www.ivtinternational.com/features/feature-what-is-holding-back-agricultural-robotics.html>]



Figure 2.10 Crawler robot [Source: <https://www.roboticsbusinessreview.com/agriculture/the-next-frontier-of-weeding-robots/>]



Figure 2.11 Legged robot [Source: <https://www.bayer.com/en/agriculture/article/ripe-robots>]

It can be observed from the Table 2.1 that wheel robots have low complexity of structure hence easy for development, low cost and long durability and acceptable good steering control for normal conditions. Crawlers are robot with high initial cost, high maintenance cost, they have superior slip control and mechanical efficiency but are heavy machines with low precision of operation. Legged robots are very rarely used because of their complexity of operation and high cost of repair, foot robot do have speed and maneuver issue due to complex operations. Hence it can be understood that wheel robots have been used very often and will remain the best choice for

different kinds of operations. Different sizes, shape of chassis and tires have been tried for various kinds of applications[36]. Differential wheel robots have been the popular choice for autonomous robots due to easy compatibility with electric motor drives[37].

Table 2.2 Prototype models of several mobile robots

<b>Activity</b>	<b>Reference</b>	<b>Mobility</b>	<b>Path Planning</b>
Weeding	SUSAN weeding robot, China	Differential drive 4 wheel steering system	Hough transform for row detection
Weeding	Phoenix robot	Continuous track vehicle	Path Tracking process
Spraying	Bonirob, Bosch	Differential drive 4 wheel steering system	Not available
Spraying	Contadino Robot	Differential drive 4 wheel steering system	Path tracking with self-positioning method
Spraying	AgxeedRobot,Dutch	Chain track wheel	Path tracking for planned trajectory
Tractor multiple operations	BRAIN GL320	4 Wheel machine	For tractor operations
Tractor multiple operations	Yanmar CT801	Crawler	For tractor operations
Tractor multiple operations	MAFF EG83	4 Wheel machine	For tractor operations

## 2.7 Regulations of Drone operations

Drones, sometimes known as “unmanned aerial vehicles” or UAVs, are essentially flying robots, therefore lessons learned from their regulation may be applied to the regulation of agriculture equipment used on the ground. Drones used in agriculture can be used to spray pesticides in hard-to-reach areas, repel birds, identify animals,

and remotely sense cropland for variable rate planting, fertilization, and crop protection. Some, like drones used to spray vineyards, take the role of helicopters, while others, like those used to identify animals, are supplemental. Space limits and the need for human intervention in drone operations are important regulatory issues. For instance, there are often higher regulations for heavier drones that carry sprayers. Drones that are very light are often free from several regulations. Attached cameras, however, may be governed by data and privacy protection laws. Drone flight space is often constrained by height, distance from airports, heliports, and air traffic control facilities, distance from crowds of people or public places on both the horizontal and vertical axes, and distance from nature preservation zones. This restricts the usage of agricultural drones, especially when people are crowded together.

Authorization for certain applications of heavier drones, activities near airports, in airspace control zones, etc., is frequently achievable but expensive and only for a limited time. For operators, there is a usual "see-and-avoid principle." A drone pilot may be able to traverse longer distances provided certain restrictions, such as those based on the expanded visual line of sight for sparsely inhabited places, are followed. When another person is present and could act, exceptions from the sight regulations—including flying while using video glasses—may be permitted. In Switzerland, for instance, administrative delays of at least 3 months are a result of strong demand for authorizations [10]. Additionally, in Switzerland, before receiving the standard authorization for drone usage, agricultural drones used for spraying must first have a spraying authorization, like that required for regular sprayers [10]. Similar practices have now been followed by other European nations. Regulations governing drones, particularly those governing on-site human supervision, may offer guidance for autonomous agricultural equipment. Adoption patterns for agricultural drones appear to be influenced by legislation. Compared the United Kingdom, where line of sight mandatory for drones and other aerial application is extremely constrained, the use of drones for other purposes besides data collection, such as pesticide spraying and the application of other inputs, has grown most swiftly in China and Brazil.

Despite their weight, agricultural drones are less dangerous than other drones since they operate near to field surfaces rather than necessary near people or sensitive

airspace. The European Union proposes to establish laws that will simply need operators to put in a statement with beyond direct line of sight for rural regions as a normal scenario, but only for drones weighing upto 25 kg, in order to prevent administrative obstacles and authorization delays for operators [11]. A North Dakota-based network of business, government, and academic institutions is combining resources to build up the infrastructure needed for bigger agricultural drones. Their regulatory strategy calls for publishing flight plans and "beyond-visual-line-of-sight" activities [12][13]. These results suggest that agricultural drone-specific legislation is developing, which can promote safe and legal usage. However, to meet the demands for monitoring, it may require more administrative expenses and staff. In reality, it is difficult to regulate larger drones that are required to apply physical farming inputs.

## 2.8 Path Tracking Controllers

Autonomous vehicles have become more popular as artificial intelligence has advanced [38]. The four components of autonomous vehicles are: sensor data collection, perception from data analysis, planning next move after perception , actuation[39]. As seen in the figure 2.12 below the important problem to be solved autonomous vehicles is Perceive and the plan stage. Stage[40]. The most crucial component of autonomous vehicles is path tracking, which aims to precisely guide the vehicle along the reference route provided by path planning [41]. One of the most widely used path tracking algorithm is pure pursuit [42]. The (PID) proportional integral- derivative, model predictive control (MPC) algorithms and linear quadratic regulator (LQR) are the other three basic geometric path tracking algorithms. Pure pursuit algorithm is simpler than other path tracking algorithms.

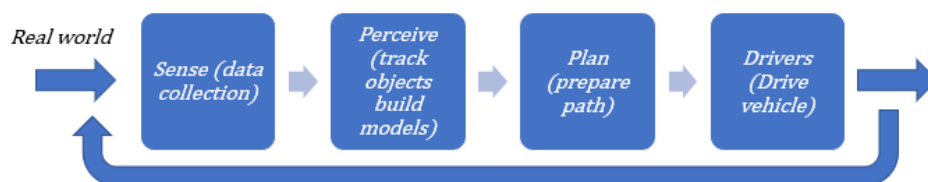


Figure 2.12 Stages of navigation process

Pure pursuit algorithm outperforms other path tracking algorithms in terms of tracking accuracy and simplicity [43][44]. Many academics concentrate their efforts on pure pursuit algorithms. To enhance tracking performance at high speeds, M. Elbanhawi et al. [45] developed a model predictive effective yaw control application of pure pursuit path tracking. In order to reduce tracking errors, Shan et al. [46] created a novel pursuit algorithm called CF-Pursuit that substituted the clothoid C curve for the circle being used pure pursuit. Numerous researchers [47][48] modify the lookahead distance which is directly proportional to its operating speed. Table 2.4 makes the brief feature comparison of different kind of algorithms for tracking autonomous vehicles. Pure pursuit, Stanley and MPC controllers are the most widely accepted methods for geometric approach for basic navigation in mobile robots[49].

Table 2.3 Feature comparison of different tracking controller

<b>Features</b>	<b>Pure Pursuit Controller</b>	<b>Stanley Controller</b>	<b>Model Predictive Controller</b>
Type of operations	Simpler geomteric controller	Improved geometric controller suitable for 2 axle vehicles	controller based on mathematical kinematic model of the system
Specific features	Proper lookahead distance tuning depending on shape and speed of vehicle	Opposite direction steering for heading error correction	MPC must be solved numerically and cannot provide a closed-form solution
Computation load	Computationally cheap	Computationally cheap	Computationally expensive
Limitations	Optimal lookahead distance may vary depending on robot design and path type	Deviations from reference path is higher	Non-linearities can effect the performance.

## 2.9 PPA applied on Autonomous NISSAN Passenger Car

An experiment was conducted to equip a Nissan Leaf electric car with autonomous features[50]. The system's prototype functioned admirably when the Zala Zone automobile industry testing facility opened. This vehicle can only operate on the university campus and proving ground, and it must go at a comparatively slow speed

of 25 km/h. The multiple-goal points-based pure pursuit is the initial and very promising alteration. It is predicated on the straightforward premise that it considers several potential goal points before choosing the one that fits best. Based on its fixed number of discretely conceivable curvatures, the realization. The wheel angle is restricted by the kinematic characteristics of the vehicle, directly determines these curvatures. Instead of utilizing an optimizer, we employed a fixed set of discrete potential curves in our realization and determined the best fitting solution. The reading were observed as given in Table 2.3. The modified Novel pure pursuit algorithm had acceptable Average deviation AvgD and Maximum deviations MaxD, the error and deviations were improved over the classical simple pure pursuit and follow the carrot methods.

Table 2.4 Deviations caused to Nissan Leaf prototype (Széchenyi Istv'an University)

Method	Lookahead Parameter	Deviation[m] ZalaZone racing track	Deviation[m] School
Follow the carrot	10	AvgD: 0.65 MaxD: 2.76	AvgD: 0.55 MaxD: 2.7
Follow the carrot	10.5	AvgD: 0.74 MaxD: 3.12	AvgD: 0.63 MaxD: 2.9
Classical pure-pursuit	2	AvgD: 0.10 MaxD: 0.56	AvgD: 0.08 MaxD: 0.5
Classical pure-pursuit	2.3	AvgD: 0.12 MaxD: 0.64	AvgD: 0.09 MaxD: 0.6
Classical pure-pursuit	2.7	AvgD: 0.13 MaxD: 0.74	AvgD: 0.11 MaxD: 0.7
Classical pure-pursuit	3	AvgD: 0.15 MaxD: 0.97	AvgD: 0.13 MaxD: 0.8
Classical pure-pursuit	10	AvgD: 0.52 MaxD: 1.90	AvgD: 0.46 MaxD: 1.9
Adaptive LH distance	2.33; 2.0	AvgD: 0.125 MaxD: 0.653	AvgD: 0.17 MaxD: 0.7
Adaptive LH distance	2.33	AvgD: 0.12 MaxD: 0.636	AvgD: 0.097 MaxD: 0.6
Multiple goal	5; 6; 4	AvgD: 0.48 MaxD: 1.40	AvgD: 0.30 MaxD: 1.34

## 2.10 Comparative Case Study on various Path Tracking Algorithm

Considering a 2-wheel robot (front and wheel) was attempted to control with the help of three different strategies of path tracking algorithm. Geometrical depiction given in figure 2.13. Taking a no. of goal points as  $N$ , assume that angle  $\alpha_i$  is inside the range  $[\alpha_{min}, \alpha_{max}]$  of all feasible angles (presumably the wheel angle limits). It is possible to calculate a series of curves, each with a radius of  $\rho_i = -2/\alpha_i$  and a centre point of  $C_i$ .  $C_i, G_k$  calculates the line segment's distance from the objective point. The d-

distance from the curvature is given by  $G_k$ , the normalised difference between the length of this segment and the radius.

$$d_{sum} = \sum_{k=0}^N |||C_i, G_k|| - \rho_i| \quad (2.19)$$

The total of the difference eqn. (2.19) for each goal point serves as its metric for choosing a proper angle. The curvature with the lowest  $d_{sum}$  is finally picked. Vehicle speed was increased for longer look-ahead distances while being kept low for shorter look-ahead distances. A brief study on different methods carried out by [49] has laid out the general characteristics Pure pursuit, Stanley and model predictive controller in following table. Using the criteria of regulating accuracy and computing economy in low-speed settings, this[51] research compared to the performances of Pure Pursuit, Stanley, Linear Model Predictive Control, and Linear Quadratic Regulator. The kinematic bicycle model, which creates its location and heading angle of vehicle in accordance with the steering command and desired velocity, was used to build the vehicle model. The controller seeks to achieve zero heading angle error and minimal lateral offset error. The lateral offset error, measured in local coordinates, is the angle between the position of the vehicle and the reference path.

**2.10.1 Pure Pursuit Controller:** To reduce the cross-track error  $d_e$ , the Pure Pursuit controller uses a lookahead point on the desired path at a fixed distance  $l_d$ .

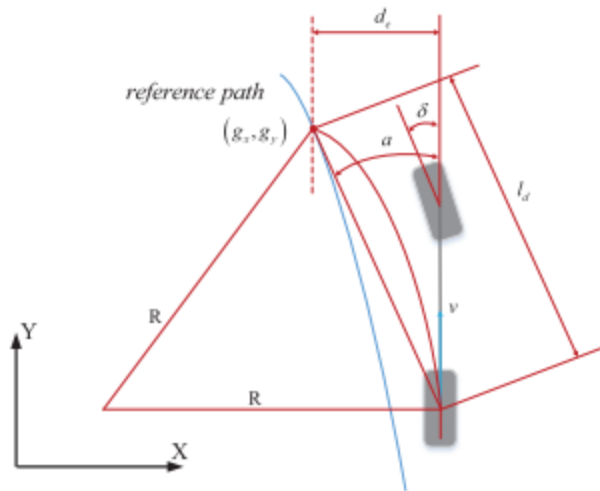


Figure 2.13 Geometrical depiction for 2-wheel robot



A lookahead distance from the position of the axle behind to the reference path determines the destination point.

The geometry relationship is given by  $\delta = \tan^{-1} l_d/R$ , where  $R$  is the radius of the curvature is determined by lookahead point  $(x,y)$  and the coordinate of its rear wheel, may be built using the vehicle bicycle model. There are also  $\sin(\delta) = d_e/l_d$ . The output of the Pure Pursuit controller for a bicycle model is

$$\delta = \theta_e(k) + \tan^{-1}\left(\frac{\lambda d_e(k)}{v(k)}\right) \quad (2.20)$$

where  $\lambda$  is gain parameter. The equation (2.20) shows that the vehicle steer towards the reference path when its crosses the track errors  $d_e$  and  $\theta_e$  increase.

**2.10.2 The LQR Controller:** Linear Quadratic regulator. It was widely used in autonomous wheeled mobile robotics, is simply an automated method of determining an acceptable state-feedback approach. The controller for track the reference trajectory will be constructed in this step. First, the error vector  $e$  is made up of the x-expressions for the position error and orientation error. The error state transformation formula (eqn. 2.21) is given by T,

$$T = \begin{matrix} s \\ d_e \\ \theta_e \end{matrix} = \begin{bmatrix} -\cos & -\sin & 0 \\ \sin & -\cos & 0 \\ 0 & 0 & 1 \end{bmatrix} e \quad (2.21)$$

The linear equation eqn. (2.22) for error is obtained as below:

$$\dot{E} = \begin{bmatrix} 0 & -v \\ 0 & 0 \end{bmatrix} E + \begin{bmatrix} 0 \\ -v/l \end{bmatrix} u \quad (2.22)$$

The aim for always optimal control issues is to establish the best control law. The closed-inputs loop's are therefore expressed in the following equation, where  $K$  is the feedback gain

$$u = KE \quad (2.23)$$

**2.10.3 Model Predictive Controller:** Predictive Linear Model Controller A trajectory error model, a system constraint, and an optimization goal make up an MPC controller. The tracking control system and its foundation are shown in the error model. The constraints of the system produce steady control signals. The system stability and path tracking speed are considered while building the optimization objective.

The autonomous vehicle model is given by eqn. (2.24)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\delta/l \end{bmatrix} v \quad (2.24)$$

Where the error state model is given by eqn. (2.25):

$$\Delta\dot{e} = \begin{bmatrix} 0 & 0 & -v \sin\theta \\ 0 & 0 & v \cos\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \theta - \theta_r \end{bmatrix} + \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ (\tan\delta)/l & v/(l\cos^2\delta) \end{bmatrix} \begin{bmatrix} v - v_r \\ \delta - \delta_r \end{bmatrix} \quad (2.25)$$

The discrete model of the error becomes as in eqn. (2.26)

$$\Delta e(k+1) = \Delta e(k) + t\bar{A}\Delta e(k) + t\bar{B}\Delta u(k) \quad (2.26)$$

These four control techniques govern the model. Another two distinct types of pathways (eight-shape path and circular path) are used to compare these controllers, with the results displayed in Figure 2.14-2.16. We specifically calculate the eight-shape path's distance error those that are indicated by  $e_x$  and  $e_y$  respectively. The simulation results are shown in Table 2.5. In conclusion, the controller for Pure Pursuit is sensitive to setting of its lookahead distance. A short lookahead distance causes quick variations in the steering angle. Although the tracking is less precise with a wide look-ahead distance, it is more stable. On a smooth path, the Stanley controller performs accurately in tracking, but a sudden change at the path's curve causes significant tracking mistakes. The Linear Quadratic Regulator controller does not take constraints into account, but it offers an explicit and linear solution with little online computational overhead. Proactive actions are taken by the Model Predictive Control, and restrictions are taken into account. It has fewer tracking mistakes than

the other controllers despite having a somewhat higher computing overhead. The graph plot taken by different algorithms are given in Figure 2.13,2.14,2.15[51] for sinusoidal, 8 and circular shape trajectory.

Table 2.5 Error measure of different techniques

<b>Error</b>	$e_x(m)$	$e_y(m)$
MPC controller	0.050	0.471
LQR controller	0.066	0.591
PPA controller	0.204	0.2314

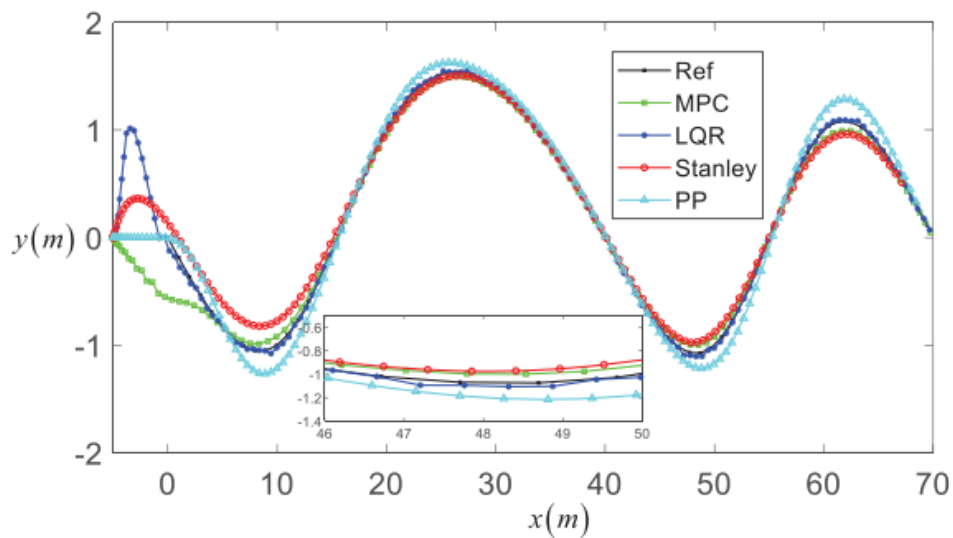


Figure 2.14 Result of tracking in sinusoidal shaped path

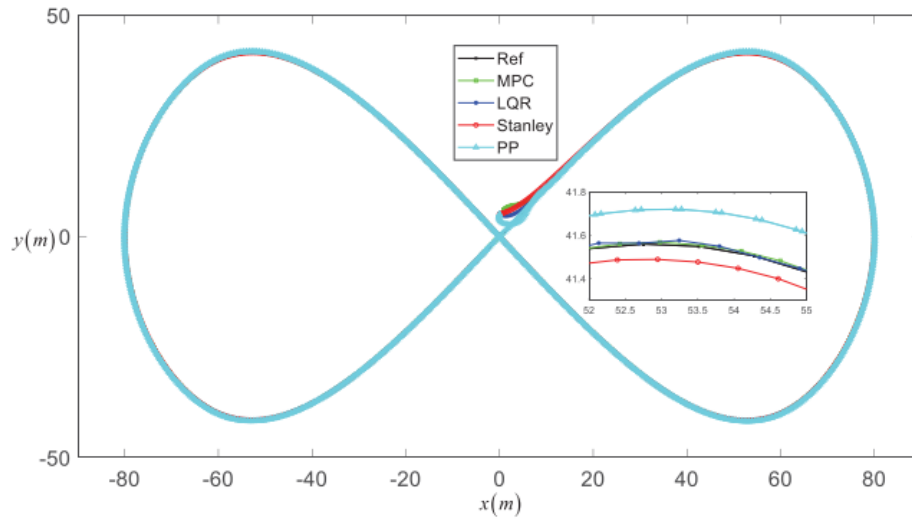


Figure 2.15 Result of tracking in 8-shaped path.

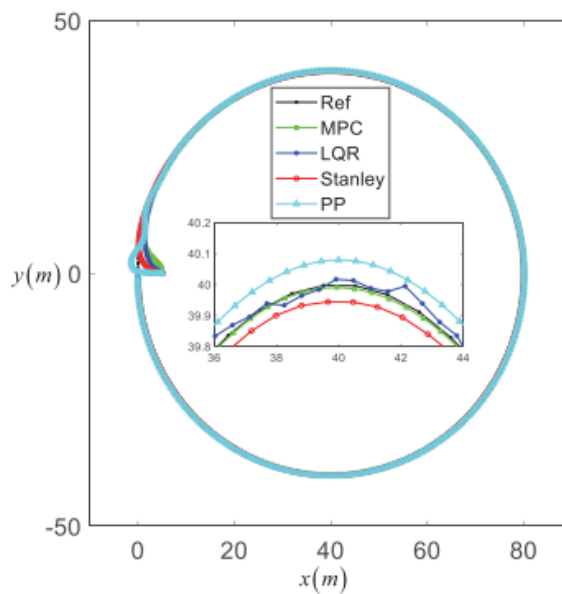


Figure 2.16 Result of tracking in circle shaped path

### 2.11 Experiment for Comparison of Path Tracking Controllers

An experimental and analysis for comparison of tracking algorithm for a different robot in farm field scenario was carried out in MATLAB software. The sample plot of the trajectory travelled by robot is given in Figure 2.15. Three different algorithms Pure Pursuit Algorithm, Model predictive control and Linear Quadratic Regulator

were tested for a differential robot of dimension wheelbase length 0.5 meters. It was tested for a field dimension of 40m x40m. Scale of map is 5 pixels per meter. The task was to cover 2 rows of plants. The plants are represented by black parts in the map. The a map was designed using grid occupancy map conversion of a actual plantation field. The simulation was carried out in a 2 D environment concerning only the lateral movements of the robot. Waypoints were created for the robot to travel beside different rows of plants. The robot was designed to perform at two levels of speed 1.5km/hr and 2.5km/hr for all three methods.

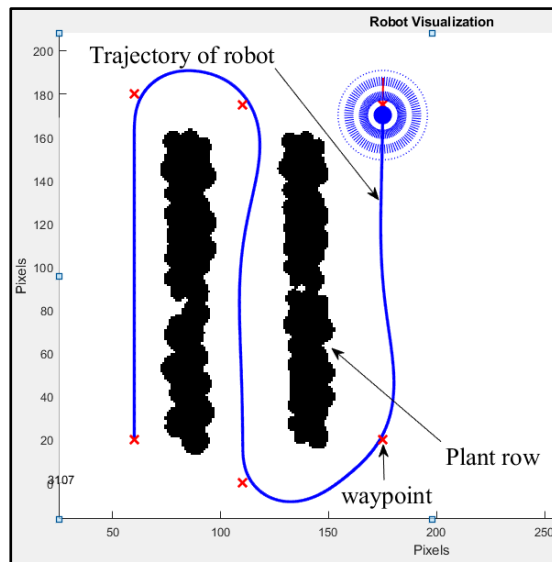


Figure 2.17 Robot trajectory for tracking analysis performed in MATLAB simulation.

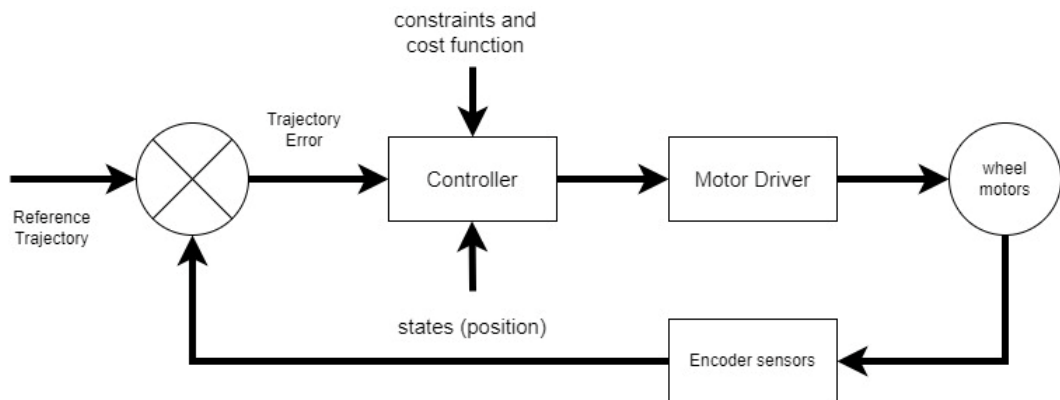


Figure 2.18 Block diagram for LQR/MPC controller.

Deviation caused in travelling the path was measured. The actual base length covering all waypoints from start to finish was 609 pixels which is equal to 122m according to the scale of the map. The total time taken by each method to complete the task was also accounted. Time measurement is important for comparison of algorithm. The lookahead distance was also varied for two different settings of 4m and 8m. This simulation is tested only for path tracking performance but not the object avoidance strategy.

## 2.12 Observations and Analysis for Comparison of Path Tracking Controllers

From the experiment important observations were conducted.  $Dist_{trv}$ ,  $Time_{exe}$ ,  $Dev\%$  are distance travelled, time of execution and deviations caused during travel. The three indicators for analyzing the performance of the algorithm. The readings are given in the Table 2.6. Distance travelled indicates the total path travelled by the robot to reach its goal. Higher  $Dist_{trv}$  indicates more overshoot of travelling caused by the method hence it represents the overall oscillation and overshoots affecting the robot. Higher value of  $Dist_{trv}$  implies for higher  $Dev\%$  and hence inefficiency faced. As the agricultural path is fairly a straight the overshoots at the corners and turns causes more deviation than the oscillations.

Table 2.6 Performance of comparison of tracking algorithms.

		<b>PPA</b>	<b>MPC</b>	<b>LQR</b>
<b>Vel1=1.5(km/hr)</b>	<b>Dist<sub>trv</sub>(m)</b>	124	123	123
	<b>Time<sub>exe</sub></b>	310	410	330
	<b>Dev%</b>	2	1	1
<b>Vel2=2.5(km/hr)</b>	<b>Dist<sub>trv</sub>(p)</b>	128	124	125
	<b>Time<sub>exe</sub></b>	182	405	302
	<b>Dev%</b>	5	2	3
<b>LhD<sub>1</sub>=4m</b>	<b>Dist<sub>trv</sub>(p)</b>	125	124	125
	<b>Time<sub>exe</sub></b>	221	405	301
	<b>Dev%</b>	3	2	3
<b>Vel=2(km/hr)</b>				
<b>LhD<sub>2</sub>=8m</b>	<b>Dist<sub>trv</sub>(p)</b>	129	124	125
	<b>Time<sub>exe</sub></b>	232	405	301
	<b>Dev%</b>	8	2	3

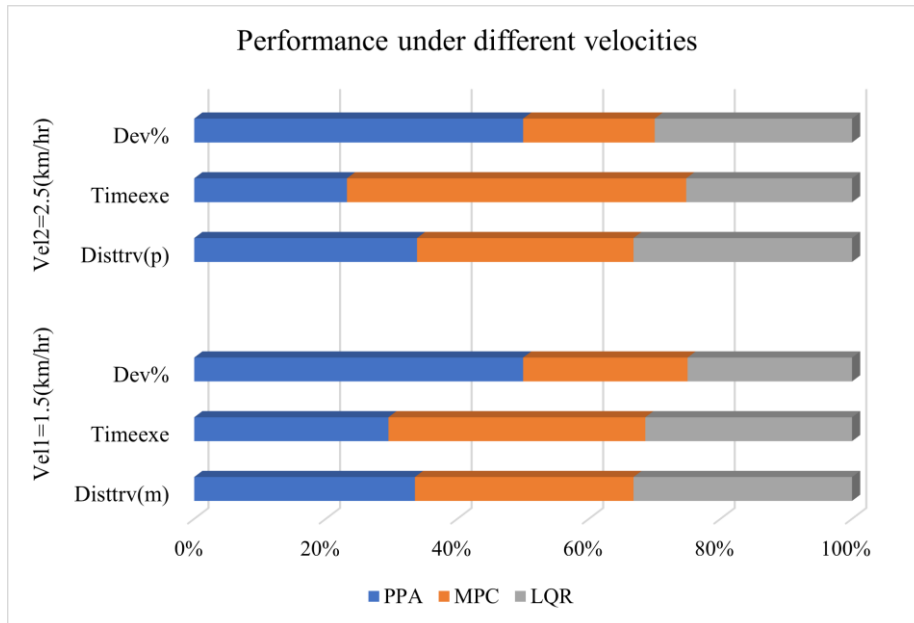


Figure 2.19 Performance chart at different velocities of robot

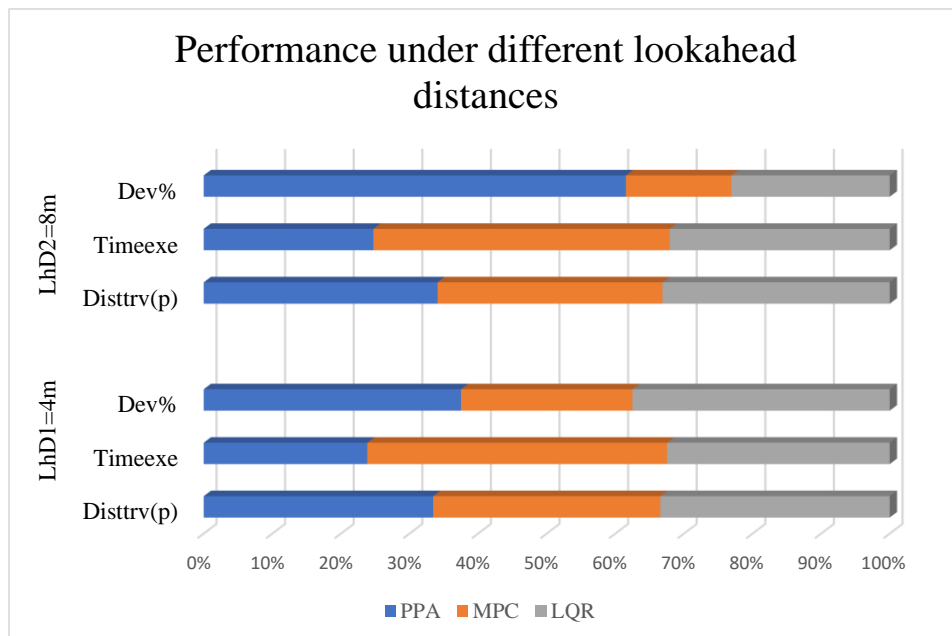


Figure 2.20 Performance chart at different lookahead distance

As it can be observed from the graph Figure that  $Dist_{trv}$  is higher (so the deviation also) for PPA algorithm when the velocity is increased but time of execution  $Time_{exe}$  has remained low compared to LQR and MPC algorithms (observed in Figure 2.7). The

lookahead parameter is an important PPA settings that highly impact the performance of PPA algorithm. With change in lookahead distance (LhD) the deviation Dev% increase in significant and begins to impact the overall travel efficiency of the robot (seen in graph Figure 2.8). Performance beyond 10% of deviation can impact the overall energy efficiency of robot significantly. Hence a proper selection of lookahead distance is crucial for performance of PPA based navigation. The lookahead distance adjustment depends on the size of field and the size of robot.

