

Istanbul Technical University
AUTOBEE Team
Intelligent Ground Vehicle Design Report

Mehmet Kabil Gültekin, Özgür Erbulan, Tanzer Atay, Giray Uzun,
Ezgi Ede

I certify that the engineering design in the vehicle by the
current student team has been significant and equivalent to
what might be awarded credit in the senior design course.



Prof. Dr. Metin GÖKAŞAN

Faculty Adviser, Autabee

Istanbul Technical University

June 2016

Contents

1	Introduction	2
2	AutoBee	2
2.1	AutoBee Project Team	2
3	AUTO-NAV CHALLENGE COMPETITION	2
3.1	Objective	2
3.2	Project Process	3
3.3	Improvements and Inovations	4
4	Mechanical Design	5
4.1	Main Considerations	5
4.2	Sizing and Designing Process	5
4.3	Mounting Process	6
5	Image Processing	7
5.1	Hardware	7
5.2	Software	7
5.2.1	ROS, OpenCV and C++	7
5.2.2	Getting image from camera via ROS	7
5.2.3	Converting ROS Image to OpenCV Image	7
5.2.4	Elimination of Gaussian Noise via Blur	8
5.2.5	Detection of White Lines	8
5.2.6	Detection Results	9
5.2.7	Image to LaserScan	10
6	Path Planning	10
6.1	Obstacle Detection	10
6.2	Path Planning	11
6.3	Mapping	12
7	Task Planner	12
8	Motor Control	13
9	Failure Modes	13
10	System Performance	13
11	Vehicle Costs	14

1 Introduction

Unmanned ground vehicles are robots that fulfill the assigned tasks without any operators and they can be remotely monitored. There is wide range of usage for this vehicles such as military applications, industry, new generation autonomous cars. They can be remotely controlled and they can maintain their operations without any intervention.

Autobee team was founded in 2015-2016 academic year and incorporated 6 members from different disciplines. The purpose of the team is by the end of the academic year accomplished full autonomous ground vehicle which can avoid the obstacles automatically on a defined path, and compete in the Intelligent Ground Vehicle Competition.

2 AutoBee

2.1 AutoBee Project Team

The team is consist of 6 members. Ezgi Ede,H. Tanzer Atay, Özgür Erbulan , Giray Uzun,M. Kabil Gültekin and Dilara İlgin study on mechanical design ,image processing , motor control, LIDAR and software, navigation , sponsorships and advertisement respectively. also the project is designated as graduation project for some of the members.

Table 1: Team Members

Student Name	Department	Project Duty
Giray Uzun	Control and Automation Engineering and Mechanical Engineering	ROS And LIDAR
Özgür Erbulan	Control and Automation Engineering	DC Motor Control
Ezgi Ede	Astronautical Engineering	Mechanical Design
H. Tanzer Atay	Control and Automation Engineering	Image Processing
M. Kabil Gültekin	Control and Automation Engineering	GPS And Navigation
Dilara İlgin	Industrial Engineering	Sponsorship and Advertisement

3 AUTO-NAV CHALLENGE COMPETITION

3.1 Objective

A fully autonomous unmanned ground robotic vehicle must negotiate around an outdoor obstacle course under a prescribed time while maintaining a minimum of speed of one mph over a section and a maximum speed limit of five mph, remaining within the lane, negotiating flags and avoiding the obstacles on the course.

The Basic Course will be primarily sinusoidal curves with series of repetitive barrel obstacles. A waypoint pair for the course will be provided prior to competition. It is mandatory for a team to complete the Basic Course to move to the Advanced Course. Maximum time for the Basic Course will be 5 minutes with a measured distance of approximately 500 feet. Pending successful running of the Basic Course the team will move on to the Advanced Course.

For the Advanced Course there will be complex barrel arrangements with switchbacks and center islands. These will be adjusted for location between runs. Direction of the obstacle course will change between runs.[1]

3.2 Project Process

Robot Operation System (ROS)[3] has been used in project and with the help of the OpenCV and ROS libraries, image processing and GPS control has been made. The flow chart of the project is given on Figure 2. Practices were implemented at in Mechatronics Education and Research Center in Istanbul Technical University. Mechatronics center provided needed motivation, equipments and the helpful study environment to the team.

