SONAR Validation for SLAM

Akshay Kumar Shastry, Sanjay Anand, V. Chaitra, K. Uma Rao, and D. R. Akshay

***Abstract*—SlAM stands for simultaneous localization and area mapping. It deals with mapping areas which are uncharted or are not easily accessible in a given environment as well as updating currently available maps, while at the same time keeping track of the current location using the map generated. This is often used with autonomous robots and unmanned vehicles, to map hostile or unfriendly or unknown environments. Commonly used SLAM hardware is very expensive. This work deals with optimizing economic constraints and quality with minimum trade-off between the two. This is done using open source software and limited number of sensors, which makes the prototype very easy to implement. It also uses a very easy, straight forward algorithm to map the surroundings using sonar. We've restricted the movement of the autonomous vehicle without compromising on the quality of the map generated.**

***Index Terms*—Mapping, localization, SONAR, open source.**

I. Introduction

SLAM-Simultaneous localization and area mapping is a method used by autonomous vehicles and mobile robots to map an unknown area or update a map of a known environment successfully while at the same time keeping track of its current location in the map generated. Mapping is the representation of an area pictorially and localization is the identification of one's position on a map accurately. The most commonly used methods for SLAM [1] are LIDAR [2], [3], Infra-red and SONAR [1], [4]. In this work we've used a single SONAR sensor, instead of multiple sensors [5], to map the surroundings which provide us with depth information using which we can generate a 2-dimensional map of the surroundings. Our current work deals with making it cost effective to implement, creating an easily understandable and readable map, at the same time keeping the number of movements of the mobile robot minimal. Further, while mapping, the number of iterations taken to generate a map is optimized as well.

II. Traditional Methods

A. LIDAR (Light Detection and Ranging)

In advanced SLAM, LIDAR [2], [3] is the most extensively used type of sensor. Laser is used to measure distances and generate a 3-dimensional image of the surroundings in which the robot is placed but the cost to implement LIDAR is very high and is quite complex as it involves extensive image processing algorithms [2], [3].

B. Infra-Red Sensor

Infra-red sensors can also be used for SLAM. Infra-red sensors have very good resolution but the range they can cover is very small relatively. They are susceptible to interference from light sources and reflective surfaces which make them very difficult to use as a sensor in a noisy background.

C. SONAR

SONAR (Sound Navigation and Ranging) gives the depth information. We can only measure the distance of the obstacle from the sensor. At best, we can only generate a 2 –Dimensional map. But with the addition of a camera [6] we can obtain an image with depth information. In our current work we focus on using a single SONAR sensor to create a 2-Dimensional map of the surroundings.

III. Functional Block Diagram

An open source arduino platform is used with an atmega-8 micro-controller. Arduino is an electronics prototyping platform with easy to use hardware and software. It receives the input from the sensor and sends it to processing. Processing is user friendly and is a java based graphics tool. We have used processing, to act as an interface to take the data as input and generate a 2-dimensional map.

Fig. 2 represents the functional block diagram. The robot when introduced into a new environment recognizes its position in the environment. This process is termed as localization. The robot then waits for the sensor to provide valid data on the surroundings including the obstacles present. The sensor, LV-Max SONAR-EZ3, which has a range of 6.45m and a resolution of 2.45 cm, has a narrow beam width compared to other sensors in the same category. It is economical, very reliable, and can take a reading every 50ms but has a dead zone of 15cm. This sensor is mounted on a servo motor on the robot, which can accurately turn one degree each step. The servo rotates from 0 degrees to 180 degrees completing one sweep while the sensor correspondingly takes 180 values. The sensor produces a voltage value which is directly proportional to the distance of the obstacle from the sensor. These values are digitized using a 10-bit ADC and sent to the computer for further processing in real time. In this work, the system has been directly connected to the microcontroller on the robot chassis, making it a wired communication strategy. However, the same can be made into a wireless system, using Zigbee. The computer processes the data from the ADC, to get an estimate of the depth of various objects. This estimate is simply a conversion

Manuscript received May 30, 2013; revised August 18, 2013.

Akshay Kumar Shastry, Sanjay Anand, and V. Chaitra are with the K.S Institute of Technology, Bangalore, affiliated to Visveswaraya Technological University (e-mails: findthisgreatguy@gmail.com).

