Autonomous Vehicle for Drug Delivery

K Nandakumar¹, S R Surya ², K Karthick³, N Raviraj⁴, B Cheran⁵, V Poovizhi⁶, S Gokul⁷, C Nimisha⁸

¹ Assistant Professor, Department of Mechatronics Engineering, Bannari Amman Institute of Technology, Erode, Tamilnadu, India.
²,⁶,⁷ Student, Department of Mechatronics Engineering, Bannari Amman Institute of Technology, Erode, Tamilnadu, India.
³,⁴,⁵ Student, Department of Electrical and Electronics Engineering, Bannari Amman Institute of Technology, Erode, Tamilnadu, India.
⁸ Student, Department of Electrical and Communication Engineering, Bannari Amman Institute of Technology, Erode, Tamilnadu, India.

nandakumar@bitsathy.ac.in, surya.mc16@bitsathy.ac.in

Abstract. Autonomous vehicles are intended to move in an environment without any human intervention. In order to determine the structure of the environment, we have to rely on sensors data to build the map and to localize the vehicle within its environment. This paper presents the autonomous vehicle built using 2D LIDAR and Robot Operating System (ROS). If the environment is to be explored is outdoor we can rely on GPS to track the location the autonomous vehicle, whereas in an indoor environment we can’t do so. For building the map of the environment and for the vehicle’s localisation simultaneous localisation and mapping (SLAM) is used. The LIDAR data is used to find out the corresponding distance of the obstacle from the robot. Based on this point cloud data map is generated denoting the obstacle, explored area, unexplored area and the location of the robot with respect to the map. Data from the encoder is used to navigate the robot. Path planning is done using A* algorithm to find out the shortest distance between the starting and the destination point. The experimental result helped us in comparing the performance of the autonomous mobile robot built using ultrasonic sensor with the autonomous mobile robot built using LIDAR.

Keywords: 2D LIDAR, Autonomous mobile robot, SLAM, ROS, A* algorithm

1. Introduction

As the future is moving towards automation, robotics has gained its importance and it is likely replacing humans. Robots perform human work with better repeatability and accuracy. So, we prefer employing robots to finish the desired work with better efficiency and to reduce time consumption. The autonomous robot built using ultrasonic sensor has wider angle of divergence. So, it is quite difficult to predict the exact location the obstacle, whereas in LIDAR the point of divergence is comparatively less, so the map generated with LIDAR data will give the accurate location of the obstacle with respect to the map. LIDAR throws out approximately 650-point cloud data for every successive revolution. The line by line execution of the python code may lead to the loss of certain LIDAR data. So we go with robot operating system (ROS) which does multiprocessing. The entire
process is divided into subprocess called nodes which gets executed concurrently. Encoder attached to the wheels of the robot is used to calculate the distance travelled by the robot. Having a dedicated robot to perform some specific application avoids the need of a person to carry out that task.

2. Literature review

Genghang Zhuang et al. conducted numerous experiments for localisation of autonomous vehicles using GPS and INS. But localising the autonomous robot without using devices like GPS for indoor application can be achieved using 2D LIDAR and MCL. Scanning of the point cloud data is done using laser and a prior map is generated using the scanned data. Once the bot moves from its position new map is generated using the scan points and this updating process is done with help of Monte Carlo Localisation (MCL) [1].

Ilya Afanasyev et al. conducted a feasibility test using gazebo simulation approach. In order to study the characteristics and behaviour of the robot under different conditions, robot simulators have been developed to pre-analyse the end results. The experiment is conducted using PR2 robot built under SLAM in ROS environment [2].

John Kerr et al. conducted a research to optimize the existing 16 ROS into a single ROS which would benefit their students. To understand the concept of ROS, working to normal operating system is studied so that the same can be implemented with ROS. The first test started with comparing the 16 ROS with criteria’s like name of the developer, latest update date, number of tutorials available. At the end of first comparison 6 ROS were eliminated. To filter the 10 ROS, second stage of testing is carried out by setting some criteria and each ROS is tested based on the criteria’s provided and scores were given out of 10. At the end of second stage of testing 7 ROS were eliminated. The final test is conducted with practical testing. The end results demonstrated that a student with a enough coding knowledge with no prerequisite knowledge in ROS was able to control a robot in a time span of 25 hours [3].