We have build the vehicle on a wheelchair. There are two reason for this. Firstly, in this way, we get desired motors and batteries according to torque calculations. Secondly, we have found a sponsor who gave us a power wheelchair to use. Vehicle has 24V 500 Watt brushed DC motors which has a 12 Km/hours speed limit and 12 degree of angle of climb. Vehicle has two 12V 72A serial connected battery.

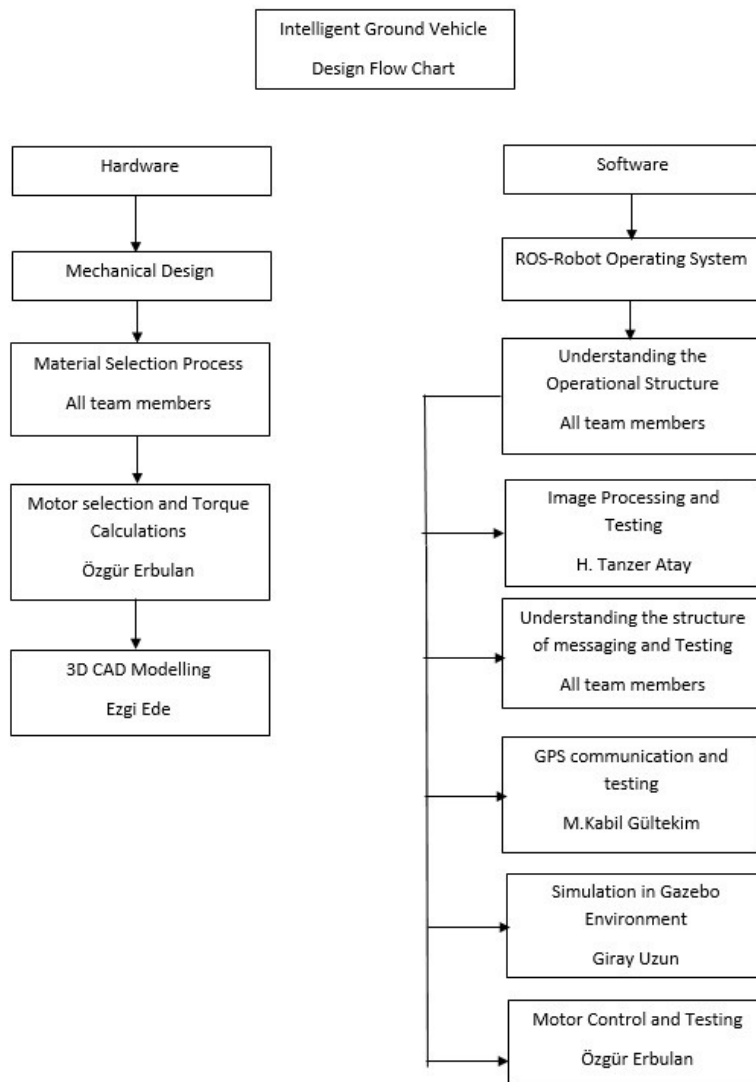


Figure 1: Flow Chart

Following figure shows the basic structure of the vehicle. LIDAR, GPS and Camera directly connected to laptop. Laptop and Arduino have bi-directional communication. Receiver, motor driver and safety light connected to arduino mega. Emergency stop cuts the power of motor driver.