### **2.13 Conclusion**

In an agricultural setting, navigation costs. All agricultural employees are concerned about this issue. WMRs with advanced mapping and localization system are expensive . The differential drive robot is the most employed robot for agriculture operations due to its simplicity of kinematic and controllability. In real scenario of agriculture, the three wheel is less used, but the 4-wheel differential drive is more used. Even in 4W-wheel design of mobile robots, the 2-wheel differential drive with 2 castor wheels placed at the rear is often used. The kinematic modeling of the mobile robot remains the same as castor wheel as free omnidirectional wheel. Unless the rear castor wheels are stopped by obstruction, the capacity of the robot remain the same. Hence study on differential drive robot is best and popular option for agricultural vehicles. The cost of agricultural robots will inevitably increase if robot navigation technology, which has been used in other industries, is brought straight to agricultural engineering. In the meantime, the real production will have an impact on the output and benefit value. Therefore it is important to implement cutting-edge technologies to the field of agricultural robots at a reasonable cost and also meet the public's need for agricultural robots with precise navigation. Although the PPA algorithm struggles with appropriately calibrating the lookahead distance, it can be used in agricultural applications where the path's nature is stable. It is possible to empirically determine the look-ahead distance that works best for general vegetable fields like object or potato plantations. Since agricultural robots are slow-moving machines, the PPA algorithm in combination with the WMRs' kinematic model can give useful results. High performance and high-speed vehicles require mathematical dynamic modelling



of the vehicles. There is some gain in performances MPC and LQR controller but PPA requires very less mathematical computation for its geometrical based system. The experiment conducted was on ideal conditions of road traction, the other tracking algorithms can suffer from unpredictable non linearities caused by slippage and minute obstructions in real scenario. Performance can also be effected by minute changes in the robot model, due replacement of some parts like chnaged tyre size, chassis. PPA algorithm being the “reach to goal “strategy based on kinematic is lesser effected by parameter variations of model due as it more position oriented than others. However proper positions sensors and obstacle avoidance algorithms must assist the PPA algorithm.

## **CHAPTER-3**

### **Study on LIDAR sensor and Obstacle Avoidance Algorithms for Mobile Agricultural Robot**

### **3.1 Introduction**

In order to navigate about the environment and avoid crashes while following a path or reaching a goal, mobile robot navigation requires obstacle avoidance method. The path planning techniques use such algorithms with localization, and the waypoints area map to determine and follow the optimum path to the target while avoiding obstacles. An understanding of different type of sensors and their suitability for appropriate type of environment and machines will be explored in this chapter. The study comparison of sensors in terms of efficiency and cost is explored for such sensors utilized in autonomous navigation. The several stages of localization and mapping is discussed. The different variety of techniques uses for SLAM technologies are explained. The complexity of such algorithms are explored and compared with simpler strategies of autonomous navigation methods. The different types of obstacle avoidance algorithm are compared and tried with an experiment and results are discussed.

### **3.2 Sensor Technologies for Agricultural Robots**

A wide range of tasks need the use of sensors in agricultural operations, including localization and mapping, navigation-guidance, obstacle and plant detection, their recognition, and environmental data monitoring (including soil, water, air and plants). They help with decision-making operations, task activation or execution, and performance assessment of the robot. According to the data they produce, sensors can be categorized into the following groups: Motion measurement methods include odometry, millimeter-wave radar scanning, laser radar systems, inertial or artificial landmarks recognition, sonar and machine vision [52]. Environmental parameters can be measured using a variety of sensors , including infrared (IR) sensors, X-ray devices, acoustic, optical, 3-D and 2-D vision systems [53]. Separating sensors depending on where they are or the data they are collecting, they can be internal and external sensors, is another option. Numerous distinct system components are monitored internally by sensors, such as encoders that report joint angles and wheel velocities, IMU accelerometers can measure linear accelerations and gyroscopes that monitor circular accelerations. Dead reckoning typically uses mentioned sensors. In order to circumvent the challenge of modelling odometer inaccuracy, inertial sensors

have now been utilized in a variety of automotive applications as a substitute to odometry data. Additionally, these sensors can be enclosed to create a sturdy packaging [52]. Internal sensors, however, are prone to drift and accumulate inaccuracies as a result of variations in climate, gravity, wheel slippage, exposure to close magnetic fields or magnetic materials. These sources of error can quickly result in significant positional drift if inertial sensor data are combined to provide position and orientation. The external sensors utilized in agricultural robots and automated systems use IR, GPS, machine vision, laser scanners (LIDAR), ultrasonic-waves systems [52], hyperspectral (Zhao et al., 2016), and environmental parameters to gather environmental information data about the state of the robot in relation to the position of the mobile robot and also the local positioning for its different components [54]. The most successful external sensors in terms of commerce for autonomous passenger cars are machine vision and GPS sensors [55]. Navigation and direction are provided by GPS sensors, which provide the system with pure positioning. Real-time kinematic RTK-GPS and differential GPS (D-GPS) are two highly developed and precise GPS technologies that enable precise real time measurements. Their high price is one of the key factor driving up the price of robotic systems for early years but which is falling due to popularity of such technology , and according to [56], they are costly to be widely utilized in farm machinery. [57] showed that RTK- GPS receivers provide incredibly accurate findings when used in broad fields. RTK-GPS was utilized by [58] to guide mobile robots. Their employment as the primary sensor for agricultural vehicle navigation systems in steering control is justified by their high level of accuracy [59]. However, since its receiver is typically situated beneath any tree canopy, which either blocks the satellite signals due to error accumulation from numerous reflections on obstacles or tree canopy, the GPS is not as reliable in covered regions like greenhouse orchards as it is in open fields [59]. Due to the construction of the greenhouse, location errors are also produced there. The GPS signal is returned in various directions. The combination of a GPS with internal sensors used for navigation decreases the mistakes collected by each and serves as a reference since the errors of a GPS and dead-reckoning sensors are often complementary to each other [9].

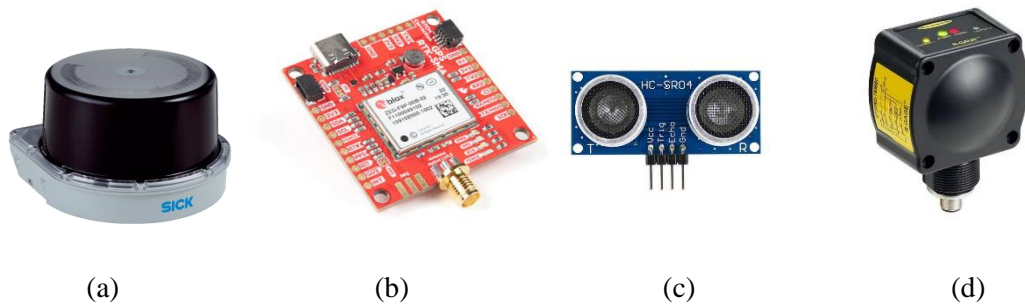


Figure 3.1(a) LIDAR Sensor 3D LiDAR sensors MRS1000, (b)Sparkfun ZF9P RTK-GPS, (c)SR04 Ultrasonic sensor, (d)Frequency Modulated QT 50 RT Radar Sensor

### 3.3 LIDAR Sensors for Agricultural Robots

LIDAR stands for Light detection and ranging sensor. Global rules regulating the use of chemical products in agriculture are becoming more stringent. For instance, the Ecophyto 2018 programme in France seeks to significantly cut back on the use of phytosanitary agents. As a result, in order to preserve production rates, several farming operations that were made easier (though still not simple) by the use of herbicides, require other alternatives. The French company Nao Technologies<sup>1</sup> created the Oz robot, an unsupervised weeding robot, in answer to this demand. This robot has a LiDAR sensor that it uses to identify the crops, enabling it to navigate the field on its own without destroying the product. The robot provided by its company requires some previous knowledge about the breadth, length and numbers of the field crop rows in order to perform an optimal autonomous navigation. This means that the precision of such in forms directly affects the navigation behavior. This work's goal is to offer a brand new autonomous navigation type of algorithm that doesn't need any prior field knowledge. Although researchers were hoping for mobile robots navigation, vision based methods produced the initial findings [60]. However, as noted in [61], the camera data are subject to lighting and climatic influences, which can affect the methodology robustness. A technology using GPS3-based navigation was considered as an alternative strategy [62]. However, without increased precision, like RTK4-GPS, conventional sensors do not have enough accuracy for navigation. Furthermore, RTK-GPS are designed for an Oz robot price system can be pricey. Although weakly responsive to outdoor sunlight, the LIDAR based technique seems

to be a cheap option, which is why many commercial robots are taking it into consideration like the Oz robot and the new French robot PUMAgri5. As previously indicated, present robots rely on the precision of those in forms and require other information about the crops (size, length, etc.) in top of the sensor data. In other words, the focus of this research is processing LIDAR data to suggest a reliable autonomous navigation system not requiring those prior details. The robot needs to find the crop rows in order to travel on its own in the field. This can be framed as a pattern fitting issue: given a set of data (LiDAR measurements), we must be able to identify the set of straight lines (the rows) dividing the data into distinct clusters (Fig. 1). An intriguing LIDAR assisted autonomous navigation algorithm is reported in [61]. This method's fundamental doesn't require is the necessitates to calibrate the algorithm. This can be related to the need for prior simulation, which we want to keep aside for the robustness reasons. According to Barawid et al., the method being evaluated in the study is based on line detection (the crops) with 2D point cloud (LIDAR readings) [63]. The Hough transform and the RANSAC line fitting algorithm are two well-known methods for line detection. According to [64]-based methods for detecting lines in a 2D point cloud are typically more effective than Hough transform methods.

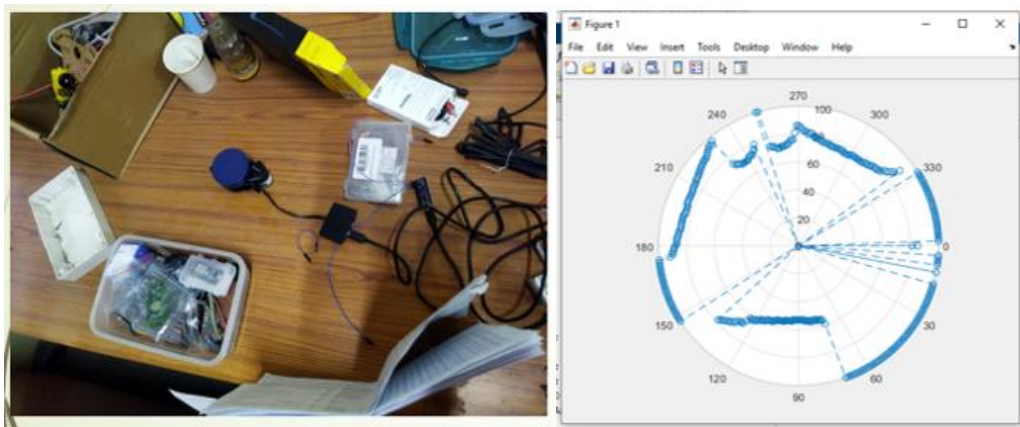


Figure 3.2(a) LIDAR Scanner YD X2 placed with objects around the environment, (b) Scanned data and point cloud plotted in MATLAB software.

A low cost 2D LIDAR mode YD X2 sensor figure 3.2(a),(b) is very popular sensor used by mobile robots. The description of the sensor's details, reading method and math computations involved is expressed below.

Table 3.1 Data format for YD X2 sensor

0		2		4		6		8		10		12		
PH		CT	LS	FSA		LSA		CS		S1		S2		...
LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	...

The above table specifies the byte organization of different data provided continuously by the sensor scanner. Data available are stream of hexadecimal numbers. Decoding of data is done to find angular readings and respective measurements. Correction calculations needed to be included to improve readings. Following table Table 3.2 will explain the description of every byte produced by the scanner.

Table 3.2 Data byte description for YD X2 sensor

Item	Name	Description
PH(2B)	Packet header	The length is 2B, fixed at 0x55AA, with the low position in the front and the high position in the back.
CT(1B)	Packet type	Indicates the type of the current packet; 0x00: Point cloud packet 0x01: Start packet
LSN(1B)	Number of samples	Indicates the number of sampling points contained in the current packet; there is only one starting point in the starting packet, and the value is 1.
FSA(2B)	Starting angle	Angle data corresponding to the first sample point in the sampled data
LSA(2B)	End angle	Angle data corresponding to the last sample point in the sampled data
CS(2B)	Check code	The current data packet check code, using double-byte XOR to verify the current data packet
Si(2B)	Sampled data	The sampling data of the system test is the distance data of the sampling point, and the interference flag is also integrated in the LSB of the Si node.

The scanner module sends random data frame size depending on its scan time available for every scan operation. The angular range of sampled scan is specified by (equation (3.1)) the range of values denoted by FSA(starting angle) to LSA(ending angle)

$$Angle_{scan} = Angle_{LSA} - Angle_{FSA} \quad (3.1)$$

Angle at a particular angle index  $i$  is given by eqn. (3.2)

$$Angle_i = \frac{Angle_{scan}}{LSN-1} * (i - 1) + Angle_{FSA} \quad (3.2)$$

Due to error caused by reflection of varying distances, the correction at particular index  $i$  is given by eqn(3.3)

$$Angle_{corr} = \tan^{-1} \left( 21.8 * \frac{155.3 - distance_i}{155.3 + distance_i} \right) \quad (3.3)$$

Starting angle is given by eqn. (3.4), last scanning angle is given by eqn. (3.5) The angle at every index  $i$  after the compensation of corrected angles is given by eqn. (3.6)

$$Angle_{FSA} = Angle_{FSA} + Angle_{corr\_FSA} \quad (3.4)$$

$$Angle_{LSA} = Angle_{LSA} + Angle_{corr\_LSA} \quad (3.5)$$

$$Angle_i = Angle_i + Angle_{corr\_i} \quad (3.6)$$

Distance values is given eqn. (3.7) directly by the  $i$ th streaming bytes

$$distance_i = S_i \quad (3.7)$$

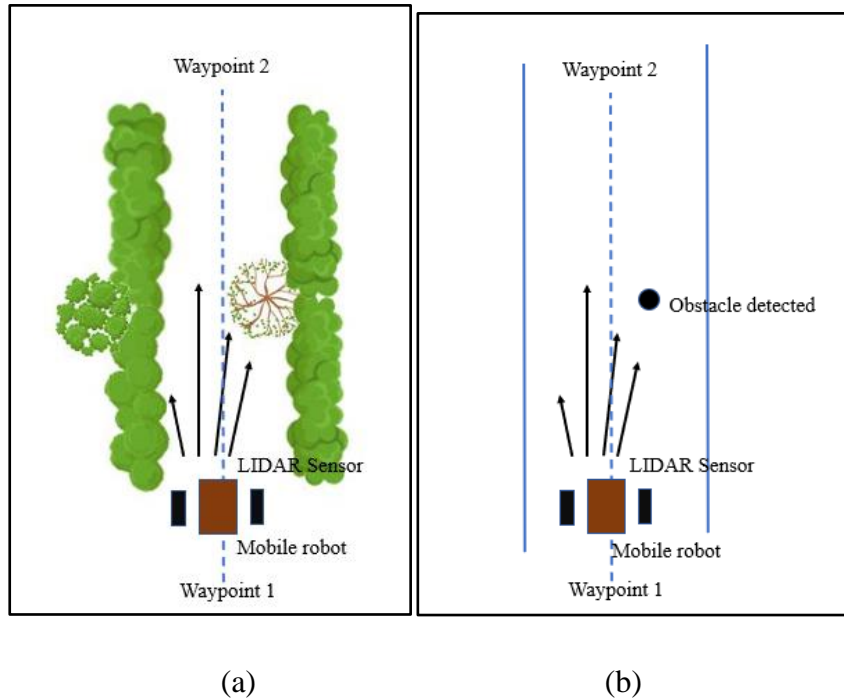


Figure 3.3(a) Mobile robot in field with LIDAR scanner, (b) Obstacle perception of robot

In an agricultural setting, LIDAR technology is more reliable because it is not affected by background illumination conditions. Additionally, the viewing area may exceed the capabilities of a camera. Despite these benefits, Light detection and ranging navigation in agriculture is not widely used, largely because of its high price. Cost savings in recent years have reignited interest in this technology. Using LIDAR, [63] created a real time guiding system for driving an automated vehicle through an farm orchard.



For vehicle navigation, vegetation rows are extracted using the Hough Transform. They said that because the algorithm can only recognize straight lines, curved rows provide a challenge. Another drawback of the approach is when the Hough transform failed to extract the proper plant rows, leading to a loss of steering system. As in the instance of Subramanian et al., LIDAR was also utilized to for obstacle avoidance while navigating. [65] created a 3D LIDAR-based navigation technique more lately, and they utilized a statistical model to identify the planted rows. In order to remove the points that belong to the ground, the LIDAR acquires a 3D point cloud. A statistical model then detects groups of points that correspond to the plants. Although the results are encouraging, because the statistical model is unique to maize plants, it will be difficult to scale the strategy to other plants. The cluster of 3D points that the statistical model is dependent on the size as well as shape of the vegetation. The system also places limitations on the operating circumstances, such as plant size and robot speed, because it is created specifically for plant phenotyping. The lack of resistance to disruptions in the proposed solutions is one of their key drawbacks. Agri-cultural environments have a number of sources of unpredictability and are dynamic and non-deterministic. For instance, the rough terrain and the many plant types' forms, sizes, and colours cause noise. A robot working in such a setting will experience wheel slippage, measurement noise, and controller and actuator noise on top of those problems. Therefore, creating a navigation system that can handle various sources of variance is a difficult challenge. The most promising navigation techniques are the probabilistic ones suggested by [66]. A 2D LIDAR model was presented by them to characterize various environmental noise kinds. For the robot to autonomously navigate through an interior or outdoor urban environment, a particle filter is employed in conjunction with the sensor model. For row following in a maize field, this project attempts to build an autonomous navigation technique for a robot outfitted with a LIDAR. The navigation technique is based on a particle filter algorithm [66], that is used to assess the robot field condition of the system, robot heading position, error of lateral deviation, and spacing between columns of plants and the ends of rows. The robot is steered using these calculated values.

### **3.4 RTK-GPS Technology for Agricultural Robots**

RTK GPS is a GPS, one that uses a terrestrial base station, which already knows its location, to produce GPS correction information and sends them to the vehicle to correct GPS readings in order to gain more precise location values. As discussed earlier, GPS determines the position and estimates the distance using a carrier waveform that has been phase-modulated with P and C/A values. The carrier phase comparison method is used to find the correction value, which is then transmitted via the RTCM data format protocol. In order to define the differential data link for real-time differential calibration of the GNSS mobile receiver, the Radio Technical Commission for Maritime Services (RTCM) was established [67]. The RTK-GPS system transmits the required data for GNSS location correction values using the accepted RTCM protocol. The specifics for two versions may be found in. Standards are mainly divided into version2 and version3 standards. Version 2 is currently seldom ever used, and version 3 is very popular. So, for implementation. RTCM propagation: Telemetry is typically heavily utilized while using RTK-GPS in UAV operations. For more precision, a platform like Q Ground Control must is required which needs to be installed base station before it can be connected to the GCS. The RTCM correction data is then streamed to the drone via the control board by the ground station using the datalink system. Due to its ability to transmit MAV Link information, telemetry is the most can be used type for communication when operating agro-robots. Without installation of the receiver near the rover or the communication equipment, this platform enables the user to receive and use GPS correction data information from the nearest mount, which means a reference station, if the Internet is connected. It uses the Networked Transport of RTCM over Internet Protocol (NTRIP) standard when supplying data [68]. NTRIP Client, its Caster, and the Server make up Network RTK. NTRIP Server transmits continual GPS data to its Caster after receiving data provided by the GPS base station. Numerous studies have been done to deal with the scenario where the aforementioned RTK-GPS system is directly endangered. First, according to research, mounting three receivers successively increases the likelihood that an RTK-GPS receiver will have an uninterrupted GPS signal despite bad environmental factors [69][70]. Even if the received GPS signal was tampered because of the

multipath effect, its three receivers in a line work together and produce accurately computed results. This method has the drawback that it solely concentrates on gathering accurate GPS data for the reception stage and is therefore unable to withstand network latency issues. The solo RTK approach is the next option, and it uses observation data recorded in the receiver for dead places to calculate the RTK-GPS data [70]. Even when the system hits a dead spot, it may still produce continuous RTK-GPS data by utilizing the earlier created GPS readings. But another drawback of this study is that it only looks on the receiver part. To make the entire

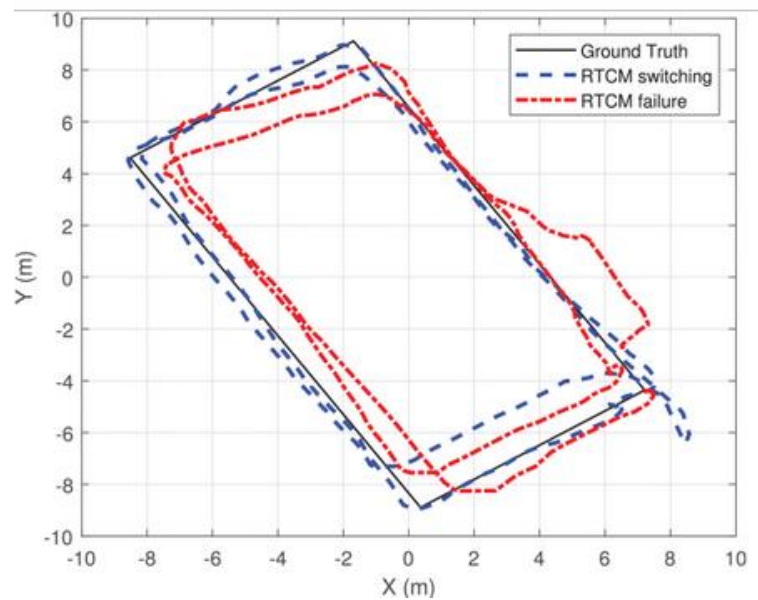


Figure 3.4 Trajectory comparison of RTCM.

RTK system robust, it is crucial to consider and improve not just the receiver end but also the transmitter end. Research that dynamically controls the data transfer rate at the transmission end of RTK-GPS positional data is also available. By managing packet-switching communication while taking environmental factors into account, this method can increase the system's effectiveness and viability from an economic standpoint.

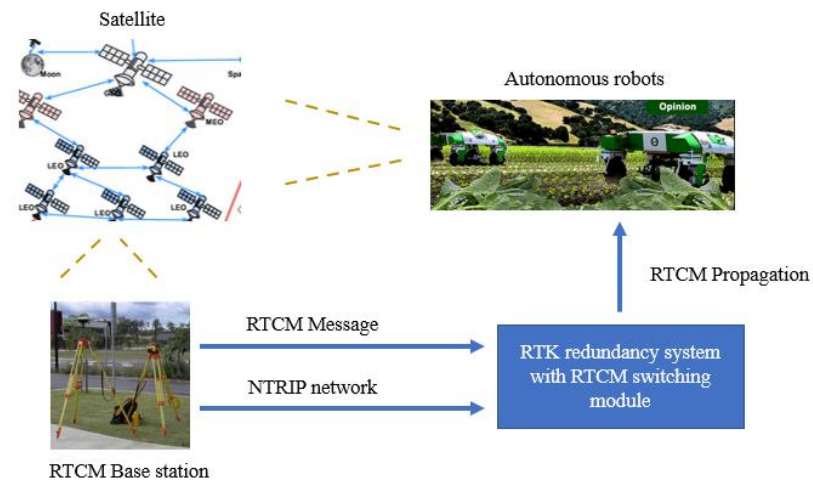


Figure 3.5 RTCM propagation system

This system can successfully transmit RTK-GPS signals in a dangerous environment by adapting the data rate in a flexible way. In contrast to earlier research, it increases the stability of the RTK-GPS system during the transmission process. However, it has a drawback in that it does not offer a system solution when the RTK-GPS data is not entirely received. As a result, there are gaps in the available research, which is left for future studies.

### 3.5 Comparative Study on Sensor Technology for Localization of Agricultural Robot

Lateral navigation (LNAV), The most popular location sensors for AVs are vision sensors, smooth navigation , X-ray pulsar-based navigation , global positioning system (GPS), differential GPS (DGPS), GNSS and real-time kinematic-GPS (RTK-GPS). Such sensors have varying accuracy, intervals, and methodologies. In order to operate the entire system using the triangulation concept, this system is made of single system which is uploaded onto a robotic platform and two static image sensors as subsystems [71]. The system uses a specified distance ( $L$ ) between two subsystems and two angles measured from the base line to a visual marker mounted on the robot to establish the robot's position. Two fluorescent bulbs with red and white radii have been utilized as

the optical marker put on the robot since the positioning algorithm detects it based on chrominance. Each subsystem is made up of a computer, a stepper motor, a CCD camera, and a rotary encoder. The rotary encoder as well as the CCD camera, respectively, detect the deflection angle of the visual marker in an image (HD) and the angulation of the CCD camera, which together measure the angle to the visual marker on the robot. The controlling system uses an image analysis to determine the position of the robot using two different angles from the sub-systems, and it transmits the location data with a radio receiver (for example, a wireless modem) [71]. Power cables were installed all around the field for the LNAV, an off-the-wire electromagnetic induction system [71]. The cables that were put around the field formed a magnetic field, and this navigation system was created by Kubota Co. With increasing distance from each cable, the resultant magnetic fields become significantly deeper. Two magnetic field sensors were utilized on either side of the vehicle for this system. The robot has to be manually taught along the field boundary in order to use this positioning technique. This method might work well in a variety of weather conditions, but big fields can significantly raise the cost of building.

The Japan Aviation Electronics Industries designed SNAV utilized DGPS, a Transcranial Magnetic Sensor (TMS), and an inertial measurement unit (IMU) (Fig. 10b) [72]. To enhance the interval of positioning system, the IMU and TMS were utilized. The expense, information service, and reference station were the main drawbacks to using SNAV, even though its accuracy was adequate. Sanyo Electric Co. produced XNAV, which was created by BRAIN-IAM. However, this positioning sensor made use of an optical measurement system, despite the fact that XNAV typically resembles an image sensor (Fig. 10c) [72]. Its reference station was used to examine the mounted target on the vehicle robot (Fig. 10 (9)), from which the diagonal length ( $L$ ) and horizontal angle could then be determined. This led to the objective being coordinated. Four crucial positioning systems are used today: GNSS, GPS, DGPS, and RTK-GPS. All share a common foundation in the global satellite navigation system (GNSS), but each differs slightly in approach, accuracy, topography, and application. GALELEO Several GNSSs, notably QZSS (Japan), NAV- STAR (US), IRNSS (India), BeiDou- 3 (China), GALELEO and GLONASS

(Russia) are currently in use (Europe). The GPS provides accuracy to within 3 m by using three satellites for attitude and longitude measurement and other satellite for height. A GNSS antenna is carried by the mobile unit (rover) to pick up GNSS signals. The base station used by DGPS to improve positional accuracy has a known positional coordination. Accuracy of 10 cm is possible with DGPS. RTK-GPS may be the most suitable choice because some agricultural applications need a minimum precision of 5 cm. Using the positional coordination of a base station, which is transmitted by a transmitting antenna, RTK-GPS corrects the location of the readings. Several businesses currently offer both antennas in a single box. table 3.3 a few positional sensors that are frequently employed in vehicle robotics labs, along with information about their accuracy and sampling rates. Table 3.3 discusses the various remote sensing positioning sensors with comparative price and accuracy.