Hugh Durrant-Whyte et al. proposed a SLAM technique that has been proposed over the past decades. SLAM technique is adopted to localize the mobile robots in an unknown environment. The environment can be indoor, outdoor, or underwater. Over the past decades the problem of SLAM has been unsolved. In the present the problems are over-thrown with the help of techniques like EKF-SLAM, Rao blackwellised filter [4].

Jie Hu et al. proposed a A-star algorithm which is widely used in unity 3 games in finding shortest path between the nodes. The possible ways to reach the destined location is analysed. Then the weightage of each node is calculated, the node will comes up with the least value is considered as the shortest path to reach that destined location. The algorithm showed best results in finding the shortest path [5].

3. Existing methodology

One of the most commonly used range/distance finding sensors is ultrasonic sensor (HC-SR04). This sensor detects the distance of the objects by emitting high frequency ultrasonic waves and waits for the waves to get reflected back from the objects. Using the time taken for the sound wave to get reflected back, the distance can be computed. This sensor has a measurement range of 2cm- 400 cm and a measuring angle of 15 degree.
The distance data is used to compute the map of the environment. Since, the angle of measurement is 15 degrees the width of the beam increases with the increase in distance (Figure 1). For example, at 4 meters the beam should be as wide as the meters i.e. the obstacle’s probability increases with distance. Thus, the map generated using ultrasonic range finding device is considerably inaccurate.

4. Proposed methodology
A laser range finder works similar to the ultrasonic sensor. But instead of ultrasonic sound waves it used finely focussed pulse of light to the target and detects the reflection. LIDAR (light detection and ranging) is one of the laser range finding sensors.

The ultrasonic sensor is replaced by a 2D LIDAR (single channel LIDAR) which has a single laser range finder rotated all through 360° taking samples at equal interval. The list of distances will be retrieved after a full rotation which will have the range data of the obstacles around. Thus, the map can be generated using the LIDAR range data. In order to localize the robot within the map, encoder is used. The encoder works on the principle of hall effect which gives 1000 pulses per revolution (which many vary according to the type of encoder used). Thus, mapping and localisation can be performed.

4.1 Localization theory
The robot is built with two motors with encoders and wheels attached to its shaft. A castor wheel is used to support the robot’s base. The robot is of differential drive type. Based on the encoder pulse received by the controller, the distance travelled by the robot could be found as below.

![Figure 1. Measuring angle of ultrasonic sensor.](image)
Let us consider the encoder pulses of left wheel be PL and the right wheel be PR. The pulses received could be related to the distance travelled by the wheels as in equation (1).

\[
NL = \frac{\pi \times W_{\text{Dia}} \times PL}{PPR}
\]

\[
NR = \frac{\pi \times W_{\text{Dia}} \times PR}{PPR}
\]

(1)

Where,

- \(NL\) = distance travelled by left wheel
- \(NR\) = distance travelled by right wheel
- PR and PL = Pulse count in right and left encoder
- \(W_{\text{Dia}}\) = wheel diameter
- PPR = pulse per revolution of encoder shaft

The robot’s base will have a frame attached to the centre of the axes of the wheel (figure 2). Any translational and rotational movement will be performed on that coordinate frame \(\{B\}\). There are two possibilities that a robot can move.

(i) A pure translation where both the motors being operated in same direction and at the same speed.
(ii) A general transformation in which motors may be operated in different direction and in different speed. In the first case robot will be moving in straight line whereas in the second, its trajectory will be a general curve.

4.1.1 \( n_L \approx n_R \) (Linear motion)
When the number of pulses in both the encoders is same, the robot would have moved in straight line (figure 3). By the equation (2), both the forward and backward motion will be calculated. To convert the motion with respect to the universal frame \( \{U\} \), equation (2) could be applied,

\[
U_{T_{B2}} = U_{T_{B2}} * B_{T_{B2}}
\]

\[
B_{T_{B2}} = \begin{bmatrix}
1 & 0 & 0 & n \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(where \( n = n_L = n_R \))

\( U_{T_{B1}} = \) Robot’s old position in the map (Initially an identity matrix)

\( U_{T_{B2}} = \) Robot’s current position in the map

\( B_{T_{B2}} = \) Robot’s displacement

\[ B_{T_{B2}} \]

Figure 3. Linear motion of robot.

