

Improved Map Generation by Addition of Gaussian Noise for Indoor SLAM using ROS

Barry Loh Tze Yuen, Khairul Salleh Mohamed Sahari, Zubaidi Faiesal Mohamad Rafeai

*Department of Mechanical Engineering, Universiti Tenaga Nasional
Selangor, Malaysia*

E-mail: khairuls@uniten.edu.my

Abstract

Rao-Blackwellized Particle Filter (RBPF) is used in this paper to solve the Simultaneous Localization and Mapping (SLAM) problem. RBPF algorithm uses particle filter where each particle carries an individual map of the environment. With the usage of Robot Operating System (ROS), GMapping package was used as a basis for map generation and SLAM. To improve the map generation, Gaussian noise was introduced to the data from laser range finder and also the odometry from the robot Pioneer P3AT's pose. The introduced algorithm was successful in decreasing the uncertainty as well as increased the knowledge of each particle in the estimation of the robot's pose, proven through practical experiment. Exploration experiments were also carried out to test the performance of P3AT based on our proposed method.

Keywords: SLAM, ROS, Gaussian Noise, map generation, exploration.

1. Introduction

Over the years, the rapid development of technology has contributed to the development of robots. The production of many new sensors including laser range finders and Kinect camera provide more precise, reliable and faster readings.

Robots equipped with sensors allow them to learn about their surrounding environment. This enables the robot to conduct various tasks that include map generation, localization and exploration, indoor services like moving from one room to another, gripping and picking up objects. Moreover, robots should be able to conduct Simultaneous Localization and Mapping (SLAM) to achieve even greater results.

SLAM, as the name suggests, allows the robot to conduct mapping of the environment while localizing itself in the map being created on-the-fly. An addition to SLAM algorithm is to utilize more than one sensor for more accurate results. Castellanos et al. presented a 2D version of SPmodel, a probabilistic representation model and EKF integration system for the usage of

multi-sensor fusion.¹ Results obtained proved that fusing two sensor data improved the localization of mobile robots even more accurately. Besides that, Arras et al. proved that using feature extraction and the Kalman filter to fuse data from a laser range finder and monocular camera increased the precision of localization significantly.² On the other hand, utilization of open-sourced systems and software are becoming very common. Zaman et al. presented their findings of utilizing open-sourced system, Robot Operating System (ROS) for mapping, localization and autonomous navigation.³ Gong et al. also presented their research on utilization of ROS for the purpose of object localization, using RFID and laser scans.⁴ Despite being open-sourced and free, there are also drawbacks.

Hence this paper is produced to test the reliability and precision of already programmed SLAM algorithms found in open-sourced software ROS⁵ using a common research robot. Based on the results obtained from experimentations, improvements were carried out to further enhance the SLAM and map generation. By studying the algorithms written by the open-sourced

community, a portion of the code was found that the addition of a Gaussian type noise will implement the usage of Kalman filter to improve the knowledge of the particle filters of the map. Besides that, exploration experiments were carried out on the robot to further test the reliability and performance of the proposed improvement. To achieve this objective, a Hokuyo URG-04LX laser range finder was used together with a Pioneer 3-AT equipped with wheel encoders only. A common laptop was used to provide the algorithms and data collection throughout the process.

2. Current Work

In this research, a ROS-based control system was used for the Pioneer 3-AT robot for localization and creation of maps in unexplored controlled and uncontrolled indoor environments.

2.1. ROS setup for Pioneer 3-AT

The two sensors used in this research are the Hokuyo URG-04LX, and the wheel encoders of the mobile robot. Linux (Ubuntu 12.04 LTS) based computer system equipped with ROS was used to control the robot. For manned maneuvers of the robot, a keyboard was used. Both the laser and the mobile robot are connected to the computer through USB ports. Based on ROS, the packages and stacks used in this research are the P2OS, Hokuyo_node, SLAM_Gmapping, Navigation stack and the Exploration stack. The P2OS and Hokuyo_node package serve as the driver to establish connection and data transfer between the laser range finder and the mobile robot to the computer. Not only that, the P2OS package provide a 3D model of the Pioneer 3-AT that can be viewed in the RVIZ Visualization tool and also allows the movement of the mobile robot to be controlled by a joystick or a keyboard. Fig. 1 shows the experimental setup and the program lifecycle for this research work.

SLAM_Gmapping package provides a highly efficient Rao-Blackwellized particle filter (RBPF) to create occupancy grid maps. The RBPF algorithm not only incorporates data fusion between the laser and robot's pose, but also solves the SLAM problem. However, the utilization of particles to carry individual map of the environment possess a big threat to the performance of the computer. This is because large computations have to be conducted. Grisetti et al. proposed adaptive techniques to be included in the

GMapping algorithm to reduce the number of particles in a RBPF for learning the occupancy grid maps.⁶ Gmapping also computes an accurate proposal distribution, considering not only the movement of the robot, but also the most recent observation. This produces drastic decrease of the uncertainty in the robot's pose, in the prediction step of the filter. Resampling operations were also carried out selectively which reduces the problem of particle depletion.

