# Obstacles invariant navigation of An Autonomous Robot based on GPS

**Md. Abdullah Al Ahasan, SK Alamgir Hossain,**
**Ahsan Ullah Siddiquee, Md. Mahbubur Rahman**
Computer Science and Engineering Discipline, Khulna University
Khulna, Bangladesh
ahasancse08@gmail.com, alamgir@cseku.ac.bd,
ahsansk84@yahoo.com, mahbubcse@cseku.ac.bd

***Abstract*—Robotics has a momentous features and application in our daily life. It can make our life faster easier and comfortable. Many researchers around the world are trying to develop an effective robotic system for household or industrial equipment management purposes. Global positioning System (GPS) is good for reaching high accuracy techniques to track the equipment current position. Motivating from this works in this paper we proposed the design and development of a prototype system which is based on GPS navigation and IR sensor for detecting and avoiding obstacle in dynamic context. As per the initial experimentation, we found out that the proposed approach of obstacle avoiding mechanism is appealing and useful to the user. Some of the potential applications of the proposed system include household appliances or hospital apparatus movement and children guidance.**

***Index Terms*—GPS Navigation, Autonomous Robot, Infrared sensors, Micro-controller**

## I. Introduction

Autonomous robots [1] are robots that can perform desired tasks in unstructured environments [2] without continuous human guidance [3], [4]. Many kinds of robots have some degree of autonomy. Different robots can be autonomous in different ways. A high degree of autonomy is particularly desirable in fields such as space exploration [5], cleaning floors [6], mowing lawns [7] and waste water treatment. To make a system autonomous we should consider two things without human guidance it can take decision in an unstructured environment. There are many different ways to make an autonomous robot. Some are being developed by human guidance also.

This paper is the prototype of the Obstacles invariant navigation of An Autonomous Robot based on GPS. So here we have mentioned the different types of research methodology to combine in one platform to make easier and faster communication process in the autonomous field. Our main contribution is to introduce a prototype system that is autonomous, more easier and cheap able in our surroundings.

Various types of works like hospital management, Office management and any public place job which is like the human labor intensive and repetitive job can be implemented by autonomous vehicle robot easily [8][9]. Robots are now widely used in many industries due to the high level of performance and reliability. Designing autonomous robot requires the integration of many sensors and actuators according to their task.

Obstacle detection is primary requirement for any autonomous robot [10]. The robot acquires information from its surrounding through sensors mounted on the robot. Various types of sensors can be used for obstacle avoiding. Methods of obstacle avoiding are distinct according to the use of different types of sensor. Some robots use single sensing device to detect the object. But some other robots use multiple sensing devices. The common used sensing devices for obstacle avoiding are infrared sensor [11], ultrasonic sensor [10], and laser range finder [12], charge - coupled device (CCD) [13] can be used as the detection device. Among them infrared sensor (IR) is the most suitable for obstacle avoidance because of its low cost and ranging capability. The IR object detection system consists of the sender and receiver to measure the distance from the obstacle. The unit is highly resistant to ambient light and nearly impervious to variations in the surface reflectivity of the detected object [14]. The paper is mentioned a type of IR sensors based wheeled mobile robot and mainly function as an obstacle avoidance vehicle.

Autonomous Remote controlled (RC) robot is not a new concept; many researchers already developed different prototype system for navigation purposes. One of the novel work presented by Chang et al. [15] where the author presented a detailed study on the usage of a navigation that essentially need the combination of real time kinematic (RTK) and GPS for reaching high accuracy techniques to track the vehicle current position.

Another notable work is [15] in which the author used a laptop, onboard PTZ camera, SICK laser, eight forward and eight rear sensors which detect obstacles from15 cm to7 m. Moreover, it has powerful motors and four robust wheels that can reach 8m/s speeds and carry up to 30kg. Also it can climb 45% slope and 9cm long. Koval et al. [16] proposed a prototype system that has built in OS, a common interface using OpenCV and a Point Cloud Library (PCL) for controlling autonomous ground vehicle navigation. To detect obstacles in its environment, the navigator uses the combination of a high-end laser range finder and a custom trinocular stereo camera.

978-1-4673-4836-2/12/$31.00 ©2012 IEEE

Xia et al. [17] developed a vision based approach is usually aimed at highway application but this project is fully autonomous and it works in urban environment which controls both short trips with low speed and long trios with high speed in global environment. So it is necessary for vehicle to estimate their position. Linear-parabolic model is often used to estimate the lane boundary. When a road is unmarked then it uses a color based approaches. It works based on three parameters: vehicle's lateral offset with respect to the vertical landmark, longitudinal offset with respect to the horizontal landmark, and heading. It represents the three parameters respectively. To extract white landmarks from gray roads, a 5x5 plot is used to detect edge points which provide vehicle's lateral and longitudinal offset with respect to landmarks.

Our contribution in this paper is three-fold. First, we present an approach for autonomous robotic system. Second, we introduce the GPS navigation based architectural design of the proposed system. Third, we evaluate our system in different metric and present our developed prototype's results.

The remainder of the paper is structured as follows. Section II describes our proposed system architecture and presents the different components of the system and their functional description. We developed a prototype based on our proposed approach. Section III describes the implementation detail of this prototype. We performed different experiment using this prototype and the evaluation results are in Section IV and finally we conclude and our future goal are describes in Section V.