Table 3.3 comparison of RTK-GPS sensor with other localizing sensors

Sensor Type	Brand	Model	Accuracy(cm)	Frequency(Hz)	Price range
SNAVsystem	JA EI	Prototype	10	2	Depends on Prototype
LNAV system	Kubota	Prototype	5	10	Depends on Prototype
Visual sensors	Prototype	Prototype	30	0.8	Depends on Prototype
RTK GPS module	Sparkfun	ZF 9P	10	20	\$0.4k-\$4k
XNAVsystem	Topcon	AP-L1	1-5	2	\$5k-\$6k
RTK GPS module	Topcon	Legacy-E	1	10	\$5k-\$16k
RTK GPS module	Topcon	AGI-3	2	10	\$5k-\$30k
RTK GPS module	Trimble	SPS855	0.8	20	\$5k-\$15k
Differential GPS	Hemisphere	V100	60	0.05	\$3k-\$8k

For military UAVs, laser sensors are mostly utilized in autonomous navigation (laser gyroscope and laser homing) due to its high accuracy in determining distance, good directional cues, and potent anti-jamming properties. They are rarely seen in agricultural devices because they require optical systems, which renders them inappropriate for situations with high humidity, severe light pollution, dust, or smoke [73]. A passive detection method with high interference resilient capabilities and effective concealment is infrared sensing technology. Even in poor weather or at night, it can estimate distances and define contours [74][75]. However, because to the system's short detecting range and the ease with which the surroundings might interfere with its light output, Structured light is emitted from the laser and, after passing through various lens structures, is converged light rays of various forms.

Table 3.4 Characteristics of obstacle sensors

Sensor Type	Max Range(m)	Merits	Demerits
RTK GPS	---	high accuracy	does not work in shelter
Ultrasonic sensor	range<20m	Low price	not reliable for close distance readings, suffer from blind acoustic spots, effected by environment
Laser/Infrared sensor	range<50m	reliable due to high resolution	required spinning mechanism, single point data is unreliable for small objects, new solid state sensor is expensive
TOF	range<10m	highly reliable	low resolution and have environmental interference
Structured light sensor	range<10m	high resolution., reliablity more than stereo vision	closely placed light sensors interfere with each other, effected by outdoor natural light
Radar system, Millimeter wave	range<250m	immune to weather conditions like heavy rain, dense fog. etc	low resolution with very high cost, limited access
Binocular(stereo) vision	range<100m	high resolution	needing proper lighting conditions, high computation required
Monocular vision	range<10m	low cost system with simple architecture, lesser computing resources than laser and binocular system	large sample database required

Line oriented light is the name given to the linear light band. The structured light sensor is made up of a laser and different structure-specific lenses [76]. The line rays optical sensor is especially suitable for robot measurement and control tasks in complicated environments because of its advantages in interference immunity capability, high precision and real-time capability. Table 3.5 compares various prototypes used for different purpose of operations, with their existing prototypes/products available globally. One of the popular ranging techniques is time of flight (TOF) ranging. It is a two-way ranging technique that gauges the separation

by timing how long it takes the measuring signal to travel between the stations. It is better suited for situations where great ranging precision is required since it uses less energy and is simpler to install than other ranging techniques. The propagation speed of the signal, which is typically electromagnetic wave signal, is almost that of light. A detecting radar that operates in the microwave range is millimeter-wavelength radar. The range of its frequency is 30 to 300 GHz.

Table 3.5 Suitability and Application of obstacle sensor

Sensor Type	Suitability for obstacle avoidance	Applied agricultural prototypes/drones
RTK GPS	on-site calibration, suitable for localisation of obstacles maps, not suitable for fast response Obstacle Avoidance	Hanche: Venus-1, AYQF: 3WQFTX-10 DJI: T16, GKXN: M23-E S40-E, XAG: P20/30 2018 P series 2019
Ultrasonic sensor	low resolution and suitable for short to mid-range operations	XAG:P20 2017
Laser/Infrared sensor	can only produce discrete information, suitable for short-distance obstacle avoidance	DJI(consumer UAVs):Inspire 2, Phantom 4Pro Hanche:CD-15,Mavic 2,
TOF	short sensing range hence suited for OA auxiliary device	---
Structured light sensor	only suitable for indoor OA with proper lighting conditions	
Radar system, Millimeter wave	more applied to agricultural UAV machines, the OA systems is cost-ineffective	XAG:P series 2018/2019, TXA:R-16, DJI: MG-1S GKXN: S40-E
Binocular(stereo) vision	suitable for most OA problem but can fail under optical illusional conditions	XAG:P20 2017, DJI:consumer UAVs, P20/30 2018 P series 2019,Hanche: CD-15, Mercury-1,
Monocular vision	has no proper depth knowledge, hence suitable mostly for static or 1D moving objects.	DJI: Mavic2

Millimeter length of wavelength radars can operate in all climate conditions due to their powerful penetrating ability, longer operating distance, dependable detection, and electromagnetic interference immunity. In contrast, due to the complex operating environment of farmland, ultrasonic and other sensors based on optical principles are easily influenced by climatic conditions [77]. Binocular stereo vision technology obtains the parallax result by binocular picture acquisition from multiple cameras, stereo matching, and other procedures, drawing on the capacity of human eyes to comprehend 3D space. The 3D information of the space scene is then recreated using the depth information of the objects. The stereo vision technology can identify and



gauge the distance between the fuselage and the obstacles because to its effective camouflage, ability to gather complete information (including color and pattern of the obstructions), and 3D depth knowledge of the scene [78].

### **3.6 Sensor Fusion for Mobile Robots**

This section focuses on data fusion and several often-used terms in the literature that occur at the perception layer. Sensor data fusion refers to processing steps that take the sensor data from multiple separate sensors and combine the information from these sensors to produce a common and unquestionably superior result compared to what the processing of each individual sensor could produce. Synergy refers to the ability of sensor data fusion to produce better or more appropriate findings than single sensor processing. Beginning with the collecting of sensor data, Figure 9 depicts the multisensory processing procedure. The sensors processing then starts to examine the sensor data, breaking it down into several jobs like calibration, feature Extraction, and object detection, etc. This process ultimately provides the application with a more or less comprehensive picture of the environment. The sensors should be calibrated individually in both time and space prior to the sensors acquisition process. Consequently, a multisensory system must be synced, or the data obtained must be time-aligned, as compared to a single sensor. The sensors must then be "space-aligned," which calls for the mathematical determination of displacements between various (spatial) sensor coordinate systems. provides a non-exhaustive illustrative overview of the tasks required for processing sensor data fusion. There are steps for following the level of abstraction in multisensory.

For instance, low level processing tasks include calibration, temporal alignment, and reconstruction, but high-level tasks include "objects identification" and obstacles categorization. According to certain viewpoints, these categories include early and late sensor data fusion.

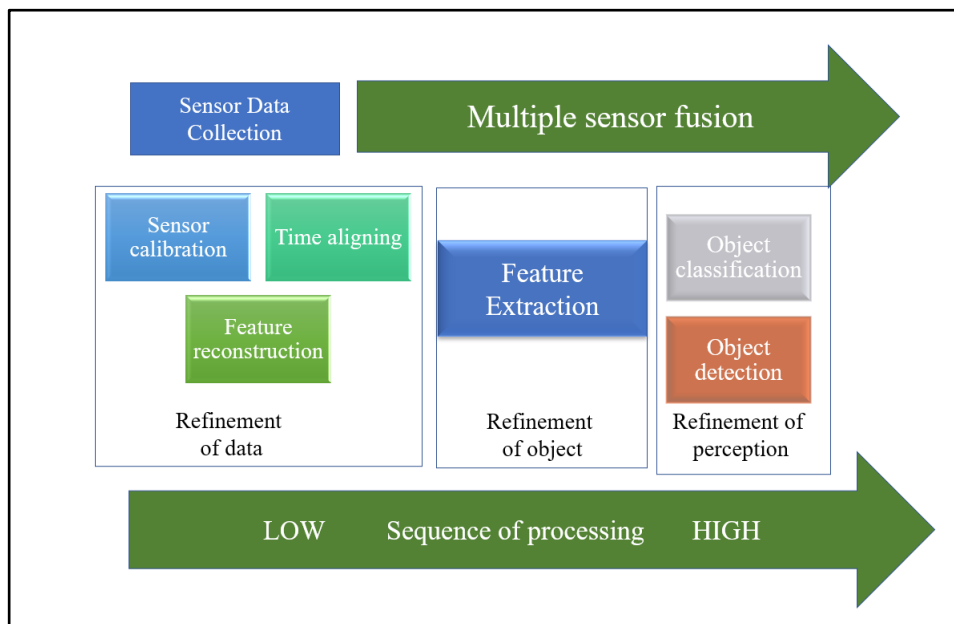


Figure 3.6 Block diagram of sensor fusion stages

**3.6.1 Sensor Refinement:** This stage addresses the three objectives of Calibration, time Alignment and reconstruction of Sensor Models. They all try to use the most accurate sensor observations. The lowest, or initial, processing steps make up most of the sensor refinement. As a result, it also creates the interface between the sensor and the sensor processing system. Broadly speaking, low level processing jobs have not been examined in as much detail as processing steps that are involved in object and scenario improvement due to their inherent complexity. These low tier jobs really have to deal with a massive amount of raw sensor data relative to the refinement level of the item and the circumstance. The raw sensor data may also seem radically different due to the very varied imaging, range, and other sensor features.

**3.6.2 Calibration and Time Alignment** is the process of figuring out how a measuring device's output (in this case, many sensors) relates to the value of the quantity or attribute supplied according to a standard measurement. A separate issue arises with time alignment. Multiple sensor systems, in general, are not synchronized, meaning that each sensor acquires its measurement at a different and distinct time. In reality, it would be challenging or pointless to develop a multisensory processing system and a view of the environment based on readings from various sensors acquired

at various times since they are likely to reveal incongruous environmental conditions. The sensor models and a standard way to interpret and describe raw data are two further groups of issues to deal with the sensor refinement level. In automotive applications, the phrase sensor model refers to a model of the sensor's data and how that data relates to the local environment, encompassing both a model of the sensor data and a model of the surrounding area of the vehicle. For some types of sensors, like RADAR sensors, this is often done separately; nevertheless, a multiple sensor system is not typically considered in its entirety. A single model that can describe all the many forms of sensor data appearing in the field of preventative safety applications on these low levels such as laser, laser, pictures, has not yet been discovered. The goal of the object refinement level is to interpret the numerous sensor observations which are actually subsets of the sensors' raw and weakly preprocessed data, as distinct objects or as elements of the environment seen (e. g. pedestrians, other vehicles, buildings, guard rails, trees, etc.). As seen in Figure 10, in during object refining process, "observations" from various sensors, which are depicted by ellipses of various colors, are recognized as objects (such as other cars or people).

**3.6.3 Extraction of Features** might be thought of as the process of determining features or attention-grabbing characteristics, such segments of sensor data related to the same item in the environment, for example. Segmentation, contour extraction, etc. are some concepts or methods that are fundamentally employed in feature extraction. The sample below illustrates feature extraction by segmentation in LIDAR sensor data: We must first execute clustering before segmenting the environment. The proximity between each pair of successive points in the scan is considered when dividing the data, such as those from a LIDAR, into sets of neighbor points. Therefore, a cluster is a group of measurements (scan points) that are sufficiently close to one another and likely represent the same item because of their proximity (Mendes et al., 2004).

**3.6.4 Object Recognition and Filtering Methods** typically approximation algorithms yield the values of specific physical entity properties or features that are especially crucial for the implementation under consideration (R. Goodman et al., 1997). For instance, estimates might be produced for the performance parameters

(such as the location and the relative velocity) of the objects seen outside the host vehicle in automotive safety and driver aid systems. The use of several time-discrete estimations of the same items found at various times is beneficial in order to enhance the prediction of the objects' characteristics or states (from objects observed surrounding the vehicle). The best case scenario would be to represent how the characteristics of the items change over time. The sequence estimation issue is a filter problem in essence. The Kalman filter and its expanded variant, the extended Kalman filter are two of the most well-known filters used in the automobile industry to address these issues (see Kalman, 1960) and (Welch & Bishop, 2001) for more).

### **3.7. Extended Kalman Filter**

Kalman filter dynamics is the end outcome of several cycles of filtering and prediction. In the context of Gaussian probability density functions, the dynamics of these cycles are calculated and understood. The Kalman filter dynamics condition is satisfied to a steady-state filter and the steady state gain is computed under additional requirements on the system dynamics. It introduces the innovation process connected to the filter, which stands for the unique information sent to the prediction by the most recent system measurement. The faulty ellipsoids connected to the included Gaussian pdf are used to analyze the filter dynamics. The conditional quantile functions that provide the minimal mean-square estimate are no longer Gaussian when the dynamics of the system state or the dynamics of the observations are neither linear nor monotonic. This non Gaussian method projection and mean evaluation using the best nonlinear filter entails a significant processing cost. The Extended Kalman filter is a suboptimal method for solving the issue within the context of linear filters (EKF). The initial nonlinear filter dynamics surrounding the earlier state estimations are linearized by the EKF, which then applies a Kalman filter for the system dynamics.

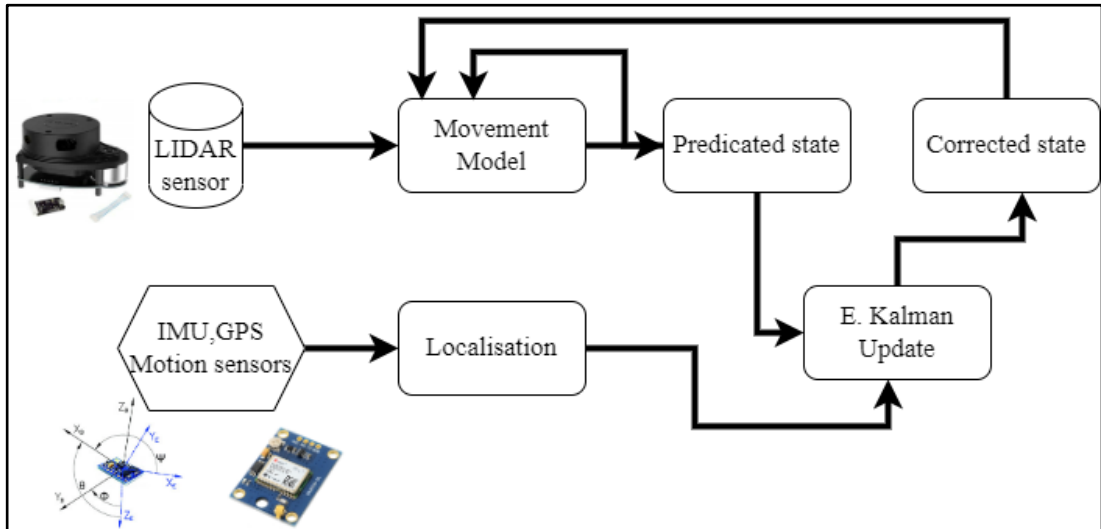


Figure 3.7 EKF sensor fusion model for autonomous vehicle.

Kalman extension filter Because the observation equation is non-linear, estimating fusion employing range, range-rate, and Cartesian measurements is a nonlinear dynamic state estimation issue. Therefore, the Kalman filter is incorrect. The typical method is to employ an analogous observation matrix in the regular Kalman filter equations after approximating by a series expansion:

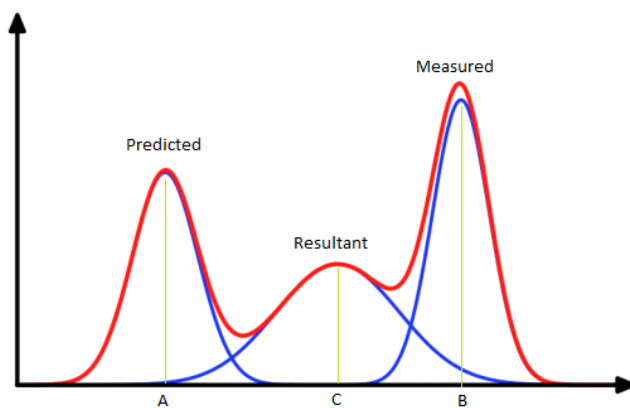


Figure 3.8 EKF probability distribution

For a first-order series expansion (linearization) of the quasi-measurement equation, an extended Kalman filter (EKF) is used. The relative estimated state projection is evaluated using the recursive equations of the EKF, which are shown below. calculates

the covariance matrix, which is supplied by the comparative state vector and its related covariance matrix. State equation of prediction is given by eqn. (3.8),

$$P_{k+1/k} = F_k P_k F_k^t - G_k Q_k G_k^t \quad (3.8)$$

And the Kalman gain is given by eqn. (3.9),

$$K_{k+1}^m = P_{\frac{k+1}{k}} (H_{k+1}^m)^t \left[ H_{k+1}^m P_{\frac{k+1}{k}} (H_{k+1}^m)^t + R_{k+1}^m \right]^{-1} \quad (3.9)$$

Updated state after fusion is given by eqn. (3.10)

$$X_{k+1}^m = X_{k+1/k}^m + K_{k+1}^m (z_{k+1} - h^m(X_{k+1/k}^m)) \quad (3.10)$$

### 3.8 Obstacle Avoidance Algorithms and Techniques

#### 3.8.1 Potential Field Algorithm

[79] has suggested a robot may be subject to fictitious forces. In this technique, the goal exerts an attractive pull on the robot while barriers repel it. For a specific robot position, a resultant force vector  $R$  is calculated, consisting of the total of a target-directed attractive force and repelling forces of obstacles. The method is then repeated while using  $R$  as the robot's acceleration force. The algorithm calculates the robot's new position for a specified amount of time. [80] has improved this idea further by considering the robot's speed close to obstacles. It provided a mixed methodology for local and global path planning that uses a potential Field approach. [81] has applied the potential field method to off-line path planning. By fusing distinct obstacle functions with logical procedures. These approaches share the presumption of a predetermined and known model of the world, in which impediments are represented by straightforward, predefined geometric shapes, and the robot's path is generated offline. All of the aforementioned techniques have indeed been improved upon, but none have been applied to a mobile robot with actual sensory data. In contrast, [82] used experimental robotic system that are outfitted with a circle of ultrasonic sensors to implement a potential field technique. Each ultrasonic range reading is

treated as a repulsive vector field in Brooks' approach. The robot pauses, turns into the direction of the resulting force vector, and continues moving if the combined strength of the repelling forces is greater than a predetermined threshold. However, with this method, only one set of range readings is considered, and earlier readings are ignored.

### 3.8.2 Virtual Force Field Algorithm

For the purpose of representing obstacles, the VFF approach employs a two-dimensional Cartesian histogram grid. Similar to CMU's certainty grid concept, the histogram grid's cells  $(i,j)$  each include a certainty value,  $c$ , which expresses the algorithm's level of assurance that an obstruction is present at those points. The construction and upkeep of the histogram grid are different from those of the confidence grid. The method developed by CMU projected a probability profile onto the cells that are impacted by a range reading; this process is computationally demanding and would take a long time to complete if real-time execution on an on-board computer were tried. Contrarily, our approach just adds one cell to the histogram grid for each range reading, resulting in a probability distribution with minimal computing cost.

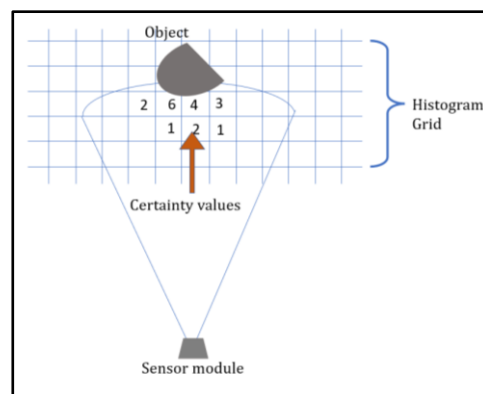


Figure:3.8 Histogram grid probability distribution of grid values

This cell is located on the sensor's acoustic axis and correlates to the recorded given distance for ultrasonic sensors. Although this method may seem oversimplified, a probability distribution is really obtained by swiftly and constantly sampling each sensor as the car moves. As can be seen in this process involves continually incrementing the same cell and its nearby cells. [83] This produces a histogram

probability distribution where high certainty values are found in cells closer to where the obstacle is actually located.

Each active cell pushes the robot away with a simulated repulsive force  $F$ . The size of this  $i,j$  force is inversely proportional to  $d$  and inversely proportional to the certainty value  $C^*$ , where  $d$  is the distance between the cell and the vehicle's center and  $x$  is a positive real number. All virtual repulsive forces are added up at each iteration to produce the final repulsive force  $F$ . A virtual attractive force  $F$  of constant magnitude is simultaneously applied to the vehicle, pulling it towards the direction of the target. The force vector  $R$  is produced by adding  $F$  and  $F$  together. Up to distinct repulsive force vectors  $F$  must be calculated and accumulated in order to calculate  $R$ .  $i,j$  Therefore, a specially created algorithm for the quick computing and summing of the repulsive force vectors serves as the computational core of the VFF method. The VFF technique has been used on a mobile robot equipped with a circle of 24 ultrasonic sensors and has been thoroughly assessed.

### **3.8.3 Vector Field Histogram (VFH) Algorithm**

When the various repulsive forces from the histogram grid cells are summed up to generate the resulting force vector  $F$ , the VFF approach has an intrinsic flaw that results in an extremely harsh feature extraction. The direction and magnitude of the  $F_r$  are reduced from hundreds of data points to just two values in one step. As a result, precise knowledge of the local barrier composition is lost. A new technique called the vector field histogram to address this flaw (VFH). Instead of one step data reduction process employed by the VFF method, the VFH technique uses a two stage approach. As a result, there are three types of data representation:

- a) The most thorough description of the robot's surroundings is kept at the top level. In this level, range data sampled by the on-board range sensors is updated regularly in real-time on the 2-dimensional Cartesian histogram grid  $C$ . The procedure for the VFF approach is also the same.
- b)
- c) A one-dimensional polar histogram  $H$  is built around the vehicle's present position at the middle level.  $H$  is made up of  $n$  angular sectors with specific



width. By mapping the active region  $C^*$  into  $H$  by a transformation, each sector  $k$  now has a value  $h$  which indicates the polar obstacle density in the direction that sector  $k$  is related.

- d) The parameters for the vehicle's drive and steer controllers, which are the VFH algorithm's output, are the lowest level of data format.

Polar Histogram Data reduction is the step where the grid coordinate based forces are converted to a polar form. The angle of the position eqn. (3.10) of grid coordinate  $(i,j)$  with respect to the sensor position is  $\beta$  where  $(x_i, y_i)$  are position of the coordinate and  $(x_0, y_0)$  is the position of sensor

$$\beta_{i,j} = \tan^{-1}(y_i - y_0)/(x_i - x_0) \quad (3.10)$$

The potential magnitude of object  $m$  is given by eqn. (3.9)

$$m_{i,j} = C_{i,j}^2(a - bd_{i,j}) \quad (3.9)$$

where  $a, b$  are position constants,  $C$  is the certainty value,  $d$  as the distance between object and sensor,

- a) The square of  $C_{i,j}$ . This demonstrates our conviction that repeated range readings as opposed to isolated distance readings, which could be the result of noise represent real impediments.
- b)  $m$  is proportionally related to  $-d$ . As a result, when occupied cells are close to the robot, they produce big vector magnitudes; when they are farther away, they produce smaller ones. To be more precise,  $a$  and  $b$  are selected so that  $a - bd = 0$ .  $m$  is equal to 0 for the farthest active cell and increases linearly for cells that are closer.

The polar density is given by eqn. (3.10):

$$h_k = \sum_{i,j} m_{i,j} \quad (3.10)$$

Due to the discrete nature of the histogram grid, the result of mapped values may appear noisy and leading to noisy errors in calculating the steering direction. Therefore, smoothing filter processing is applied to the value of  $h$ , which is defined by eqn. (3.11)

$$h'_k = \frac{h_{k-1} + 2h_{k-1+1} + \dots + lh_k + \dots + 2h_{k-1+1} + h_{k+1}}{2l+1} \quad (3.11)$$

where  $h'_k$  is the smoothed polar obstacle density (POD). Value of  $l$  can be decided as per the smoothness required and computation expendable.

The transformation and the representation can be understood by placing a robot in an obstacle course as in Figure 3.10(a)(b). A robot with a 360-degree scanner sensor is placed in the center. Object A, B, and C are placed around the vehicle. Object A is a small sized object, while B and C are larger. Figure 3.9

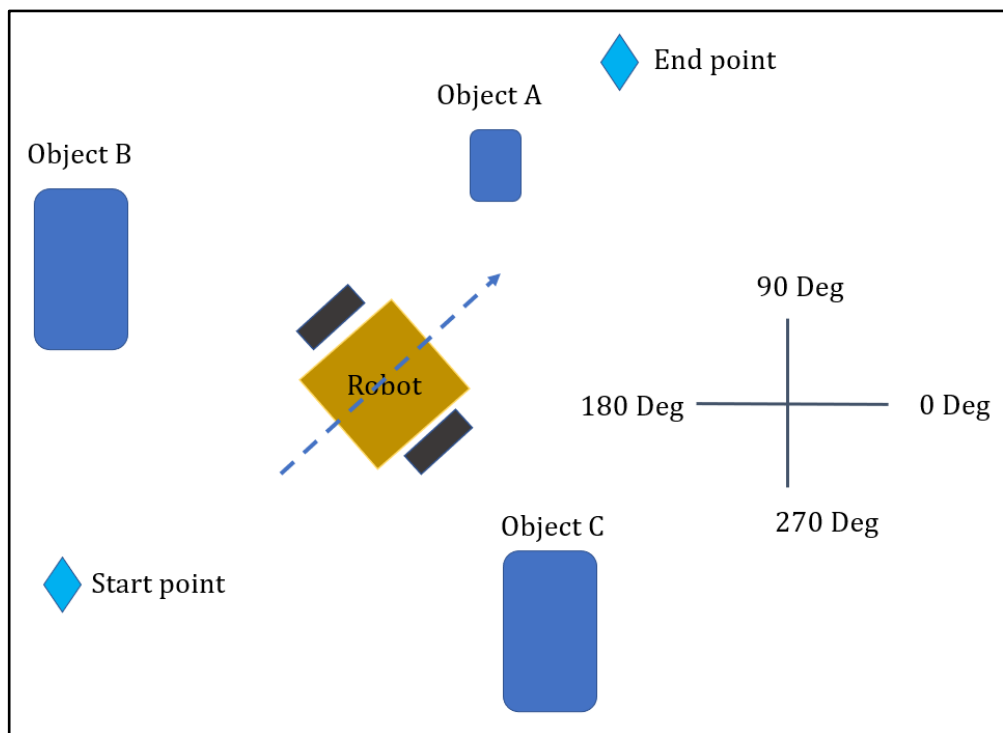
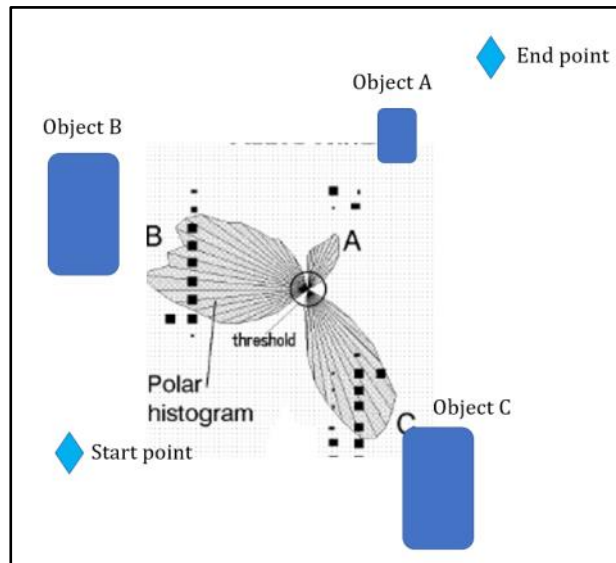
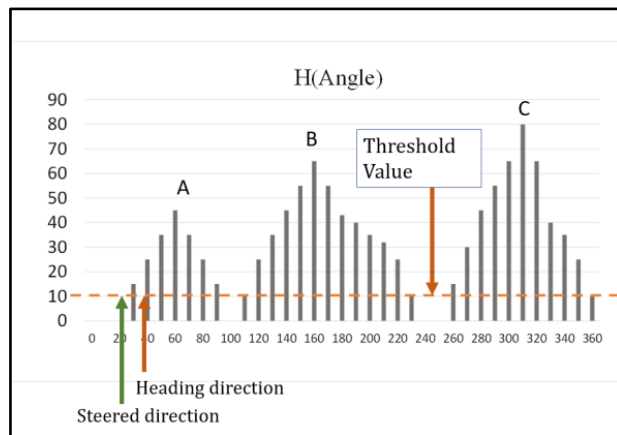


Figure 3.9 Robot in obstacle course



(a)



(b)

Figure 3.10(a) Polar histogram in polar form, (b) Polar histogram (filtered) represented w.r.t robot orientation

The breadth of the robot was not considered by the original VFH approach. Instead, it adjusts for the robot width and smooths the polar histogram using an analogically derived low-pass filter. The primary challenge in putting the original VFH method into reality is the adjustment of this filter [84]. Even for a fine-tuned filter, the robot still has a propensity to scrimp. In comparison, the VFH+ approach accounts for the breadth of the vehicle by using a low pass filter that has been computed theoretically.