4.1.2 \( n_L \neq n_R \)
If the wheel’s direction and speed are not same, the robot will be moving in a general curved path. It could be considered that the robot rotates with respect to an imaginary frame \( \{c\} \) located at instantaneous central point of the rotational motion. Translation of robot frame could be found by finding instantaneous centre \( \{c\} \), angle of rotation \( \theta \) and radius of trajectory of the robot \( r \) as in equation (3).

\[
r = \frac{n_L * W}{n_R * n_L}
\]
The transformation of robot’s current position \( \{B_1\} \) with respect to the centre \( \{c\} \), \( ^cT_{B1} \), can be calculated using equation (4). From that, the robot’s current position \( \{B_2\} \) with respect to the centre \( \{c\} \) can be calculated by rotating the \( \{B_1\} \) with respect to \( \{C\} \) about \( \theta \) (figure 4).

\[
^cT_{B2} = ^cT_{B1} \cdot \text{Rot}_Z(\theta) \]

\[ \theta = \frac{nR \cdot nL}{W} \]

\[
^cT_{B1} = \begin{bmatrix}
1 & 0 & 0 & n \\
0 & 1 & 0 & -r - (w/2) \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

(3)

(4)

The transformation of robot’s current position \( \{B_1\} \) with respect to map \( \{U\} \) can be calculated as shown below.

\[
^UT_{B2} = ^UT_{B1} \cdot ^{B1}T_C \cdot ^cT_{B2}
\]

Where,

\( ^UT_{B1} = \) Robot’s previous position (Initially set to identity matrix)
\[ T_C = (C_{B1})^{-1} \]

Thus, using equation (6), the robot can be localised within the map and \(^{U}T_{B2}\) will be the current transformation which will define the translation and rotation of the robot with respect to the frame \(\{m\}\).

4.2 Mapping

Mapping is the process of generating a map of the environment which can be used to navigate within the same. In this work, grid mapping is used i.e. the obstacles in the work environment are mapped in matrix of dimension 1000X1000. The range data provided by the LIDAR will be based on the frame \(\{L\}\) attached to the LIDAR (figure 5).

![Figure 5. Illustration of frame\{L\}.](image)

According to the frequency of the LIDAR, the \(n\) samples (spaced equally per revolutions) are retrieved. Angle and distance \(d\) defines the exact position of the obstacles in the environment. The map will be stored as a 2-D array which will have a height \(h\) and width \(w\). In order to generate the map some assumptions are made and those are the following,

- Initially the robot is placed at the centre of the map and mapping starts from there.
- Each cell will represent the distance of resolution(m) per cell of the array.
- Array index starts from 0.
- Number of columns defines width and number of rows defines the height of the map.
- A frame \(\{G\}\) is attached to start of the map (i.e. row = 0 and column = 0).
- A frame \(\{m\}\) is attached to the centre of the map (row = height/2 and column = width/2).
- Robot’s transformation is recorded with respect to this frame.
Figure 6. Representing bot’s position in map.

The map can be imagined as shown in the above figure 6. Thus resolution affects the accuracy of the map considerably.

Resolution $\alpha$ (accuracy)$^{-1}$

\[\text{(7)}\]

The resolution, height and width of the map determines the size and of the map. For each revolution the distance points are retrieved along with the angle. The corresponding $x, y$ coordinate of the obstacle can be retrieved using the equation (8),

\[X = \frac{d}{\text{resolution}} \times \cos \theta\]

\[Y = \frac{d}{\text{resolution}} \times \sin \theta\]

(8)

These points are based on the frame $\{L\}$ and in order to update into the map, the points need to be represented in the frame $\{B\}$. $U_T^B$ will also need to be converted from distance to cell conversion by dividing the $x_0$ position by revolution before proceeding.

