Autonomous Rover Navigation Using GPS Based Path Planning

Abul Al Arabi, Hasib Ul Sakib, Pranabesh Sarkar, Tanjina Piash Proma, Jahedul Anowar, M Ashraful Amin
Computer Vision and Cybernetics Group, Department of Computer Science and Engineering
Independent University Bangladesh, Bashundhara R/A, Dhaka, Bangladesh
email: (arabi,1520858, 1330232, 1320810, 1530887, aminmdashraful)@iub.edu.bd

***Abstract*—Nowadays, with the constant evolution of Artificial Intelligence and Machine Learning, robots are getting more perceptive than ever. For this quality they are being used in varying circumstances which humans cannot control. Rovers are special robots, capable of traversing through areas that are too difficult for humans. Even though it is a robust bot, lack of proper intelligence and automation are its basic shortcomings. As the main purpose of a rover is to traverse through areas of extreme difficulties, therefore an intelligent path generation and following system is highly required. Our research work aimed at developing an algorithm for autonomous path generation using GPS (Global Positioning System) based coordinate system and implementation of this algorithm in real life terrain, which in our case is MDRS, Utah, USA. Our prime focus was the development of a robust but easy to implement system. After developing such system, we have been able to successfully traverse our rover through that difficult terrain.**

It uses GPS coordinates of target points that will be fed into the rover from a control station. The rover capturing its own GPS signal generates a path between the current location and the destination location on its own. It then finds the deviation in its current course of direction and position. And eventually it uses Proportional Integral Derivative control loop feedback mechanism (PID control algorithm) for compensating the error or deviation and thus following that path and reach destination. A low cost on board computer (Raspberry Pi in our case) handles all the calculations during the process and drives the rover fulfilling its task using an microcontroller (Arduino).

***Keywords*—Rover and robotics, autonomous traversal, autonomous path planning, GPS path planning, GPS path follower.**

I. Introduction

Autonomous robots fundamental quality is their ability to do something on their own. Due to their ability to accomplish any task without human intervention, they are a compelling section in the field of robotics. Todays Technology is persistently shifting towards automation with less and less human interaction and achieving ease in functionality. Autonomous Rovers are exploration robots dwelling on ground in absence of human with different autonomous capabilities. Currently there are ongoing developments of autonomous vehicles. Some work has been carried out for the development of semi-autonomous robots on uneven terrains [1, 22]. There are works that have been carried out [2, 20, 21] to implement autonomous capability in case of rescue robots. There are several other works of artificial neural networks in autonomous driving domain. Some uses vision with intersection detection [4], [8], [9], some uses neural networks based on major features of specific road type [5], [6], [7] and others used anticipated road marking [10]. Some of these autonomous robots navigate using various sensors i.e. ultrasonic sensors [11] but with lower precision.

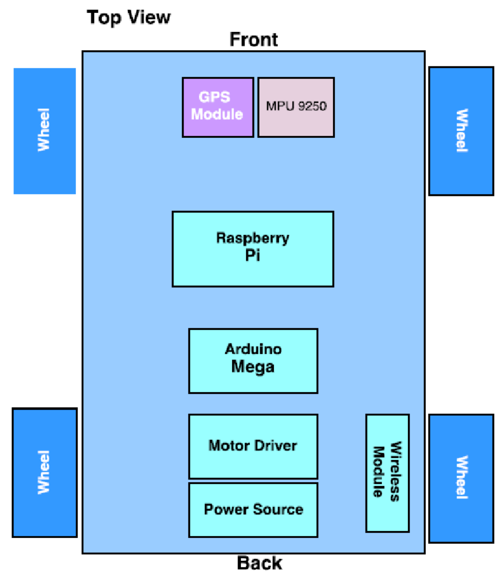

Fig. 1. Overview of the Rover

Moreover, there have been experiments regarding the application of behavioral cloning to autonomous navigation of robots in unstructured environments [12]. But the proposed method ensures a simple yet ideal and inexpensive method for navigation completely based on GPS. Now the contemporary neural network and vision based systems do give better performance and results for on-road situations, but in case of off-road scenarios GPS works better. Although by adding image processing layer with it, the whole system can be given further advancement. But in case of long path planning, only the image processing based system cant reach the destination and generate the desired path. For that situation, the GPS based global localization system is mandatory.