## II. Proposed System Architecture

In this section we present the different components of the system and their functional description. The components of the system are depicted in Figure 1 as a diagram. In this diagram as a first step into the building of the autonomous system we present different parts of our whole system and its development options. We have developed three modules. These are IR obstacle detection and avoidance module: This is mainly working when an obstacle came in front of the vehicle. The working process of different modules are as follows:

- **Sensing Unit (Rx/Tx):** This unit mainly takes the information from the environment and sense whether any obstacle exists or not.
- **Micro-controller Unit (decision Unit):** Take the necessary decision according the result of the sensing unit.
- **Hardware control:** To perform the task from the command of the decision making unit.
- **Wireless Controller:** Perform the work according the input from hardware control.
- **GPS navigation module:** Navigate our vehicle according the result of the GPS. It is the combination of software and hardware module.
- **GPS device (hardware):** Here a GPS navigator is always navigates the current position of our vehicle and update the location time to time.
- **Software module (calculate shortest distance):** This gets the update information from the GPS device and calculates the shortest path. According to this algorithm shortest path direction of the vehicle mainly denotes which side it should be to turn.

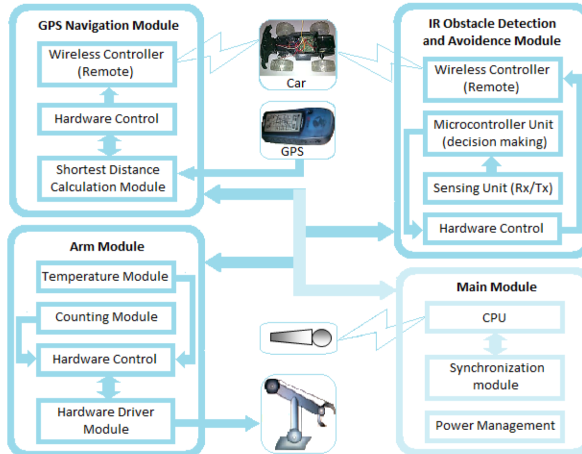


Fig. 1: Proposed system architecture

- **Hardware control:** To perform the task from the command of the software module.
- **Wireless Controller:** Perform the work according to the input from hardware controller.
- **Arm module:** This is the largest part of our system which consists of both the software and hardware modules as follows:
- **Temperature module:** Sensing the current temperature of the environment.
- **Counting Module:** Count the equipments of the currently holding or any countable materials.
- **Hardware driver module:** It is mainly stepper motor controller for the hand. Perform the task from the decision previous state.
- **Hand:** To take up or take off the materials.
- **Hardware controller:** Perform the task from the command of hardware driver module.
- **Main module:** This is the software portion and main controller Unit. In this module functional units are:
- **CPU (Central Processing Unit):** All of the software portions are working combined here.
- **Synchronization module:** Synchronize all modules according to their working priorities.
- **Power management:** The entire device take the power from this.
- **Natural Language processing Unit:** Here in this module CPU unit gets the command to start the autonomous robot instantly. This command is like “GO”, “STOP”, “LEFT”, and “RIGHT”.

We illustrate those details further in Section II-A. Afterwards, in Section II-B we describe the autonomous vehicle system and its communication interfaces to the system. Further in Section II-C, we present the rendering techniques, access control mechanism, and GPS tracking and way layout techniques. Finally, in Section II-E we present the four interaction modalities in detail.

### A. Human interaction with robotics

In modern technology has changed the robotics environment. In the past robots were predominately employed for assembly and transportation purposes in settings like factory floors, industrial robot etc. a new generation of service robot has begun to emerge, designed to assist people in everyday life [18][19] [20][21]. So the modern robot has developed their shape and functionality. In public place it is the main problem to deign such a robot that can adapt the human appearance. It is our main challenge to design such an autonomous robot that can adapt the human appearance. We have solved this problem to identify our current position. That states at first identify our location and find the environment. It may the place of Hospital, Office or a school where many types of public jobs are done. In a Hospital our autonomous robot can act as a nurse, in an office it can act as a office assistant, in a school it can act as a guardian of a school children.

### B. Vehicle tracker

There are many approaches have been developed to detect the real time vehicle on the real position [22][23][10]. To observe the color and shape [24] vehicle may be tracked. Four different features are extracted for each input frame first. In each feature space, a mean-shift estimator finds a potential target position based on the corresponding statistical model. Finally the target location is determined by dynamically fusing the tracking outputs from the four feature spaces. Feature Space is the first criteria of this paper. To detect a vehicle the distance between the model and the candidate is very important. In this work, a separate probabilistic model of the target is built in each of the following four feature spaces: HSV color space, vertical edge space, horizontal edge space, and diagonal edge space. Detecting and after calculating this features it can conclude to a decision of an upcoming vehicle. Color Feature is another criteria of this paper. First detect a vehicle and illustrate its original RGB color then transforms to HSV color model. It is difficult to detect a vehicle in high or low brightness. That's why it takes a threshold value which ignores hue pixels and extreme brightness values. And it has limitation that it appearance must be changed for its different position of the different location. Here we have used GPS interfacing with our autonomous robot vehicle for tracing the real time real position. It has many advantages over other approaches.

### C. RC controlled vehicle

The availability of the RC controlled system and its semi-autonomous feature [25] inspired us to use it. We have designed our control unit as maintaining similarity with our autonomous robot vehicle features [22]. Human operator is difficult for the controlling of a robot vehicle [26] because of time delay between the transmitter(s) and receiver(s). For the remote vehicle a term has been introduced to combine the interaction of humans with remote, intelligent, partly autonomous system is "Teleautonomous" control [27].