The robot radius  $r_r$ , which is the distance from the robot center to its farthest boundary point [85], enlarges obstacle areas on the map.

The certainty cells are literally made larger by a radius of  $r_{r+s}=r_s+ d_s$  where  $d_s$  is the smallest distance between the robot and an obstacle, for further security. The robot can be thought of as a point-like vehicle because the barriers have been expanded by  $r_{r+s}$ . For mobile robots whose shape can be roughly described by a disc, this method works well. The barriers cells must be increased in size in accordance with the measurements and current orientation of the robot if its shape is extremely asymmetrical. By growing the barriers while the principal polar histogram is constructed, this width adjustment approach is much effectively executed. Unlike the previous VFH approach, which updated just one histogram sector for respective cell, this method updates all histogram sectors that correspond to the expanded cell are updated. For each cell the enlargement angle is given by eqn. (3.12):

$$\varphi_{i,j} = \arcsin \left( \frac{r_{s+s}}{d_{i,j}} \right) \quad (3.12)$$

Then the sector value  $h$  is defined by eqn. (3.13),

$$h'_{i,j} = 1 \text{ if } k. \alpha \in [\beta_{i,j} - \varphi_{i,j}, \beta_{i,j} + \varphi_{i,j}] \quad (3.13)$$

Else as given by eqn. (3.14)

$$h'_{i,j} = 0 \quad (3.12)$$

The VFH+ approach determines a collection of potential suitable directions after first locating all openings in the masked polar histogram. These potential trajectories are then subjected to cost function that accounts for more than simply the difference between the candidate and the goal direction. The new direction of motion is then decided upon based on which candidate direction has the lowest cost.

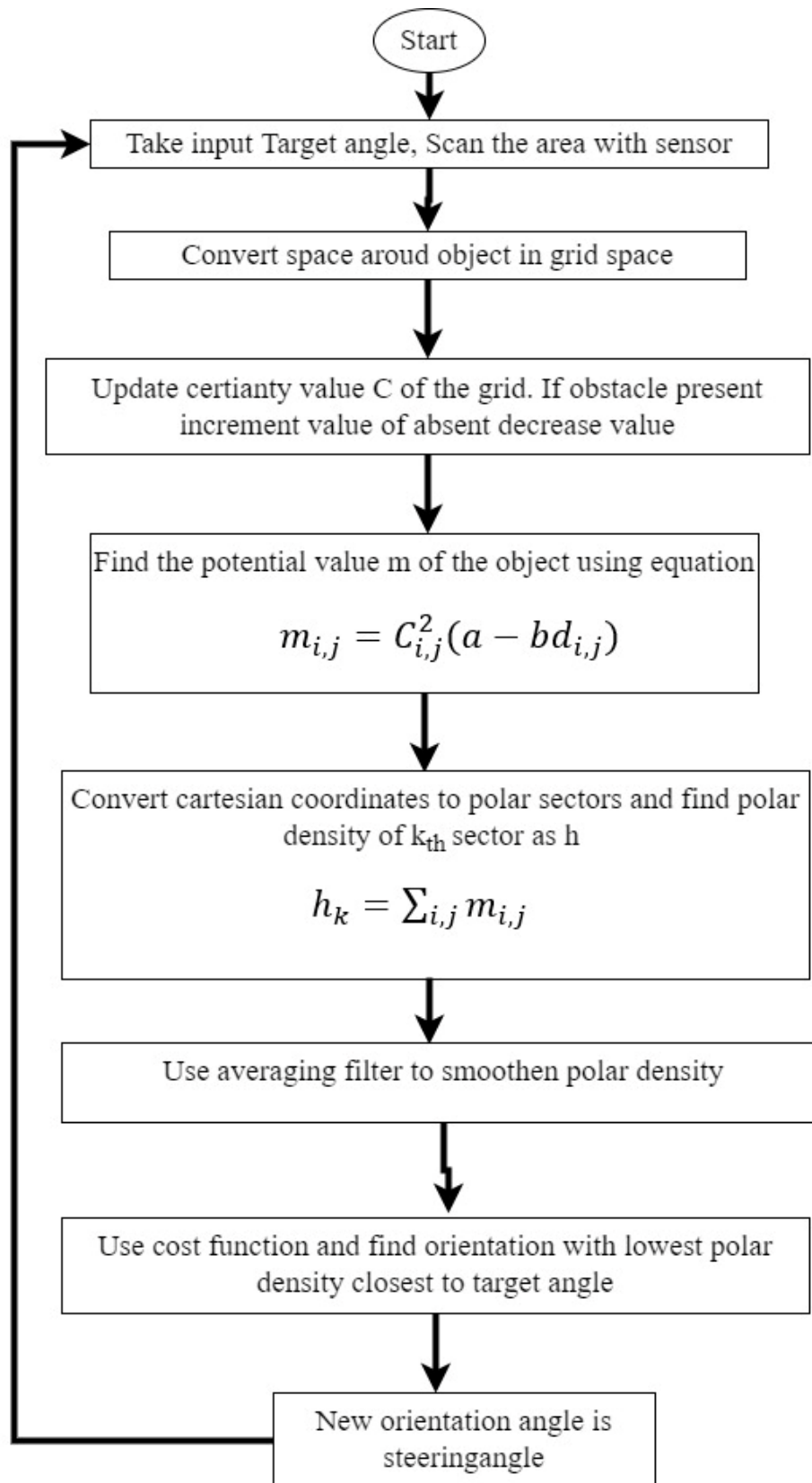


Figure 3.11 Flowchart for VFH Algorithm

### 3.8.4 Bubble Band Technique

This technique, first out by Khatib and Quinlan in [86], establishes a "bubble" containing the greatest amount of open space surrounding the robot that may be traversed without colliding. A simplified representation of the vehicles geometry and the range data supplied by the sensors are used to estimate the shape and size of the bubble. This idea allows for the planning of a path between a starting point and a destination using a group of these bubbles. However, we also include it in this concise presentation due the concept of a bubble, viewed as a subset of empty space surrounding the robot. Obviously, this technique is more of a problem of offline path planning than one of obstacle avoidance.

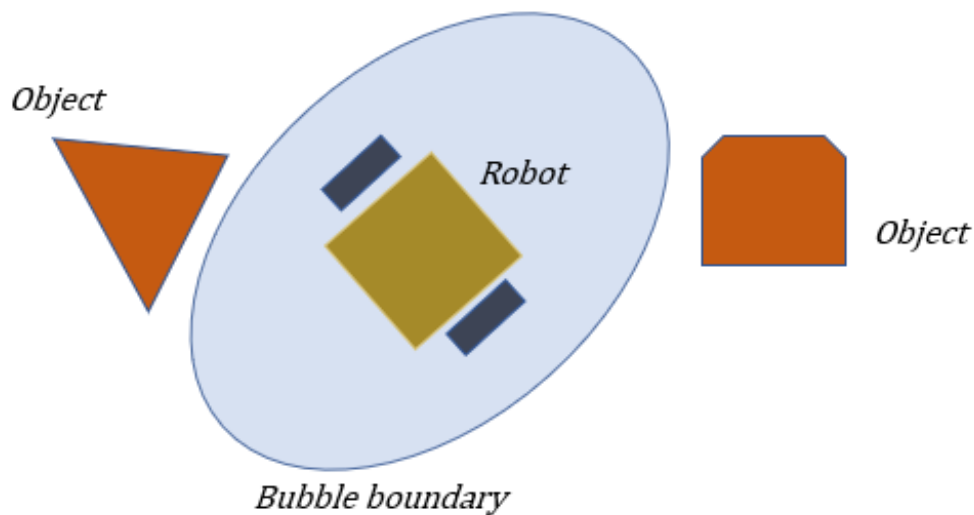
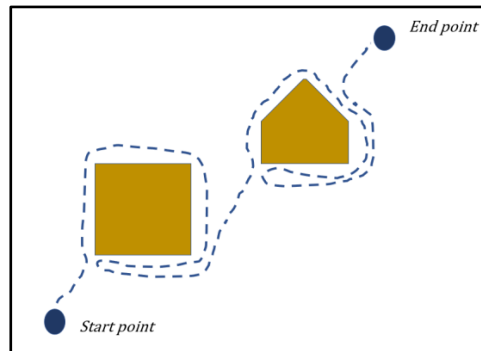


Figure 3.12 Bubble band strategy

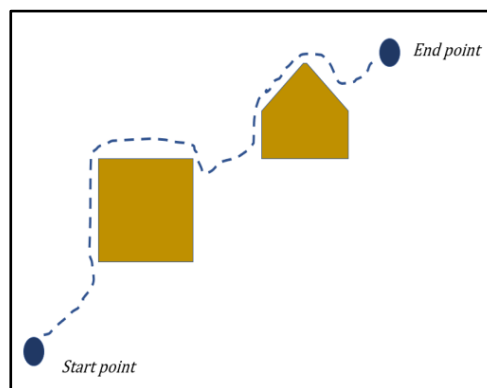
### 3.8.5 BUG Algorithm

The "bug algorithm" is the most basic obstacle avoidance algorithm ever documented. It states that when a barrier is met, the robot entirely circles the item to determine the location with the closest proximity to the objective. From present location, the robot then will exit the barrier as an algorithmic bug. Because of how ineffective this method is, various changes have been suggested [87]. The robot begins off following the obstacle's border in the "bug2" method but it immediately abandons it as it crosses the line between the start point and the objective. Even though the bug-type algorithms

are extremely simple, they come with some significant drawbacks, including the following: (1) they only take into account the most recent sensor readings, which significantly reduces the overall achievement of the robot; (2) they are slow; and (3) they do not take into account the actual kinematics of the robot, which is important with non-holonomic robots.



(a)



(b)

Figure 3.13 (a)BUG1 path strategy, (b) BUG2 path strategy

There are also few of additional intriguing algorithms for avoiding obstacles. However, only a small portion of them is appropriate for real-time, embedded applications. A extension of the fuzzy route following issue discussed in [31] is the integration of fuzzy logic solutions like those reported in [88]

Table 3.6 Comparison of various avoidance algorithm

Algorithm	Merits	Demerits
Bug1	It is complete finite algorithm which finds a	Significant memory and computation can be utilized
Bug2	As being a greedy search algorithm, it first takes what look better. It requires less computation and is easy to implement.	The greedy strategy can fail in some cases exhibiting in completeness.
AFF	A simplistic and easy method to implement	Robot cannot overcome local minima. Oscillations can be issue when tries to handle obstacles in narrow passage.
VFH	It removethe problem of sensor noise. polar histogram provides the probability of presence of obstacle in particular direction	Local minima might not be negotiated if its very less magnitude. It ignores the dynamics of mobile, not suitable for performance vehicle
Bubble band technique	It can outperform VFH when passing through	Output steering provided is not smooth operation. It requires

### 3.9 Localization and Mapping Strategies of Agricultural Robots

SLAM (simultaneous localization and mapping) is a technique for autonomous cars that enables simultaneous map construction and vehicle localization. The vehicle can map out uncharted terrain with the help of SLAM algorithms. The map data is used by engineers to perform activities like routing and obstacle detection and avoidance.



### 3.9.1 Mono SLAM

The foundation of Mono SLAM strategy, as in [11], is a probabilistic feature-based map that captures the position of the single camera, all relevant characteristics, and, most importantly, the uncertainties in those estimations at any given moment. The system uses only single camera for mapping. The Extended Kalman Filter updates the map constantly and continuously, yet it is initialized at system startup and endures until the operation is finished. During camera motion and feature extraction, the probabilistic state estimates of the camera and features are revised. The map is expanded with new states as new features are seen, and features may also be removed if required. A first order uncertainty distribution defining the number of potential departures from these values, as well as the propagation through time of the average best estimates of the camera and feature states, are what give the map its probabilistic nature. The map is mathematically expressed as the state vector  $x$  eqn. (3.13), and covariance matrix  $P$  eqn. (3.14) is a square matrix of equal dimensions that may be divided into submatrix parts.

$$\hat{x} = \begin{bmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{bmatrix} \quad (3.13)$$

$$P = \begin{bmatrix} P_{xx} & P_{xy1} & P_{xy2} & \dots \\ P_{y2x} & P_{y1y1} & P_{y1y2} & \dots \\ P_{y2x} & P_{y2y1} & P_{y2y2} & \dots \\ \vdots & \vdots & \vdots & \dots \end{bmatrix} \quad (3.14)$$

The probability distribution of the overall map is calculated is a single multivariate gaussian distribution in a space of size as vector size, following state vector eqn. (3.15) is derived

$$x_v = \begin{bmatrix} r_w \\ q^{WR} \\ v^W \\ \omega^R \end{bmatrix} \quad (3.15)$$

Where  $r_w$  is the 3D position,  $v^w$  is the velocity value,  $q^{wr}$  is the orientation vector and  $\omega^R$  is the angular velocity vector. Figures 3.10 demonstrate the manner of search

evolved over several frames and how the distribution typically changed over time from uniform to sharp peak, respectively. The distribution is securely approximated as Gaussian and the characteristic is initiated as a point into the map when the ratio of the standard deviation of depth estimates falls below a threshold (presently 0.3). As the camera travels and more standard measurements are taken, the depth uncertainty of features that have recently exceeded this threshold generally gets less which shows numerous randomness ellipsoids extended along the approximate camera viewing direction. Figure 3.14 depicts the depth perception possibility with Mono LIDAR which is often limited in short ranges[89].

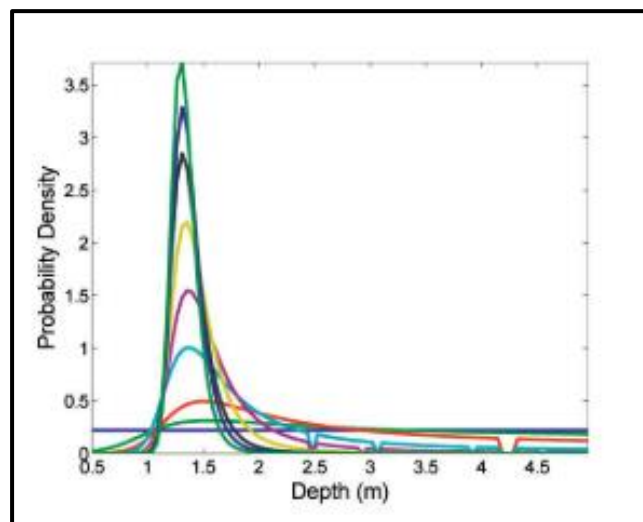


Figure 3.14 Evolution of the probability density varying depth signified by object sets.

### 3.9.2 Binocular SLAM

On picture segmentation, binocular matching was carried out to produce depth maps of the objects. Multi-fruit matching and combination stereo matching were the first two phases in this process. The first phase included determining which object in a set of stereo pictures on the left matched to the object on the opposite side of the identical pair of stereo images. An object in the left picture can match with more than one object in the right image and vice versa because of this phase reducing the multiple matching

between items in the left image and objects in the right image. Through stereo matching between objects in the left picture and its corresponding object in the right image, different maps of various items were acquired in the second stage. The triangle extending method might then be used to acquire depth maps of objects. To lower the likelihood of multiple matching, the multi-fruit matching strategy first published in the literature (Plebe and Grasso, 2001) was applied. Typically, there is more than one object in either the left or right image in a stereo pair. Multiple screening then would take place. That is, using the disparity limitations, more than single object in one picture of this pair of stereo images was matched with an object in the other image.

$$\Delta x_{min} \leq \Delta x \leq \Delta x_{max} \text{ (where } |\Delta y| < \Delta y_m \text{)} \quad (3.16)$$

where  $\Delta x$  and  $\Delta y$ , correspondingly, represent the horizontal and vertical discrepancies. The greatest and minimum horizontal disparities, or  $\Delta x_{max}$  and  $\Delta x_{min}$ , correspond to the closest and farthest working distances, respectively.  $\Delta y_m$  is the extreme value of vertical disparities because of the occlusion's fluctuation in the vertical coordinates of object centroids in each picture of a pair of stereo images. The disparity restrictions for a pair of stereo pictures might be used to generate an original difference matrix.

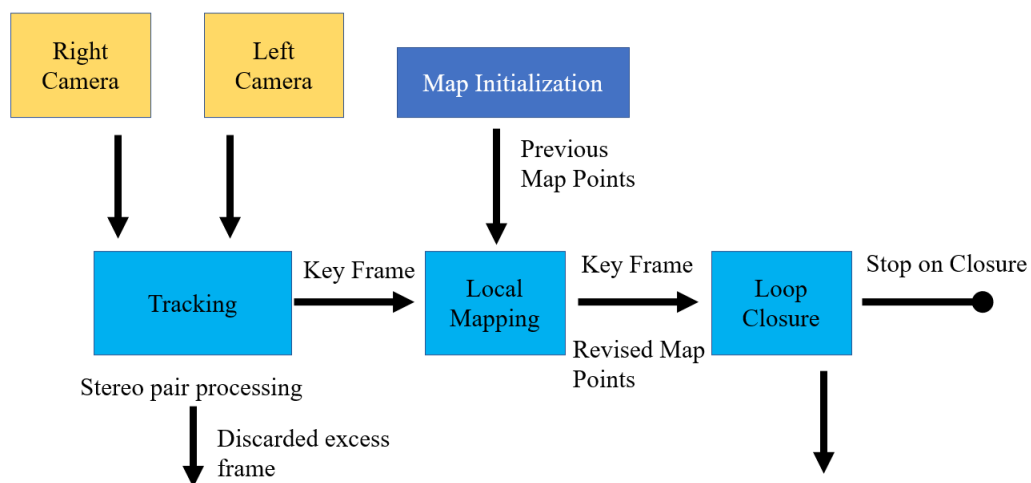


Figure 3.15 Stereo Visual SLAM processing pipeline

The disparity of each object in the right picture fulfilled the disparity requirements with a object in the left image for every element in a row of the original discrepancy.

In comparison, every element in a column of the original difference was equal to or more than a object in the right picture for each object in the left image. Eventually, using the triangle ranging method, depth maps were obtained. The positioning mistakes of the localization technique (the discrepancy between the depth value of aobject obtained using the localization method based on binocular stereo vision and the actual depth value of this object)

### **3.9.3 LIDAR-SLAM**

Because optimization is preferred over filtering in the most advanced visual SLAM, effective management of the recorded data in the scene map is essential to the precision attained (can be either be Kalman filter or particle filter). By using resilient cost functions, it is possible to jointly optimize the sensor (robot) trajectory and its map, revise the linearization points, and mitigate the impacts of estimates and inaccurate readings. On the other hand, basic point clouds are used to describe the map in most LiDAR-based SLAM systems. This is frequently a sensible decision because the basic point-based depiction doesn't rely on the presence of any particular features of the environment then produces exceptionally accurate maps when paired with pose based optimization and loop closure detection[21]. However, real-time operation of methods like those in requires enormously parallel processing on basic purpose graphic processing unit (GPU). The number of limitations that may be created in the map and then used by optimization is ultimately reduced because raw point-based depictions are unable to distinguish important features like objects of defined classes or geometric characters, which makes it difficult to track such characteristics over multiple scans similarly to visual tracking. The use of surfel-based map representations derived from LiDAR data has been attempted in a number of earlier research. [90] employed a voxel grid-based representation having lower resolution point clouds. Additionally, scan to-model registration and compact 3D model (map) representation were both accomplished using the normal distribution transform. Grid-based map representations have the issue of being computationally wasteful since global closest neighbor's search must be employed to determine similarities between the scan points and its model. The use of surface elements for map representation in SLAM has gained popularity. ElasticFusion's effective RGB-D implementation has

highly inspired work on LiDAR-based systems. A local, robot-centric multi-resolution grid map and the surface representation were combined by [91]. It is feasible to efficiently accumulate the scanned points to surfels due to the map's better resolution close to the LiDAR, then to employ the local maps in a multiple level graph-based SLAM. [92] were able to take use of the spatial data linkage between the laser scan and the rendered presentation from the map thanks to the surfels based depiction, enabling real-time mapping at a wide scale. Recent attempts to apply deep learning techniques to address the LiDAR-based odometry challenges also make use of volumetric representations. Sadly, point clouds, which are common LiDAR data formats, lack differentiation. Therefore, cutting-edge neural models like PointNet [93] or using a voxel-based approach ensure invariance to the point order permutation while processing range data. 2021, 21 and 3445 sensors 4 of 31 successful RGB-D-adopted ElasticFusion techniques, which directly influenced the development of LiDAR-based systems. A local, robot-centric multi-resolution grid map and the surfel representation were combined by [91]. It is feasible to efficiently aggregate the scanned points to surfels thanks to the map's better resolution close to the LiDAR, then to employ the local maps in a multiple level graph-based SLAM. Behley and Stachniss were able to take use of the projective data linkage between the laser scan data and rendered view from the map because of the surfel based representation, enabling real-time mapping on a wide scale. Recent attempts to apply deep learning techniques to address the LiDAR-based SLAM or LiDAR-based odometry challenges also make use of volumetric representations. Sadly, point clouds, which are common LiDAR data formats, lack differentiation.

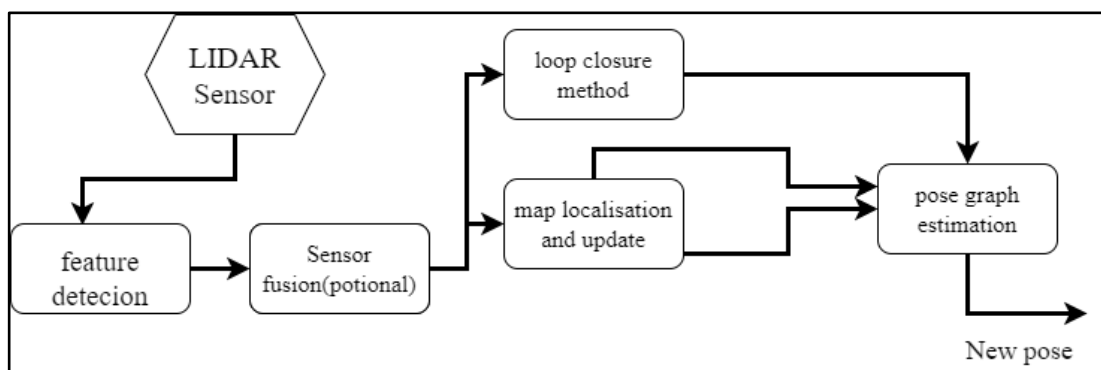


Figure 3.16 Block scheme for lidar Architecture

Modern neural models like PointNet or those that use voxel representations provide invariance to the point-order permutation while processing range data.

Convolutional networks can be used to handle point clouds using the PointNet technique and its variations, but they are thought to consume too much computer power to be used for real time SLAM. The recent L3-Net system combines a light version of the Point Net with another neural network which accumulates the matching costs provided by all the features in a volumetric model, it then outputs the pose as one of the few attempts for solving LiDAR-based localization in a new environment (either odometry or SLAM) with neural network methods. transformed the pose estimation problem into a classification task. Unfortunately, recasting the pose estimation issue as cost accumulation reduces the pose estimate's accuracy since, depending on the data format, the output is confined to discrete values. The most recent Deep LO method used an unsupervised loss function derived from the iterative closest points formulation and offers pose estimation accuracy on par with model-based systems. It also makes use of the normal vectors consistency for obtaining a confidence map that gives the loss function weighting factors. Although unsupervised learning is a significant advancement for learnable LiDAR SLAM, due to the fact that the Deep LO had to be trained independently for each of the three different data sets where it was tested on highlights a drawback of fully learnable LiDAR-based SLAM: It can have issues if it encounters a previously undiscovered environment.

The optimization problem in the odometry step can therefore be formulated as eqn. (3.17):

$$T^* = \operatorname{argmin}(\sum_i f(p_i, t_i, T) + \sum_j g(p_j, t_j, T)) \quad (3.17)$$

Where  $f(p_i, t_i, T)$  represents distance from point  $p_i$  to the line where  $g(p_j, t_j, T)$  represents the earlier scan.  $T^*$  is the best pose estimate. The lidar frame of reference during the measurement is  $p^m$  and at the end of the scan is given by  $p^e$  which is given by eqn. (3.18)

$$p^e = T_{i,i+1}^{-1} \exp\left\{\frac{t_i}{t_s} \log T_{i,i+1}\right\} p^m \quad (3.18)$$

Where  $t_i$  is the time of measurement at  $i$ -th turn,  $t_s$  is the duration of overall scan,  $T_{i,i+1}$  is the pose by the end of the scan.

### 3.10 Experiment for Comparison of Obstacle Avoidance Algorithms

An experiment was conducted to find the performance comparison of different obstacle avoidance methods. The most popular three algorithms Rapid exploring random tree RRT, Vector Field Histogram VFH and Dijkstra's were compared for tackling obstacles. Simulation comparison was carried out in MATLAB program. Robot used was differential 2-wheel robot of base length 0.5 meters. While VFH is a "face and deal obstacle" algorithm, the RRT and Dijkstra are preplanning search algorithms. The Fig. 3.17(a) indicates robot and obstacle position, Fig. 3.17(b) and search algorithm nodes created during execution. They need constant knowledge of map and object while moving. They also require preparation time once an obstacle is encountered. Performance indicators used are  $Time_{mPlan}$  is the time taken for the robot to prepare strategy to overcome the obstacle.  $Time_{trv}$  is the time taken by the robot to travel the path once the planning is ready.  $Time_{total}$  is sum  $Time_{mPlan} + Time_{trv}$  of the time taken for the overall operation to complete. The performance of search algorithm is highly dependent on the size of area of operation and complexity of obstacle structure. VFH algorithm remains stable within large range of area provided the sensor is suitably designed for the size of the robot. The time of execution  $Time_{mPlan}$  is approximate time taken for a Raspberry board computer. The processing speed of various CPUs and microcontrollers are given in Table 3.7[94] expressed in MIPS that is million instructions per second.

Table:3.7 Speed comparison of CPUs.

Processor	MIPS	Produce Year
ARM Cortex A7	2,850 MIPS at 1.5 GHz	2011
Texas Instruments TMS320C20	12.5 MIPS at 25 MHz	1987
Intel Core i5-2500K (Quad core)	83,000 MIPS at 3.3 GHz	2011
Intel Core i7 4770K	133,740 MIPS at 3.9 GHz	2013
Raspberry Pi 2	4,744 MIPS at 1.0 GHz	2014
Arduino Board UNO	16MIPS at 16Mhz	2009

The experiment was conducted in i7 processor laptop computer can be compared with computer board Raspberry Pi board 2 a very popular compact board used for robotic. The speed differences come with a factor of 28. Real differences of computation may vary as it is dependent on several factors of an operating system software, however the speed of algorithm will be significantly lower than performed in CPU.

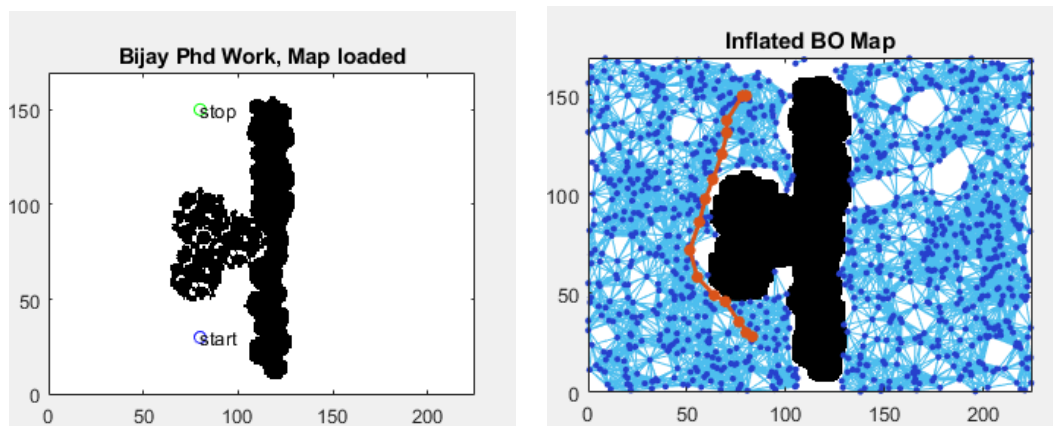


Figure 3.17(a) Field obstacle with start-end points, (b)Plot for search algorithms

### 3.11 Observations and Analysis for Comparison of different Obstacle Avoidance Algorithms

The results were observed for reading taken at different size of the area of operations, where the algorithms would begin to vary significantly. As understood from Table 3.8 the time taken by algorithm are of much significance. Search and plan algorithm require computation time to first plan and go over the obstacle. The algorithms are probabilistic method hence can give different deviations and time of travel for every trial, though alterations in outcome is less, one set of reading has been noted by taking average of 5 different trials in order to average the output effect. The branch length limit for search algorithm was set as 3meters.