”University Rover Challenge” (URC) is an international premier robotics competition organized by The Mars Society annually in MDRS- Mars Desert Research Station to build next generation of Mars rover that will one-day work

2376-1172/17 $31.00 © 2017 IEEE
DOI 10.1109/AMS.2017.22

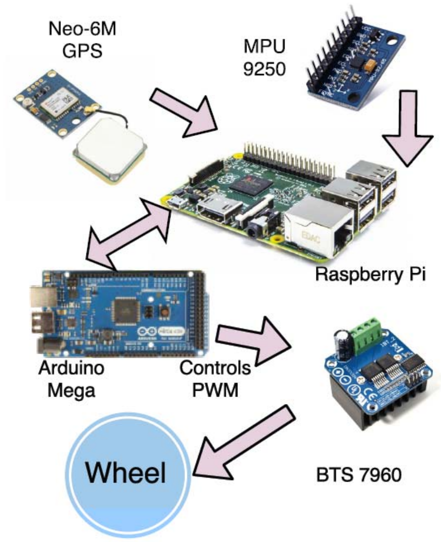

Fig. 2. Overview of System Components

alongside humans in exploration of Mars. In the main event of the competition there is a section called Autonomous Traversal where the rover must go from point to point with only the given GPS coordinates without any human interfering. This Navigation system is a brainchild to accept the challenge to navigate using GPS signals. Using the proposed system, our rover has successfully traversed on the terrain in the main event of the competition in MDRS, Utah, USA and on the terrain in Bangladesh.

II. Approach

The aim of the project is to design a simple, low cost and a straight forward way of building a navigation system for the rover to go from one point to another point solely based on GPS coordinates. The model uses very simple method of calculations which can be implemented in very small sized computers and chips such as Raspberry Pi and Arduino Mega.

The rover comprises of both mechanical and electrical system components such as controller, wheel, motors etc. It can be viewed from figure 1. The on-board computer captures GPS and some telemetry data. Having computed all the necessary components of the algorithm, the rover's on-board computer sends signal to the slave controller (Arduino in this case). It thus controls the motor drivers hence the motors. The overall system can be viewed as per figure 2. The whole system is a closed loop system with feedbacks from different sensors and GPS. The whole approach is represented in the upcoming sub sections.

A. Aspects of Navigation System

There are two main aspects of any Autonomous Navigation.

1. Current Location of Rover (Localization)

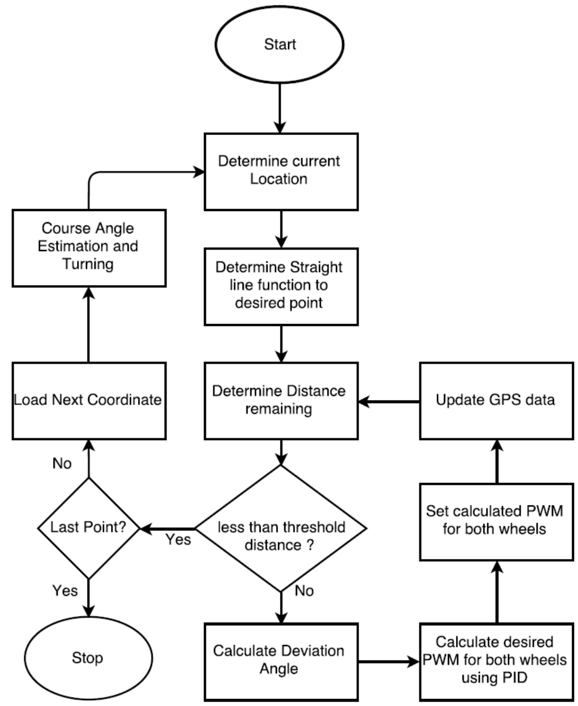

Fig. 3. Flow chart for the simplest overview

2. Heading towards the desired Location (Path formation algorithm)

These two core problems need to be solved to design a working navigation system for the rover.

1) Localization: It is the procedure of determining where the rover is currently positioned in a specific time. There are different ways of Localization and can be achieved in different ways. After registering an initial location, new position can be approximated by measuring distance covered with speed and direction of movement. But this system suffers accumulated error after some time and can deviate largely.

GPS comes in handy during localization. It has been used extensively in cars and other vehicles. The accumulated error does not concur with GPS because GPS data is not based on the previous record. But there is some trade-off. GPS has low accuracy and precision because it needs many satellites to determine current position and everything in its surrounding affects its stability. In spite of these limitations, it is highly reliable in open area and highly accurate when position correcting systems are available like WAAS (Wide Area Augmentation System), that makes it a perfect and ultra-low-cost device for our system.

