Outdoor Robot Navigation Using Gmapping Based SLAM Algorithm

B.L.E.A. Balasuriya[1], B.A.H. Chathuranga[2], B.H.M.D. Jayasundara[3], N.R.A.C. Napagoda[4], S.P. Kumarawadu[5], D.P. Chandima[6] and A.G.B.P. Jayasekara[7]

Department of Electrical Engineering
University of Moratuwa
Moratuwa, Sri Lanka

eranga.balasuriya@gmail.com[1], heshanchathurangeba@gmail.com[2], duminda902@gmail.com[3], asankanapagoda@gmail.com[4], sisil@elect.mrt.ac.lk[5], chandima@elect.mrt.ac.lk[6], buddhika@elect.mrt.ac.lk[7]

***Abstract*— This paper presents the complete methodology followed in designing and implementing a tracked autonomous navigation robot which can navigate through an unknown outdoor environment using ROS (Robot Operating System). The concept is based on the mapping process using SLAM (Simultaneous Localization and Mapping) GMapping Algorithm. Implementation of the robot on the ROS platform is presented in this paper and experimental results are also presented validating the accuracy of the algorithm.**

Keywords*— *Autonomous Robot; SLAM; GMapping; Laser Scanner; Rao - Backwellized Particle filter

I. Introduction

Generating a map of the unknown environment and localizing the position of the robot are the major tasks of mobile robots. Many researches are focused on the problem of how to represent the environment as well as robot's position. SLAM (Simultaneous Localization and Mapping) algorithm is one of the most widely researched topic in robotics when it comes to mapping unknown environments and localizing the robot's position. Extraction unknown environment using SLAM is based on some sensors such as laser scanners, kinects or ultrasonic sensors [1, 2, 3].

Nowadays with rapid improvement of advanced sensors, SLAM algorithm is being used in real life applications such as search and rescue field (natural disasters such as fire, flood and earthquake), military field, public servants (guide people in public areas such as super markets and museums) [4].

The Robot Operating System (ROS) is the most developed and popular robotics framework for robotics applications nowadays. Researchers prefer the usage of ROS because of several advantages. ROS enables researchers to quickly and easily perform simulation and real world experiments using gazebo and rviz [5]. On the other hand, these kinds of autonomous mobile robots deal with a lot of sensors and each process needs to be functioning properly in the real time. Sensors and drivers used are needed to be updated every 10 – 15 milliseconds and we need a type of operating system that gives us this privilege. This is because even split second lag can result in a disaster for the robot. If the standard microcontrollers are used with self-defined timings for this purpose, it would be very hard to handle multiple processes that need to be run [6].

Mapping and localization are achieved in this study by using the GMapping algorithm (a variant of the SLAM algorithm) with single laser sensor. GMapping algorithm is based on Rao-Blackwellized particle filter (RBPF). As proper detection of surrounded obstacles is required for the accuracy of the algorithm, the RPLIDAR 360^0 laser scanner which provides both distance and bearing angle measurements to the nearby objects.

In this study, implementation of SLAM algorithm is focused on long distance outdoor robot navigation. The robot platform used here is a tracked mobile robot which is specifically built for outdoor navigation. Different SLAM algorithms have been applied for outdoor robotic applications but were unable to give expected results especially in unstructured environments. Finding a solution to the SLAM applying GMapping based SLAM algorithm on a tracked mobile robot could be highlighted as the new contribution to the study as the proposed method could handle many challenges in the outdoor robot navigation

In the scope of this paper, the basic theory behind SLAM algorithm is explained in section II. Moreover, the theory of Rao-Blackwellized particle filter, which is the base of GMapping algorithm is explained in the same section. Implementation of SLAM algorithm in tracked mobile robot and the experimental results are illustrated in section III. Control method of the robot and the conclusion are given in section IV and section V respectively.

II. Slam Algorithm

Localization and mapping are always coupled problems. In the localization problem, a map $\boldsymbol{p}$ of landmark location is known. Then the pose of the robot $\boldsymbol{x}_v(k)$ is determined using the sensor measurements of landmark location $\boldsymbol{z}k$).