### D. GPS interfacing with autonomous robot vehicle

GPS based tracking autonomous vehicle is a modern approach for tracking vehicles [9]. It is a customized and improved version of Vehicle tracking system (VTS) under GPS. The cheaper design, cohesive architecture and the faster processing system makes its different from others. An application that needs real-time, fast, and reliable data processing is GPS-based vehicle tracking. It is generally build on a recently produced VTS (The Aram Locator) offering a SOC (System-on-chip) replacement of the micro-controller-based implementation. Although the micro-controller-based system has acceptable performance and cost, an FPGA-(Field Programmable Gate Array) based system can promise less cost and a more cohesive architecture that would save processing time and speeds up system interaction [28][29]. The VTS is the Aram Locator. It consists of two main parts, the Base Station (BS) and the Mobile Unit (MU). The BS consists of a PIC Microcontroller based hardware connected to the serial port of a computer. The MU is a self-contained PIC Micro-controller based hardware and a GPS module. The latter would keep track of all the positions traversed by the vehicle and records them in its memory. The system has a great storage capacity, and could perform a significant recording with a considerable sampling rate. The mobile unit (MU) of the addressed Aram Locator consists of two communicating micro-controllers interfaced with memory. There is also a GPS unit and RF transceiver. The micro-controllers make use of the same memory using a specific protocol. The system is performing properly and has a big demand in the market. However, FPGAs promise a better design, with a more cohesive architecture that would save processing time and speeds up system interaction. The two micro-controllers along with memory would be incorporated into or better supported with a high-density PLD. This will transform the hard slow interface between them into a faster and reliable programmable interconnects, and therefore makes future updates simpler. This design estimated to save a considerable percentage of the overall cost of one working unit. For a large number of demands, there would be a significant impact on production and profit. Hence, PLDs such as FPGAs are the way, for a better design in terms of upgrade-ability and speed, and it is a promising advancement for the production cost and revenue. The memory controller navigates the proper memory addressing. Multiplexers are distributed along with the controller to make the selection of the addressed memory location and do the corresponding operation.

### E. Integrated with obstacle detection and avoidance

There are two modern approaches of vehicle detection. The first approach is monocular vision [12] based on the use of specific model for vehicles and the second approach is based on the use of stereo vision which refine the coarse results obtain by the former [13]. The former is based on the processing of monocular images and the use of a specific model for vehicles. The results of the computation are fed to the latter that, conversely, is based on the use of stereo-vision and does not rely on a specific model for obstacles. Both systems have been installed and tested on ARGO (the experimental vehicle), an experimental vehicle equipped for testing vision algorithms and autonomous driving. Here we have introduced IR sensors for detecting the obstacle in real environment.

### *F. Integrated natural language processing (NLP)*

One of the common buzz word in robotics is bioinformatics. Today's researcher are trying to develop Andro Humanoid Robot [30] what can interact as like as human. We have introduced interacting with a machine in human way. Also discusses the processing of various inputs to produce an appropriate response from the machine. It uses 'Synthetic' method to make a text output command from an input speech. First it takes speech as an input and decodes phonetics from the speech. Phonetics decoding is important because it will help to find out the exact meaning of the speech. Utterance recognition will be after the finishing of decoding. Utterance test makes to recognize things identically. Such as example, The box in the center is red. For this sentence 'The box in the center' is clear to the machine but what is red. To identify these things we need utterance test. After that user focal response with vocal is also important in this gesture case [11]. Such as example, 'Turn left and Go over there. Turn left can easily be identified by the machine but when you say Go over there then it is important to take data not only from the speech you spoke but also from the direction your head turns. For this case the focal response resolution data is also important to demonstrate a perfect command. After all these taken data ALU outputs a text command for the machine to do.

## III. Implementations

This navigation algorithm has introduced and implemented on our autonomous robot. Then, once our IR sensor was defined and set with our mobile robot, we had to define quantitatively its range of detection consisting of a 3-Dimensional area in which obstacles must be detected reliably and speedily. We defined such an area with features, such as the detection distance of an obstacle. It is an autonomous vehicle system in which we have used some kinds of mechanism, by which it is able to reach its goal from current location. In this time, it uses such sensor that it is able to detect any obstacle on its path and avoid those obstacles.

This system is divided into two parts. First is to find default direction (Left or Right turn) of the vehicle using by GPS navigation for mapping the shortest path on its goal (Figure 3).Here we consider three instance of the goal as the sample in figure Goal-1, Goal-2 and Goal-3.In the figure longitude and latitude (LL). But in really it would be any place of the real coordinate system. So here figure we have supposed to three different position of the goal in different module. Here also we have showed the angle of the vehicle to turn initially to reach the goal direction that is denoted by the L for different Goal in the different position. It would be clearer when we