Table 3.8 Performance comparison for different obstacle avoidance algorithms

		Time <sub>mPlan</sub> (sec)	Time <sub>trv</sub> (m)	Dist <sub>trav</sub> (m)	Time <sub>tot</sub> (sec)	Dev%	Time <sub>mPlanR</sub> (sec)	Mapping req
30x30m <sup>2</sup>	Dijkstra	5	58	29	300	16	140	Yes
	RRT	3	55	26	286	4	84	Yes
	VFH	0	59	30	287	20	0	No
		Time <sub>mPlan</sub> (sec)	Time <sub>trv</sub> (m)	Dist <sub>trav</sub> (m)	Time <sub>tot</sub> (sec)	Dev%	Time <sub>mPlanR</sub> (sec)	Mapping req
60x60m <sup>2</sup>	Dijkstra	21	116	59	617	16	588	Yes
	RRT	8	114	56	595	9	224	Yes
	VFH	0	118	59	588	16	0	No

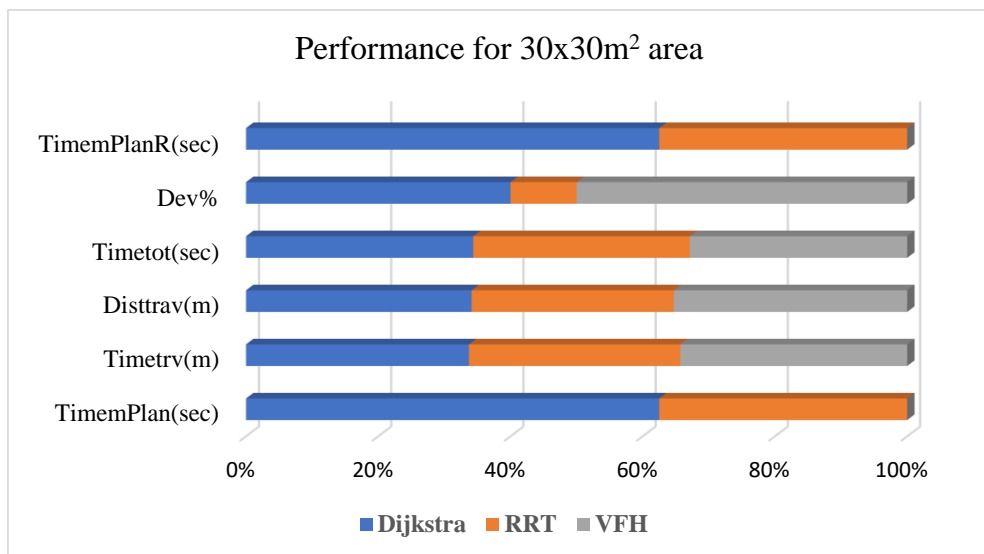


Figure 3.18 Performance indicators for Dijkstra vs RRT vs VFH at 900m<sup>2</sup> area.

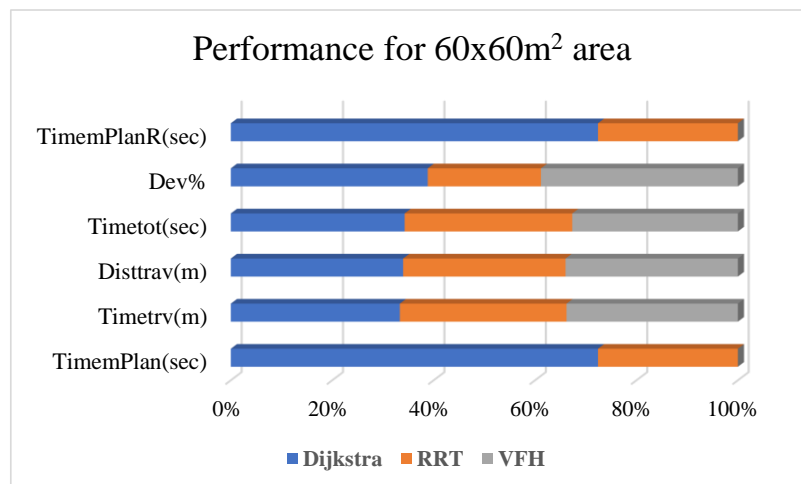


Figure 3.19 Performance indicators for Dijkstra vs RRT vs VFH at 3600m<sup>2</sup> area.

As seen in graph plot of Figure 3.18 the deviations dev% for VFH is highest, Dijkstra and RRT have lower deviations as they optimize for the shortest path to curve the obstacle and reach the stop point. RRT is very fruitful algorithm in such case as it expands its tree branches towards the direction target point rather than other direction hence reaches the end point quickly.  $Time_{mPlan}$  of Dijkstra is much higher than RRT due to obvious reasons that its branches expand overall directions to reach the end point, this can waste the time, but such Dijkstra is favorable in places of narrow path and complex object placements like in a labyrinth.  $Time_{trv}$  is mostly proportional to the distance explored by the robot.  $Time_{total}$  becomes very high for Dijkstra as the area grows as it requires more time to expand its branches in all directions, the execution time of algorithm is dependent area of coverage. Dev% for robot rises for RRT for larger area as it has to grow longer, and deviation can increase with longer branches.  $Time_{tot}$  total time can be higher than calculated in the simulation, as real applications of search algorithms need constant map information which needs computation and updating. Real time mapping is not taken into account for this experiment. Therefore, the time required for real operation to plan the map is much costly in terms of hardware and computation resources.

### **3.12 Conclusion**

Study on different sensors available for mobile robots were studied. Their usage and capacity to be employed in a variety of perception and localization were addressed. LIDAR sensor can be concluded as best suitable sensor for agricultural outdoor operations, with a respectable range and adaptability to different environments. Drawbacks of other sensors have been compared and discussed in the chapter. The utilization of RTK-GPS can be best suitable for outdoor vehicles as it requires minimum setup and is very accurate. Though simulation experiment was conducted in different chapters, the localization of differential robot has been taken as an ideal case based on wheel odometry but practical operating machine in field require more accurate sensor to work properly. In such case a RTK-GPS would be the ideal choice for the robot to be localized properly in outdoor farm area. Various localization and

mapping methods has been discussed each with their several complexity and stages of computation. VFH has been compared with RRT and Dijkstra algorithm with simulation experiments and results. It can be understood how search algorithm are intelligent algorithm to reduce and tackle objects, they need significant preparation time and continuous map information to handle dynamic obstacles. Choosing VFH algorithm to be a suitable algorithm for dealing dynamic obstacles at low computation and design a hybrid algorithm.

## **CHAPTER-4**

### **Hybrid Algorithm using PPA-VFH-Bug algorithms for Autonomous Navigation of Agricultural Robot**

## **4.1 Introduction**

A cost-effective and effective navigation system for differential wheeled mobile robots is essential since autonomous robots for diverse locations within the operating field area. It is also important that navigation system be simple, robust and predictive for such robots to be accepted by farmers. In this chapter, a hybrid algorithm for autonomous navigation system using the pure pursuit algorithm (PPA) and vector field histogram is proposed (VFH). While the VFH algorithm aids in avoiding obstacles, the PPA algorithm drives the vehicle autonomously in the direction of waypoints. For observation, the VFH approach employs 2-D light detection and ranging (LiDAR) sensors. PPA prescribes a minimal number of waypoints to simplify the map setup. Also too much waypoints can increase the probability of waypoints being covered by obstacles and hence causing problem to reach the goal point. A number of performance indicators, such as the robot's trip distance, the number of iterations required to finish the route, are assessed using the variable settings of the PPA algorithm.

The precision application of fertilizers and pesticides by mobile unmanned vehicles is a requirement for agricultural automation. Further, as this study investigated method for avoiding barriers based on vector field histograms (VFH) it came across other issues. Local minima generated by U shaped traps could get the robot stuck in the space, a hard to recover situation. The proper tuning of VFH parameter is required to mitigate such traps. For effective travel and efficient pesticide application, shortest route strategy to waypoints is a disadvantage. The challenge is overcome by integrating BUG algorithm to the hybrid system. MATLAB simulation was carried out for expected outcomes.

## **4.2 Unique Challenges of Obstacle Avoidance Algorithms for Agricultural Robots**

Robots must make quick collision avoidance decisions based on timely awareness of the nearby environment since the environment can be continually changing naturally to maintain crop and the body safety. Obstacle detection and collision detection control must be done in proper order to properly work a obstacle avoidance problem. The fundamental need is to calculate the least avoidance distance and the maximum

avoidance angle while taking into account its self-mobility and posture. The most widely used algorithms for obstacle avoidance control include the VFH algorithm[95] but it has no intelligence of objects character. Visibility technique, Artificial potential field method, Dynamic window method are also integrated with the recent research state in the field. Through various researchers' extensive effort, it is easy to see that each of the five approaches has a distinct advantage as well as demerits. This is just an application in the general environment; the dynamic window method has been discussed and applied to sample the speed over a predetermined time interval for the obstacle avoidance problem in the agricultural setting. The speed calculation problem is challenging when employed in the agricultural context because odometry and inertial measurement module mistakes would accumulate in long time, making it more challenging to apply the dynamic window method. Implementation of the BUG approach supported by visible image method in agriculture has some potential, but there are several issues for proper execution and proper choice of computation. The limited operating space of agricultural robots is a result of the crops' intensive growth. When obstacle avoidance is required, obstacles need to be avoided the best way possible, the robots must maintain relatively close proximity to the impediments before bypassing them, which places a high demand on the sensor's accuracy and the choice of decision points to turning back to the original path after diversion. The effectiveness of motion planning is directly impacted by this factor. Because of the thick impediments, the artificial potential field method cannot be used in agricultural settings and is ineffective due to its sensitivity in a noisy irregular environment. The VHF method can be used as its primary purpose is to represent the environmental map by occupying the grid to represent the connection between obstacle and the robots to direct the robot's motion to the obstacles. However, as the requirements can become more stringent, such as when the quantity of obstacles increases it will place a significant load on the calculation, this is a problem that should be fully taken into account. In other words, even though some obstacle avoidance algorithms have not been tried to use in agriculture, their positive performance traits in different environments guide researchers studying towards the right direction for solving the problem.

### **4.3 PPA-VFH Hybrid Algorithm**

The method presents on modeling and simulation of agriculture robot whose navigations is based on PPA algorithm. The system is aided for steering guidance using VFH algorithm. A 2D 360-degree LiDAR sensor is used to scan the field area. PPA is a simple algorithm following the predesignated waypoint coordinates which are consecutively reached till the last point. The algorithm is executable in a 16/32-bit microcontroller thus applicable in a modest autonomous robot system. In the real world waypoints for PPA is set as different target coordinate (x,y) points similar to GPS locations coordinates. VFH is a fast algorithm which helps robot to steer away when detects an obstacle with LiDAR sensors. The performance is satisfactory with the rare appearing smaller obstacles ahead of the robot.

### **4.4 Methodology for PPA-VFH Hybrid Algorithm**

The simulation is carried out in the Matlab/SIMULINK environment. During execution, the real-time values of observations were logged and saved in a CSV (Comma separated values) data file. The loop iterations were sampled typically every 120 milliseconds. The navigation platform toolbox is used to implement the PPA and VFH algorithm functions. For the LiDAR sensor application, the mobile robotic simulation toolbox is used. PPA is used in the simulation to achieve autonomous lateral navigation. The PPA the waypoint seeking system was guided by a steer guidance VFH algorithm against collision with obstacles. As shown in Figure 3, different waypoints ( $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$ ,  $P_5$ , and  $P_6$ ) are constructed. With a change in size of coordinate data, these waypoints can be converted to GPS tagged points for real-world applications. The vehicle's robot localization is expected to be efficient, ignoring wheel slippage and drift. Odometry sensors, GPS, and motion sensors could be used in real-world models. The navigation route will cover points  $P_1$  through  $P_6$  in order. At the U turn of the row of plants shown in Figure 4.2, the waypoints coordinates are formed as tip points alternately. The robot will navigate the field while avoiding various items; huge unavoidable obstructions will create a strong repulsive force field surrounding the vehicle, which will cause the robot to become immobile. The flowchart in Figure 4 depicts the overall operation. The experiment is conducted

on a vegetable field measuring 50 x 50 m<sup>2</sup>. The robot experiment is carried out at various speeds. For each trial, the robot's speed was changed from 0.5 to 2.75 km/hr with a fixed speed. This speed range is ideal for slow operating irrigation robots that operate slowly. The robot will proceed in a zigzag cycle path to cover all rows of plants. The differential wheel robot has a 0.40 m wheelbase and a 10 cm wheel radius. After multiple trials and observations, the VFH parameters of histogram threshold value and number of angular sectors were calibrated and locked at an optimal value of 20. Even though the LiDAR sensor is two-dimensional, a synthetic two-dimensional imaging approach for navigation can be constructed from popular three-dimensional data [96]. This method also minimizes the data and complexity associated with cloud point mapping.

Using image conversion from colour to black-and-white, a field image (Figure 4.1) is converted to a grid occupancy map (Figure 4.2). The occupancy map is a binary map with black parts representing obstructions and white parts representing open moveable space. Different obstacles, like as stones, solid items, or overgrown plants, may be present on the path as the robot advances along row gaps. The VFH algorithm assists in detecting and tactically avoiding static and moving impediments. The algorithm is based on the repellent forces produced by nearby obstructions. Every trial is recorded with scanned LiDAR sensor readings and pre-set PPA parameter values. The total distance  $D_c$  traversed by the robot is used to calculate the robot's deviation while travelling. More oscillations with a larger average deviation are indicated by a large distance  $D_c$  travelled by the robot value. PPA characteristics that is desired linear velocity ( $D_{LV}$ ), maximum angular velocity ( $AV_{max}$ ), and minimum radius for turn ( $R_{min}$ ) are tested in various ways. Sensor readings from location and object proximity are taken to determine optimal performance. The robot's total distance travelled in relation to time is an essential performance indicator.



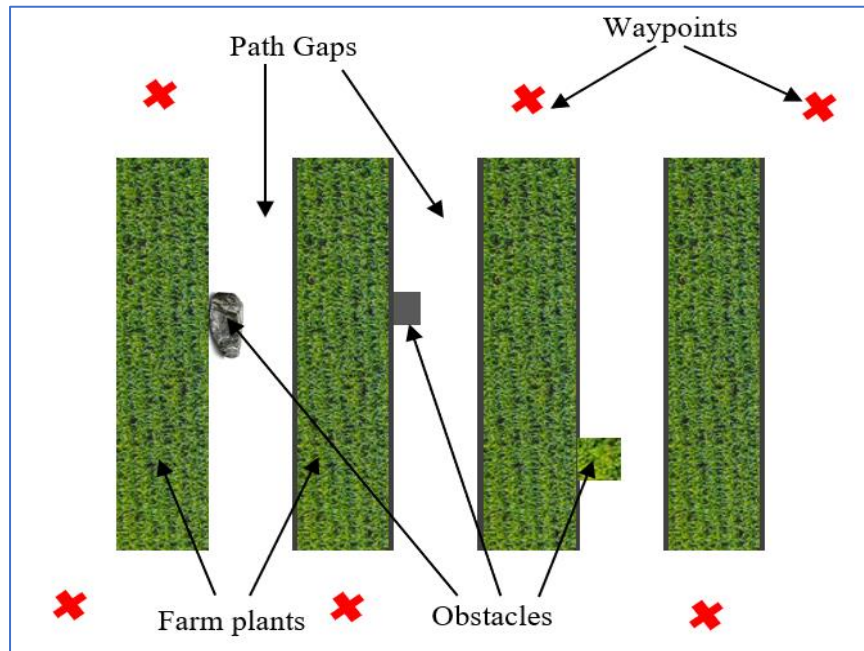


Figure 4.1 Real field image scenario.

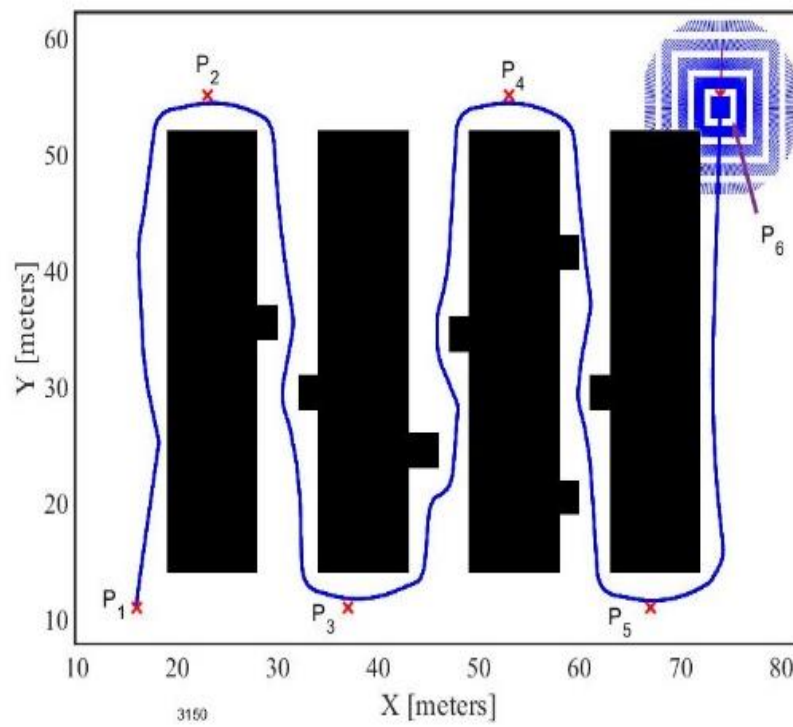


Figure 4.2 Field image with grid occupancy map

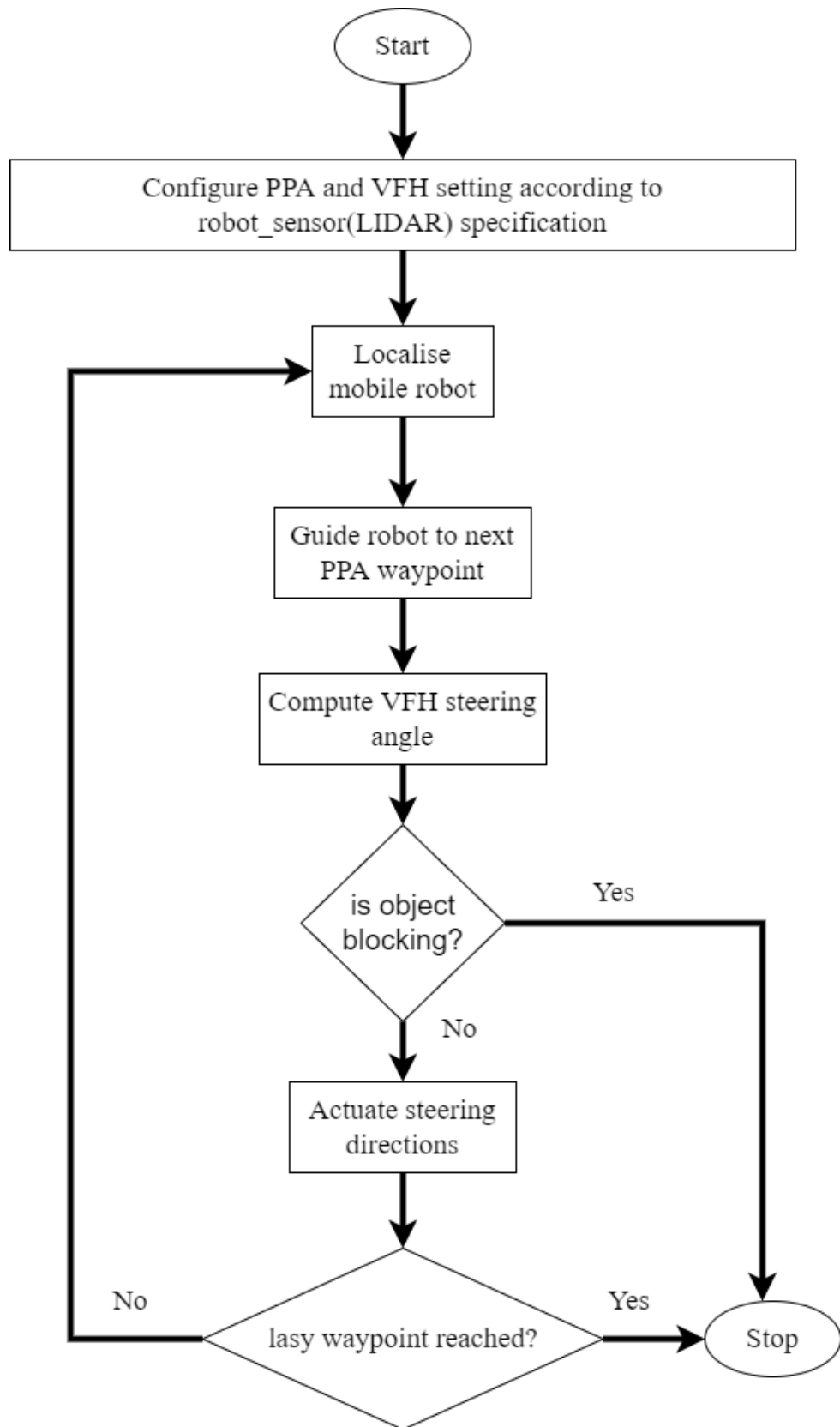


Figure 4.3 Flowchart of overall operation

The robot was expected to follow straight trajectories along the narrow gap. Marked trailed behind by the mobile robot help to visualize the deviation. The robot is expected to drive sharp close at U turns sections to prevent overshoot.

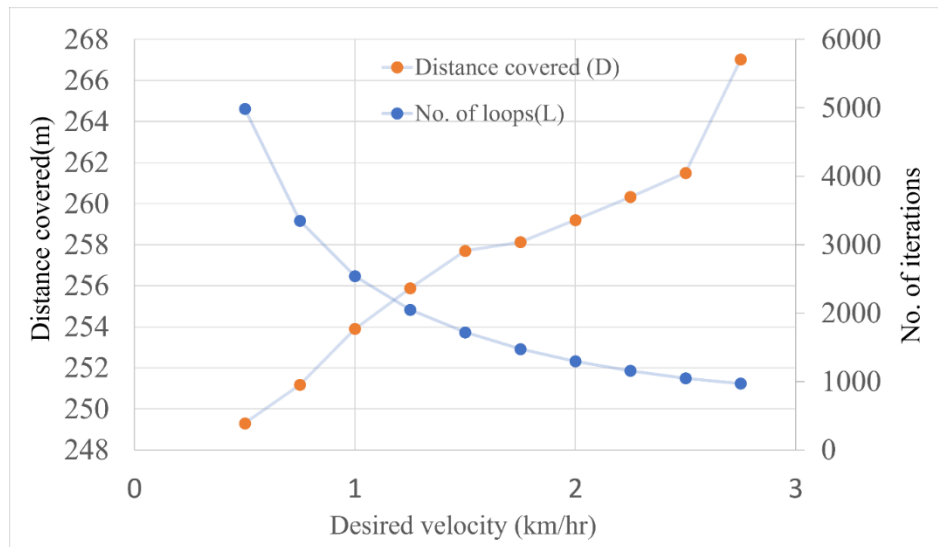


Figure 4.4. Performance plot under varying velocity of robot

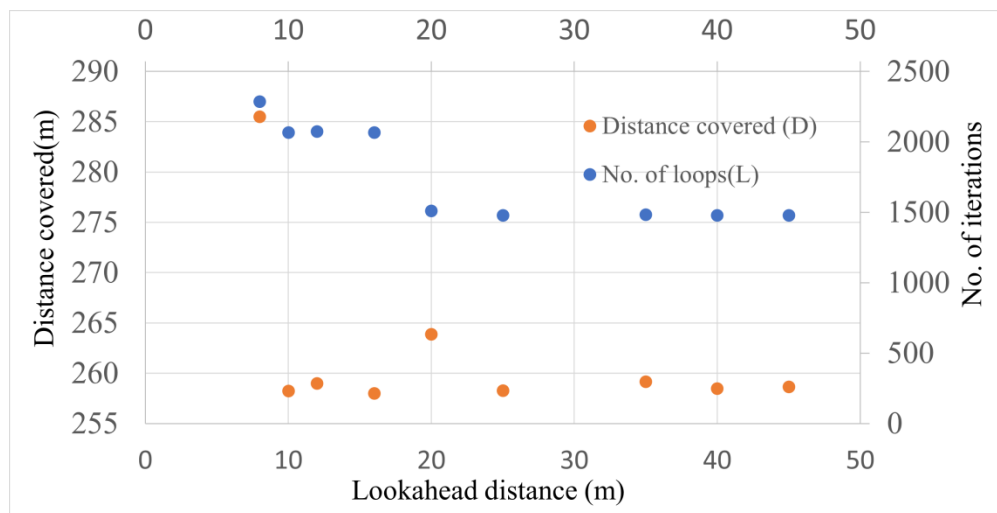


Figure 4.5 Performance plot under varying lookahead distance

Table 4.1 Performance data with lookahead distance

$AV_{max}$ (rad/s)	$N_L$	$D_C$ (m)	$OD_{avg}$ (m)	$OD_{min}$ (m)	$OD_{max}$ (m)
2	2076	259	1.16	0.81	1.80
4	2076	259	1.16	0.81	1.80
6	2076	259	1.16	0.81	1.80
8	2076	259	1.16	0.81	1.80
10	2076	259	1.16	0.81	1.80
11	2088	260	1.20	0.81	1.51
12	2076	259	1.53	0.31	3.06
0.01	2093	261	1.25	0.81	1.57
0.06	2089	260	1.20	0.81	1.49
0.08	2085	260	1.20	0.81	1.51
0.001	2086	260	1.15	0.81	1.47

#### 4.5 Results and Observations for PPA-VFH Algorithm

On various PPA settings, several sets of readings are taken. The desired linear velocity (DLV), Angular velocity maximum ( $AV_{max}$ ), Lookahead distance (LD), and Minimum Radius are all varied in a total of 34 trials ( $R_{min}$ ). Real time proximity distances are used to measure closeness of obstacles around the vehicle: Obstacle distance average ( $OD_{avg}$ ), obstacle distance minimum ( $OD_{min}$ ), and obstacle distance maximum ( $OD_{max}$ ).  $OD_{max}$  is the robot's maximum or farthest distance maintained by vehicle from obstacles while travelling along the path. The nearest distance for the same is  $OD_{min}$ . The average obstacle distance maintained is  $OD_{avg}$ . Figures 3 and 4 as well as Tables 4.1 and 4.2 depict observations. In each simulation,  $N_L$  is the number of loops or iterations consumed for completion of a trial, it is an accurate indicator of the experiment's time consumption. It represents the time it takes the mobile robot to get from waypoint P1 to P6. The distance travelled by the robot as it moves across the field is called  $D_C$ , and a smaller value is expected as usual for movement without oscillations.

The desired linear velocity ( $D_{LV}$ ) is the speed of the moving vehicle. Slower speed produces lesser inertia of motion and hence lesser oscillations, however the time taken for completion of path is high. Such movement is suitable in slow field operations like watering of plants, weed removing etc. Force fields created by the obstacle objects are subtle, and the steering against obstacles are done smoothly, a more stable non deviated travel. The least obstacle distance  $OD_{avg} = 0.98\text{m}$  when the  $D_{LV}=0.5\text{Km/Hr}$ . The maximum value of  $OD_{avg} = 1.40\text{m}$  when  $D_{LV}=1.5 \text{ Km/Hr}$ . Minimum value of  $N_L=973$  when  $D_{LV}=2.75 \text{ Km/Hr}$ . Maximum value of  $N_L=4988$  when  $D_{LV}=0.5 \text{ Km/Hr}$ . Least distance  $D_C = 249\text{m}$  when  $D_{LV} = 0.5 \text{ Km/Hr}$  and maximum  $D_C=267\text{m}$  when  $D_{LV}= 2.5 \text{ Km/Hr}$ .

Maximum angular velocity ( $AV_{max}$ ) is highest limited value of angular velocity for the vehicle. It has a threshold value up to which performance is not much altered, but is restricted beyond the limit. Mentioned observations are given in Table 4.1. The least value of  $OD_{avg} = 1.16 \text{ m}$  when  $AV_{max}= 2$ , the highest  $OD_{avg} = 1.53\text{m}$  when  $AV_{max} = 12$ . Minimum value of  $N_L = 2076$  at  $AV_{max}=2 \text{ rad/sec}$ , maximum value of  $N_L=2093$  at  $AV_{max} = 0.01 \text{ rad/sec}$ .

Lookahead distance ( $L_D$ ) is an important factor effecting the lateral behavior of the vehicle.  $L_D$  is kept as constant value set before executing experiment. It is varied and tested at speed  $D_{LV} = 1.5\text{Km/Hr}$ , an average speed chosen for the mobile robot. Lower value of  $L_D$  produces more oscillations but distance against obstacles and plants are minimum. High oscillations consumes more iteration and time. Observations are presented in Figure. 4. Minimum value of  $OD_{avg} = 1.16\text{m}$  occurred when  $L_D = 12\text{m}$ .  $OD_{avg} = 1.39\text{m}$  (highest) when  $L_D = 45\text{m}$ . Maximum number of loops  $N_L = 2286$  when  $L_D = 8\text{m}$ , the minimum value of  $N_L=1510$  when  $L_D = 20\text{m}$ . Maximum value of  $D_C = 285\text{m}$  when  $L_D = 8\text{m}$ , minimum value  $D_C = 258$  when  $L_D = 40\text{m}$ . An optimum value of  $L_D = 20\text{-}25\text{m}$  is chosen, where  $N_L=1478$  is minimum and  $OD_{avg}=1.37\text{m}$ .

Minimum radius  $R_{min}$  is another threshold value for turning radius scale. Its value had no impactful effect on the mobile robot performance. Insignificant changes are observed under distance of 3 meters (seen on Table 4.2) after which if increased the robot movement is disturbed due to large turn near row-end U sections.