In the mapping problem, vehicle pose $\boldsymbol{x}_v(k)$ is known. It is then only needed to take measurements of landmark location $\boldsymbol{z}(k)$ and build map $\boldsymbol{p}$ using $\boldsymbol{z}(k)$. Fig. 1. Illustrates the measurements of landmarks $\boldsymbol{z}(k)$, the vehicle pose $\boldsymbol{x}_v(k)$ and the landmark location $\boldsymbol{p}(k)$.

978-1-5090-0645-8/16/$31.00 ©2016 IEEE

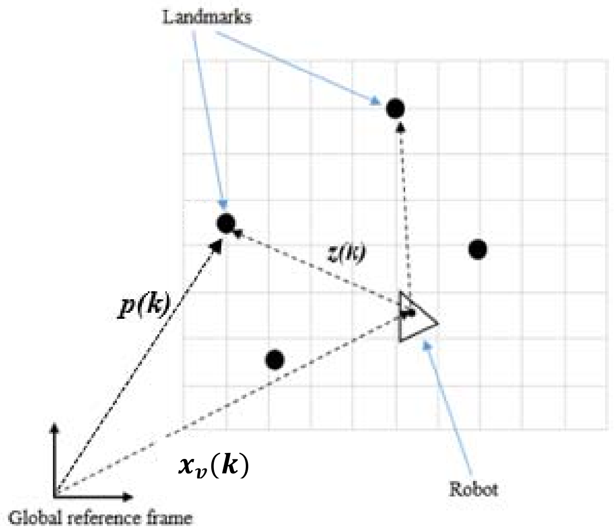

Fig. 1. Parameters of SLAM algorithm

This SLAM algorithm can be stated as the solution of the localization and mapping problems considered together.

A. Process And Observation Models

As stated before, SLAM problem takes measurements of landmark location $\boldsymbol{z}(k)$ to build the map $\boldsymbol{p}$ and to determine vehicle pose $\boldsymbol{x_v}(k)$ using $\boldsymbol{z}(k)$. Here, a robot with a known kinematic model starts at an unknown location and then moves through an environment containing several features or landmarks.

The platform is equipped with a RPLIDAR 360^0 laser scanner which takes relative measurements of the relative locations between any individual landmark and the robot. Absolute landmark locations are not available. Both the process model and the observation model are non-linear models that can be directly incorporated into the GMapping based SLAM algorithm without losing original information in the models as linearization of the models is not done.

- Process Model

The state of the system consists of the position and the orientation of the vehicle together with the positions of all landmarks. The following kinematic equations are used to predict the state of the robot from the steering (γ) and velocity (*V*) inputs.

$$\begin{aligned} \dot{x_v} &= Vcos(\varphi) \\ y_v &= Vsin(\varphi) \\ \dot{\varphi_v} &= \frac{Vtan(\gamma)}{L} \end{aligned} \tag{1}$$

Here, L is the wheel-base length of the vehicle. Now, the discrete-time process model can be written as follows,

$$\begin{bmatrix} x_v(k+1) \\ y_v(k+1) \\ \varphi_v(k+1) \end{bmatrix} = \begin{bmatrix} x_v(k) + \Delta TV(k)cos(\varphi_v(k)) \\ y_v(k) + \Delta TV(k)sin(\varphi_v(k)) \\ \varphi_v(k) + \frac{\Delta TV(k)tan(\gamma(k))}{L} \end{bmatrix} \tag{2}$$

The location of the i^{th} landmark is denoted $\boldsymbol{p_i}$. The state transition equation for the i^{th} landmark is as follows,

$$\boldsymbol{p_i}(k+1) = \boldsymbol{p_i}(k) = \boldsymbol{p_i} \tag{3}$$

This is due to all the landmarks are assumed to be stationary. If there are N number of landmarks in the environment, the vector of all N landmarks is denoted,

$$\boldsymbol{p} = [\, \boldsymbol{p}_1^T \ldots \boldsymbol{p}_N^T \,]^T \tag{4}$$

Then, the augmented state vector containing both the state of the vehicle and the state of all landmark locations becomes,

$$\boldsymbol{x}(k) = [\, \boldsymbol{x}_v^T(k) \;\; \boldsymbol{p}_1^T \ldots \boldsymbol{p}_N^T \,]^T \tag{5}$$