2) Path Formation Algorithm: The next challenge that comes in autonomous navigation is choosing an algorithm by which the desired path will be formed to go from one point to another point. The instability of GPS readings is needed to be introduced in the algorithms because GPS data

Fig. 4. Rover traversing with autonomous system at MDRS

are always unstable in certain degrees. So, by solving this problem we can essentially build an autonomous navigation system.

Before developing the algorithm, further analysis is needed for GPS and its inner working to determine our work. To receive GPS signal, ground receiver does not need to transmit any data. It receives data from visible 7 Satellites to determine its position according to those satellites.

There are different formats of GPS signal. Two main formats are used. Decimal Degrees (ex 23.8095012 N and 90.4292258 E points somewhere in Dhaka). Other one is represented by Degree, Minute and second format ($23°48'34''$ Lat $90°25'45''$ Long) corresponds to the same location mentioned before.

The master algorithm combines these small sub systems to devise the main autonomous navigation system. The summarized and simplest representation of the algorithm can be viewed from the figure 3.

B. Mathematical Modelling

For mathematical modelling two coordinates are taken into account for calculation simplicity. This mathematical model runs in a sequential way with every two consecutive points.

Let's assume two coordinates as (Lat1,Long1) and (Lat2,Long2). (Lat represents the Latitudes and Long represents Longitudes). Let R be the Earths radius (Average Radius 6371000 meters).

1) Distance Measurement: To implement the algorithm, distance between the current location and desired location is to be measured. Although Latitude and Longitude are like x, y coordinates in the graph. But normal two-dimensional formulas cannot be used because Earth is a sphere, not a plane. To overcome this, Haversine formula is used [4]. This also has a degree of approximation as Earth is not a perfect sphere.

$$\triangle Lat = Lat2 - Lat1 \tag{1}$$

$$\triangle Long = Long2 - Long1 \tag{2}$$

Now the Haversine of central angle [4] between those 2 points is found with the following equation 3.

$$\begin{aligned} havAngle =& sin^2(\triangle Lat/2) \\ &+ cos(Lat1) * cos(Lat2) * sin(\triangle Long/2) \end{aligned} \tag{3}$$

Now the distance between the two points can be found from 4.

$$D_{pointA\ to\ pointB} = R * k \tag{4}$$

$PointA$ and $PointB$ has the coordinates of $Lat1, Long1$ and $Lat2, Long2$ respectively, where the value of k can be found from equation 5.

$$k = 2 * atan2(\sqrt{havAngle}, \sqrt{1 - havAngle}) \tag{5}$$

Where the function $atan2$ is Four-quadrant inverse tangent.

2) Deviation Measurement: After that, an algorithm can be conceptualized whenever the rover is not on any point on the path line between those two given coordinates. When the rover is not on the path line, deviation from the path is measured in the form of distance. This deviation is treated as the error that will be fed into the PID algorithm later. There can be two types of deviation. Suppose the rover is on the left side of the path line between the points or on the right side. This creates complexities in the four quadrant system as the perception of left or right is essentially different in those different quadrants.

To eradicate this problem an approach emphasizes on using a relative deviation rather than absolute left or right.

In this case a relative deviation angle between the rover's current heading and the path line's course direction is measured. Previously mentioned equation 4 can be used to determine the distance of the rover from the start point of the current line. These two data can be incorporated to determine the current deviation of the rover.

If the relative deviation angle with respect to the path line to be followed be $\triangle\theta_r$ then,

$$\triangle\theta_r = atan2(\frac{Lat_e - Lat_s}{Long_e - Long_s}) - atan2(\frac{Lat_r - Lat_s}{Long_r - Long_s}) \tag{6}$$

Here the subscript $e,\ s,\ r$ stands for the GPS coordinates $endpoint,\ startingpoint,\ roverpoint$ respectively. Now if the deviation from the path be $\triangle d_{line}$ then it is found from equation 7.

$$\triangle d_{line} = sine(\triangle\theta_r)\ *\ D_{StartPoint\ to\ Rover} \tag{7}$$

Here $D_{StartPoint\ to\ Rover}$ is the distance of the rover from the start point of the current line. This equation finds the perpendicular distance of the rover from the line and has a positive value if the rover is on the relative right side with respect to the course direction and negative otherwise. This essentially acts as the $error$ value for the PID algorithm to function.