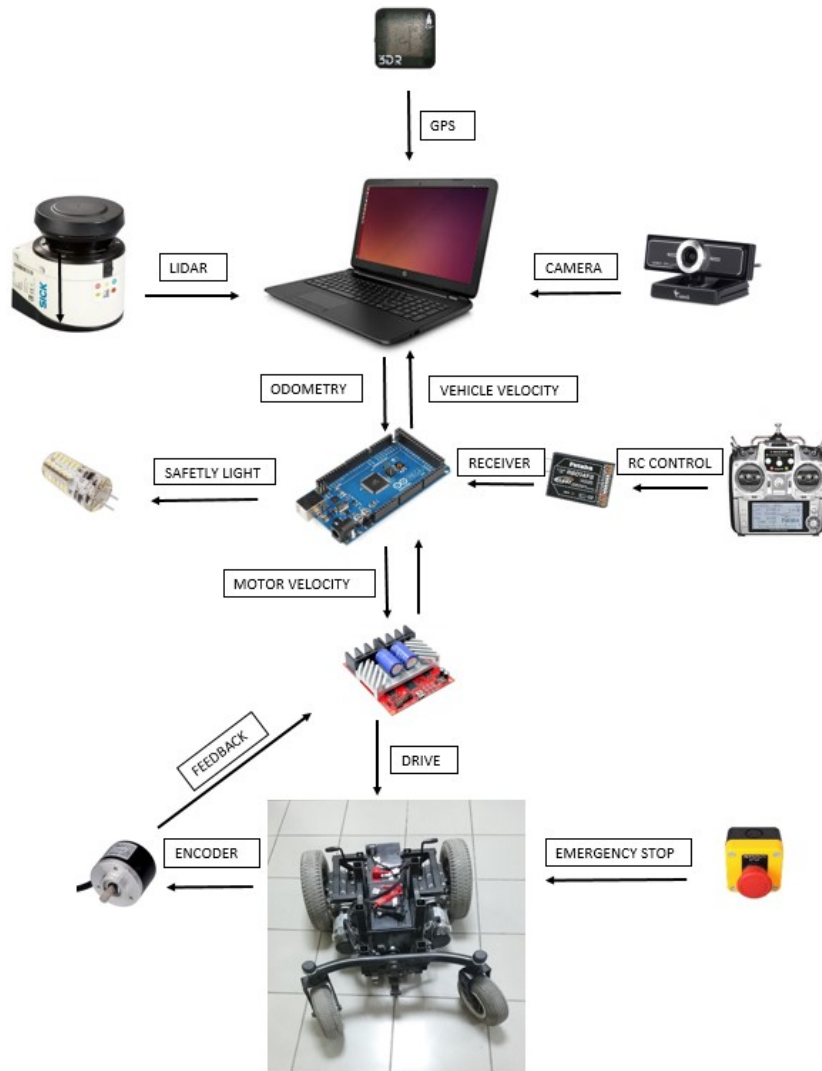


Figure 2: Hardware Scheme

3.3 Improvements and Inovations

Our team founded in 2015-2016 academic year and all developments and works started from scratch. The team developed all of the mechanical structures and the software structures without any previous vehicle as an example. Works and developments is going to explain under the chapters for each aspects of vehicle.

4 Mechanical Design

4.1 Main Considerations

For the mechanical design, main concerns are budget, material finding, mountability, visual quality and reability. Mountability is one of the issue because of the long distance travels. Vehicle's bottom part is power wheelchair and that part is receipted as lower chassis. Top parts of the vehicle is design as mountable. The size of the vehicle determined according to needed equipment and payload requirement.

4.2 Sizing and Designing Process

Vehicle has two floor. Payload, motor controller, arduino mega are on the first floor. Payload is at the center of the floor and the height of the first floor determined according to payload. Motor controller and arduino are at the front part on the floor. Floor's behind structure designed as circular shape and it is openable. CATIA used for the CAD drawings.(Figure 3)