- The Observation Model

Observations $\boldsymbol{z}(k)$ are taken by the laser scanner. We can write the observation model relevant to the i^{th} landmark as follows,

$$\boldsymbol{z_i}(k) = H_i\boldsymbol{x}(k) + \boldsymbol{w_i}(k) \tag{6}$$

$$= H_{pi}\boldsymbol{p} + H_v\boldsymbol{x_v}(k) + \boldsymbol{w_i}(k) \tag{7}$$

where $\boldsymbol{w_i}(k)$ is a vector of uncorrelated observation errors with zero mean and variance $\boldsymbol{R_i}(k)$. H_{pi} is the jacobian matrix of the observation model with respect to landmark state, H_v is the the jacobian matrix of the observation model with respect to vehicle state, H_i is the observation matrix that relates to the sensor output, $\boldsymbol{z_i}(k)$ to the state vector $\boldsymbol{x}(k)$ when observing the i^{th} landmark.

$$H_i = [-H_v \quad ,0 \cdots 0, \quad H_{pi} \quad ,0 \cdots 0] \tag{8}$$

Thus it is obvious that the observations are relative between the robot and the landmarks.

B. The Estimation Process

In this study, the GMapping algorithm is used to solve the SLAM problem. Grid map based GMapping algorithm and scan-matching method are used for map generation and pose estimation in this study respectively.

General probabilistic Bayes filter as in (9) is mostly the base for SLAM algorithms. This uses some measurements to estimate the density of unknown probability. [7]

$$p(k) = \textstyle\sum_{i=1}^{N} w_i \delta_s(x) \tag{9}$$

The map generated using the GMapping algorithm is presented as grid maps. To learn grid maps from laser scan data, GMapping algorithm uses Rao-Blackwellized particle filter (RBPF). In RBPF the joint posterior (10) between map data p_t,

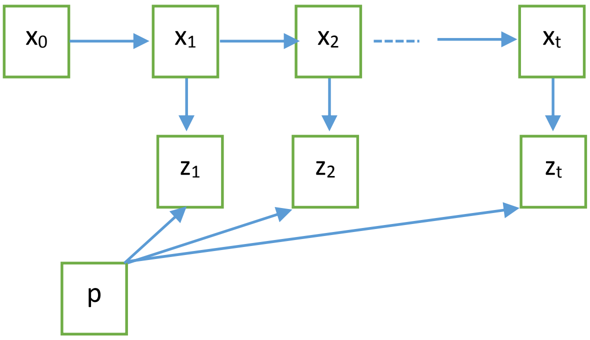

Fig. 2. GMapping algorithm without odometer data

and the state $x_{1:t} = x_1, \dots, x_t$ of the robot is being estimated by using measurement data from laser scan $z_{1:t} = z_1, \dots, z_t$ [7].

$$p(x_{1:t}, \mathrm{p}|z_{1:t}, u_{1:t-1}) \tag{10}$$

To represent a non-gaussian distribution as in this study, RBPF is a better way compared to EKF and other Kalman filter based algorithms which are suitable for representing linearized distribution [8]. Basic principle of RBPF is to set state of hypotheses. Each particle keeps a state with the measurements obtained by the laser scanner. Each landmark is associated with the corresponding particle. The strongest hypotheses is kept with given weights and the weak one is omitted after resampling [9, 10].

The particles can be defined as potential poses of the robot and each state represents the posterior.

$$p(x_{1:t}, \mathrm{p}|z_{1:t}, u_{0:t-1}) = p(x_{1:t}|z_{1:t}, u_{0:t-1}) \times p(\mathrm{p}|x_{1:t}, z_{1:t}) \tag{11}$$

Particles are being selected with a probability proportional equation as in (11), comparing the observations with its own map and selecting the most likely particle where each particle filter carries a map.

This method provides a costless solution for one of the main localization problems of SLAM. GMapping algorithm avoids control data for verification in defining the pose from odometer or other control data. The robot pose estimation is defined using single laser sensor scan data.

In Fig. 2, x_t refers to state of the vehicle and the landmark. z_t refers to the measurement and p_t represents the obtained map. As it seems, the control data is not being used in this flow chart. Thus it shows the GMapping algorithm without control data.