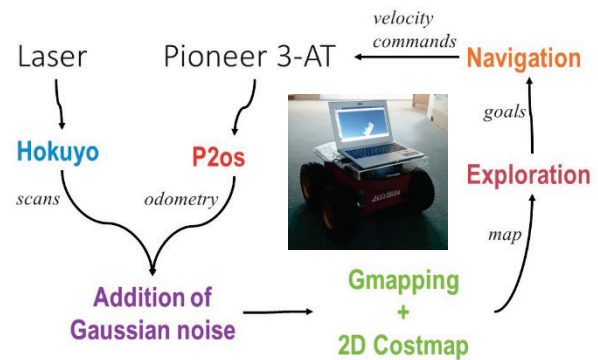


Fig. 1: Experimental setup and program lifecycle

Following up the algorithm of Gmapping, individual importance weightage is assigned to each particle based on the significance of the sampling principle. The weightage allows resampling that performs particle filtering when the target distribution differs from the proposal distribution. Instinctively, the more efficient proposal distribution approximation of the target distribution, the better is the performance of the filter. Montemarlo et al. presented RBPF that uses a Gaussian approximation of the improved proposal in the context of landmark-based SLAM.⁷ However, Grisetti et al. found that GMapping uses the same principle of Gaussian approximation of improved proposal but in the context of dense grid maps.⁶

The Gaussian computes for each particle using a Kalman filter that estimates the pose of the robot. This technique is utilized whenever the map is represented by a set of features and the errors affecting the feature detection is assumed to be Gaussian. From the understanding, to fully utilize the Kalman filter, a Gaussian distribution or also known as normal distribution type of noise was added to both the laser range finder sensor data and the wheel encoders' sensor data. The Box-Muller transform was used to add a Gaussian distribution noise to the sensor data. In Ref. 8, Box and Muller mentioned that the Box-Muller

transform is a pseudo-random number sampling method to generate pairs of standard, independent, normally distributed random numbers when provided uniformly distributed random numbers. The Box-Muller transform can be expressed in two different forms, the basic and polar form. The polar form calculations were adapted in the source codes due to its advantages of being arguably faster and cheaper to the performance of the CPU. The following equations explain how Gaussian Noise was added to both the odometer and laser range scanner data associated with this paper using Box-Muller Transform.

First, random numbers in the range from $[-1, +1]$ for variables x_1 and x_2 are generated and substituted into equations (1) and (2), to generate values of u and v with standard normal distribution and standard deviation of 1.

$$u = \sqrt{-2 \ln x_1} \cos(2\pi x_2) \quad (1)$$

$$v = \sqrt{-2 \ln x_1} \sin(2\pi x_2) \quad (2)$$

The value of s is then calculated based on equation (3). If value equals to 0 or more than 1, generate x_1 and x_2 until condition is fulfilled for value of s .

$$s = u^2 + v^2 \quad (3)$$

$$U = x_1 \cdot \sqrt{\frac{-2 \ln s}{s}} \quad (4)$$

$$V = x_2 \cdot \sqrt{\frac{-2 \ln s}{s}} \quad (5)$$

Values of U and V are then calculated using equations (4) and (5). The values of U and V are then multiplied with a specified standard deviation to produce Gaussian noise. For this paper, 0.01 is used. Gaussian noise is then added to the sensor data as shown in Fig. 2 and Fig. 3.

Algorithm 1: Adding noise to laser scans

```

input : Laser scans
output: Laser scans with added Gaussian noise

begin
    subscribe to /scan topic;
    obtain ranges parameter from sensor_msg/LaserScan message;
    forall the elements of ranges do
        calculate independent variable from uniform distribution using
        Box-Muller transform;
        add to each ranges value in ranges array;
    publish to /scan_noise topic;

```

Fig. 2: Algorithm for addition of Gaussian noise to laser range scanner data

Algorithm 2: Adding noise to odometry data of robot's pose

```

input : Odometry from robot's pose and velocity of robot
output: Odometry data with added Gaussian noise

begin
    if new velocity of robot is published then
        compare velocity from /cmd_vel and velocity from odometry;
        obtain differences and calculate the weightage;
        sample weightage and calculate independent variable from
        uniform distribution using Box-Muller transform;
        add to the speed;
        if linear speed = 0 then set linear speed = 0 ;
        if angular speed = 0 then set angular speed = 0 ;
        publish to topic /pose_noise and tf transform of /odom_noise to
        /base_link;

```

Fig. 3: Algorithm for addition of Gaussian noise to odometry sensor data

The navigation stack provides a costmap that is built around the mobile robot as it moves to provide information of any nearby obstacles. The navigation stack also provides Dijkstra algorithm, Trajectory Rollout and Dynamic Window Approach for local and global path planning, and obstacle avoidance.