3) PID control for Deviation Compensation: After these calculations, a simple algorithm can be devised where the deviation of location of the rover from the straight line can be sent to a PID function to determine the speed of both the right wheel and left wheel to adjust the course so that the rover heads towards the straight line and follow that straight line.

The compensation value can be found from equation 8.

$$\begin{aligned} error &= \triangle d_{line} \\ \triangle error &= error - previous\ error \\ p &= kp\ *\ error \\ d &= kd\ *\ \frac{\triangle error}{\triangle time} \\ i &= ki\ *\ \sum_n error \\ e_{compensation} &= p + d + i \end{aligned} \tag{8}$$

Here $kp,\ kd,\ ki$ are the PID proportionality constants. The motor PWM (pulse width modulation) speed controlling is done by adjusting the compensation value, $e_{compensation}$, with the $base\ speed$ of the motors. Base speeds are the PWM values when there is no deviation from the line. The left and right side motors' PWM value can be found by the adjustment as per equation 9.

$$\begin{aligned} Right\ PWM &= base\ speed + e_{compensation} \\ Left\ PWM &= base\ speed - e_{compensation} \end{aligned} \tag{9}$$

These two PWM values are constrained in between a max value and min value to avoid excessive sharp turns and to provide a minimum torque respectively.

4) Course Angle Estimation: After reaching the destination of current end-point, the next-straight line equation needs to be calculated. Since the course direction of the current line and next line to be followed is essentially different hence the rover needs to align to the new course direction. The deviation angle between two successive straight line is found as per equation 11.

$$\begin{aligned} &deviation\ angle = \\ &new\ course\ direction - current\ course\ direction \end{aligned} \tag{10}$$

If the difference in next point latitude and current end point latitude is $\triangle Lat_{new}$ and the difference in current end point latitude and starting point latitude is $\triangle Lat_{old}$ and so on, then this equation can be summarized as,

$$\begin{aligned} &\triangle\theta = \\ &tan^{-1}\frac{\triangle Lat_{new} * \triangle Long_{old} - \triangle Lat_{old} * \triangle Long_{new}}{1 + \triangle Lat_{new} * \triangle Lat_{old}} \end{aligned} \tag{11}$$

The rover's on-board compass then finds the rover's current heading and hence sets the motors of one side to rotate clockwise and others counter clockwise to fix into the new course direction.

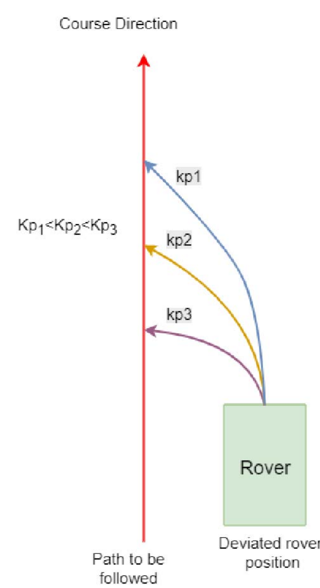

Fig. 5. Effect of kp on Rover's navigation

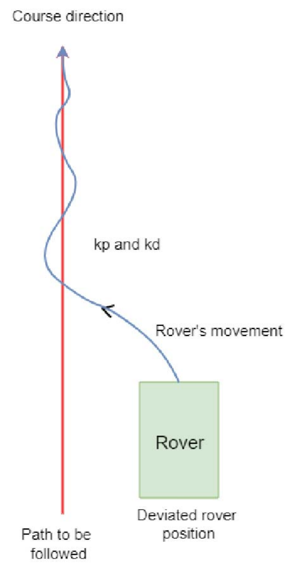

Fig. 6. Compensating deviation with both kp and kd

III. Implementation and Field Testing

This whole algorithm has been implemented on a custom-made rover system. The rover weight was approximately 34 kilograms. It had a dimension of 3ft*2.5ft approximately. The four-wheel system with a counter-link mechanism as suspension system has been able to traverse through extreme rough terrains.

The control system comprises of a central computing system (Raspberry Pi-3 Model B in our case), high current motor drivers, GPS device (Ublox Neo 6m), MPU9255 as accelerometer and gyro sensor, current feed-backs from individual motors and this whole system has been powered via Lithium Polymer Batteries with a capacity of 20,000 mAh that made the rover capable of running at least 1 hour.