The algorithm can be ordered according to three main steps. Those are:

- Initial state guess. The pose of the robot is obtained from the previous pose with measurement z_{t-1} from laser scanner.
- Scan-matching algorithm obtains map p_{t-1} from initial state guess.
- Updates of particles. Particles are updated based on measurements z_t. The map p_t of particle is updated based on state x_t and measurement z_t.

Aligning currently scanned map with constructed map is done using scan-matching. Scan-matching was used to align current scan z_t with given initial state x_{t0} and map p_{t-1} [10, 11]. Starting from initial state x_t (12), scan-matching algorithm is being applied on map p_{t-1}. Aligning is only done for limited areas of state x_t by the algorithm.

$$x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$$

$$x_t'^{(i)} = \arg max_x p(x|p_{t-1}^{(i)}, z_t, x_t'^{(i)}) \tag{12}$$

Scan-matching selects the particles with high hypotheses of state x_t (12). The next state is being calculated according to this result (13).

$$p(z_t|p_{t-1}, x_j) \rightarrow p(x_j|x_{t-1}, u_{t-1}) \tag{13}$$

At this stage a normalizer is used to compute the weights (14).

$$n^{(i)} = \sum_{j=1}^{K} p(z_t|p_{t-1}^{(i)}, x_j) p(x_j|x_{t-1}^{(i)}, u_{t-1}) \tag{14}$$

Finally the map p_t of the particle at t is updated according to the state x_t and the measurement z_t [12].

III. Implementation of the SLAM Algorithm

This section describes a practical implementation of the simultaneous localization and mapping on a robot platform. The

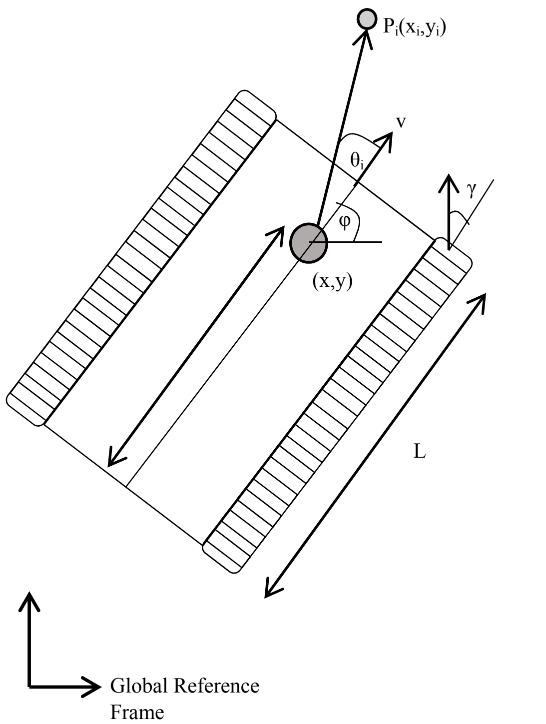

Fig. 3. Schematic diagram of the robot

robot is equipped with a laser scanner and a combination of sonar sensors is used as a backup obstacle avoidance system. If needed, the robot can be manually navigated via a Bluetooth remote running on an Android smartphone.

In this experiment, we are evaluating the accuracy of the map given by the SLAM GMapping algorithm. Issues encountered in landmark extraction and data association are discussed further in following sections.

A. Experimental Setup

The robot platform equipped with the laser scanner and a laptop is mounted on the robot in order to execute the program on the ROS platform. The robot moving on a predetermined path with the landmark points, obstacles such as walls are detected by the laser scanner.

- The Process Model:

Fig. 3 shows a schematic diagram of the process model of the robot observing a landmark. To predict the state of the robot from steering and velocity inputs the kinematic equations (2) are used.

The landmarks are assumed to be stationary point targets. Therefore the landmark process model is as follows,

$$\begin{bmatrix} x_i(k+1) \\ y_i(k+1) \end{bmatrix} = \begin{bmatrix} x_i(k) \\ y_i(k) \end{bmatrix} \tag{15}$$

- The Observation Model:

Using the distance $r_i(k)$ and bearing $\theta_i(k)$ to a landmark i measured by the laser scanner, the observation model $H_i(.)$ can be written as,

$$\begin{aligned} r_i(k) &= \sqrt{(x_i - x_r(k))^2 + (y_i - y_r(k))^2} + w_r(k) \\ \theta_i(k) &= \arctan\left(\frac{y_i - y_r(k)}{x_i - x_r(k)}\right) - \varphi(k) + w_\theta(k) \end{aligned} \tag{16}$$

where $w_r(k)$ and $w_\theta(k)$ are the noise sequences associated with the distance and bearing measurements. The location of the laser scanner on the robot is given by,

$$\begin{aligned} x_r(k) &= x(k) + a.cos(\varphi(k) - b.sin(\varphi(k) \\ y_r(k) &= y(k) + a.sin(\varphi(k) - b.cos(\varphi(k) \end{aligned} \tag{17}$$

B. Experimental Results

- Vehicle Localization Results:

The generated map of the indoor environment (Fig. 4) is shown in Fig. 5. The red arrow in the map given in Fig. 5, shows the current pose by successful localization of the robot with the grid on. Localization is done using the encoder readings of the left and right motors. In this experiment, the robot moved in a straight line.

Fig. 4. Indoor environment

- Environment Mapping Results:

Experiment 1

In this experiment, the robot was navigated in an indoor environment as shown in Fig. 4.

Every 5 seconds, the map is updated with the readings taken by the laser scanner. Outer edges of the walls are successfully recognized by the laser scanner and marked with black dots as shown in Fig. 5. Bottom half of the map shows an open area with the corridor. Middle part of the map shows the corridor with a stair case on the right side. Top part of the map shows the end of the corridor and an opening to a room on left.

There we can see that the scanner measurements are poor as it goes out of its range of 6 m at some points. In such open areas

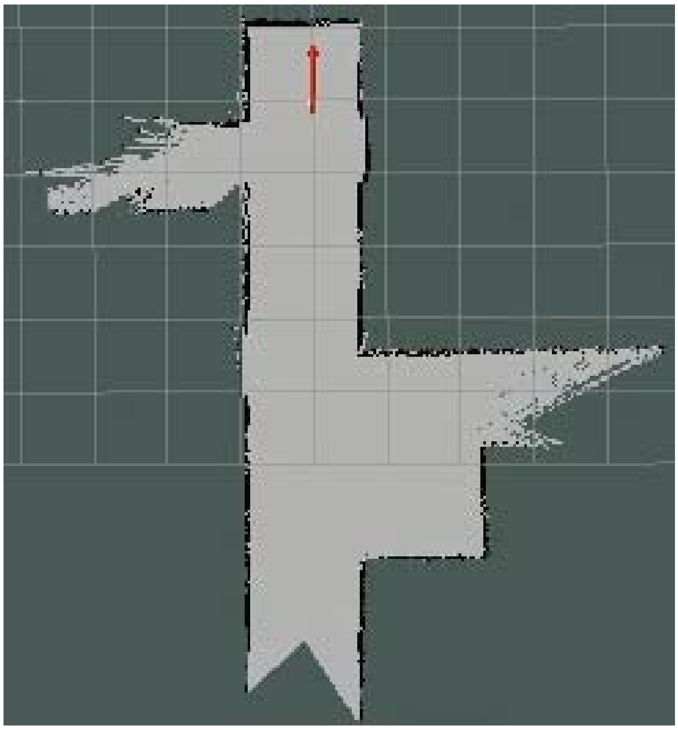

Fig. 5. Indoor mapping results

Fig. 6. Robot platform

shown in Fig. 4, where the laser scanner could go out of range, we see that the sensor measurements are highly noisy.

Experiment 2

The tracked mobile robot used in experiment 1 and experiment 2 is shown in Fig. 6. This mobile platform is specifically built for navigation in outdoor environments.

In this experiment, the robot was navigated in an artificially created sandy environment using barricades as shown in Fig. 7.

As shown in Fig. 8, the outdoor navigation was very much successful as it is given less errors and noise. The blue line in Fig. 9 represents the path of the robot in the mapping process. This path can be obtained in Rviz (ROS visualization) [5] by publishing the path topic. There is a little difference between the path obtained by Rviz as in Fig. 9. and the actual path of the robot in real environment. This is because there is less slip of the tracks when it runs on sandy terrain. This error between the actual path and the path generated in the map can be taken as the localization error of the algorithm.