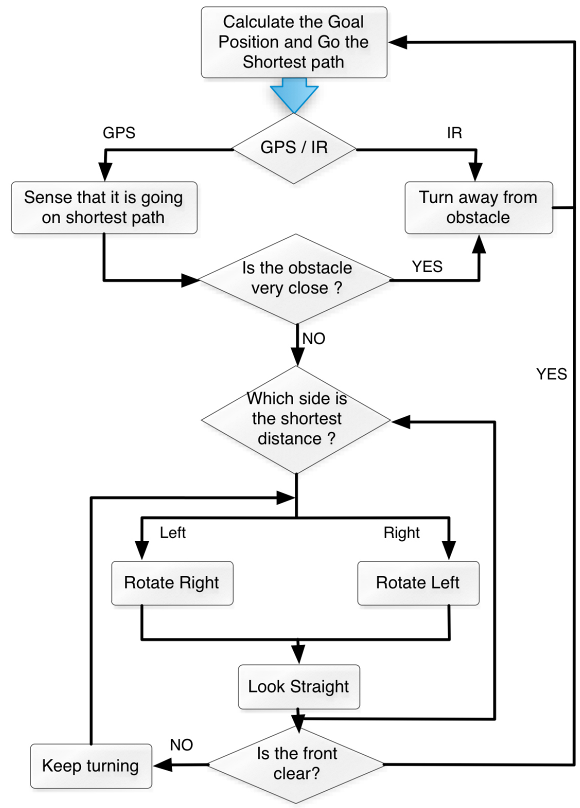


Fig. 2: The Navigation algorithm of our IR and GPS module.

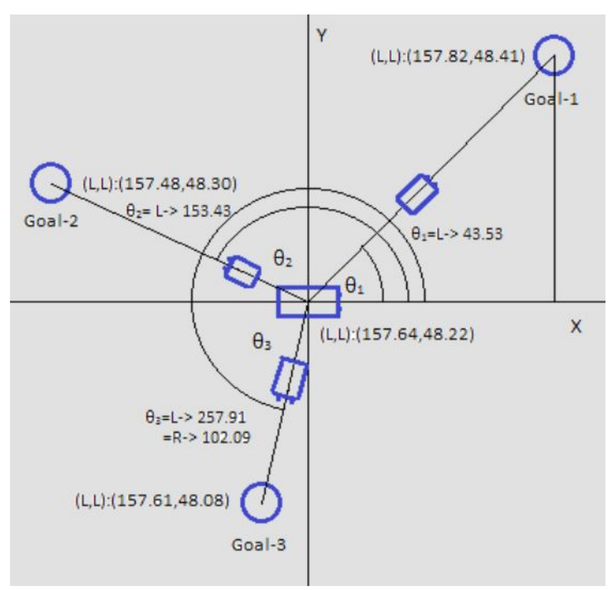


Fig. 3: The coordinate part of the start and goal State.

will observe the default direction algorithm(Figure 6) and the example of the figure of algorithm(Figure 7). Secondly, it tries to move forward, avoiding detecting preventive obstacles from front left, front right or front middle (Figure 4). The detection of the goal position and measure the distance of goal from the start position with the GPS device, viewed from the robot center, relies deeply on the intersection of the robot center line and the shortest distance of it. Calculate angle, this will depend on the rotational angles, as our GPS sensor's components are activated by these actuators. In figure 3, is the respectively the rotational angle between the center position and the goal of my final destination. The distance D is the shortest distance of the robot has to pass the path to reach the goal.

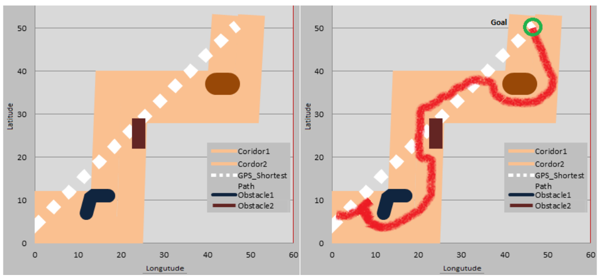


Fig. 4: A sample corridor and its start state and goal state.

## A. Default Direction Algorithm

When the vehicle changes its current location then the controller calculates the default direction continuously, on every moment. Default direction helps to make a map for the shortest path on its goal. In this study, we use one kind of blind search that is described below. To find out the default direction it needs a subtraction of two types of angle, both ranges are 0 to 359 degrees, first is the angle ( Goal) of the goal with XY axis, and second is the angle ( Car) of the moving vehicle direction with XY axis which it gets from a built-in sensor. After getting two angles, the "Default Direction" can be found using this algorithm. It turns on default direction (left or right) when the angle difference is more than 15 degrees with the goal. The vehicle works with this process continuously.

## B. Obstacle detection and avoidance algorithm

In this part, it uses four IR sensors to detect obstacles. Those sensors are placed on the vehicles front side, front left side, front right side and back side. When it tries to move forward, if any obstacle prevents its movement it tries to turn its default direction on (if there is no obstacle prevention on that default direction). If there are obstacle prevention on that default direction, it tries to turn the opposite direction on of the default direction and this turning on is a continuous process before frequently moving forward to the goal. If turning or moving is not possible then it tries to move backward, otherwise it stops. It moves backward until it can turn toward left or right (if it moves forward, it goes back to the previous location unnecessarily). Every moment, the vehicle controller calculates the default direction continuously.