Table 4.2 Performance data with minimum turning radius.

$R_{min} (m)$	$N_L$	$D_C (m)$	$OD_{avg} (m)$	$OD_{min} (m)$	$OD_{max} (m)$
2	1478	258	1.33	0.94	1.67
4	1477	258	1.38	0.94	1.93
5	1494	261	1.35	0.97	1.85
3	1478	258	1.33	0.94	1.67

#### 4.6 Specific Observations

The nearest obstacle distance  $OD_{avg}$  is significantly influenced by desired linear velocity  $D_{LV}$  and lookahead distance  $L_D$ . It is to observe oscillations with respect to the different values of  $D_{LV}$ . A high value of 2 m is observed at linear velocity of 1.5 Km/Hr which again drops to 1.5 m at the velocity of 2.25 Km/Hr, again rising to higher value of 2.11m at 2.75Km/Hr.  $OD_{avg}$  increases with respect to value of  $L_D$ .  $OD_{avg}(=1.39)$  is highest at  $L_D=45m$  and the lowest  $OD_{avg}(=1.17)$  at  $L_D=10m$ . The shortest path taken by the robot is observed as 249 m low value with  $D_{LV}=0.5$  m/s.

The fastest coverage of field is conducted within  $N_L=973$  loops at  $D_{LV}= 2.75$  Km/Hr. The slowest coverage is done in  $N_L= 4988$  loops achieved at a low desired linear velocity of 0.5 Km/Hr (Figure 3). The vehicle gives an optimum speed  $D_{LV}=1.25$ km/hr with 2049 number of iteration loops. It can be seen in graph shown at Figure. 3, where the crossover point of two identifiers  $N_L$  and  $D_C$ . The  $N_L$  and  $D_C$  are intersecting for best performance results. Optimum lookahead distance is  $L_D=20$  from where the number of loops is low with acceptable stability in its trajectory as seen in Figure 4.4

#### 4.7 Path Planning Algorithms for Circumnavigation of Obstacles

The algorithm of route planning is to accomplish obstacle avoidance by determining the ideal path, while the path planning problem of robots is a hot subject of the

composition approach to investigate the beginning and end path. So, the effectiveness of path planning directly depends on the algorithm's optimal grade. Wheeled mobile robots operate in low-speed environments with few dynamic difficulties, making path planning for them easy in terms of dynamics as it is botherless. Here, I describe and analyze some of the popular path planner algorithm and obstacle avoidance search methods. The path planner algorithm locates the destination node in accordance with the beginning circumstances and specific criteria, comparable to graph navigation. The search method is often divided into two major categories: precise algorithm and estimated algorithm. The depth first searching algorithm and the breadth-first searching algorithm both relate to the precise algorithm, and the Dijkstra algorithm is an example of a breadth first searching implementation. The estimated algorithm, on the other hand, is represented by the relatively well-established D\* algorithm and A\* algorithm. Mentioned four techniques are based on clear maps which typically employs a mapping information. In addition, researchers have shown that the RRT algorithm may be used to design paths even if the clear map is not provided. The BFS algorithm spreads out from the starting point and nearby waypoints without taking the optimum into account. To quickly reach the targeted point, the DFS algorithm is able to calculate the cost for a node to that goal point. Dijkstra algorithm can obtain the best solution by concentrating on the closest route from the starting point to rest of the locations. The A\* algorithm can be thought of as an enhanced Dijkstra algorithm because it exposes the heuristic idea of knowledge about the neighboring nodes near the beginning and target nodes, focusing on the closest path from one point to another. The RRT algorithm uses a tree structure that spreads outward from a reference point, with the direction of the spread being chosen at random from its modelling space. These algorithms Dijkstra algorithm, DFS algorithm, RRT algorithm and A\* algorithm have four different types of uses in agriculture. The D\* algorithm is not appropriate for an agriculture environment because the BFS algorithm absorbs too many computer power and significantly raises the application cost, making it impractical in agricultural settings. The method may be used in a field setting since its main purpose is to solve the optimal path issue in a volatile environment. According to recent studies, this algorithm is rarely used in the field of agricultural robotics for outdoor environmental path planning. Table 4.3 shows that these reasonably advanced search

algorithms had certain drawbacks in the general context and were subsequently improved. The original approximation method has been presented and the accuracy of path planning has been somewhat improved by new algorithms like Theta\* and Phi\* that have been proposed in current years. However, after summarizing and analyzing the traits of the abovementioned algorithms, while also combining them with the difficulty of the agricultural location, we provide near possibilities for the implementation of these enhanced algorithms and better search methodologies in agriculture in future. The three different kinds of algorithms that were enhanced inspired from A\* can be used in the facility environment. In order to identify the ideal path more quickly, the IDA\* method combines the benefits of A\* and DFS. The LPA\* method may be utilised for local pathfinding, which is crucial in the facility context, despite its poor handling of path smoothing. The Bidirectional A\* can accelerate computation and simplify the algorithm. The benefits of IDA\* could also be combined with this approach. We can significantly increase the A\* algorithm's precision and speed by building a two-way depth search, which is also necessary in a complicated agricultural context. In smart agriculture, the three different forms of better algorithms for D\* can be momentarily disregarded. Since the field environment is dynamic but its versatile aim is manageable and the universal D\* method is capable, we should concentrate on promoting and using the D\* algorithm [19]. Theta\* and Phi\*, two novel search algorithms, were presented within the previous ten years but still haven't found widespread use. The qualities of mentioned algorithms, however, can be used in agriculture in near future. First off, the alteration techniques relying on A\* algorithm can be used in different environments that A\* algorithm is ideally applicable to. Secondly, they can be used to solve the edge as well as the angle problem of the map that the original path search algorithm which cannot effectively handle in practical uses, significantly reducing the path of error improvement. Finally, compared to the original, these methods' computational cost is far reduced. All of these benefits show that they have promising application possibilities in agricultural settings. In addition to the benefits of these new and enhanced techniques, their drawbacks are also extremely clear. Future researchers on the study on agricultural robots should focus on reducing or avoiding these implementation flaws. BUG algorithm can be employed to solve the issue of circumnavigation without having to deal mapping computations and



information.

Table 4.3 Comparison of different search algorithms

<b>Algorithm Type</b>	<b>Limitation</b>
A* algorithm	evaluation function is difficult to set, smoothness of operation is low there not applicable, bi directional search needs to be constrained. Load distribution is complex
D* Algorithm	Node are assigned with complexity, difficult to decide node distribution, time taken for operation is high
Theta Algorithm	No guarantee of shortest path, parent nodes must be closer to new node, path can be longer than necessary
Phi Algorithm	can only repeat single execution for any orientation of path, time required for vertex extension is irregular

#### 4.8 Efficiency Issues of Autonomous Robot and Precision Farming

- Autonomous robot need to be more efficient during application of pesticides and irrigation.
- Obstacles (others than plants in row) must not be sprayed with pesticide during its avoidance over a pre-planned path.
- The distance travelled by the mobile vehicle around the obstacle must be minimum so that over distance is not travelled by the robot. It causes wastage of fuel, time, and energy.
- Efficient way of travel is to circumnavigate the obstacle closely as possible until the pre-planned trajectory of the robot is reached.
- Efficient way also improves application of pesticides/water as it reaches the back side of the obstacle from where more coverage of plants along the row is

accessed.

- Most planning algorithm tries to reach the closest goal after obstacle avoidance which is not preferred for an efficient application of pesticides/water. Planning algorithm work based on global knowledge of the map which is required acquired or supplied to the system, may require to be constantly updated while driving, a costly computational task for simple robots.
- The onset/offset of obstacles is detected by the activity of bug algorithm in the hybrid algorithm.

The figures given below shows the comparative strategies of movement of mobile on encountering obstacle, Figure 4.6(a) represents path taken by conventional planner search algorithms while Figure 4.6(b) shows the BUG strategy which has higher coverage for efficient application of pesticides and water.

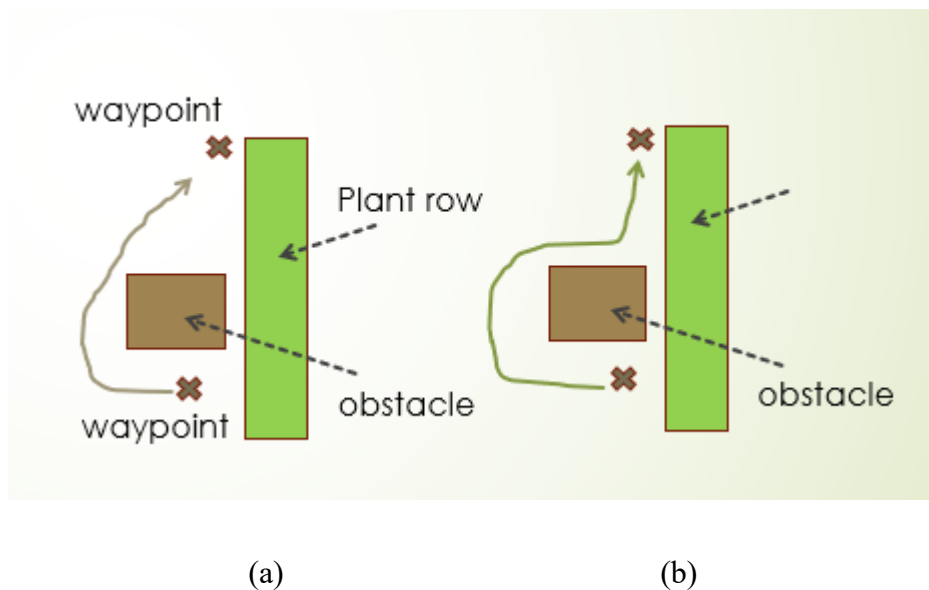


Figure 4.6 (a) Shortest reach strategy, 4.6 (b) Bug strategy,

The figures a given below shows how in Figure 4.7 (a) the robot tend to move in local traps which can cause to failure, being trapped inside the local minima. The more efficient manner of mitigating the local miming would be to detect it and circumnavigate the obstacle beyond it as in figure 4.7(b).

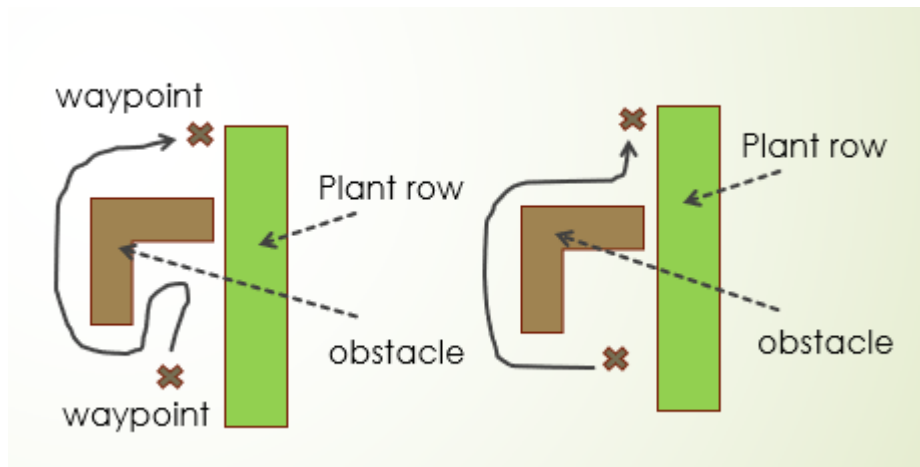


Figure 4.7(a) U shaped trap local minima, (b) Avoiding local minima

#### 4.9 PPA-VFH-BUG Algorithm

Vector Field Histogram (VFH+) is a popular algorithm to find avoidable directions against obstacles. However, since it is dependent on potential fields produced by barriers, it is drawn to local minima produced by U-shaped obstructions. This work is designed for the robot to move out of such pitfalls. This work focuses on fine-tuning an adaptive VFH+ algorithm to solve such issues. Until the required trajectory for the original route is regained, the algorithm maintains the robot near to the obstacle. It promises efficient travel after encountering obstacles. PPA algorithm is used to guide the robot from one waypoint to another. The adaptive VFH+ algorithm oversees controlling steering upon detection of obstacles. A 360-degree LIDAR sensor is used to get the position of obstacles around the robot. The data is fed to VFH+ algorithm to find the steerable direction.

#### 4.10 Methodology for PPA-VFH-BUG Algorithm

Utilizing the Pure Pursuit algorithm, a moving trajectory for the robot to move toward the target point is created. PPA method provides simple steering guidance in the direction of the way point. The scenario is tested in converted image of Grid Occupancy map created out of real farm field, the black part represents the objects and plants, the while region implying free space for robot to travel.

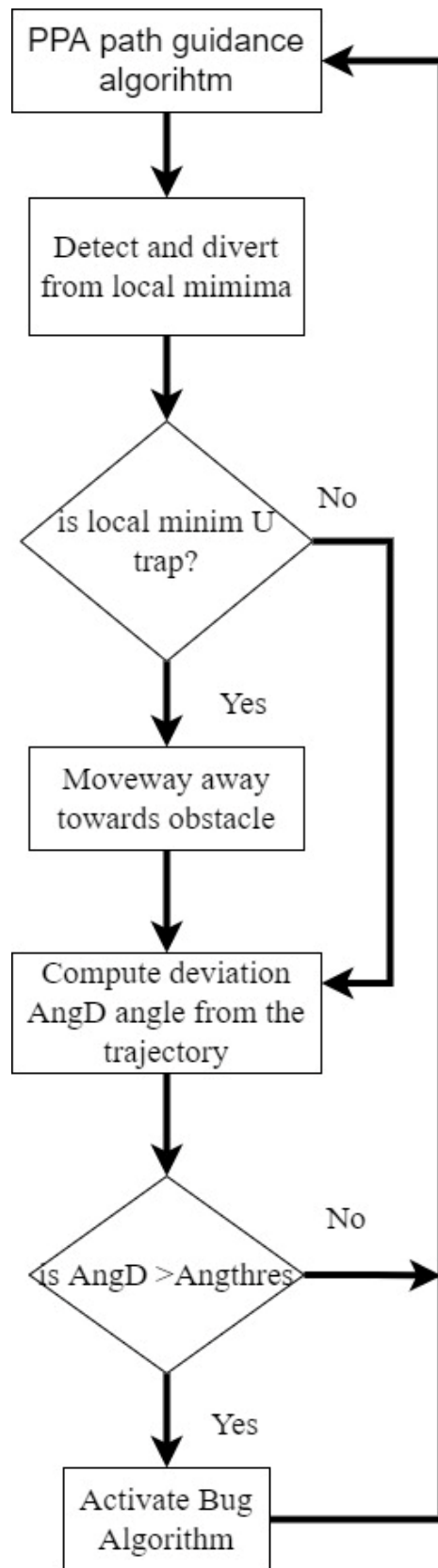


Figure 4.8 PPA-VFH-BUG Algorithm

A narrow 2-wheel Differential wheeled robot with a dimension of wheelbase length as 30cm and radius of wheel as 20cm was considered for the simulation. The start-destination waypoints point is preserved at the end of the plant's row. The VFH+ directs the robot to go straight along the row gaps. The VFH+ algorithm guides the robot parallel to rows even if the actual route from one way point to another is not vertically parallel with the rows. The approach is based on monitoring the mobile robot's orientation angle with respect to angle between start-destination trajectory. If the orientation deviates from a predetermined threshold value, the robot enters a mode in which the target angle value is slightly increased in the direction of the obstacle, causing it to steer around the side of the obstruction. The robot maintains the constant target angle until it reaches the obstacle's end and then returns to its original trajectory. In the presence of large obstacles and U-shaped obstacles, the VFH+ algorithm's lack of a constant target angle and low threshold values cause the robot to do U turns. Figure 2 explains the method of circumnavigation with fixed target angle. Such instances prevent the VFH+ algorithm from reaching its goal. As a result, VFH+ is maintained sensitive with low threshold values to enable the detection of U-shaped local minima using a suitable range of LIDAR sensor. Different ranges of LIDAR ranges were set to understand the effect and limits for avoiding local minima points caused by the U traps. The set of suitable values of LIDAR range was observed. Variations of robot's maneuvering capability were observed for different settings of LIDAR ranges settings. Figure. 3 illustrates the path travelled by the robot in simulation grid occupancy map.

For measuring the parameters of the results of simulation, observations were taken at different  $\mathbf{Ang}_{Thres}$ . The  $\mathbf{Dist}_{cov}$  is total distance travelled by the robot after avoiding the obstacle.  $\mathbf{Dist}_{obs}$  is the average distance of proximity with obstacles and plant. Its higher value indicates inefficiency operation and overshooting away from obstacles and plants.  $\mathbf{Dist}_{over}$  is the percentage of distance over travelled by the robot around the object. Higher value of overtravel indicates inefficient operation.

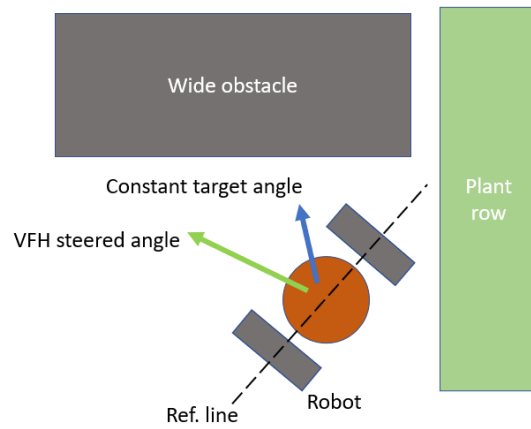


Figure 4.9 Robot control strategy with large obstacles.

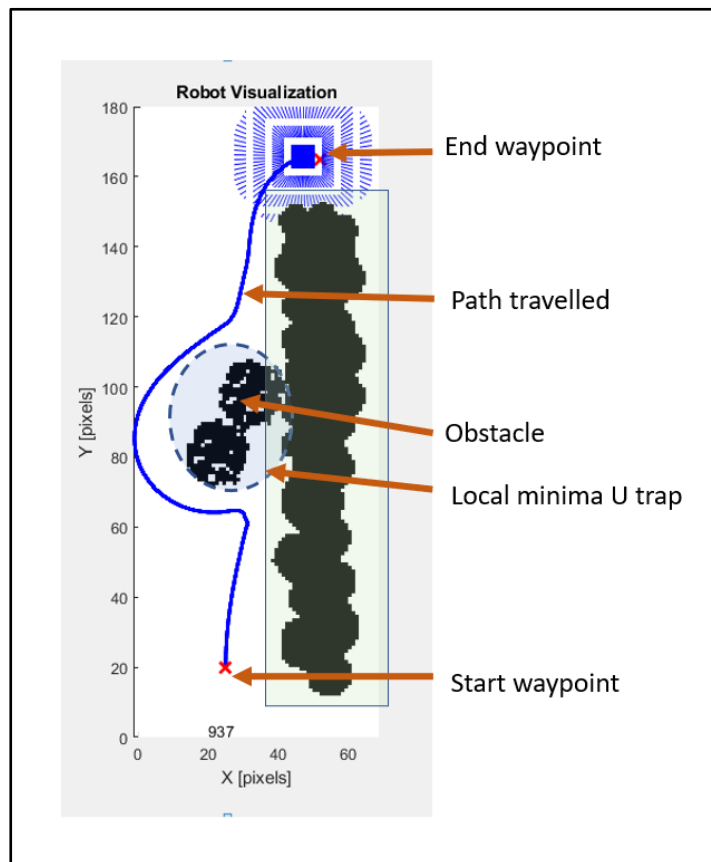


Figure 4.10 Illustration of MATLAB Simulation

#### 4.12 Observations and Results for PPA-VFH-BUG Algorithm

Optimal reading was taken between 5.7-40.1 degrees with **Dist<sub>obs</sub>** travelled at consistent best possible level, **Dist<sub>over</sub>** is also consistent at the same range of angles. The **Dist<sub>over</sub>** have lower attractive results for angles greater than 74 degrees but is inconsistent. Inconsistent output is result shapes and position of obstacles. Some readings can be abrupt depending on the structure of field of operation. Data discussed in present in Table 4.4. and figure 4.11 of this document. Different settings of LIDAR range effected the performance of the robot table 4.5. The range of successful travelling from start point to end point was 34-46cm. Increasing the value of the range or decreasing the range made the robot journey unsuccessful. They would overshoot or hitting and enter the obstacles region. Lidar setting of 40cm gave the best result with minimum over travel of 26% and 20cm obstacle distance. Observation mentioned in the Table 4.4.

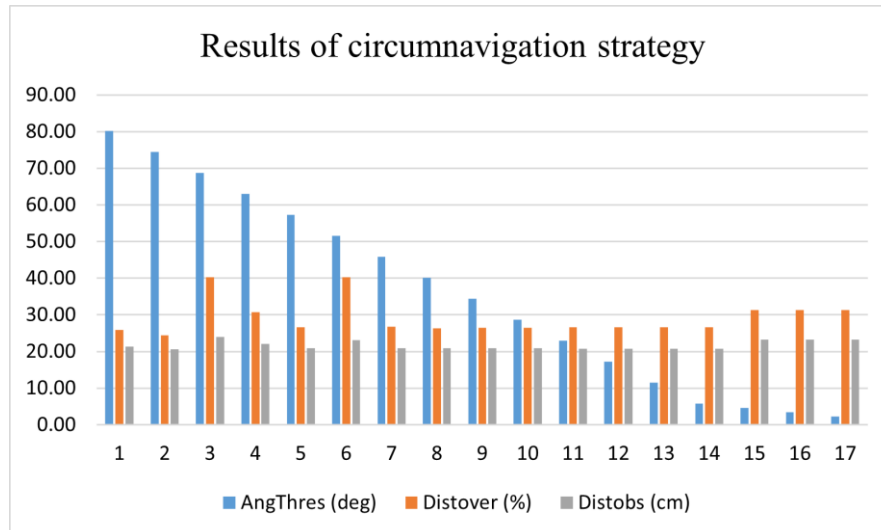


Figure 4.11 Graph plot for results of circumnavigation against obstacles.

Table 4.4: Characteristics of robot control with different  $Ang_{Thres}$  values.

<b>Ang<sub>Thres</sub> (deg)</b>	<b>Dist<sub>cov</sub> (cm)</b>	<b>Dist<sub>obs</sub> (cm)</b>	<b>Dist<sub>over</sub> (%)</b>
80.2	358.4	21.4	25.8
74.5	354.4	20.7	24.4
68.8	399.6	24.0	40.3
63.0	372.4	22.1	30.8
57.3	360.4	20.8	26.5
51.6	399.6	23.1	40.3
45.8	361.2	20.9	26.8
40.1	359.6	20.9	26.3
34.4	360.0	20.9	26.4
28.6	360.0	20.8	26.4
22.9	360.4	20.8	26.5
17.2	360.4	20.8	26.5
11.5	360.4	20.8	26.5
5.7	360.4	20.8	26.5
4.6	374.0	23.2	31.3
3.4	374.0	23.2	31.3
2.3	374.0	23.2	31.3

Table 4.5 Performance readings of robot control with different LIDAR Range(cm) values.

<b>LIDAR Range(cm)</b>	<b>Dist<sub>cov</sub> (cm)</b>	<b>Dist<sub>obs</sub> (cm)</b>	<b>Dist<sub>over</sub> (%)</b>
46	352	19.8839501	23.76933896
44	355.2	19.908054	24.89451477
42	374	24.0435684	31.50492264
40	360.4	20.7890909	26.72292546
38	378.8	24.1426793	33.19268636
36	383.6	24.3145833	34.88045007
34	399.6	24.7818018	40.50632911



Table 4.6 Sample of real time data reading taken in MATLAB program simulation

index	mn	md	minR	maxR	CPx	CPy	CPd	LPx	LPy
2	16	16	16	16	25	20	90	28.7	39.7
3	16	16	16	16	25	20.2	90	28.7	39.9
4	16	16	16	16	25	20.4	90	28.7	40
5	16	16	16	16	25	20.6	90	28.8	40.2
6	16	16	16	16	25	20.8	90	28.8	40.4
7	16	16	16	16	25	21	89	28.8	40.6
8	16	16	16	16	25	21.2	89	28.9	40.8
9	16	16	16	16	25	21.4	89	28.9	41
10	16	16	16	16	25	21.6	89	28.9	41.2
11	16	16	16	16	25	21.8	89	29	41.4
12	16	16	15.99	16	25	22	89	29	41.6
13	16	16	15.99	16	25	22.2	89	29.1	41.8
14	16	16	15.98	16	25	22.4	89	29.1	42
15	16	16	15.98	16	25	22.6	89	29.1	42.2
16	16	16	15.97	16	25	22.8	88	29.2	42.4
17	16	16	15.96	16	25	23	88	29.2	42.6
18	16	16	15.96	16	25	23.2	88	29.2	42.8
19	16	16	15.95	16	25.1	23.4	88	29.3	43
20	16	16	15.94	16	25.1	23.6	88	29.3	43.2
21	16	16	15.94	16	25.1	23.8	88	29.3	43.3
22	16	16	15.93	16	25.1	24	88	29.4	43.5
23	16	16	15.92	16	25.1	24.2	88	29.4	43.7
24	16	16	15.91	16	25.1	24.4	88	29.5	43.9
25	16	16	15.91	16	25.1	24.6	88	29.5	44.1
26	16	16	15.9	16	25.1	24.8	87	29.5	44.3
27	16	16	15.89	16	25.1	25	87	29.6	44.5
28	16	16	15.88	16	25.1	25.2	87	29.6	44.7
29	16	16	15.87	16	25.1	25.4	87	29.6	44.9
30	16	16	15.86	16	25.1	25.6	87	29.7	45.1
31	16	16	15.85	16	25.2	25.8	87	29.7	45.3
32	16	16	15.84	16	25.2	26	87	29.7	45.5
33	15.99	16	15.83	16	25.2	26.2	87	29.8	45.7
34	15.99	16	15.82	16	25.2	26.4	87	29.8	45.9
35	15.99	16	15.8	16	25.2	26.6	87	29.9	46.1
36	15.99	16	15.79	16	25.2	26.8	86	29.9	46.3
37	15.99	16	15.78	16	25.2	27	86	29.9	46.5
38	15.99	16	15.77	16	25.2	27.2	86	30	46.7
39	15.99	16	15.76	16	25.2	27.4	86	30	46.9
40	15.99	16	15.74	16	25.3	27.6	86	30	47
41	15.99	16	15.73	16	25.3	27.8	86	30.1	47.2
42	15.99	16	15.72	16	25.3	28	86	30.1	47.4
43	15.99	16	15.7	16	25.3	28.2	86	30.1	47.6
44	15.99	16	15.69	16	25.3	28.4	86	30.2	47.8
45	15.99	16	15.67	16	25.3	28.6	86	30.2	48
46	15.99	16	15.66	16	25.3	28.8	85	30.3	48.2
47	15.98	16	15.64	16	25.4	29	85	30.3	48.4
48	15.98	16	15.63	16	25.4	29.2	85	30.3	48.6
49	15.99	16	15.61	16	25.4	29.4	85	30.4	48.8
50	15.99	16	15.6	16	25.4	29.6	85	30.4	49

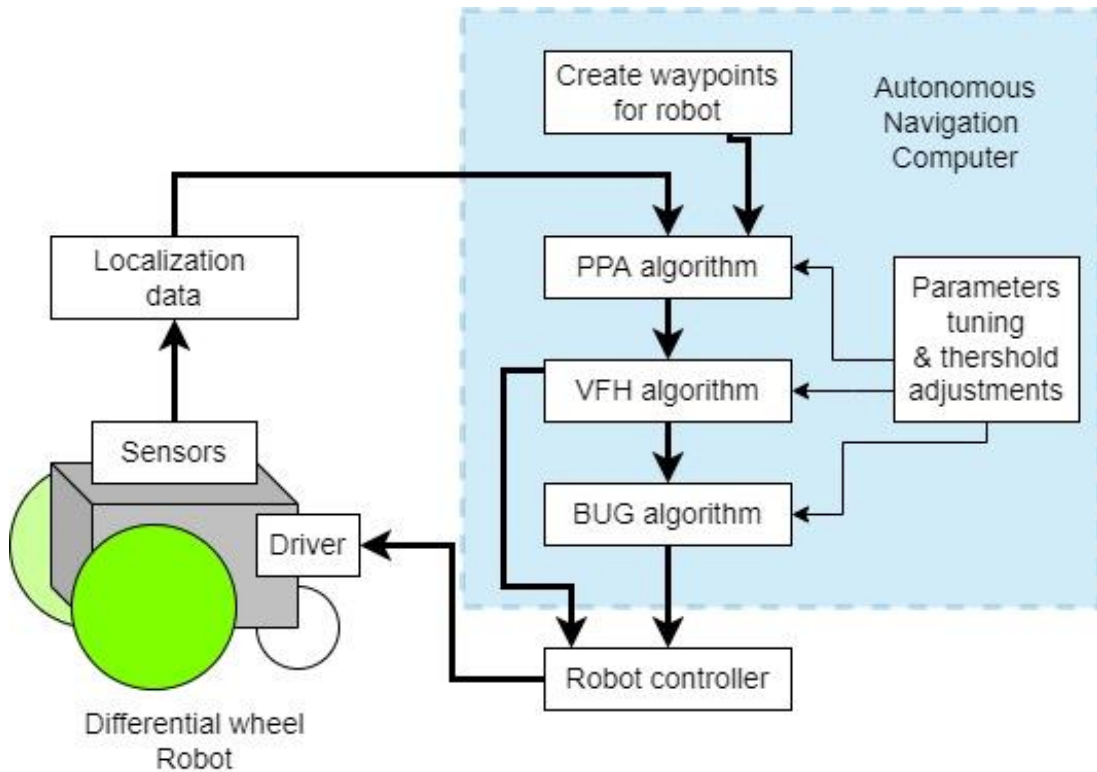


Fig 4.12 Pilot scheme diagram for PPA-VFH-BUG system.