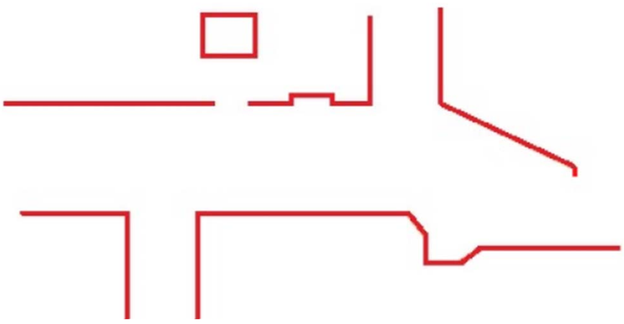

Fig. 7. Outdoor environment

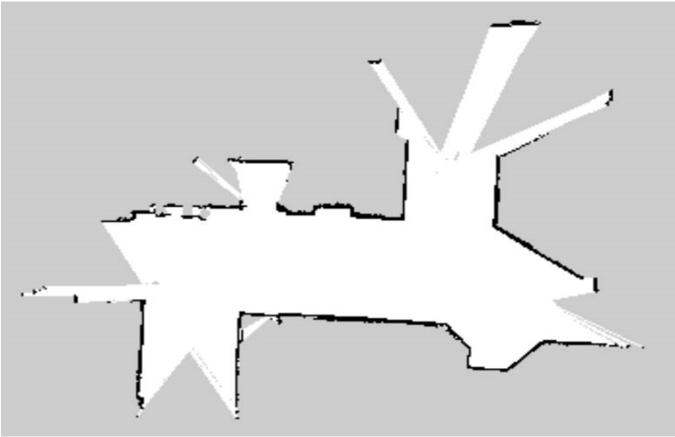

Fig. 8. Outdoor mapping results

It was also needed to cover the top of the laser scanner in order to reduce the effect of the sun which affects the readings and the detection of objects negatively.

The terrain was flat at most points but there were uneven areas at some parts. There was no problem in navigating through such premises as the odometer data were accurately calculated.

Before implementing the navigation process, the maps should be more closely checked for accuracy in both localization and mapping, to prevent the robot from crashing into obstacles. Using the scale of pixels in the map, comparisons can be made between distances from the map and measurements of the actual mapped areas. If these values match with negligible error, then the map can be deemed sufficiently accurate for use with autonomous navigation.

IV. Control Process of the Robot

Manual control is effective when running the robot in predetermined paths to obtain data for experimental purposes such as identification of errors with sensors, wheels, track and for further improvements.

Manual navigation is done using an Android based application which communicates with the Arduino control

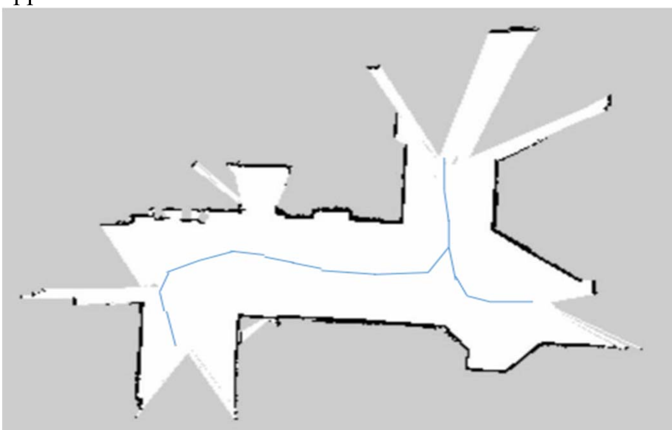

Fig. 9. Robot navigating path

board via Bluetooth. A Bluetooth module (HC-06) is used to fulfill the communication between android application and control board. Android phone is used to control.

Control method used for mapping process is different from manual control method based on Bluetooth module. For the mapping process we have given a command velocity to the robot using rqt console in ROS [5]. This is the same command velocity acquired by the robot when it is navigating autonomously in the mapped environment. For this purpose we acquired the relationship between speed of the motors and the input PWM to the motors (18).

$$v = 0.0032P - 0.4045 \tag{18}$$

The relationship above is assumed as a linear relationship though it is not exactly linear. But this relationship is suitable for the PWM control of the motors since a little variation in PWM can be neglected. In the above equation v is the speed of the robot and P is the PWM.