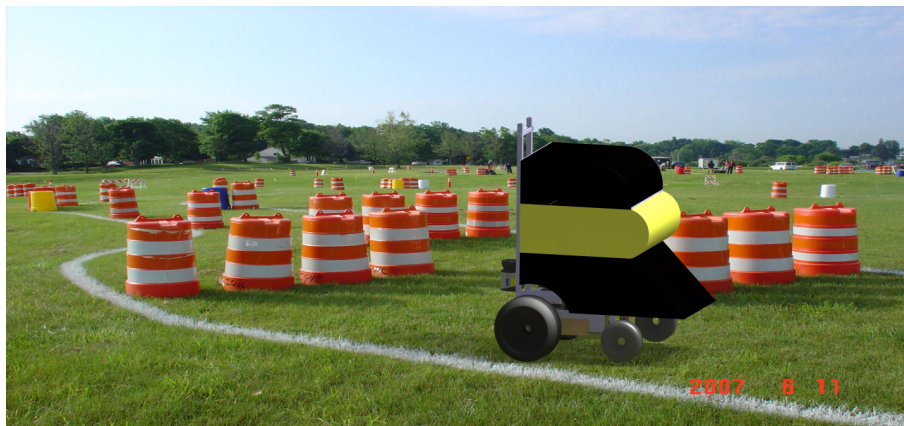


Figure 3: CATIA Rendering View

LIDAR is mounted at the front of the vehicle at a needed height and each floor supported by columns which are mountable also. At the second floor, laptop and the fan system is located. While determining the height of the floor, laptop is considered as a closed and with the fan height, floor height is designed. The back cover of the floor designed as quarter circular shape and it is also mountable and openable for attempt of intervention.(Figure 4)

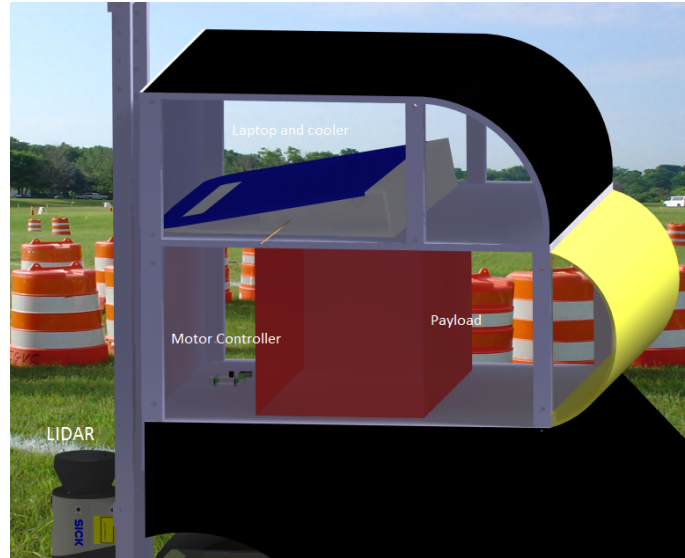


Figure 4: Equipment Locations

Visuality is an important aspect for design. Due to our university logo consist of a bee, we choose our name as Autobee and geometry of the vehicle is inspired by bee. With colors and circular edges, the vehicle achieved much more modern look.

4.3 Mounting Process

For the structure material aluminum and plexiglas are selected. For the bottom of the floors, aluminum is used. The cause of this selection is weight. It is also very easy to process for mounting. Plexiglas is selected for the side, top and back walls. Plexiglas is reliable, strong and lightweight material. The transparent color of the material gives a visual clearness. The material frequently used in vehicles and aeronautical application. (Figure 5)



Figure 5: Vehicle Montage Stage

5 Image Processing

5.1 Hardware

In the image processing, hardware is also as important as software. Because, the hardware, usually camera, is the link of the real world and computer environment. In this link, there is a tradeoff between quality and speed. If the image quality and resolution are increased, the processing time will increase or vice versa. In this project and competition, processing time is very important for the task management so this tradeoff must be optimized.

After extensive research conducted, Genius 120-degree ultra wide angle, 12MP Full HD webcam (Figure 6) were founded as proper.



Figure 6: Camera

5.2 Software

5.2.1 ROS, OpenCV and C++

OpenCV (Open Source Computer Vision Library) is an open source machine learning and computer vision software library, originally developed by Intel's research center in Nizhny Novgorod (Russia), later supported by Willow Garage and now maintained by Itseez.[2]

In consequence of the Linux support, connection between ROS via cvbridge package, C++ interfaces, good performance at real-time vision applications, the image processing procedure is carried on C++ interfaces of OpenCV libraries.

5.2.2 Getting image from camera via ROS

In the part of image processing, the robotic sensor that we used is an USB camera. There is ROS driver which name is usb_cam. This driver contains usb_cam_node interfaces with standard USB cameras using lib_usb cam and publishes images as sensor_msgs::Image.

5.2.3 Converting ROS Image to OpenCV Image

In ROS, there was a CvBridge which converts to ROS image messages to OpenCV images. The main structure is given in Figure 7.