#### 4.13 Conclusion for Hybrid Algorithm

PPA-VFH algorithm was used well in conjunction with PPA to avoid obstructions along the plantation gap. Keeping small number of waypoints reduces the mobile vehicle's setup time and oscillations. (2) Choosing an appropriate lookahead distance and velocity are important parameters which are influenced by the robot's design, path size, maneuverability, and field type. We did a basic performance assessment of our LiDAR-based navigation system based on the needs of a simplified navigation system for an autonomous irrigation robot. The 2D grid occupancy mapping provides a straightforward method for overcoming autonomous navigation with less complex sensor data analysis. The procedure may be carried out on 16/32-bit computer with minimum power usage. The range and performance of electric vehicles are largely dictated by their energy usage. This method may be useful for agricultural specific electric cars because of its low computational demand. It can increase the mileage and

energy contribution of other field operations, such as pump machines and robotic arms.

PPA-VFH-BUG Algorithm was run using a variety of LIDAR parameter settings and circumnavigation algorithm threshold angles. The optimum LIDAR range for avoiding local minima was 40 cm, which was also closer enough to plant fields to be successfully applied by pesticides. The ideal stable value for the threshold angle was less than 40 degrees, but not less than 5 degrees, since this lead to overreaction in the behavior of the robots. The approach may be used for bigger scale for actual robots and field though the experiment was done on a smaller scale. The operator may use the simulation experiment for directing the robot around an agricultural farm effectively. The technique uses fundamental mathematical procedures to lower hardware costs. Additionally, it helps ease the strain on the most sophisticated, complicated systems utilized for most autonomous navigation.

**CHAPTER-5:**  
**Conclusion and Future Scope**

## 5.1 Conclusion

- Autonomous navigation system was developed effectively by traversing all plant rows in a zigzag pattern. Different algorithms were compared for finding best low-cost application. Multiple tests were conducted to evaluate the performance of the system. The least travel oscillations were seen at a speed of 0.5 km/h, while the highest speed covered was 2.75 km/h.
- The experiment has been conducted on barriers that are straight and rectangular, but it should function better in the actual world. Plants have rough surfaces on their leaves, stems, and branches, which improve readings of laser reflecting sensors. Planer surface deflect light signals at higher angle of incidence. Ignorable objects, such as little branches or leaves, might cause sensors to respond with an overly acute proportion causing the robot to stop. This issue can be resolved by using more effective planting procedures with well-maintained inter-row spaces for mobile robot.
- Remote monitoring and control override by human operator can be used to assist the robot in overcoming hurdles. The designed algorithm can certainly work for Level 2-3 autonomy of mobile robots which are more achievable than fully autonomous robots, the operator needs to occasionally control the robots. The strategy is based on “face and drive” method like humans where obstacles are encountered in real time with low knowledge of cloud point mapping required.
- The hybrid algorithm has integrated and advanced LIDAR sensor with a classical mathematical approach to solve negation problem in agriculture robotics. There are scopes to improve the sensory readings and analyze the data to observe different shapes and types of obstacles. Small U-shaped obstacles can trap the robot, algorithms can be developed to detect and divert from such objects at early. Th algorithm can also be studied and improved to different mobile robot configuration of 4-wheel drive systems (like Bonirob and Harvey robots).

## 5.2 Future Scope

- ▶ Algorithm has been tried with LIDAR and Accurate localization information. Real life scenario may require sensor fusion of different sensors (like RTK-GPS, 3D Lidar, IMU etc) for proper localization. Study on and utilization of appropriate filters (like EKF) will be required for sensor fusion.
- ▶ Memorization algorithm of obstacles can lead to faster navigation and obstacle mitigation for multiple operations in a given area.
- ▶ Fine tuning and sensitivity need to be studied for practical applications; stray small objects can obstruct an over sensitive mobile robot.
- ▶ Fusion of advanced sensors based on LIDAR, Visual, EKF and other proximity sensor may make SLAM algorithm more robust for localisation and mapping in outdoor farm operations.
- ▶ Compatibility and tuning of the algorithm need to be tested for robots with different shape/wheel configurations of robot.

## Appendices

### Program code (for PPA-VFH-BUG Algorithm):

```
clear all;
clc;
%%%%%load image and convert to bw grid%%%%%%%%%
image = imread('mapsMoreObstacles.jpg');
grayimage = rgb2gray(image);
grayimage = imresize(grayimage,.2);

bwimage = grayimage<150;
imshow(grayimage)
map1 = binaryOccupancyMap(bwimage);
%figure('Name','Bijay Rai Robot work')
%subplot(1,2,1)
show(map1)
%%%%%%%%%

%%%%%WAY POINTS CREATION%%%%%%%%%
pathWayPoints = [
16 11;
23 55;
37 11;
53 55;
67 11;
74 55;
];

pause(0.0001);
%hold on;
%plot(pathWayPoints(:,1), pathWayPoints(:,2),'k--d')

%xlim([0 13])
%ylim([0 13])
title('Waypoints Map')
robotInitialLocation = pathWayPoints(1,:);
%%%%%WAY POINTS CREATION ENDS%%%%%%%%%

%%%%%ROBOT DIMENSION%%%%%%%%%
% Define Vehicle dimension
R = 0.05; % Wheel radius [m],0.05 original
L = 0.40; % Wheelbase [m],0.13 original
%%%%%%%%%

% %%%%%%%%%VFH SETUP%%%%%%%%%
maxRange=8;
vfh = controllerVFH;
% vfh.TargetDirectionWeight=8;
% vfh.CurrentDirectionWeight=3;
% vfh.PreviousDirectionWeight=3;

%%%%%%%%%default settings VFH+%%%%%%%%%
vfh.TargetDirectionWeight=5;
vfh.CurrentDirectionWeight=2;
```

```

vfh.PreviousDirectionWeight=2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vfh.DistanceLimits = [0.1 maxRange];
vfh.NumAngularSectors = 20;%36 preiousval
vfh.HistogramThresholds = [10 20];%5 7 default
vfh.RobotRadius = L;
vfh.SafetyDistance = L*1.1;
vfh.MinTurningRadius =1.2;%0.2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%LIDAR SETUP%%%%%%%%
lidar = LidarSensor;
lidar.sensorOffset = [0,0];
lidar.scanAngles = linspace(-pi,pi,180);
lidar.maxRange = maxRange;

viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map1';%'map'
attachLidarSensor(viz,lidar);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%log file name logger%%%%%%%%
[ctimetf] = clock;
fileName=int2str(ctime(1));
fileName=strcat(fileName,'-',int2str(ctime(2)));
fileName=strcat(fileName,'-',int2str(ctime(3)));
fileName=strcat(fileName,'-',int2str(ctime(4)));
fileName=strcat(fileName,'-',int2str(ctime(5)));
fileName1=strcat(fileName,'-',int2str(ctime(6)))
fileName=strcat(fileName1,'-', 'AgroLogger.CSV')
%L = log4m.getLogger(fileName);
%%log file name logger%%%%%%%%

%startLocation = pathWayPoints(1,:);
endLocation = pathWayPoints(2,:);
initialOrientation = 0; %initial angle of pose of robot
robotCurrentPose = [robotInitialLocationinitialOrientation]%'
intialpostion[ x, y, theta]
robot = differentialDriveKinematics("TrackWidth", 1,
"VehicleInputs", "VehicleSpeedHeadingRate");

controller = controllerPurePursuit;
controller.Waypoints = pathWayPoints;

%%%%%%%%FOR NON ADAPTIVE PPA Settings%%%%%%%%
controller.DesiredLinearVelocity = 0.8 %original 0.8
controller.MaxAngularVelocity = 1
controller.LookaheadDistance =20
%%%%%%%%FOR NON ADAPTIVE PPA Settings%%%%%%%%

LinVel=controller.DesiredLinearVelocity; %0.8 original

```



```

AngVel=controller.MaxAngularVelocity;
LookDis=controller.LookaheadDistance;

goalRadius = 0.1;
distanceToGoal = norm(robotInitialLocation - endLocation)
sampleTime = 0.1;
vizRate = rateControl(1/sampleTime);
frameSize = robot.TrackWidth/0.8;
frameSize = 10;

initPose = [robotInitialLocation(1);robotInitialLocation(2);1.57];
prePose=initPose
pose(:,1) = initPose;
msg=text(10,10,'xx')

logData=sprintf('Range:%i, VfhSectors:%i, VfhRoRad:%i,
VfhMinRad:%i,LiVel:%.1f,
LookDis:%.1f',maxRange,vfh.NumAngularSectors,vfh.RobotRadius,vfh.Min
TurningRadius,...
controller.DesiredLinearVelocity,controller.LookaheadDistance)
%L.info('Settings',logData);
startTime=tic
sumdist=0

for idx=2:8000
%for idx=2:20

delete(msg)
msg=text(20,2,num2str(idx))

prePose = pose(:,idx-1)
ranges = lidar(prePose);
id = isnan(ranges);
ranges(id) = maxRange;
ranges;
mn=mean(ranges)
md=median(ranges)
minR=min(ranges)
maxR=max(ranges)
index=idx

[vRef,wRef,lookAheadPt] = controller(prePose)

AnglesR=lidar.scanAngles;
degAngles=AnglesR*180/pi;
prePose365=wrapTo2Pi(prePose(3))
curAngle=prePose365*180/pi
%[ranges AnglesR' degAngles']

targetDir = atan2(lookAheadPt(2)-prePose(2),lookAheadPt(1)-
prePose(1))
targetDir365 =wrapTo2Pi(targetDir)
targetAngle=targetDir365*180/pi

```

```

%code for lidar scan setting -pi/2 to pi/2, 51 setting
if(prePose365>pi)
approachAngle=2*pi-prePose365+targetDir365
else
approachAngle=targetDir365 - prePose365
end

approachAngle=targetDir365 - prePose365
approachAngleDeg=approachAngle*180/pi

vfhSteer = vfh(ranges,AnglesR,approachAngle)
vfhSteerDeg = vfhSteer*180/pi;
totalSteer=vfhSteer
totalSteerDeg=totalSteer*180/pi

if ~isnan(totalSteer) && abs(totalSteer-targetDir) > 0.1
wRef = 1.0*totalSteer
%wRef = 0.5*newSteer
end

velB = [vRef;0;wRef] % Body velocities [vx;vy;w]
vel = bodyToWorld(velB,prePose) % Convert from body to world

% Perform forward discrete integration step
pose(:,idx) = prePose + vel*sampleTime;
nowPose = prePose + vel*sampleTime

%%%%%%%%%%PRINTING COMMANDS%%%%%%%%%%
loggerData1=sprintf(' ,ind:%i,mn:%.2f,md:%0.2f,minR:%0.2f,maxR:%0.2f,
CPx:%0.1f,CPy:%0.1f,CPd:%0.0f,LPx:%0.1f,LPy:%0.1f',
idx,mn,md,minR,maxR,prePose(1),prePose(2),curAngle,lookAheadPt(1),lo
okAheadPt(2))
%
% loggerData2=sprintf(' ,nowError:%.4f,preError:%.4f, S1:%.1f,
%%%%%%%%%%END OF PRINTING COMMANDS%%%%%%%%%%

xlim([0 100])
ylim([0 70])

viz(pose(:,idx),pathWayPoints,ranges)
% hold on;
% plot(lookAheadPt(1),lookAheadPt(2),'*');

distRobotToLastWaypoints=norm([nowPose(1),nowPose(2)]-
[pathWayPoints(6,1),pathWayPoints(6,2)])

if(distRobotToLastWaypoints<1 | mn<1)
k='touched'
break;
end

```

```

%distWayPoints=distance([prePose(1),prePose(2)], [pathWayPoints(:,1),
pathWayPoints(:,2)])
for i=1:6
    distWayPoints(i)=norm([nowPose(1),nowPose(2)]-
[pathWayPoints(i,1),pathWayPoints(i,2)]);
end
%distWayPoints
nearestWaypoint=min(distWayPoints)

prePos=[prePose(1),prePose(2)]
nowPose(1:2)
distStep = norm(prePos-[nowPose(1),nowPose(2)])
sumdist=sumdist+distStep

waitfor(vizRate);

end
fileName=strcat(fileName, '-', 'AgroLogger.png')
processTime=toc(startTime)
DeviationFromCSV(strcat(fileName, '-', 'AgroLogger.CSV'));
saveas(2, fileName)

return

%%%%%%%%%%%%PRM CODE%%%%%%%%%%%%
startLocation = pathWayPoints(1,:);
endLocation = pathWayPoints(2,:);
%inflate(map, robot.TrackWidth/2);
newMap=binaryOccupancyMap(bwimage(:,1:200));
prm = robotics.PRM(newMap);
prm.NumNodes = 50;
%prm.NumNodes = 200;
prm.ConnectionDistance = 100;
path = findpath(prm, startLocation, endLocation)
%distances = hypot(diff(path(:,1)), diff(path(:,2)))
%totalDistance=sum(distances)
hold on;
subplot(1,2,1);
show(prm)
%%%%%%%%%%%%PRM ENDS%%%%%%%%%%%%

```

### Program code (for PPA-VFH Algorithm):

```

clear all;
clc;
%%%%%%%%load image and convert to bw grid%%%%%%%%
image=imread('imageNoObstacle.jpg');
image=imread('imageOneObstacle3.jpg');

```

```

%image=imread('preSub1.jpg');
grayimage = rgb2gray(image);
grayimage = imresize(grayimage,.5);

bwimage = grayimage<180;
imshow(grayimage)
map1 = binaryOccupancyMap(bwimage);
%figure('Name','Bijay Rai Robot work')
%subplot(1,2,1)
show(map1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%4 ROWS WAY POINTS CREATION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pathWayPoints = [
25 20;
52 165;
110 5;
175 160;
];

pause(0.0001);
%hold on;
%plot(pathWayPoints(:,1), pathWayPoints(:,2),'k--d')

title('Waypoints Map')
robotInitialLocation = pathWayPoints(1,:);
%%%%%%%%WAY POINTS CREATION ENDS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%ROBOT DIMENSION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define Vehicle dimension
%scale assumed is 1 pixel dist=2 cms
R = 10; % Wheel radius [m],0.05 original
L = 6; % Wheelbase [m],0.13 original
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% %%%%%%%%%VFH SETUP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
maxRange=21;%20 original for conf2, 15 failed, 25 failed, 16 failed,
24 failed
vfh = controllerVFH;
% vfh.TargetDirectionWeight=8;
% vfh.CurrentDirectionWeight=3;
% vfh.PreviousDirectionWeight=3;

%%%%%%%%%%%%default settings VFH+%%%%%%%%%%%%
vfh.TargetDirectionWeight=5;
vfh.CurrentDirectionWeight=5;
vfh.PreviousDirectionWeight=2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
HistoLowThres=20;

vfh.DistanceLimits = [0.1 maxRange];
vfh.NumAngularSectors = 20;%36 preiousval
vfh.HistogramThresholds = [HistoLowThres (HistoLowThres+5)];%5 7
default
vfh.RobotRadius = L/2;
vfh.SafetyDistance = L*1.1/2;

```

```

vfh.MinTurningRadius =1.2;%0.2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%LIDAR SETUP%%%%%%%%
lidar = LidarSensor;
lidar.sensorOffset = [0,0];
lidar.scanAngles = linspace(-pi,pi,90);
lidar.maxRange = maxRange;

viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map1';%'map'
attachLidarSensor(viz,lidar);

AnglesR=lidar.scanAngles;
degAngles=AnglesR*180/pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%log file name logger%%
[ctimetf] = clock;
fileName=int2str(ctime(1));
fileName=strcat(fileName,'-',int2str(ctime(2)));
fileName=strcat(fileName,'-',int2str(ctime(3)));
fileName=strcat(fileName,'-',int2str(ctime(4)));
fileName=strcat(fileName,'-',int2str(ctime(5)));
fileName1=strcat(fileName,'-',int2str(ctime(6)))
fileName=strcat(fileName1,'-', 'AgroLogger.CSV')
L = log4m.getLogger(fileName);

%%log file name logger%%

%startLocation = pathWayPoints(1,:);
endLocation = pathWayPoints(2,:);
initialOrientation = 0; %initial angle of pose of robot
robotCurrentPose = [robotInitialLocationinitialOrientation]%'
intialpostion[ x, y, theta]
robot = differentialDriveKinematics("TrackWidth", 1,
"VehicleInputs", "VehicleSpeedHeadingRate");

controller = controllerPurePursuit;
controller.Waypoints = pathWayPoints;

%%%%%%%%FOR NON ADAPTIVE PPA Settings%%%%%%%%
controller.DesiredLinearVelocity = 2 %original 0.8
controller.MaxAngularVelocity = 4
controller.LookaheadDistance =20 %20 initial best
%%%%%%%%FOR NON ADAPTIVE PPA Settings%%%%%%%%

LinVel=controller.DesiredLinearVelocity; %0.8 original
AngVel=controller.MaxAngularVelocity;
LookDis=controller.LookaheadDistance;

```

```

goalRadius = 0.1;
distanceToGoal = norm(robotInitialLocation - endLocation)
sampleTime = 0.1;
vizRate = rateControl(1/sampleTime);
frameSize = robot.TrackWidth/0.8;
frameSize = 10;

initPose = [robotInitialLocation(1);robotInitialLocation(2);1.57];
oldPose =initPose
pose(:,1) = initPose;
msg=text(10,10, 'xx')

logData=sprintf('Range:%i, VfhSectors:%i, VfhRoRad:%i,
VfhMinRad:%i,LiVel:%.1f,
LookDis:%.1f',maxRange,vfh.NumAngularSectors,vfh.RobotRadius,vfh.Min
TurningRadius,...
controller.DesiredLinearVelocity,controller.LookaheadDistance)
L.info('Settings',logData);
headerData=sprintf(', ind,mn,md,minR,maxR,CPx,CPy,CPd,LPx,LPy');
L.info('Settings',headerData);

startTime=tic
sumdist=0

foridx=2:1000
%foridx=2:20

delete(msg)
msg=text(20,2,num2str(idx));

%prePose = pose(:,idx-1);
prePose = oldPose
ranges = lidar(prePose);
id = isnan(ranges);
ranges(id) = maxRange;
ranges;
mn=mean(ranges);
md=median(ranges);
minR=min(ranges);
maxR=max(ranges);
index=idx;

[vRef,PPARef,lookAheadPt] = controller(prePose)

prePose365=wrapTo2Pi(prePose(3))
curAngle=prePose365*180/pi;
%[ranges AnglesR' degAngles']

targetDir = atan2(lookAheadPt(2)-prePose(2),lookAheadPt(1)-
prePose(1));
const=polyfit([pathWayPoints(1,1)
pathWayPoints(2,1)],[pathWayPoints(1,2) pathWayPoints(2,2)],1)
interceptC=const(2)
CurrentInterceptC=prePose(2)-const(1)*prePose(1)

```

```

diffIntercept=interceptC-CurrentInterceptC
targetDir365 =wrapTo2Pi(targetDir)
targetAngle=targetDir365*180/pi;
approachAngle=targetDir365 - prePose365
approachAngleDeg=approachAngle*180/pi;
thresAngle=0.2;
if(approachAngle<-thresAngle || approachAngle>thresAngle)
    targetDir365=-0.7
end

vfhSteer = vfh(ranges,AnglesR,targetDir365)

vfhSteerDeg = vfhSteer*180/pi;
totalSteer=vfhSteer;
totalSteerDeg=totalSteer*180/pi;
if ~isnan(totalSteer) && abs(totalSteer-targetDir) > 0.1
    wRef = 0.8*totalSteer
    wRef = PPAwRef+1*totalSteer;
end
velB = [vRef;0;wRef]; % Body velocities [vx;vy;w]
vel = bodyToWorld(velB,prePose); % Convert from body to world
% Perform forward discrete integration step
oldPose = prePose + vel*sampleTime;
%pose(:,idx) = prePose + vel*sampleTime;
%nowPose = prePose + vel*sampleTime;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
loggerData1=sprintf('%i,%.2f,%.2f,%.2f,%.2f,%.1f,%.1f,%.0f,%.0
.1f,%.1f',
idx,mn,md,minR,maxR,prePose(1),prePose(2),curAngle,lookAheadPt(1),lo
okAheadPt(2));

%logData=strcat(loggerData1,loggerData2)
L.info('readings',loggerData1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xlim([0 70])
ylim([0 180])
%viz(pose(:,idx),pathWayPoints,ranges)
viz(oldPose,pathWayPoints,ranges)
% hold on;
% plot(lookAheadPt(1),lookAheadPt(2),'*');
distRobotToLastWaypoints=norm([oldPose(1),oldPose(2)]-
[pathWayPoints(2,1),pathWayPoints(2,2)]);
if(distRobotToLastWaypoints<5 | mn<1)
    k='touched'
break;
end
prePos=[prePose(1),prePose(2)]
oldPose(1:2)
distStep = norm(prePos-[oldPose(1),oldPose(2)])
sumdist=sumdist+distStep
waitfor(vizRate);
end
sumdist
RealDist = norm([pathWayPoints(1,1),pathWayPoints(1,2)]-
[pathWayPoints(2,1),pathWayPoints(2,2)])
return

```

## References:

- [1] T. Duckett *et al.*, “Agricultural Robotics: The Future of Robotic Agriculture,” Jun. 2018, Accessed: May 19, 2021. [Online]. Available: <http://arxiv.org/abs/1806.06762>
- [2] J. Backman, T. Oksanen, and A. Visala, “Navigation system for agricultural machines: Nonlinear Model Predictive path tracking,” *Comput. Electron. Agric.*, vol. 82, pp. 32–43, Mar. 2012, doi: 10.1016/J.COMPAG.2011.12.009.
- [3] X. Kan, T. C. Thayer, S. Carpin, and K. Karydis, “Task Planning on Stochastic Aisle Graphs for Precision Agriculture,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3287–3294, Apr. 2021, doi: 10.1109/LRA.2021.3062337.
- [4] T. S. Abhishek, D. Schilberg, and A. S. Arockia Doss, “Obstacle Avoidance Algorithms: A Review,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1012, p. 012052, 2021, doi: 10.1088/1757-899x/1012/1/012052.
- [5] J. K. Yoo and J. H. Kim, “Gaze Control-Based Navigation Architecture with a Situation-Specific Preference Approach for Humanoid Robots,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2425–2436, Oct. 2015, doi: 10.1109/TMECH.2014.2382633.
- [6] J. Borenstein, Y. Koren, and S. Member, “The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots,” *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, 1991.
- [7] J. P. Vasconez, G. A. Kantor, and F. A. Auat Cheein, “Human–robot interaction in agriculture: A survey and current challenges,” *Biosyst. Eng.*, vol. 179, pp. 35–48, Mar. 2019, doi: 10.1016/J.BIOSYSTEMSENG.2018.12.005.
- [8] A. O. Fulop and L. Tamas, “Lessons learned from lightweight CNN based object recognition for mobile robots,” *2018 IEEE Int. Conf. Autom. Qual. Testing, Robot. AQTR 2018 - THETA 21st Ed. Proc.*, pp. 1–5, Jul. 2018, doi: 10.1109/AQTR.2018.8402778.
- [9] L. R. Khot, L. Tang, S. B. Blackmore, and M. Nørremark, “Navigational context recognition for an autonomous robot in a simulated tree plantation,” *Trans. ASABE*, vol. 49, no. 5, pp. 1579–1588, 2006.
- [10] Y. Morales, A. Carballo, E. Takeuchi, A. Aburadani, and T. Tsubouchi, “Autonomous robot navigation in outdoor cluttered pedestrian walkways,” *J. F. Robot.*, vol. 26, no. 8, pp. 609–635, 2009.
- [11] Y. Koren, J. Borenstein, and others, “Potential field methods and their inherent limitations for mobile robot navigation.,” in *ICRA*, 1991, vol. 2, pp. 1398–1404.
- [12] J. Borenstein and Y. Koren, “Obstacle avoidance with ultrasonic sensors,”



*IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 213–218, 1988.

- [13] H. Noborio, T. Yoshioka, and T. Hamaguchi, “On-line deadlock-free path-planning algorithms by means of a sensor-feedback tracing,” in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, 1995, vol. 2, pp. 1291–1296.
- [14] S. M. Noorhosseini and A. S. Malowany, “GORP: a new method for mobile robot path-planning problem,” in *Mobile Robots VII*, 1993, vol. 1831, pp. 37–44.
- [15] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robot. & Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.
- [16] W. Li, “A hybrid neuro-fuzzy system for sensor based robot navigation in unknown environments,” in *Proceedings of 1995 American Control Conference-ACC’95*, 1995, vol. 4, pp. 2749–2753.
- [17] H. Najjaran and A. Goldenberg, “Real-time motion planning of an autonomous mobile manipulator using a fuzzy adaptive Kalman filter,” *Rob. Auton. Syst.*, vol. 55, no. 2, pp. 96–106, 2007.
- [18] X. Yang, R. V Patel, and M. Moallem, “A fuzzy--braitenberg navigation strategy for differential drive mobile robots,” *J. Intell. Robot. Syst.*, vol. 47, no. 2, pp. 101–124, 2006.
- [19] X. Gao *et al.*, “Review of wheeled mobile robots’ navigation problems and application prospects in agriculture,” *IEEE Access*, vol. 6, pp. 49248–49268, 2018, doi: 10.1109/ACCESS.2018.2868848.
- [20] D. G. Nedumaran and M. M, “E-Agriculture and Rural Development in India,” *SSRN Electron. J.*, Jan. 2020, doi: 10.2139/SSRN.3551994.
- [21] Y. Ji, S. Li, C. Peng, H. Xu, R. Cao, and M. Zhang, “Obstacle detection and recognition in farmland based on fusion point cloud data,” *Comput. Electron. Agric.*, vol. 189, p. 106409, Oct. 2021, doi: 10.1016/J.COMPAG.2021.106409.
- [22] J. Lopez, P. Sanchez-Vilarino, R. Sanz, and E. Paz, “Efficient Local Navigation Approach for Autonomous Driving Vehicles,” *IEEE Access*, vol. 9, pp. 79776–79792, 2021, doi: 10.1109/ACCESS.2021.3084807.
- [23] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, 2006.
- [24] J. V. Miró, T. Taha, D. Wang, and G. Dissanayake, “An adaptive manoeuvring strategy for mobile robots in cluttered dynamic environments.,” *Int. J. Autom. Control.*, vol. 2, no. 2/3, pp. 178–194, 2008.
- [25] A. M. Alajlan, M. M. Almasri, and K. M. Elleithy, “Multi-sensor based

- collision avoidance algorithm for mobile robot,” in *2015 Long Island Systems, Applications and Technology*, 2015, pp. 1–6.
- [26] M. M. Almasri, A. M. Alajlan, and K. M. Elleithy, “Trajectory planning and collision avoidance algorithm for mobile robotics system,” *IEEE Sens. J.*, vol. 16, no. 12, pp. 5021–5028, 2016.
- [27] J. Lowenberg-DeBoer *et al.*, “Lessons to be learned in adoption of autonomous equipment for field crops,” *Appl. Econ. Perspect. Policy*, vol. 44, no. 2, pp. 848–864, Jun. 2022, doi: 10.1002/AEPP.13177.
- [28] R. S. Wallace, A. Stentz, C. E. Thorpe, H. P. Moravec, W. Whittaker, and T. Kanade, “First Results in Robot Road-Following,” in *IJCAI*, 1985, pp. 1089–1095.
- [29] O. Amidi, “Integrated Mobile Robot Control,” 1990.
- [30] R. C. Coulter, “Implementation of the pure pursuit path tracking algorithm,” 1992.
- [31] A. Ollero, A. García-Cerezo, and J. L. Martínez, “Fuzzy supervisory path tracking of mobile reports,” *Control Eng. Pract.*, vol. 2, no. 2, pp. 313–319, Apr. 1994, doi: 10.1016/0967-0661(94)90213-5.
- [32] B. M. Leedy, J. S. Putney, C. Bauman, S. Cacciola, J. Michael Webster, and C. F. Reinholtz, “Virginia tech’s twin contenders: A comparative study of reactive and deliberative navigation,” in *The 2005 DARPA grand challenge*, Springer, 2007, pp. 155–182.
- [33] A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, “A system for semi-autonomous tractor operations,” *Auton. Robots*, vol. 13, no. 1, pp. 87–104, 2002.
- [34] J. Wit, C. D. Crane III, and D. Armstrong, “Autonomous ground vehicle path tracking,” *J. Robot. Syst.*, vol. 21, no. 8, pp. 439–449, 2004.
- [35] J. L. Martinez, A. Pozo-Ruz, S. Pedraza, and R. Fernandez, “Object following and obstacle avoidance using a laser scanner in the outdoor mobile robot Auriga-/spl alpha,” in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, 1998, vol. 1, pp. 204–209.
- [36] J. G. Bellingham and K. Rajan, “Robotics in Remote and Hostile Environments,” *Science (80-. )*, vol. 318, no. 5853, pp. 1098–1102, Nov. 2007, doi: 10.1126/SCIENCE.1146230.
- [37] O. Bawden, D. Ball, J. Kulk, T. Perez, and R. Russell, “A Lightweight, Modular Robotic Vehicle for the Sustainable Intensification of Agriculture”.
- [38] Y. Huang, H. Ding, Y. Zhang, ... H. W.-I. T., and undefined 2019, “A motion planning and tracking framework for autonomous vehicles based on

artificial potential field elaborated resistance network approach,” *ieeexplore.ieee.org*, Accessed: Aug. 27, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8643096/>

- [39] C. Sun, X. Zhang, Q. Zhou, Y. T.-I. Access, and undefined 2019, “A model predictive controller with switched tracking error for autonomous vehicle path tracking,” *ieeexplore.ieee.org*, Accessed: Aug. 27, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8695742/>
- [40] N. D. Tillett, T. Hague, and J. A. Marchant, “A robotic system for plant-scale husbandry,” *J. Agric. Eng. Res.*, vol. 69, no. 2, pp. 169–178, 1998, doi: 10.1006/jaer.1997.0245.
- [41] B. Zhang, C. Zong, G. Chen, B. Z.-I. Access, and undefined 2019, “Electrical vehicle path tracking based model predictive control with a laguerre function and exponential weight,” *ieeexplore.ieee.org*, Accessed: Aug. 27, 2022. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8611335/>
- [42] J. Jes’, J. Morales, J. L. Martínez, M. A. Martínez, and A. Mandow, “Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner,” *Springer*, vol. 935237, 2009, doi: 10.1155/2009/935237.
- [43] Z. Wang, Y. Bai, J. Wang, and X. Wang, “Vehicle path-tracking linear-time-varying model predictive control controller parameter selection considering central process unit computational load,” *J. Dyn. Syst. Meas. Control*, vol. 141, no. 5, p. 51004, 2019.
- [44] Q. Yao and Y. Tian, “A model predictive controller with longitudinal speed compensation for autonomous vehicle path tracking,” *Appl. Sci.*, vol. 9, no. 22, p. 4739, 2019.
- [45] M. Elbanhawi, M. Simic, and R. Jazar, “Receding horizon lateral vehicle control for pure pursuit path tracking,” *J. Vib. Control*, vol. 24, no. 3, pp. 619–642, 2018.
- [46] Y. Shan, W. Yang, C. Chen, J. Zhou, L. Zheng, and B. Li, “CF-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles,” *Int. J. Adv. Robot. Syst.*, vol. 12, no. 9, p. 134, 2015.
- [47] H. Ohta, N. Akai, E. Takeuchi, S. Kato, and M. Edahiro, “Pure pursuit revisited: field testing of autonomous vehicles in urban areas,” in *2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2016, pp. 7–12.
- [48] M.-W. Park, S.-W. Lee, and W.-Y. Han, “Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm,” in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014, pp. 1443–1447.
- [49] “Three Methods of Vehicle Lateral Control: Pure Pursuit, Stanley and MPC |

by Yan Ding | Medium.” <https://dingyan89.medium.com/three-methods-of-vehicle-lateral-control-pure-pursuit-stanley-and-mpc-db8cc1d32081> (accessed Aug. 27, 2022).