The exploration stack implements a frontier-based entropy approach with loop closing abilities for further stability. The exploration node scans the environment around the current robot's position for reachable empty spaces within the map. Nearest goals will be selected while unreachable areas are ignored. The exploration node work very closely with the costmap created by the navigation stack.

Throughout the experiments, the configuration files for all the packages were tweaked to provide improvements and enhance the results obtained.

2.2. Experiments

Two experiments were carried out in different types of environments. The map area, as shown in Fig. 4, is a self-created controlled environment. It is an empty room, filled with cardboard boxes, with a dimension of 7m x 5m. The purpose of the controlled environment is to ensure a safe and high performance working automatic robot system before proceeding with more complex situations. Fig. 5 shows the map generated which only took about 90 seconds.

The other experiment was carried out in an uncontrolled environment, that is the South Wing corner of the College of Engineering, UNITEN with total estimated dimensions of about 41m x 25m. During this

experiment, to prove that the map generation will be increased further, the mobile robot was manually driven twice. Once without the addition of Gaussian noise to the sensor data, and another time with the addition of Gaussian noise. Few parameters like the configurations, path taken by the mobile robot and the speed of the mobile robot were kept constant throughout the entire experiment to obtain results with integrity. Fig. 6 and Fig. 7 show the map generated without addition of Gaussian noise and addition of Gaussian noise, respectively.



Fig. 4: Self-created controlled environment



Fig. 5: Map generated of the controlled environment

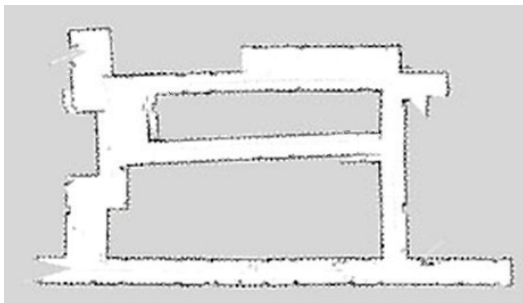


Fig. 6: Map generated for uncontrolled environment without addition of Gaussian noise

For the final experiment, the mobile robot was equipped to utilize the exploration stack in the same uncontrolled environment. During this experiment, the

mobile robot was allowed to be independent in conducting path planning and obstacle avoidance. The entire map generation took about 35 minutes and shown in Fig. 8. Throughout all the experiments, there were many issues faced and observations made that will be discussed further in the next section.

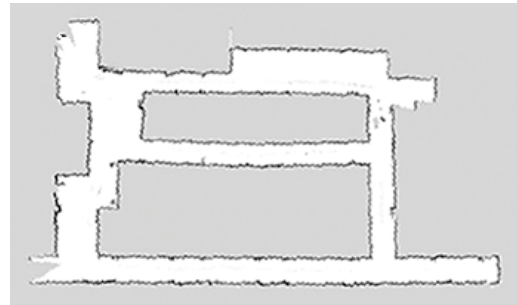


Fig. 7: Map generated for uncontrolled environment with addition of Gaussian noise

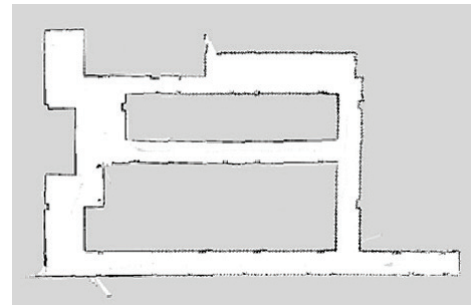


Fig. 8: Map generated for uncontrolled environment with addition of Gaussian noise and exploration stack

3. Discussion

3.1. Observations

In smaller environments, the mobile robot is able to navigate around smoothly and generate accurate maps. This is mainly due to the fact that in that short period of time, not many errors were accumulated by the wheel encoders. Not only that, the laser range finder is powerful enough for the small environments. However, in bigger environments like the uncontrolled environment, the map generated tends to have a lower resolution and is inconsistent. As the mobile robot continues to navigate and generate map of its surrounding environments, the wheel encoders tend to accumulate more and more errors, mainly due to

slippage. Hence, to counter the accumulating errors, the sensor data from the wheel encoders were fused together with sensor data from the laser range finder. However, as shown in Fig. 6, the effort still may not be enough as the map generated is not very satisfactory. With the addition of Gaussian noise to both the sensor data, the map generation improved quite significantly due to the usage of Kalman filter.