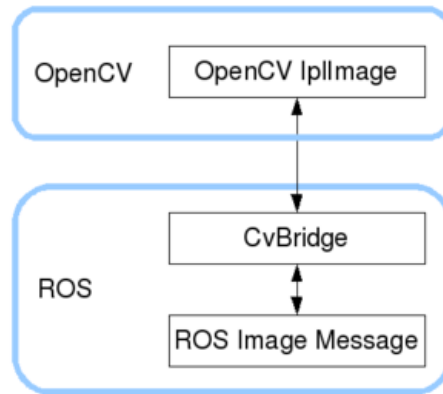


Figure 7: Converting ROS Image to OpenCV Image

5.2.4 Elimination of Gaussian Noise via Blur

Removing Gaussian Noise contains smoothing the inside salient region of an image. For this classical linear filters such as Gaussian filter reduces noise adequately but blur the edges substantially. (Figure 8)

5X5

0	0	1	0	0
0	1	2	1	0
1	2	-16	2	1
0	1	2	1	0
0	0	1	0	0

Figure 8: Gaussian filter

$$G_0(x, y) = A e^{-\frac{-(x-\mu_x)^2}{2\sigma_x} - \frac{-(y-\mu_y)^2}{2\sigma_y}}$$

5.2.5 Detection of White Lines

In literature of detecting white lines, generally, the first step contains classifying pixels as being curve point or not be attached to whether they lie on some curve in input image. A good and common curve tracing algorithm, proposed by Steger [5], used a local dealing to this issue.

- Detecting Curve Points

The most important point is that exhibit a characteristic 1D line profile in the finding lines in 2D images. The point (x, y) is considered a curve point if (p_x, p_y) falls within the pixel boundaries. And also the orientation, (x, y) , of the curve point can be derived by

$$\theta(x, y) = \tan^{-1}\left(\frac{-n_x}{n_y}\right)$$

- Combining Curve Points

The linking terminates when there is no possible curve point in the neighborhood of the current curve point. The linking process is shown in Figure 9.

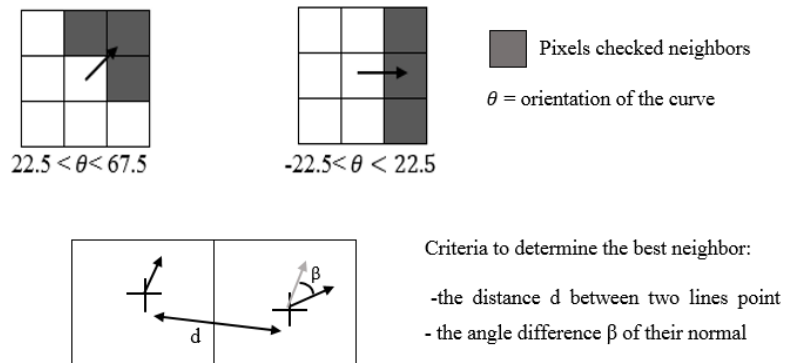


Figure 9: Linking Process

5.2.6 Detection Results

The detected image results are shown in Figure 10.

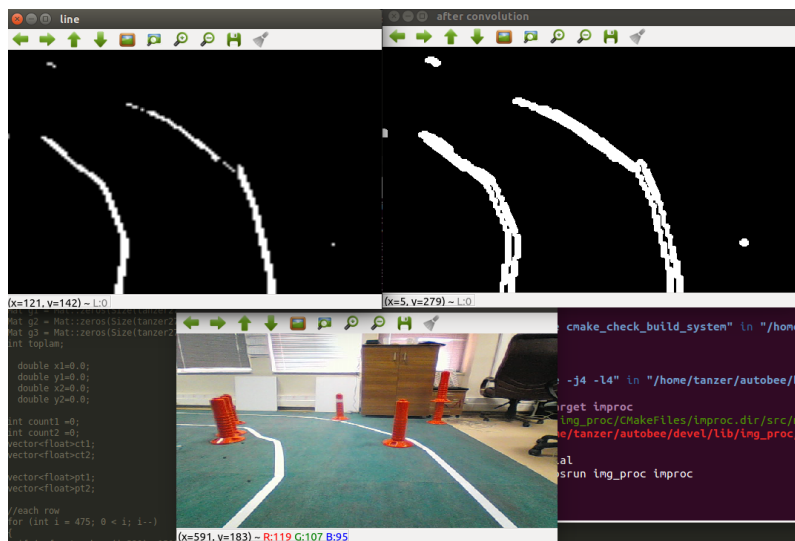


Figure 10: Detection Result

5.2.7 Image to LaserScan

After detected the lines, to eliminate conflict between laser and image data, we decided to convert image frames to LaserScan format. The results and Rviz displays are shown in Figure 11.