V. CONCLUSION

It can be observed that, using ROS and its features, we can generate consistent and stable robot location estimates with bounded errors affected by a minimum amount of noise. As stated earlier, there is a small localization error incorporated with the encoder values due to the slippage occurring constantly between the tracks and the wheels of the robot platform. This could also be affected by the PID controller which is not tuned perfectly at the moment causing non-smooth movement of the robot. It is seen that both in indoors as well as in outdoors, the objects like walls that should be presented as straight lines are actually not presented as exact straight lines in the generated map. This is also due to the above reasons.

Sensor fusion is one method of dealing with the slippage of the tracks. Using CMOS cameras to count the cycles of the belt, the absolute linear distance traveled by the robot can be measured and by that, the slippage can be measured. Slippage can be reduced by applying grease inside tracks and reducing the speed, especially in indoors. This reduces not only the localization errors but also the mapping errors in both indoor and outdoor.

The two experiments were done only with stationary landmarks in the environment. GMapping algorithm copes with both stationary and non-stationary landmarks. In the mapping process the detected non-stationary landmarks will be removed from the created map and only stationary landmarks will be added to the state vector

"Loop Closure" is a challenging problem in SLAM that could be solved with the proposed SLAM algorithm using integer programming based multi-frame data association [13].

This can be further improved to navigate autonomously in an unknown environment by adding "move_base node" [5] which enables obstacle avoidance for the robot with the help of odometers and the generated map.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to academic staff and our colleagues of Department of Electrical engineering, University of Moratuwa for their guidance and valuable support throughout the course of this project work.

REFERENCES

[1] Rui-Jun Yan; Jing Wu; Sung-Jin Lim; Ji-Yeong Lee; Chang-Soo Han, " Natural corners-based SLAM in unknown indoor environment," Ubiquitous Robots and Ambient Intelligence (URAI), vol., no., pp.259,261, 26-28 Nov. 2012

[2] Yang, P., "Efficient particle filter algorithm for ultrasonic sensor based 2D range-only simultaneous localization and mapping application, " Wireless Sensor Systems, IET , vol.2, no.4, pp. 394,401, December 2012

[3] Zhe Zhang; Hong Guo; Nejat, G.; Peisen Huang, "Finding Disaster Victims: A Sensory System for Robot-Assisted 3D Mapping of Urban Search and Rescue Environments," Robotics and Automation, 2007 IEEE International Conference , vol., no., pp. 3889,3894, 10-14 April 2007

[4] Ventura, R.; Lima, P.U., "Search and Rescue Robots: The Civil Protection Teams of the Future," Emerging Security Technologies (EST), vol., no., pp.12, 19, 5-7 Sept. 2012

[5] ROS [online]. Available: http://wiki.ros.org/

[6] Why-is-Robot-Operating-System-ROS-preferred [online]. Available: https://www.quora.com/

[7] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, 1st ed. The MIT Press, 2005.

[8] Biber, P.; Strasser, W.,"The normal distributions transform: a new approach to laser scan matching," Intelligent Robots and Systems, 2003. (IROS 2003)., vol.3, no., pp.2743,2748 vol.3, 27-31 Oct. 2003

[9] Grisetti, G.; Stachniss, C.; Burgard, W.,"Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," Robotics and Automation, ICRA 2005 , vol., no., pp.2432,2437, 18-22 April 2005

[10] Burgard W., Stachniss C., Bennewitz M., Arras K., "Introduction to Mobile Robotics – Bayes Filter, Particle Filter and Monte Carlo Localization", Lectures, Uni Freiburg.

[11] F. Lu and E. Milios. "Globally consistent range scans alignment for environment mapping". Journal of Autonomous Robots, 4:333–349, 1997.

[12] Censi, Andrea; Iocchi, L.; Grisetti, G.,"Scan Matching in the Hough Domain," Robotics and Automation, ICRA 2005, vol., no., pp.2739,2744, 18-22 April 2005

[13] Chandima Dedduwa Pathiranage, Keigo Watanabe and Kiyotaka Izumi, "A fuzzy logic based approach to the SLAM problem using pseudolinear models with multiframe data association," *Artificial Life and Robotics*, vol. 13, no. 1, pp. 155-161, Dec. 2008.