- [50] C. Hajdu, “Novel Pure-Pursuit Trajectory Following Approaches and their Practical Applications,” pp. 597–602, 2019.
- [51] J. Liu, Z. Yang, Z. Huang, W. Li, S. Dang, and H. Li, “Simulation performance evaluation of pure pursuit, stanley, LQR, MPC controller for autonomous vehicles,” *2021 IEEE Int. Conf. Real-Time Comput. Robot. RCAR 2021*, no. October, pp. 1444–1449, 2021, doi: 10.1109/RCAR52367.2021.9517448.
- [52] T. Hague, J. A. Marchant, and N. D. Tillett, “Ground based sensing systems for autonomous agricultural vehicles,” *Comput. Electron. Agric.*, vol. 25, no. 1–2, pp. 11–28, 2000.
- [53] Z. Li, C. Vigneault, and N. Wang, “Automation and robotics in fresh horticulture produce packinghouse,” 2010.
- [54] Z. Li, N. Wang, G. S. V. Raghavan, and C. Vigneault, “Ripeness and rot evaluation of ‘Tommy Atkins’ mango fruit through volatiles detection,” *J. Food Eng.*, vol. 91, no. 2, pp. 319–324, 2009.
- [55] D. C. Slaughter, D. K. Giles, and D. Downey, “Autonomous robotic weed control systems: A review,” *Comput. Electron. Agric.*, vol. 61, no. 1, pp. 63–78, 2008.
- [56] F. Rovira-Más, I. Chatterjee, and V. Sáiz-Rubio, “The role of GNSS in the navigation strategies of cost-effective agricultural robots,” *Comput. Electron. Agric.*, vol. 112, pp. 172–183, 2015, doi: 10.1016/j.compag.2014.12.017.
- [57] Y. Nagasaka, N. Umeda, Y. Kanetai, K. Taniwaki, and Y. Sasaki, “Autonomous guidance for rice transplanting using global positioning and gyroscopes,” *Comput. Electron. Agric.*, vol. 43, no. 3, pp. 223–234, 2004.
- [58] C. Zhang, N. Noguchi, and L. Yang, “Leader--follower system using two robot tractors to improve work efficiency,” *Comput. Electron. Agric.*, vol. 121, pp. 269–281, 2016.
- [59] V. Subramanian, T. F. Burks, and A. A. Arroyo, “Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation,” *Comput. Electron. Agric.*, vol. 53, no. 2, pp. 130–143, Sep. 2006, doi: 10.1016/J.COMPAG.2006.06.001.
- [60] J. B. Gerrish, “Mobile robots in agriculture,” 1984.
- [61] S. A. Hiremath, G. W. A. M. van der Heijden, F. K. van Evert, A. Stein, and C. J. F. Ter Braak, “Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter,” *Comput. Electron. Agric.*, vol.

- 100, pp. 41–50, Jan. 2014, doi: 10.1016/J.COMPAG.2013.10.005.
- [62] T. Bell, “Automatic tractor guidance using carrier-phase differential GPS,” *Comput. Electron. Agric.*, vol. 25, no. 1–2, pp. 53–66, 2000.
- [63] O. C. Barawid, A. Mizushima, K. Ishii, and N. Noguchi, “Development of an Autonomous Navigation System using a Two-dimensional Laser Scanner in an Orchard Application,” *Biosyst. Eng.*, vol. 96, no. 2, pp. 139–149, Feb. 2007, doi: 10.1016/J.BIOSYSTEMSENG.2006.10.012.
- [64] L. Jacobs, J. Weiss, and D. Dolan, “Object tracking in noisy radar data: comparison of Hough transform and RANSAC,” in *IEEE International Conference on Electro-Information Technology, EIT 2013*, 2013, pp. 1–6.
- [65] U. Weiss and P. Biber, “Plant detection and mapping for agricultural robots using a 3D LIDAR sensor,” *Rob. Auton. Syst.*, vol. 59, no. 5, pp. 265–273, May 2011, doi: 10.1016/J.ROBOT.2011.02.011.
- [66] S. Thrun, “Probabilistic robotics,” *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [67] R. S. Committee and others, “RTCM standard 10403.3 differential GNSS (global navigation satellite systems) services-version 3,” *RTCM Spec. Comm.*, no. 104, 2016.
- [68] G. Weber, D. Dettmering, and H. Gebhard, “Networked transport of RTCM via internet protocol (NTRIP),” in *A Window on the Future of Geodesy*, Springer, 2005, pp. 60–64.
- [69] M. Bakula, R. Pelc-Mieczkowska, and M. Walawski, “Reliable and redundant RTK positioning for applications in hard observational conditions,” *Artif. Satell.*, vol. 47, no. 1, p. 23, 2012.
- [70] T. Speth, A. Kamann, T. Brandmeier, and U. Jumar, “Precise relative ego-positioning by stand-alone RTK-GPS,” in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, 2016, pp. 1–6.
- [71] K. ISHII, H. TERAOKA, and N. NOGUCHI, “Studies on Self-learning Autonomous Vehicles (Part 3) Positioning System for Autonomous Vehicle,” *J. Japanese Soc. Agric. Mach.*, vol. 60, no. 1, pp. 51–58, 1998.
- [72] M. Monta, N. Kondo, and Y. Shibano, “Agricultural robot in grape production system,” in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, 1995, vol. 3, pp. 2504–2509.
- [73] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha, “Scene understanding in a large dynamic environment through a laser-based sensing,” in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 127–133.
- [74] S. Huang, T. Xia, and T. Zhang, “Passive ranging method based on infrared images,” *Infrared Laser Eng.*, vol. 36, no. 1, p. 109, 2007.

- [75] H. Wang, J. Wang, and Z. Shen, "Helicopter Pods-based Obstacle Avoidance Technology Using Infrared Imaging and Radar," *Sci. Technol. Innov. Her.*, vol. 29, pp. 56–59, 2014.
- [76] Y. Wang, J. Liu, Q. Zeng, and others, "3D environment restructure method with structured light for indoor vision/inertial navigation," *J. Chin. Inert. Technol.*, vol. 1, pp. 51–58, 2016.
- [77] N. Long, K. Wang, R. Cheng, W. Hu, and K. Yang, "Unifying obstacle detection, recognition, and fusion based on millimeter wave radar and RGB-depth sensors for the visually impaired," *Rev. Sci. Instrum.*, vol. 90, no. 4, p. 44102, 2019.
- [78] S. Ding, X. Li, S. Wei, X. Kong, and W. Yan, "Fusing Stereopsis & Corner Sparse Optical Flow for Real-time Obstacle Avoidance of Unmanned Aerial Vehicle," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, pp. 790–795.
- [79] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, Springer, 1986, pp. 396–404.
- [80] B. Krogh, "A generalized potential field approach to obstacle avoidance control," in *Proc. SME Conf. on Robotics Research: The Next Five Years and Beyond, Bethlehem, PA, 1984*, 1984, pp. 11–22.
- [81] C. Thorpe and L. Matthies, "Path relaxation: Path planning for a mobile robot," in *OCEANS 1984*, 1984, pp. 576–581.
- [82] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autom.*, vol. 2, no. 1, pp. 14–23, 1986.
- [83] J. Borenstein, Y. Koren, and S. Member, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," 1991. Accessed: Jun. 04, 2021. [Online]. Available: [http://www-personal.umich.edu/~ykoren/uploads/The\\_Vector\\_Field\\_HistogramuFast\\_Obstacle\\_Avoidance.pdf](http://www-personal.umich.edu/~ykoren/uploads/The_Vector_Field_HistogramuFast_Obstacle_Avoidance.pdf)
- [84] A. Manz, R. Liscano, and D. Green, "A comparison of realtime obstacle avoidance methods for mobile robots," in *Experimental Robotics II*, Springer, 1993, pp. 299–316.
- [85] S. M. Udupa, "Collision detection and avoidance in computer controlled manipulators," California Institute of Technology Pasadena, CA, 1977.
- [86] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 1993, pp. 802–807.
- [87] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," in *Proceedings of IEEE*

*International Conference on Robotics and Automation*, 1996, vol. 1, pp. 429–435.

- [88] T. S. Hong, D. Nakhaeinia, and B. Karasfi, “Application of fuzzy logic in mobile robot navigation,” *Fuzzy Logic-Controls, Concepts, Theor. Appl.*, pp. 21–36, 2012.
- [89] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007, doi: 10.1109/TPAMI.2007.1049.
- [90] F. Yan, Y. S. Liu, and J. Z. Xiao, “Path planning in complex 3D environments using a probabilistic roadmap method,” *Int. J. Autom. Comput.*, vol. 10, no. 6, pp. 525–533, Dec. 2013, doi: 10.1007/S11633-013-0750-9/TABLES/2.
- [91] D. Flavigné and M. Taïx, “Improving motion planning in weakly connected configuration spaces,” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 5900–5905, 2010, doi: 10.1109/IROS.2010.5650612.
- [92] S. Konduri, E. O. C. Torres, and P. R. Pagilla, “Dynamics and Control of a Differential Drive Robot with Wheel Slip: Application to Coordination of Multiple Robots,” *J. Dyn. Syst. Meas. Control. Trans. ASME*, vol. 139, no. 1, 2017, doi: 10.1115/1.4034779.
- [93] Y. Ji, S. Li, C. Peng, H. Xu, R. Cao, and M. Zhang, “Obstacle detection and recognition in farmland based on fusion point cloud data,” *Comput. Electron. Agric.*, vol. 189, no. August, p. 106409, 2021, doi: 10.1016/j.compag.2021.106409.
- [94] K. Uchiyama *et al.*, *The Gmicro/500 Superscalar Microprocessor with Branch Buffers*, vol. 13, no. 5. 1993. doi: 10.1109/40.237998.
- [95] Vayeda Anshav Bhavesh, “Comparison of Various Obstacle Avoidance Algorithms,” *Int. J. Eng. Res.*, vol. V4, no. 12, pp. 629–632, 2015, doi: 10.17577/ijertv4is120636.
- [96] M. Samuel *et al.*, “A Review of some Pure-Pursuit based Path Tracking Techniques for Control of Autonomous Vehicle,” *Int. J. Comput. Appl.*, vol. 135, no. 1, pp. 975–8887, 2016.

### **Biodata of Candidate**

1. Name: Bijay Rai
2. Father's Name: Chandra Bahadur Rai
3. Mother's Name: Manrupa Rai
4. Address: Todey Bazar, Via Jaldakha, Kalimpong, Dist: Kalimpong, Pin: 734503, West Bangal, India
5. Designation: Assistant Professor, Sikkim Manipal Institute of Technology
6. Date of Birth: 29/09/1987
7. Educational qualification:  
  
Class 10 (Indian Certificate of Secondary Education)  
Class 12 (Indian Certificate of Secondary Education)  
B.Tech (Electrical and Electronics Engineering)  
M.Tech (Power Electronics Engineering)
8. Work experience: 11 Years
9. Area of Interest: Embedded systems, microcontrollers, robotics and computer programming.

Candidate signature



### **List of Publications**

1. Bijay Rai, Amarendra Matsa, Asim Datta, “**Impact of agricultural robots towards biodiversity and the necessary regulations**”, Current Trends in Biotechnology and Pharmacy (Scopus),ISSN: 2230-7303
2. Bijay Rai, Amarendra Matsa, Asim Datta, “**Obstacle Detection for Agricultural Robot Based on Vector Field Histogram**”, Mathematical Statistician and Engineering Applications(Scopus), ISSN: 2094-0343
3. Bijay Rai, Amarendra Matsa, Asim Datta, “**Steer guidance of autonomous agricultural robot based on pure pursuit algorithm and LiDAR based vector field histogram**”,Journal of Applied Science and Engineering (ESCI, EI and Scopus) ISSN:2708-9967

### **Conferences presented.**

1. Bijay Rai, Amarendra Matsa, Asim Datta, “**Obstacle detection for agricultural robot based on vector field histogram**”, International Conference on Science, Engineering & Technology-ICSTE, 30<sup>th</sup> July 2022, Coimbatore,
2. Bijay Rai, Amarendra Matsa, Asim Datta, “**Sensing of local minima and circumnavigation of obstacles using VFH+ algorithm for agricultural robots**” International Conference on Advanced Research in Computer Science and Information Technology (ICARCSIT),Kolkata, India August 28th, 2022.



# CERTIFICATE OF PRESENTATION

International Conference on Science, Engineering &  
Technology (ICSTE)

30th July 2022, Coimbatore, India

This is to certify that ..... **Bijay Rai** .....

of ..... **Sikkim Manipal University** ..... has done his/her

**2022**

excellence in presenting the research paper titled.....

..... "Obstacle Detection For Agricultural Robot Based On Vector Field Histogram" .....

on 30th July 2022 at Coimbatore, India.

Dr. Allena Whitfield  
President



Ananya S  
Co-ordinator



## ASIAN SOCIETY FOR ACADEMIC RESEARCH

International Conference on

Advanced Research in Computer Science and Information Technology

# Certificate

This is to certify that *Bijay Rai* has presented a paper entitled "*Sensing of Local Minima and Circumnavigation of Obstacles using VFH+ Algorithm for Agricultural Robots*" at the International Conference on Advanced Research in Computer Science and Information Technology (ICARCSIT) held in Kolkata, India on 28<sup>th</sup> August, 2022.



Associated with



[www.asar.org.in](http://www.asar.org.in) | [papers.asar@gmail.com](mailto:papers.asar@gmail.com)

  
Conference Coordinator  
Asian Society for Academic Research

  
Chairman  
Asian Society for Academic Research

# Mathematical Statistician and Engineering Applications

ISSN : 2094-0343

Email : [info@ardaconference.com](mailto:info@ardaconference.com)

Website : [www.philstat.org.ph/index.php/MSEA](http://www.philstat.org.ph/index.php/MSEA) / Contact No : +91 93456 84472



MATHEMATICAL STATISTICIAN AND  
ENGINEERING APPLICATIONS  
(MSEA)  
ISSN 2094-0343

Date: 18<sup>TH</sup> Aug 2022

## Acceptance Letter

Dear Author(s): Bijay Rai, Dr. Amrendra Mahtsa, Dr. Asim Datta

Paper ID	SCIENCESOCIETY_98526
Paper Title	Obstacle detection for agricultural robot based on vector field histogram

This is to enlighten you that above manuscript reviewed and appraised by the review committee members of ARDA and it is accepted for the purpose of publication in the “**Mathematical Statistician and Engineering Applications**” with ISSN: 2094-0343 that will be available at <https://www.philstat.org.ph/index.php/MSEA> .

You have to send following documents at [event@sciencesociety.co](mailto:event@sciencesociety.co) on or before 22<sup>ND</sup> Aug 2022.

1. Final Paper | Ms Word .doc/.docx file
2. Proof of Registration | Scanned | Online Received Email

### **Note: Please read carefully**

1. Above manuscript will be published.
2. Mathematical Statistician and Engineering Applications is a SCOPUS Indexed Journal.
3. Author(s) will receive Publication information and Published Paper link through ARDA
4. You may see more about the journal at: <https://www.philstat.org.ph/index.php/MSEA>
5. You will receive Volume/ Issue information of your paper very soon.

Sincerely

Dr. Simpson Rodricks

President,  
ARDA.





**Dr Sudhakar Poda**  
Associate Editor  
Association of Biotechnology and Pharmacy  
Current Trends in Biotechnology and Pharmacy  
email:ctbpress2020@gmail.com/sudhakarpodha@gmail.com

To

**Bijay Rai, Amarendra Matsa and Asim Datta**

Department of Electrical and Electronics Engineering, Sikkim Manipal Institute of Technology, Rangpo, Sikkim-737136, India

Department of Electrical Engineering, Mizoram University, Mizoram, Aizawl-796004, India

Department of Electrical Engineering, India, Tezpur University, Tezpur, Assam-784028, India

Sub: Acceptance of your paper entitled "Impact of Agricultural Robots towards Biodiversity and the Necessary Regulations" – Reg.

Ref: Manuscript Ref. CTBP22-R9061.

---

With reference to the subject cited above, I am very happy to inform you that your MS entitled "Impact of Agricultural Robots towards Biodiversity and the Necessary Regulations" (Manuscript Ref. CTBP22-R9061) by Bijay Rai, Amarendra Matsa and Asim Datta, has been accepted for publication in Current Trends in Biotechnology and Pharmacy and the same has been scheduled for publication. I will get back to you soon after the proofs are ready.

Best regards. Yours Sincerely,

\_\_\_\_\_  
**Dr. Sudhakar Poda**  
Associate Editor  
CTBP Journal  
Association of Biotechnology and Pharmacy

## Steer Guidance Of Autonomous Agricultural Robot Based On Pure Pursuit Algorithm And LiDAR Based Vector Field Histogram

Rai Bijay<sup>1\*</sup>, Matsa Amarendra<sup>2</sup>, and Datta Asim<sup>3</sup>

<sup>1</sup> Department of Electrical and Electronics Engineering, Sikkim Manipal Institute of Technology, Rangpo, Sikkim- 737136, India

<sup>2</sup> Department of Electrical Engineering, Mizoram University, Mizoram, Aizawl- 796004, India

<sup>3</sup> Department of Electrical Engineering, India, Tezpur University, Tezpur, Assam-784028, India

\* Corresponding author. E-mail: bijay.r@smit.smu.edu.in

Received: Mar. 24, 2022; Accepted: Nov 05, 2022

The application of autonomous robots has been increasing in agriculture sector to substitute human labour and to improve the production yields. A self-sufficient robot is intended to accomplish specific jobs in different locations of the working field area, thereby an economical and effective navigation system for differential wheeled mobile robots is a paramount importance. In this paper, an autonomous navigation system of an agricultural mobile robot is proposed using pure pursuit algorithm (PPA) which is also assisted by vector field histogram (VFH). PPA autonomously guides towards waypoints, whereas the VFH algorithm helps the vehicle steer away for obstacles. The 2-dimensional light detection and ranging (LiDAR) sensors are used to monitor through the VFH algorithm. Minimum number of waypoints are set in PPA for convenience on map setup. Several indicators such as distance covered by robot, number of iterations required for completion of travel, etc., are investigated with the variable settings in PPA algorithm. Result analysis shows that mobile robot can travel at speed range of 2.5-25 km/hr with obstacle evasion which is adequate for agricultural mobile robots.

**Keywords:** agricultural autonomous robot, vector field histogram (VFH), pure pursuit algorithm, light detection and ranging (LiDAR), obstacle avoidance

© The Author(s). This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

[http://dx.doi.org/10.6180/jase.202310\\_26\(10\).0002](http://dx.doi.org/10.6180/jase.202310_26(10).0002)

### 1. Introduction

Despite being an agricultural superpower, India lacks efficient agricultural water management. Rain provides 62 percent of the water, while irrigated supplies provide 37 percent [1]. In rural places, the majority of water (about 85 percent) is generally wasted. Younger generations are gravitating toward urban-type jobs, away from agriculture, resulting in labour shortages for farmers. As a result, autonomous navigation robots will play an essential role in agriculture, filling the hole and increasing output yields. Agriculture always necessitates a large amount of water and chemicals. The traditional method of agriculture wastes a lot of these resources. Precise farming procedures help you save money by reducing resource consumption.

The use of autonomous robots for irrigation and pesticide and fertiliser application can help farmers be more precise [2].

Easy procedures, such as the application of pesticides and irrigation, need simple navigation. India's tough environment has exacerbated the difficulties of agricultural labour; exceptional midday heat waves make it challenging for farmers to work in the field. Automated machines can play a crucial role in such operational areas. A recent government survey suggested that 32 percent of the rural population of India is illiterate, compared to 15 percent of the urban population [3]. India is a delicate market where technological acceptance is not dependent just on performance, but also on simplicity of makeup, affordability, and

**Particulars of the candidate**

NAME OF THE CANDIDATE: BIJAY RAI

DEGREE: DOCTOR OF PHYLOSOPHY (Ph.D.)

DEPARTMENT: Electrical Engineering

TITLE OF THESIS: Development of Autonomous Mobile Robot for Pest Control  
Irrigation in Agriculture.

DATE OF ADMISSION: 08/08/2019

APPROVAL OF RESEARCH PROPOSAL:

1. DRC: 23/04/2020

2. BOS: 28/04/2020

3. SCHOOL BOARD: 30/04/2020

MZU REGISTRATION NO: 1904980

Ph.D. REGISTRATION NO. & DATE: MZU/Ph.D./1732 OF 08.08.2019

EXTENTION (IF ANY): N/A

HOD Signature

**ABSTRACT**  
**DEVELOPMENT OF AUTONOMOUS MOBILE ROBOT FOR PESTICIDE**  
**APPLICATION AND IRRIGATION**

**AN ABSTRACT SUBMITTED IN PARTIAL FULFILLMENT OF THE**  
**REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**

**BIJAY RAI**

**MZU REGN. NO. 1904980**

**PH.D. REGN. NO. MZU/Ph.D./1732 OF 08.08.2019**



**DEPARTMENT OF ELECTRICAL ENGINEERING**  
**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**FEBRUARY 2023**



**DEVELOPMENT OF AUTONOMOUS MOBILE ROBOT FOR PESTICIDE  
APPLICATION AND IRRIGATION**

BY

**BIJAY RAI**

Department of Electrical Engineering

Name of Supervisor: Dr. Amarendra Matsa (Mizoram University, Mizoram)

Name of Co-Supervisor: Dr. Asim Datta (Tezpur University, Assam)

Submitted

In partial fulfillment of the requirement of the Degree of Doctor of Philosophy in  
Electrical Engineering of Mizoram University, Aizawl

**Abstract:**

The deployment of autonomous robots has increased in the agricultural industry to assist human labour operations and increase production yields. Autonomous robot can work in difficult conditions of weather for very long hours. Autonomous mobile robot can be an ideal vehicle for transportation and delivery of water and spraying of pesticides as it is one of the most labour-intensive work at agricultural field. Most autonomous robot has been studied with the similar approach to autonomous passenger car like Tesla, Waymo. However, this approach is creating problem is acceptance of robots by famers. Involvement of AI, machine learning and cloud computing are simply too expensive and sophisticated technology for farmers to operate. Such systems require rigorous training of machine and data collection in suitable environment which is mostly difficult to be executed in agricultural land.

Drone technologies have been popular but has security, safety issues and well as inefficiency of delivery. The drones used are planes or multiple rotor helicopters. Delivery from air to ground is done at high rate without capability to spray at precise location of plants. Wind turbulence added by flying machine decrease the precision of operations. Drones are facing new issues of security as they can harm safety of people and infiltrate security of sensitive places, for example many north-eastern states have very less area permissible to fly drones due to presence of close international borders.

Wheeled robots can deliver at very precise locations of plants at slower rate, they are also energy efficient machines compared to drones. 2-wheel differential drive robot is a good choice for agricultural purpose due to its maneuverability and simpler mathematical kinetic model. Farm mobile robots are slow operating machines, hence their kinematic model is enough for implementation of prototypes. The robot can orient and move to any positions based on control of its two drive wheels, supportive dummy castor wheels are necessary to running 2-wheel differential drive stable. Such type of drive is simpler and cheaper to design in hardware compared to other legged and crawler robots.

In this research study an attempt to design an autonomous navigation system for a 2-wheel differential drive agricultural robot is carried out to design a system based on fast and simple geometrical algorithms using high advanced LIDAR sensors. The tracking algorithms popularly used are PPA, LQR and MPC controllers. These algorithms are compared to each other to find their suitability to my study. MPC and LQR are possible solutions for fast accurate tracking control but are only suited for ideally modelled conditions. Disturbances caused by nonlinear factors while driving the robot (like slippage or wheel lock) could make the system unpredictable and unstable. Model based algorithms also take considerable time for its operations challenging to be used in 8-16 bit micro controllers. PPA however has been favored as more suitable method due to its simplicity of computation, however it does need proper tuning of its lookahead distance based on size of robot and the farm rows.

There are various kinds of sensors and methods discussed for localization and mapping. LIDAR has been popularly used for simultaneous localization and mapping, the method is interesting however is very heavy in computation, It requires continuous 2D or 3D modelling of working space increasing load of computation and amount of memory required. To remove errors caused by faulty sensors and non-linearities of data sensed, such method also required data from several many sensors and fusion of those multiple data using Extended Kalman filter. All such intelligence added to the system increases the overall operations to the robot computer. To design a simple navigation system workable in a structured farm field a well-tuned obstacle avoidance algorithm should be sufficient.

For the operation of obstacle avoidance very popular algorithms like RRT, Dijkstra's and AVFH algorithms have been compared. Dijkstra and RRT are branching search algorithm where the path to avoid the obstacle is based on the knowledge of map and its obstacle. A global perception of the map needs to be continuously updated to such algorithms for overcoming obstacles. Based on the knowledge and map available, these algorithms will compute a path to make a new path to avoid the obstacle and reach the next waypoints in the fastest shortest route. Planning is a crucial step in such algorithms where it takes considerable time based on the size of the area handles and type of parameters set to these algorithms for tuning sensitivity. Sensitive search

algorithms in a large map take considerable time before being able to mitigate the obstacle. The constant demand for knowledge of the map of surrounding also adds to computational load. VFH on the other hand works like a human driver where his knowledge of the object face become more important than knowing the whole map. VFH is inspired for potential force strategy where the obstacle in proximity exerts repulsive certainty force. The VFH algorithm converts the cartesian map of surrounding and obstacles to a polar form having sectors. It then helps robot find the closest angle or sector to given target angle, that which is having least certainty value. The robot can take the angle to avoid obstacle. Therefore, VFH has been proposed to be used with PPA for critical decision for obstacle handling.

The pesticide spraying and irrigation requires peculiar attention even after obstacle avoidance. Obstacle are measured and understood only from the front face by a robot travelling towards it. Once obstacle detected, apart from steering away from it going to next waypoint, the proper application of water and pesticides on plants at the rear side of obstacle is very important. For efficiency in application of pesticides and water, a strategy to circumnavigate the obstacle is carried out using BUG algorithm. This strategy overcomes the problem of neglected spraying operation on plants at rear areas of obstacles. This negligence happens in most search plan algorithm which try to reach next waypoint using the shortest route possible.

Using a combined low computation hybrid PPA-VFH-BUG algorithm, the simulation was successfully carried in MATLAB program environment with acceptable results of performance. Simulation was carried out for low-speed mobile robots with travel speeds up to 2.5 km/hr.