Uma Rao K. is with the Department of Electrical & Electronics Engineering, R.V. College of Engineering, Bangalore.

from the sensor unit to a standard unit. The sensor used here, gives a numerical value which is twice the actual distance in inches. This data is used to generate a 2-dimensional map.

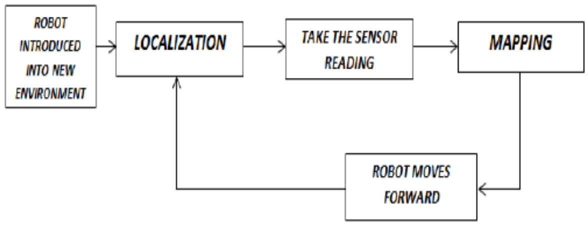

Fig. 1. Functional block diagram of the prototype.

The map is continuously updated as the sensor sweeps over 180°. The robot moves forward a specified distance, after the first sweep is completed. The distance can be programmed depending on the application. The whole process is repeated again and the new information from the surroundings is updated in the generated map.

IV. Technical Challenges

In any SLAM technique, there are a few technical challenges that must be addressed and overcome. One of the most important areas of concern is whether mapping must be done first or localization? The answer to this question is not very straightforward as it depends on various factors such as mobility, accuracy of motors and sensors *et al*.

A. Mapping

In slam the challenge is to generate an unbiased map for localization and an accurate position estimate. However, both these parameters have inherent errors due to the sensor accuracy and Odometry errors [5]. The map is generated neglecting the errors. Every iteration of the algorithm, affects the inaccuracy in localization and mapping, due to these errors. In this work, the error in mapping has been reduced by optimizing the number of iterations.

B. Sensing

The different sensors used, have their own advantages and disadvantages. Depending upon the requirement we select suitable type of sensors. In every data that the sensor acquires there is an error factor. This error factor depends on the sensitivity and resolution of the sensor [5]. There is also an error due to the analog to digital conversions as well.

Most sensors have dead zones or areas where it cannot judge the accurate distance of the object or obstacle but simply recognizes the presence of an object and only gives a minimum value as output. The sensor which we've currently used for the prototype has a dead zone of 15cm. Due to the divergent nature of the beam from the sonar; noise due to reflection from an object just within the beam width is added. There is also noise due to vibrations from the servo motor.

C. Localization

The robot has to recognize its exact location on the map which has been generated but faces various issues as it has no accurate data on how much it has moved from the last location. There are many related technical issues such as errors in the DC motors, Odometry errors. Exact location of the robot cannot be determined precisely and requires complex calculations hence an estimation of where the robot might be after it has moved is calculated, based on the last position and the programmed distance that it moves. After approximating the new position of the robot on the generated map, sensor readings are again taken and the new information is added to the existing map.

V. Proposed Algorithm

In our proposed work the robot when introduced into an unknown environment, localizes itself first and then generates the 2-dimensional map around it without actually moving. The robot assumes that it is at the origin of the map and maps all the obstacles in the first two quadrants of the Cartesian plane.