Let us assume a obstacle (x) located at coordinate at $m$ and $n$ with respect to $\{L\}$

\[L_x = \begin{bmatrix} m \\ n \\ 0 \end{bmatrix}\]
For each revolution ‘N’ number of points which will be retrieved and we can represent it in frame \{L\} as,

\[
LP = \begin{bmatrix}
m_1 & m_2 & m_3 & \ldots & \ldots & m_n \\
n_1 & n_2 & n_3 & \ldots & \ldots & n_n \\
0 & 0 & 0 & \ldots & \ldots & 0
\end{bmatrix}
\]

Since the LIDAR is mounted on the top of the robot rigidly, \{L\} frame’s transformation with respect to \{B\} frame will be a constant.

\[
BTL = \begin{bmatrix}
-1 & 0 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Both frames origin coincide since both the LIDAR and the robot is assembled in such a way as shown in the map visualization frame \{M\} is fixed rigidly on the frame \{G\} and hence the transformation will be

\[
^GTM = \begin{bmatrix}
1 & 0 & width/2 \\
0 & 1 & height/2 \\
0 & 0 & 1
\end{bmatrix}
\]

To convert the distance from the \{C\} frame to the \{G\} frame, equation (9) is used.

\[
G_P = ^GTM * ^MTb * ^BTl * LP
\]

The first row of the \(G_P\) matrix will represent the columns of the array and the second row of the \(G_P\) matrix will represent the rows of the array where the distance points from the LIDAR is converted with respect to the grid on applying the above transformations. Thus, the data from the array can be updated after this conversion.

To generate the map, assume the maximum range for the LIDAR (i.e. above which the range data will be discarded). In the map, the obstacle will be marked as 0, floor area will be marked as 1 and unexplored will be marked as -1. Initially the whole map will be set as unexplored since the exploration is not yet started. As the robot moves in the area, the LIDAR data is used to find out the obstacles at that particular time. Since the obstacles around the robot may be dynamic, the presence of object in the map must be checked every time when the obstacles come into the vicinity of LIDAR range.

Each and every time the map has to be updated by the latest LIDAR distance data of obstacles. For this a circular region of radius LIDAR maximum range will be replaced by newly computed data based on the following equation (10).

\[
\text{New Update} = \text{(old obstacle > New floor)} \ || \ \text{New obstacle}
\]

(10)
4.3 Calculating old obstacle

Obstacle is calculated by taking the values from the map for only the circular region of radius of current maximum LIDAR range and robot position (figure 7).

![Figure 7. Calculating old obstacle.](image)

4.4 Calculating new floor

New floor area can be calculated by altering the screen range data and the altered data can be used to find the boundary of the floor. The LIDAR range value having greater than the LIDAR maximum range is replaced with the LIDAR maximum range (figure 8).

![Figure 8. Calculating new floor.](image)

By connecting all the data points, a polygon is made within which all the values are 1 and outside the polygon all the values are 0. This is the new floor data we needed.

4.5 Calculating new obstacles

New obstacles will be calculated by discarding the LIDAR range scan distance data greater than the LIDAR maximum range and setting the data points with the value ‘1’ and the other points as ‘0’ (figure 9).

![Figure 9. Calculating new obstacle.](image)
4.6 Updating map with new obstacle

Thus, the new value to be updated can be calculated. For uncovered area, the older uncovered data and the new uncovered data are taken and binary AND operation is performed to compute the newer uncovered area. Thus, the value is updated with the newer circular region (figure 10)

![Figure 10. Combining all together.](image)

4.7 Path finding

In the generated map, the robot has to navigate in order to explore the unexplored area in the map and to reach another point in the map. In order to achieve this task we are in need of a path from source(robot’s current location) to the destination point.

![Figure 11. Path finding.](image)

The path can be traced using A* algorithm which is build according to our map representation (figure 11). The algorithm iterates through the possible oaths and finds the appropriate and shortest path among them. A* algorithm is an extension of edsger wybe dijkstra’s algorithm which uses heuristic algorithm to estimate the cheapest or shortest path to the destination point from the current position by avoiding the other possible points. Using the path the robot can navigate within the map by calculating the required location.

5. Result and analysis

The end results demonstrated that the autonomous mobile robot built using 2D LIDAR generated map with comparatively higher accuracy when compared to the map generated using ultrasonic sensor’s data. It is found that the angular divergence of ultrasonic sensor is 15 deg where as when LIDAR is used the divergence come to below 1deg.
The position of the obstacle is well defined in the currently generated map, whereas in the ultrasonic sensor it depicts the possibility of obstacle’s presence within its entire angle of divergence range (figure 12).