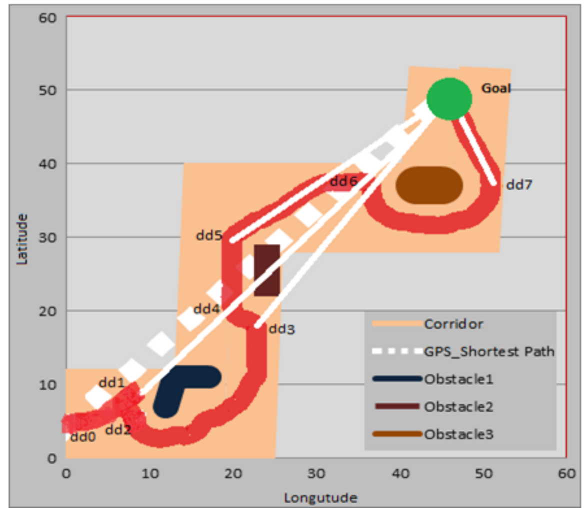


Fig. 5: The graphical part of the start and goal state.

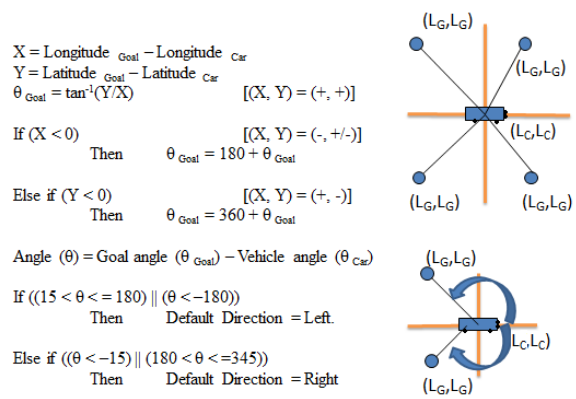


Fig. 6: Default derection algorithm.

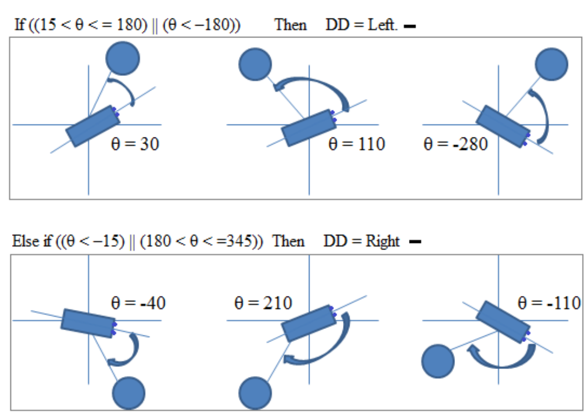


Fig. 7: Example of default derection algorithm.

TABLE I: The robot behaviors of depending on obstacles position, in case of the obstacles avoidance task.

| FS | FR | FL | SL | SR | BS | BR | BL | Obstacles avoidance |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Freely Front movement |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 Front Right Movement |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 30 Front Left Movement |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Back Straight Movement |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | Back Straight Movement |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | Back Straight Movement |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 30 Back Right Movement |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 30 Back Left Movement |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | STOP |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Front Straight Movement |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Front Straight Movement |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | Front Straight Movement |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | Front Straight Movement |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | Front Straight Movement |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | Front Straight Movement |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Front Straight Movement |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 30 Front Right Movement |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 30 Front Left Movement |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | STOP |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 30 Front Left Movement |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 30 Front Left Movement |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 30 Front Left Movement |
| 1 |  | 0 | 0 | 0 | 1 | 1 | 1 | 30 Front Left Movement |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 30 Front Left Movement |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | Front Straight Movement |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | Front Straight Movement |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 30 Front Right Movement |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 30 Front Left Movement |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 30 Front Left Movement |

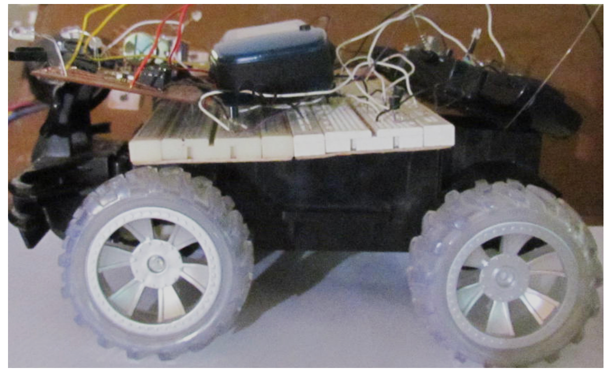

Fig. 8: The top view of our autonomous robot main part.

Note: FS - Front Straight, FR  Front Right, FL  Front Left, SL  Side Left, SR  Side Right, BS - Back Straight, BR - Back Right, BL - Back Left.

In the figure 6 the default algorithm is implemented here. This figure is explained the pseudo code of our algorithm and it is very simple and easy to understand. In the figure 7 has explained the algorithm with the example of this.

Our second part of the implementation is GPS based navigation. Here we have used Garmin product model etrex LEGEND H.GPS-based navigation in non-urban area is almost solved successfully. But in urban area GPS accuracy is drops down because of multiple signal reflection that is called multipath error. When two buildings are still close then path identity is very challenging problem. This project has done a great approach, which reduces path error by the measurement of GPS based result. We have tried to integrate of GPS and odometry information, and improved sensor model. The Unscented Kalman Filter (UKF) is the estimation of the state of a dynamic system which gives a sequence of observations and control inputs [31].