The motors have a RPM of 60 with 32 kg-cm torque. The MPU9255 is a 9 DOF (degree of freedom) chip (3 DOF compass, 3 DOF gyro, 3 DOF accelerometro). Hence it has an excellent capability of calculating current heading with a great degree of accuracy. For compensating error while generating heading that arises due to pan and tilt position of the sensor, sensor fusion algorithm has been used that incorporates different sensors with the compass [3].

As a terrain we have used two places. One in Purbachal, Dhaka, Bangladesh and another was at ”MDRS”, Utah, USA. In those two place the GPS coordinates was essentially different along with the N, S, E, W of the GPS system. The coordinates of different points are fed into our GPS map that is on the control station side. These coordinates are then transmitted over the wireless communication to the rover. The raspberry Pi on the rover then accumulates those coordinates and runs the rover.

IV. Result and Analysis

The rover needed lots of calibration with different variables while implementing the algorithm into the rover system. The first thing that was mandatory is the proper communication between different controllers and communication systems. There were two communication, one is the internal data handling between the rover controllers and another was the manual overriding from the control station to stop in any emergency scenario. We used wired method for the internal communications and 915 MHz transceivers for the communication between rover and control station.

While testing the rover we used a moderately accurate GPS (Ublox Neo-6m) device. This gave us an accuracy of 3m and in wide open area less than 1m. Although with implementation of high precision GPS devices, the system will be much more stabilized with higher degree of accuracy, our system was robust enough to comply to the path. It eradicates the ’must have’ necessity of high precision GPS device. The GPS modules have different refresh rate and accuracy settings. There was a trade off between the accuracy and refresh rate. Hence even with higher accuracy it is not possible to run the rover in higher speed as it may over shoot from the path due to the time latency induced due to reduction in refresh rate. Again, with higher refresh rate the accuracy was compromised. We, hence, implemented the moving average in the GPS data with a high refresh rate. It highly compensated the trade off although there was another latency coming from the moving average system.

The PID system needs to be properly calibrated for proper mobility along the path line. The values of proportionality constants kp, kd and ki were found from iterations. It was found that the rover tends to follow line strictly with higher value of kp but it suffers from over shoots and another special problem that we named perpendicular returning to line (PRL). With higher value of kp the rover follows different curvature while returning to the line from a deviated position, figure 5. The curvature tends to increase with the higher values of kp. In a certain circumstance the rover returns to the path in a perpendicular direction (kp3 in the figure 5). In this case rover then fails to align to the path. To get over this problem the kp was set to a moderate value along with kd which gave a much stable system (figure 6) removing the PRL problem. With further tuning to the values of those constants the rover followed the path with a minimum amount of overshoot while returning from high deviations (figure 6).

The base speed was set to 80 percent of max speed. Having a diameter of 30 cm of the wheels and 60 RPM motors, the maximum attainable speed on each wheel is 3.4 km/h. The rover was found to be able to follow long paths, as long as 1 km which is not possible via image processing. It may follow paths longer than that but due to limitations of field area it was our maximum length to test. Overall, we set markers at the end points of each path and found the rover stopping with a distance ranging from 0.20m to 4.10m from those markers.

V. Conclusion and Future Work

A very efficient method has been presented for autonomous rover navigation. It only uses GPS coordinates to navigate from one location to another. The prime feature of this project is the accurate distance calculation of the current and desired location of the rover solving two major complications i.e. Localization and Path Formation. The distance measurement has been accomplished by deploying Haversine formula using the latitude and longitude which has its maximum degree of approximation ensuring maximum accuracy. Even though there are circumstances where the rover deviates from the right track, the PID control algorithm corrects the error by calculating the deviation angle. The overall proposed system is considerably more methodical and cost efficient than the contemporary complex systems. Despite the fact that it is only driven by the GPS signals, it is more reliable for its higher accuracy rate for outdoor navigation.

With the proper development of an algorithm and successful implementation in real life condition, we are focused in further development of this work. We already have conceptualized a polynomial based path generation method that will eradicate the limitations of our current straight line based system. Along with that in case of rough terrain some path may be too much dangerous to be traversed. A future work will be conducted to include an intentional deviation on such cases and regenerating polynomial path from another point avoiding those inaccessible paths. Also, we are working with implementation of image processing in our current work. This work is in progress and beta has been tested. It will add more accuracy to proper path following with land mark detection or directional indication detection.

Acknowledgment

This work is fully funded by the Independent University Bangladesh.

References

[1] Nagatani, K., Yamasaki, A., Yoshida, K., Yoshida, T., Koyanagi, E. Semi-autonomous traversal on uneven terrain for a tracked vehicle using autonomous control of active flippers, 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, 2008, pp. 2667-2672.