![Figure 12. Map detecting the obstacle probability of LIDAR and ultrasonic.](image)

**Table 1** Encoder data analysis.

| Trials | BOT (X axis in m) | BOT (Y axis in m) | Encoder output (X axis in m) | Encoder output (Y axis in m) |
|--------|------------------|------------------|-----------------------------|-----------------------------|
| 1      | 1.77             | 0.17             | 1.78                        | 0.16                        |
| 2      | 1.86             | 0.27             | 1.89                        | 0.20                        |
| 3      | 1.91             | 0.40             | 1.93                        | 0.38                        |
| 4      | 1.57             | 0.24             | 1.57                        | 0.25                        |
| 5      | 1.54             | 0.23             | 1.54                        | 0.24                        |
| 6      | 1.66             | 0.30             | 1.65                        | 0.29                        |
| 7      | 1.60             | 0.29             | 1.60                        | 0.26                        |
| 8      | 1.54             | 0.24             | 1.54                        | 0.23                        |
| 9      | 1.56             | 0.24             | 1.56                        | 0.23                        |
| 10     | 1.54             | 0.30             | 1.55                        | 0.24                        |

From the experiments conducted to calibrate the encoders (table 1), it is found that the positional coordinated obtained from using encoders are sufficiently accurate.

### 6. Conclusion

This project has described the localisation of autonomous mobile robot using 2D LIDAR. The autonomous mobile robot can be used to perform indoor applications like drug delivery, taking pictures in a hazardous environment and updating the map of that environment. This project can be further developed and can be made to explore the vast area. For achieving better accuracy IMU can be incorporated with encoder in further developments.

Due to the financial constraints, this work is limited to 2D LIDAR. Using costlier 3D LIDAR enable us to go for SLAM based localization and mapping. The localization and mapping relies on the
encoder data obtained from the wheel encoders. Slippage of wheels causes error accumulation during long run. This could be reduced by using SLAM along with IMU and encoder.

Acknowledgement
We would like to express our heart filled gratitude to our Professors, fellow faculty members, Graduate engineering trainees and students, for their constructive ideas, inspirations, encouragement, excellent guidance and much needed technical support extended to complete our project.

References

[1] Genghang Zhaung, Shengjie Chen, Jianfeung and Kai Haung, . (2017), A Real-Time Embedded Localization in Indoor Environment using LiDAR Odometry”. National Conference on Embedded System Technology, pp. 210-222. Springer, Singapore

[2] Ilya Afanasyev, Artur Sagitov, and Evgeni Magid: “ROS-Based SLAM for a Gazebo- Simulated Mobile Robot in Image-Based 3D Mode of Indoor Environment”. International Conference on Advanced Concepts for Intelligent Vision Systems, pp. 273-283. Springer, Cham, (2015).

[3] John Kerr and Kevin Nickels: “Robot Operating Systems: Bridging the Gap between Human and Robot” 44th IEEE South eastern Symposium on System Theory in the proceedings of (2012).

[4] Hugh Durrant-Whyte, and Tim Bailey: “Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms” IEEE International Conference, Robotics and Automation magazine, (2015).

[5] Jie Hu, Wang gen Wan and Xiaoqing Yu: “A Pathfinding Algorithm in Real-time Strategy Game based on Unity3D” audio, language and image processing of international conference in the year (2012).

[6] Biswas J and Veloso M.: “localization and navigation for autonomous indoor mobile robots”. International journal of Computing Communications & Instrumentation Engineering (2015).

[7] Wang C C, Thorpe C and Thrun S: “Online simultaneous localization and mapping with detection and tracking of moving objects”, IEEE Conference on Robotics & Automation (2013), IEEE.

[8] Deans M and Hebert M: “Experimental comparison of techniques for localization and mapping using a bearing-only sensor”. In International Conference on IEEE on Robotics and Automation (Cat. No. 03CH37422), vol. 1, pp. 842- 849. IEEE, (2015).

[9] Kim J and Sukkarieh S: “Airborne simultaneous localisation and map building”. In IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), vol. 1, pp. 406-411. IEEE, (2013).