### *C. Conventional GPS sensor model*

The Global Positioning System (GPS) became a synonym for satellite-aided global localization systems. GPS currently consists of 31 satellites orbiting at about 20, 000 km providing global coverage with free access for civilian usage. Anywhere in the world at least six satellites are visible at all times. The signals of four satellites are necessary for a GPS receiver to estimate its position. There are two major factors aecting the accuracy of GPS. a) The geometric constellation of the satellites represented by a numeric value termed Dilution of Precision (DOP). b) The errors in the pseudo range estimation, referred to as user-equivalent range error. Map Generation After getting the required at least six satellites signal response from the GPS as drawing a map of the current location of our autonomous robot. The Unscented Kalman Filter (UKF) Let Xt be the state, Ut the control input, and Zt the observation at time t. Furthermore, assume that state transitions are given by a function g and observations by a function h, both corrupted by Gaussian noise. That is, Xt = g(Xt 1, Ut 1) + t , (1) Zt = h(Xt) + t , (2) Where t and t are zero-mean Gaussian

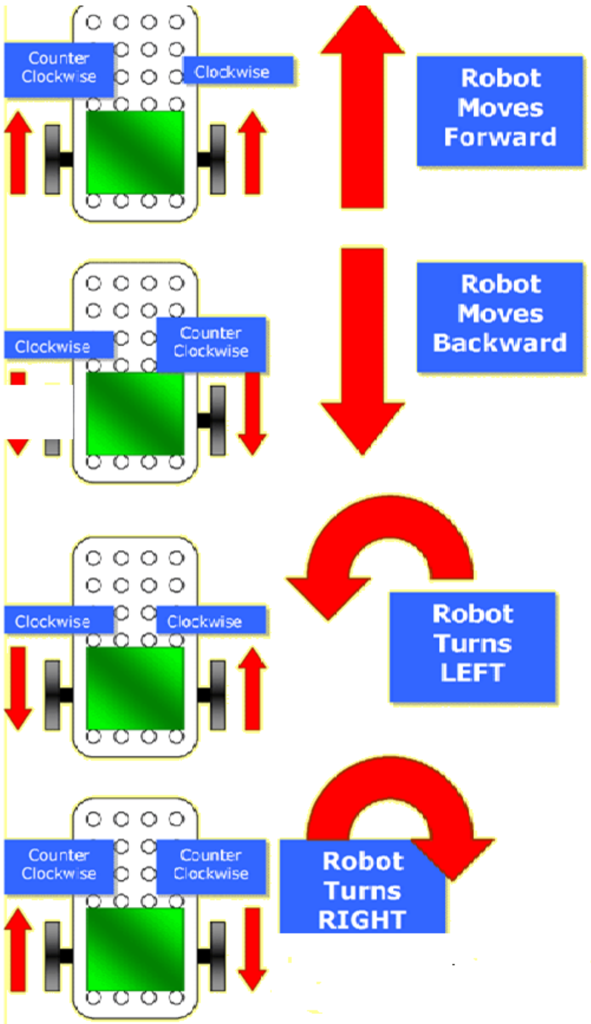


Fig. 9: Obstacle avoidance decision for the different angle of the obstacle existence.


Fig. 10: The GPS device that we have used for our autonomous robot.

noise variables with covariance Qt and Rt, respectively.) Lastly we have discussed about the natural language processing where our autonomous robot is controlled by human voice. Our functionality of the working procedure as a flowchart is described below.

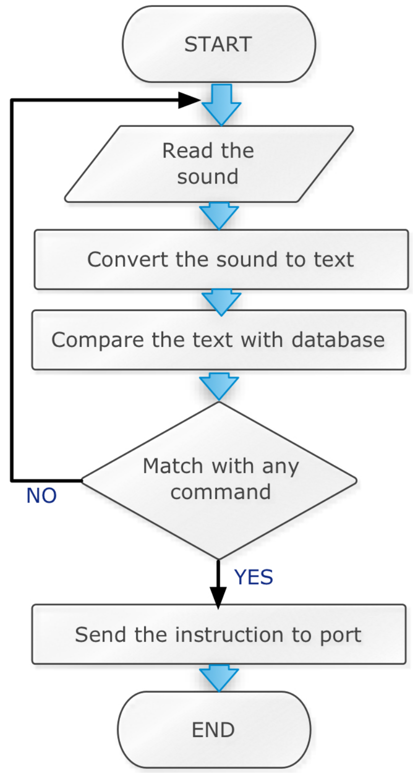


Fig. 11: Flowchart of the natural Language processing by human interaction.

In figure 9 is a sample of the rotation of a car this is explained here. It is simply show that how a car is rotate called left but it is really right. It is related to the Table I and table II.

In the figure 11 is the flowchart of the speech recognized of an autonomous robot. Here this shows how the speech SDK is implemented here to introduce the human speech recognize by an autonomous robot.

## IV. Results

In this section we discuss about our experiential result. At first controller calculate the default direction. Every obstacle positions the GPS and the IR working independently. According the priority of the calculation our autonomous vehicle has to take decision what is the right choice of the right time. Every circle position is the decision making approach where the vehicle turn the current position to take the right decision to pass shortest path as the time pass. We have taken here eight sample position where the vehicle have to take the decision in which directions is the right at the different position of the vehicle.