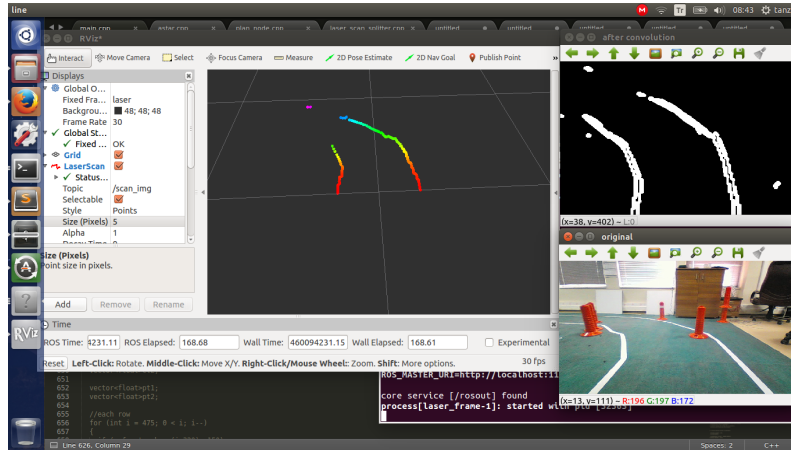


Figure 11: Image to LaserScan

6 Path Planning

6.1 Obstacle Detection

We use Sick LMS151 LIDAR for obstacle avoidance since its quality is enough for us and laboratory gave us permission to use it. Sick is a 2D laser sensor and it has 50meter range. Ground clearance of laser sensor is 30 centimeters. Height of obstacles must be high than ground clearance of laser sensor. Thus, laser sensor can detects easily the obstacles. We use Rviz for simulation of laser data. Rviz is 3D visualization tool for ROS. Following picture is the indoor working area of the team. We have simulate the IGVC path like in the photo. We have used Optitrack motion capture system for fake gps. We can see the vehicle position in the safe area and give a proper goal to go like competition via optitrack and vrpn_client node which is provided by clear path robotics. This system provides us fast and easy working environment.(Figure 14)(Figure 12)



Figure 12: Obstacle Detection

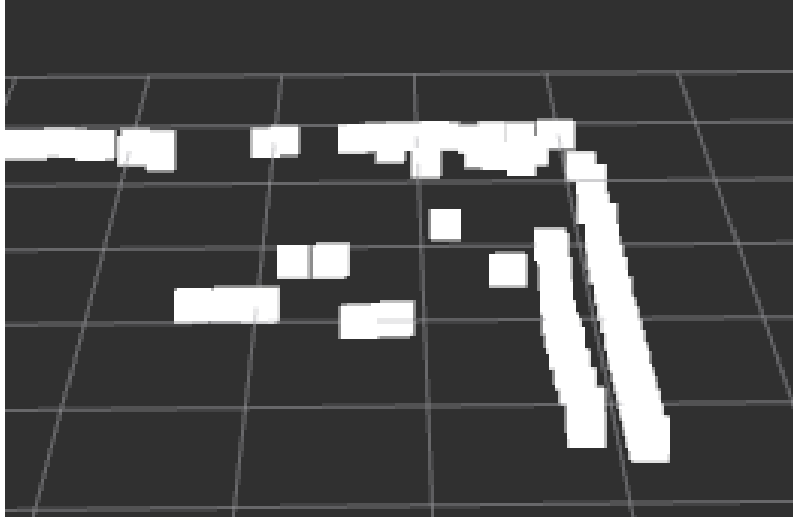


Figure 13: Obstacle Detection

6.2 Path Planning

Aim of the path planning is reach the goal with optimal cost. Path planner selects shortest path between waypoints and uses secure route which is avoidance from obstacle and stays in the lanes. We use follow the gap algorithm that published by our faculty advisor[4]. In the FGM algorithm, LIDAR detects obstacles and gaps. After the detection, biggest gaps is determined and vehicle moves to that direction. Orientation of the vehicle is controlled by proportional controller. We have also tried open source ROS global planner packets like DWA planner, Voronoi planner. Final algorithm will be selected after outdoor tries.