[2] Edlinger, R., Zauner, M., Rokitansky, M. INTELLIGENT MOBILITY - New approach of robot mobility systems for rescue scenarios, 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linkoping, 2013, pp. 1-5.

[3] https://github.com/kriswiner/MPU6050/wiki/affordable-9-dof-sensorfusion.

[4] Ivis. F. Calculating geographic distance: concepts and methods. In Proceedings of the 19th Conference of Northeast SAS User Group, 2006.

[5] Jochem T., Pomerleau D. ”Vision-Based Neural Network Road and Intersection Detection” Intelligent Unmanned Ground Vehicles. The Springer International Series in Engineering and Computer Science (Robotics: Vision, Manipulation and Sensors), vol 388. Springer, Boston, MA, 1997.

[6] Jochem, T. M., Dean A. P., Charles E. T. ”MANIAC: A next generation neurally based autonomous road follower.” Proceedings of the International Conference on Intelligent Autonomous Systems. IOS Publishers, Amsterdam, Pittsburgh, PA, 1993.

[7] Jochem, T., Baluja, S. Massively parallel, adaptive, color image processing for autonomous road following. No. CMU-RI-TR-93-10. CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1993.

[8] Pomerleau, D. A. Neural network perception for mobile robot guidance. Vol. 239. Springer Science & Business Media, 2012.

[9] Dickmanns, E. D., Zapp, A. ”Autonomous high speed road vehicle guidance by computer vision.” International Federation of Automatic Control. World Congress (10th). Automatic control: world congress. Vol. 1. 1988.

[10] Kenue, S. K. ”Lanelok: Detection of Lane Boundaries and Vehicle Tracking Using Image-Processing Techniques. Part I: Hough-Transform, Region-Tracing, and Correlation Algorithms.” Proc. SPIE Conference on Mobile Robots IV. Vol. 1195. 1989.

[11] Kluge, K. YARF: An open-ended framework for robot road following. No. CMU-CS-93-104. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1993.

[12] Wrathall, N. ”An Autonomous Navigation Algorithm using GPS and Digital Compass Sensors, University Of Toronto, 2008,Sept,30.

[13] Kadous, M. W., Sammut, C., Sheh, W. ”Autonomous traversal of rough terrain using behavioural cloning.” The 3rd International Conference on Autonomous Robots and Agents. 2006.

[14] Edlinger, R., Zauner M., Rokitansky, W. ”INTELLIGENT MOBILITY - New approach of robot mobility systems for rescue scenarios,” 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linkoping, 2013, pp. 1-5.

[15] Chu, k., Lee M., Sunwoo, M. ”Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles,” in IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1599-1616, Dec. 2012.

[16] Hadsell, R., Sermanet, P., Ben, J., Erkan1, K., Scoffier, M., Kavukcuoglu, K., Muller, u., LeCun, Y. ”Learning long-range vision for autonomous off-road driving”, Journal of Field Robotics, vol. 26, no. 2, issn 1556-4967, pages 120–144, 2009.

[17] Guivant, J., Nebot, E. Implementation of simultaneous navigation and mapping in large outdoor environments . Robotics Research: The Tenth International Symposium, Springer. 2003.

[18] Kelly, A., Stentz A., Amidi, O., Bode M., Bradley, D. ”Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments”, The International Journal of Robotics Research, vol. 25, no. 5-6, pages 449-483, 2006.

[19] Alon, Y., Ferencz, A., Shashua, A. ”Off-road Path Following using Region Classification and Geometric Projection Constraints,” 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 689-696.

[20] Saha, A., Paul, S. K., Hasan, M., Amin, M. A. ”Android Based Autonomous Arduino Bot” CEET, 2015.

[21] Al Arabi, A., Sarkar, P., Ahmed, F., Rafie, W. R., Hannan, M.,Amin, M. A. ”2D mapping and vertex finding method for path planning in autonomous obstacle avoidance robotic system” Control and Robotics Engineering (ICCRE), 2nd IEEE International Conference on, pp. 39-43, 2017.

[22] Bashar, M R., Al Arabi, A., Tipu, R. S., Sifat, M. T. A., Alam, M. Z. I., Amin, M. A. ”2D surface mapping for mine detection using wireless network”, Control and Robotics Engineering (ICCRE), 2nd IEEE International Conference on. pp. 180-183, 2017.