TABLE II: The table shows the value and the angle of the direction at the individual obstacle position result.

| Pos | Angle | FS | LS | RS | BS | DD | Turn | Move |
|---|---|---|---|---|---|---|---|---|
| dd0 | 50 | 0 | 0 | 0 | 0 | Left | Left | Forward |
| dd1 | 0 | 1 | 1 | 0 | 0 | Left | No | Toward Back |
| dd2 | -260 | 0 | 0 | 0 | 0 | Left | Right Turn | Opposite Turn of DD |
| dd3 | -30 | 1 | 0 | 1 | 0 | Right | Left Turn | Opposite Turn of DD |
| dd4 | -95 | 0 | 0 | 0 | 0 | Right | Right Turn | Forward |
| dd5 | 10 | 0 | 0 | 0 | 0 | Right | Right Turn | Forward |
| dd6 | 40 | 1 | 1 | 0 | 0 | Left | Right Turn | Opposite Turn of DD |
| dd7 | -70 | 0 | 0 | 0 | 0 | Left | Left Turn | Forward |

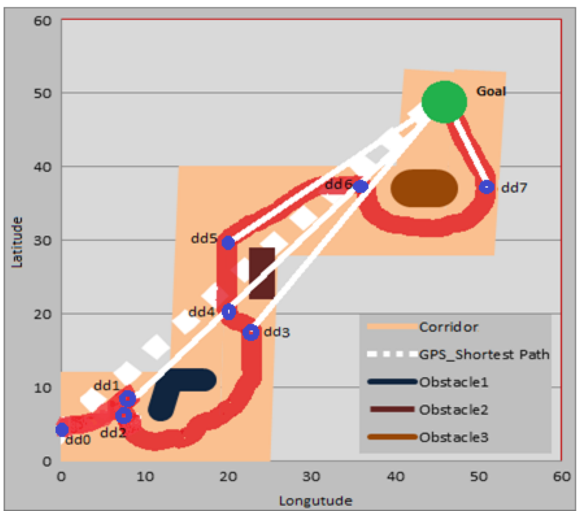


Fig. 12: Path with eight gps point sample position of a result.

- Case 0 :( dd0) Here as the = 50, this is turning point and the vehicle has to 3 times left turn to his position update according the goal direction. This is not very tough or time complexity.
- Case 1 :( dd1) Here as the = 0, this is turning point but the vehicle has recognized that his position is updated according the goal direction.
- Case 2 :( dd2) Here as the = -260, this is turning point and the vehicle has to take 6 times right turns to his position update according the goal direction. This is not very tough or time complexity.
- Case 3 :( dd3) Here as the = -30, this is turning point and the vehicle has to take 2 times right turns to his position update according the goal direction. This is not very tough or time complexity.
- Case 4 :( dd4) Here as the = -90, this is turning point and the vehicle has to take 6 times right turns to his position update according the goal direction. This is not very tough or time complexity.
- Case 5 :( dd5) Here as the = -50, this is turning point and the vehicle has to take 3 times right turns to his position update according the goal direction. This is not very tough or time complexity.
- Case 6 :( dd6) Here as the = 55, this is turning point and the vehicle has to take 6 times right turns to his position update according the goal direction. This is not very tough or time complexity.
- Case 7 :( dd7) Here as the = 35, this is last turning point and the vehicle will reach its goal position update according the goal direction. It has to take 2 times left turn to reach the Goal.

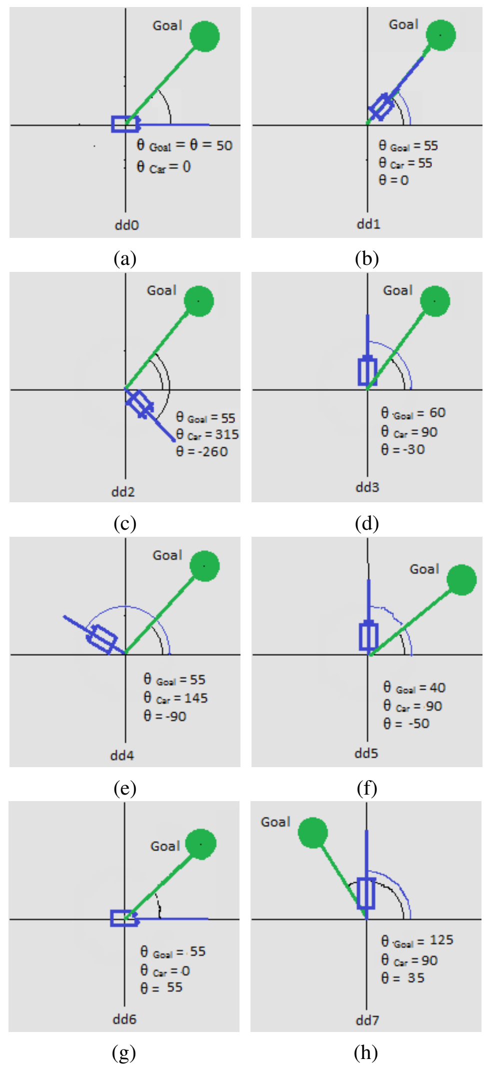


Fig. 13: The direction of goal and vehicle of the eight sample position.

The table shows the value and the angle of the direction at the individual obstacle position result.

## V. Conclusion

In this paper we presented a prototype for an autonomous robot which is based on GPS navigation. We design a new approach of autonomous robot and implement it on the real life implementation. We have developed such a system easier to control, easier to maintenance and friendly for environment. Our mainly focused of the general people adapted with the robotic environment. However our future work we wish to provide a developed version of this autonomous robot which can solve many tasks. We are working to add new hardware features into the autonomous robot to accommodate the recommendations of the users.