On the other hand, Fig. 8 displays one of the best maps generated. The exploration stack from ROS implements a frontier-based exploration algorithm, which also includes loop closing methods. It formulates the entropy of the environment as the mobile robot is navigating. Higher entropy generated signifies a higher uncertainty of the knowledge of the environment. This allows the mobile robot to recalculate and navigate through the most optimal path with low entropy to generate the most accurate map.

3.2. Issues

Despite providing the algorithms and hardware for optimal map generation, there were still some factors that were affecting the results. Most of these factors involve the aspects related to both the environment and the mobile robot. These issues include:

A. Height and orientation of the laser range finder

The laser range finder was placed 20cm above the ground and onto the platform of the mobile robot. The laser range finder also scans the environment at a plane parallel to the ground. Due to that, there were certain occasions whereby the mobile robot could not navigate away from an obstacle as the obstacle's height was below the scanning plane of the laser. Furthermore, the laser range finder has a blind spot at the back and would not be able to detect obstacles behind the robot.

B. Objects with wider lower parts

Certain objects in the environment like chairs and tables affect the navigation of the mobile robot. The mobile robot tends to have the perception that the path planned is clear. However, the chair in Fig. 9 has its frame sticking out and the mobile robot has a very high chance of actually colliding with it.

C. Surface difference and inclination

As shown in Fig. 10, the difference in surface type and its inclination caused a disturbance in the sensor

data of the laser range finder, generating maps with structural irregularities. Besides the laser range finder, the wheel encoders also produce a much higher error count whenever the mobile robot is navigating through this area. Moreover, the main plug hole covers which are not properly fitted a shown in Fig. 10 (right) also contributed to inconsistency in the map generated.



Fig. 9: Chair with a wider lower part



Fig. 10: Left picture shows the different surface types and inclination. Right picture shows the main plug hole covers

D. Overbearing dynamic obstacles

While conducting the exploration experiment, there may be times whereby the mobile robot may stop for a long period of time before being able to navigate itself. When there are too many people walking pass by in the room, the costmap generated by the navigation stack tends to get very congested and not being refreshed quickly. The mobile robot will then assume that there are many obstacles very close to it in all directions.

4. Conclusion

Maps of the environment are very important to the mobile robot. With accurate maps, the mobile robot may be able to conduct proper path planning, obstacle avoidance and precise localizations.

The GMapping node from ROS utilizes the RBPF SLAM. Not only that, it also uses the EKF for multi-sensor fusion to reduce the amount of errors accumulated by the sensors. Based on the algorithm, it detects that if the sensor data contains error of Gaussian distribution, a Kalman Filter is used to compute each particle that estimates the robot's pose. Given the linearity and advantages of Kalman Filter, the estimation would be more accurate to correct the localization. In turn, it will generate more precise and consistent maps as the particles contain more valuable information than before. Hence, two nodes were written in C++ language to perform addition of uniform distribution values using the Box-Muller transform to both the laser range finder sensor data and the wheel encoders sensor data. Experiments were then carried out to prove the improvement of the map generated which turned out to be successful as shown in Fig. 7 and Fig. 8.

However, this improvement is only limited to applications using RBPF-based grid-mapping algorithm and the usage of GMapping in ROS. Further implementations in the future may include a visual sensor for calculating depths to enhance further the knowledge of the particles and reduce the accumulated errors. Not only that, it is also recommended to look for improvements in the costmap generated, reliability of the sensor data, and even various mapping strategies that could be implemented.

Acknowledgements

This work is supported by Universiti Tenaga Nasional through research grant: J510050673.

References

1. J. Neira, J. Horn, J. Tardos and G. Schmidt, Multisensor mobile robot localization, in *Proc. IEEE Int. Conf. on Robotics and Automation*, (1996), vol. 1, pp. 673-679
2. K. Arras, N. Tomatis and R. Siegwart, Multisensor on-the-fly localization using laser and vision, in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, (2000), vol. 1, pp. 462-467
3. S. Zaman, W. Slany and G. Steinbauer, ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues, *Electronics, Proc. of Saudi Int. Electronics Communications and Photonics Conf.*, (2011), pp. 1-5
4. S. Gong, H. Liu, Y. Hu, J. Zhang, ROS-based object localization using RFID and laser scan, *Proc. of Int. Conf. on Information and Automation*, (2012), pp. 406-411
5. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, ROS: an open-source robot operating system, *ICRA Workshop on Open Source Software*, (2009)
6. G. Grisetti, C. Stachniss and W. Burgard, Improved techniques for grid mapping with rao-blackwellized particle filters, *IEEE Trans. on Robotics* **23**(1) (2007) 34-46
7. M. Montecarlo, S. Thrun, D. Roller and B. Wegbreit, Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence*, (Morgan Kaufmann, 2003), pp 1151-1156
8. G. E. P. Box and M. E. Muller, A note on the generation of random normal deviates, *The Annals of Mathematical Statistics* **29** (1958) 610-611