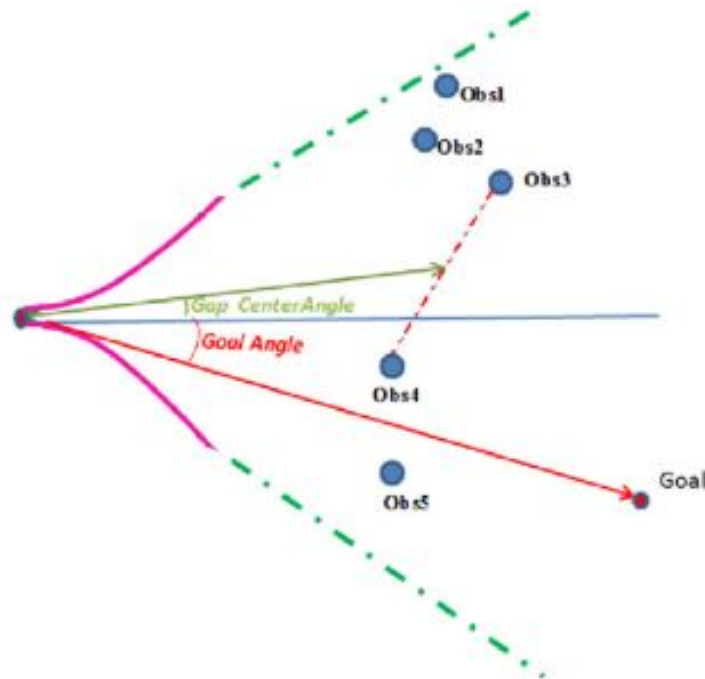


Figure 14: Follow the Gap Algorithm

6.3 Mapping

Mapping is important for obstacle avoidance because without clear map, path planner will not give good results. We use ROS open source mapping packets which are work on costmap. Costmap creates safe road for vehicle. Following figure is the mapping with planner in the mechatronics research center. Results of path is good enough for us.(Figure 15)

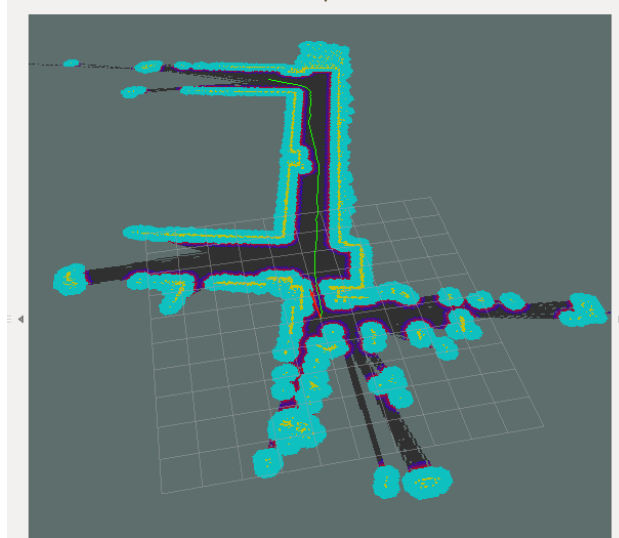


Figure 15: path planning on costmap

7 Task Planner

We had written a task planner to go to the spesific waypoint. ROS-service is used for it. There are some different service to do varied things, such as add,erase,list waypoints and start/stop services. Using method is written below:

- To add new waypoint : ROSservice call /addTask 1 2
- To erase all list : ROSservice call /deleteTask 1
- To list all waypoints : ROSservice call /List 1
- To start waypoint : ROSservice call /StartStop 1
- To start waypoint : ROSservice call /StartStop 0

Error should be smaller than to release next waypoint. We are calculating an error by

$$((currlocx - waypointlocx)^2 + (currlocy - waypointlocy)^2)^{0.5}$$

Then absolute valuse of error is important for us. Because we draw a circle to success the mission. It means if vehicle enter in our circle, vehicle reach the goal. So error is smaller than threshold, then next waypoint is given.