## REFERENCES


[1] J. Kim, Y. Kim, and K. Lee, "The third generation of robotics: Ubiquitous robot," in *Proc of the 2nd Int Conf on Autonomous Robots and Agents*. Citeseer, 2004.

[2] G. Mester, "Intelligent mobile robot motion control in unstructured environments," *Acta Polytechnica Hungarica*, vol. 7, no. 4, pp. 153–165, 2010.

[3] E. Littmann, A. Drees, and H. Ritter, "Robot guidance by human pointing gestures," in *International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*. IEEE, 1996, pp. 449–457.

[4] J. Stuckler and S. Behnke, "Following human guidance to cooperatively carry a large object," in *11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2011, pp. 218–223.

[5] D. Maluf, Y. Gawdiak, and D. Bell, "On space exploration and human error-a paper on reliability and safety," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS'05*. IEEE, 2005, pp. 79–79.

[6] L. Noakes and R. Kozera, "The lawn-mowing algorithm for noisy gradient vector fields," CITR, The University of Auckland, New Zealand, Tech. Rep., 1999.

[7] E. Arkin, S. Fekete, and J. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geometry*, vol. 17, no. 1, pp. 25–50, 2000.

[8] S. Thrun, J. Schulte, and C. Rosenberg, "Interaction with mobile robots in public places," *IEEE Intelligent Systems*, pp. 7–11, 2000.

[9] A. Yaqzan, I. Damaj, and R. Zantout, "Gps-based vehicle tracking system-on-chip," in *Proceedings of the World Congress on Engineering*, vol. 1, 2008.

[10] G. Lefaix, T. Marchand, and P. Bouthemy, "Motion-based obstacle detection and tracking for car driving assistance," in *16th International Conference on Pattern Recognition*, vol. 4. IEEE, 2002, pp. 74–77.

[11] D. Kortenkamp, E. Huber, R. Bonasso *et al.*, "Recognizing and interpreting gestures on a mobile robot," in *Proceedings of the National Conference on Artificial Intelligence*, 1996, pp. 915–921.

[12] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous systems*, vol. 32, no. 1, pp. 1–16, 2000.

[13] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Toulminet, "A cooperative approach to vision-based vehicle detection," in *IEEE Intelligent Transportation Systems*. IEEE, 2001, pp. 207–212.

[14] L. Astudillo, O. Castillo, and L. Aguilar, "Intelligent control for a perturbed autonomous wheeled mobile robot: a type-2 fuzzy logic approach," *Nonlinear Studies*, vol. 14, no. 1, p. 37, 2007.

[15] J. Chang, S. Kang, P. Liu, S. Hsieh, T. Huang, and P. Lin, "An autonomous robot equipped with the gps virtual reference station (vrs) system to perform pavement distress surveys," in *25th International Symposium on Automation and Robotics in Construction*, 2008, pp. 141–147.

[16] M. Koval, "Vision-based autonomous ground vehicle navigation," 2011.

[17] T. Xia, M. Yang, and R. Yang, "Vision based global localization for intelligent vehicles," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2006, pp. 571–576.

[18] S. Nof, *Handbook of industrial robotics*. Wiley, 1999, vol. 1.

[19] G. Lacey and K. Dawson-Howe, "The application of robotics to a mobility aid for the elderly blind," *Robotics and Autonomous Systems*, vol. 23, no. 4, pp. 245–252, 1998.

[20] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte *et al.*, "Towards personal service robots for the elderly," in *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, vol. 25, 2000, p. 184.

[21] R. Schraft and G. Schmierer, "Serviceroboter: produkte, szenarien, visionen," *Recherche*, vol. 67, p. 02, 1998.

[22] M. Betke, E. Haritaoglu, and L. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, vol. 12, no. 2, pp. 69–83, 2000.

[23] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in *Fourth IEEE Workshop on Applications of Computer Vision, WACV'98*. IEEE, 1998, pp. 8–14.

[24] K. She, G. Bebis, H. Gu, and R. Miller, "Vehicle tracking using on-line fusion of color and shape features," in *the 7th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2004, pp. 731–736.

[25] M. OLSON, "Remote control of a semi-autonomous robot vehicle over a time-delayed link," Ph.D. dissertation, University of Saskatchewan, 2001.

[26] J. Adams, "An investigation of the effects of the time lag due to long transmission distances upon remote control. phase i. tracking experiments," DTIC Document, Tech. Rep., 1961.

[27] R. V. L.Conway and M. Walker, "Teleautonomous systems: Projecting and co-ordinating itelligent action at a distance," *IEEE transaction on Robotics and Automation*, vol. 6(2), pp. 146–158, April 1990.

[28] M. Chughtai and A. Yaqoob, "An approach to task allocation for dynamic scheduling in reconfigurable computing systems," in *IEEE 9th International Multitopic Conference, INMIC*. IEEE, 2005, pp. 1–6.

[29] V. Subramanian, J. Tront, C. Bostian, and S. Midkiff, "A configurable architecture for high speed communication systems," in *Military and Aerospace Applications of Programmable Logic Devices International Conference*, vol. 3, 2002, p. E11.

[30] D. Perzanowski, A. Schultz, and W. Adams, "Integrating natural language and gesture in a robotics domain," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*. IEEE, 1998, pp. 247–252.

[31] D. Maier and A. Kleiner, "Improved gps sensor model for mobile robots in urban terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 4385–4390.