8 Motor Control

In the first design, we have decided to use px4 autopilot card and roboclaw motor driver. Px4 is a autopilot card that has a real-time operating system called nuttx. One of the benefits of the px4 is that it can communicate with ROS via mavROS messages and it sends GPS data to ROS master via usb connection. It provides waypoint list to use as goal. We use futaba receiver and transmitter to control the vehicle when it is not on the autonomous mode. Receiver was connected to px4 was driving the vehicle with pwm outputs. One switch is assigned to change mode of the vehicle. Basically switch off means listen the cmd_vel topic via rcoverride function and switch on means listen the RC transmitter channels to drive the car. At the same time, this switch controls the safely light. Unfortunately, autopilot and motor driver damaged at test drive in the past weeks. It was a challenge for us because neither px4 nor 60A motor drivers are available in Turkey. It takes at least weeks to get from abroad online stores. We have decided to use cyberfet motor driver which is on process of development and arduino mega board. In the present scheme arduino reads cmd_vel topics and transmit the motor speeds to the cyberfet driver by serialport.

Since we have two drive motor, we used differential drive. 400 pulse rotary encoders used to read speed. We decided to send velocity references to motor controller as final design. Motor controller reads the encoders and drive motors with closeloop. At the same time, motor controller shares the linear and angular velocities back to ROS to create odometry node.

9 Failure Modes

Since we use a gps that designed for hobby drones, lost of gps signal is a problem that we could encounter. In this situation, algorithm will stop the vehicle until gps comes back. Another problem could be bright sunshine. Even we protect the camera from sun, it may affect vision. In this case, incoming frame is regulated. In case of inability to follow the path, we will reduce upper limit of linear velocity.

10 System Performance

System performance table is given below. Our algorithm do not drive vehicle in the reverse direction, so deadends performance is less than others. Parameters are determined by both autonomous and remote control. Battery life, ramp climbing ability and motor spesifications are taken from datasheet.

Table 2: Overall System Performance

Parameter	Value
Top speed	12 km/h
Ramp climbing ability (12 Degree)	9 km/s
Reaction time	100 ms
Physical obstacles detection range	50 m @ 270 degree
Visual obstacles detection range	5 m @270 degree
Battery life	8 hours
Maximum Distance	25 km
Motor Idle Speed	4600 rpm
Accuracy of arrival at navigation waypoint	%95
Effectiveness in dealing with deadends	%90
Effectiveness in dealing with traps	%95
Effectiveness in dealing with potholes	%99
Effectiveness in dealing with center islands	%95
Effectiveness in dealing with switchbacks	%95

11 Vehicle Costs

Most of the Autobee vehicle equipment is provided by Mechatronics Education and Research Center in İstanbul Technical University and some part of the costs are provided by sponsors.

Number	Equipment	Amount (Dollar)	Cost to team (Dollar)
1	GPS	90	0
2	DC motor	500	0
2	Encoder	50	50
1	LIDAR	7000	0
1	Motor controller	300	300
2	Battery	200	0
1	Mechanic Equipment	250	250
2	Back Wheel	80	0
2	Front wheel	35	0
1	Laptop	1200	0
1	Px4 autopilot	250	0
1	E-stop	15	15
1	Cooler	35	35
1	Usb port	5	5
1	Radio Control	650	0
1	Arduino Mega	30	30
1	Camera	46	46
1	Elektronik equipment	100	100
TOTAL		10836	826

References

- [1] *The 24th Annual Intelligent Ground Vehicle Competition*. 2015.
- [2] *Itseez leads the development of the renowned computer vision library OpenCV*. 2015.
- [3] Enrique Fernández et al. *Learning ROS for Robotics Programming - Second Edition*. 2015.
- [4] Volkan Sezer and Metin Gokasan. A novel obstacle avoidance algorithm:“follow the gap method”. *Robotics and Autonomous Systems*, 60(9):1123–1134, 2012.
- [5] Carsten Steger. “An unbiased detector of curvilinear structures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1998.