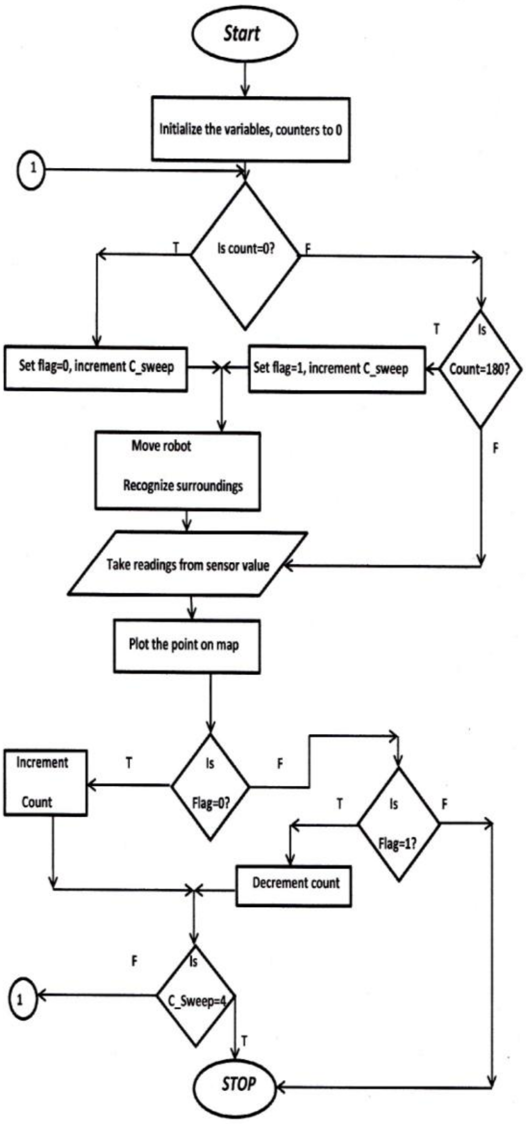

Fig. 2. Flowchart of the proposed algorithm.

Steps Involved in Mapping

1) The variables and counters are initialized to zero. C_sweep is used to keep track of the number of sweeps and is constantly updated when the servo motor is at 0° or

180°. Count is used to keep track of the angular displacement of the servo motor. Flag is used to keep track of direction of mapping and is reset or set when the servo motor is at 0° or 180°.

2) Position of the servo motor is checked i.e. whether the servo motor is at 0° or 180° and the flag is set to 0 or 1 respectively. Depending on the value of the flag, mapping is done from left to right when flag is set and from right to left if flag is reset.
3) The robot moves only when there is a change in the flag register, it localizes after moving and takes in the sensor reading. The sensor readings are plotted on the map.
4) If flag=1: count is decremented, if flag=0: count is incremented, every time a sweep is completed the flag is complimented. The count value gives the theta of the servo motor.
5) If the no. of sweeps exceeds 3 then it exits from the loop.

It only has to move a few steps to update the generated map with new information which was not visible from the initial location. The perpendicular distances of the objects around it are converted into polar co-ordinates and are plotted as a function of angle theta of the servo motor. After the robot completes one sweep, a skeletal map is generated. Then the robot moves forward for a predetermined distance in the forward direction.

From this new point it localizes itself in the map and then starts to perform the second sweep. At this point it increases various counters and sets flags to keep track of the direction and number of sweeps. On the completion of the second sweep, new data is added and the existing map is updated. Robot checks the number of valid iterations and various conditional flags. Then it moves forward again and the whole process is repeated. After an iteration there is an addition of information but if the number of iterations exceeds a certain number, no new information is added to the map and there is an increase in the number of errors or noise in the map. This is kept in check by reducing number of iterations. This number of optimal iterations was obtained by trial and error and was found to be dependent on the environment, usually two to three iterations were found to be sufficient.

VI. Results and Applications

In our current work we have extensively focused on Sonar validation for SLAM. We have mapped various scenarios with our prototype where we've generated an environment in which we expect our prototype to be deployed. We have focused on a non GPS domestic environment where our prototype can be put to use.

In Fig. 3 our prototype has mapped a nearly empty room with a single object. As indicated in the Fig. 3 there is noise present, this is due to the divergent beam of the sonar. We tried to remove this noise by using an averaging filter but there was no noticeable difference. Hence the filter was removed.

In Fig. 4 multiple obstacles have been placed, and the obtained mapping is after two sweeps of the sensor. For the second sweep you can notice that the robot has recognized itself in the map in its new position as well as its initial position.

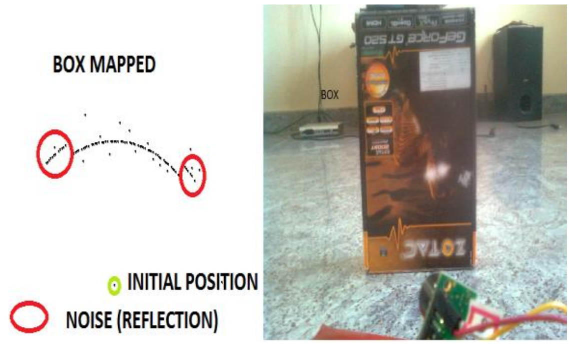

Fig. 3. Single sweep mapping of an object.

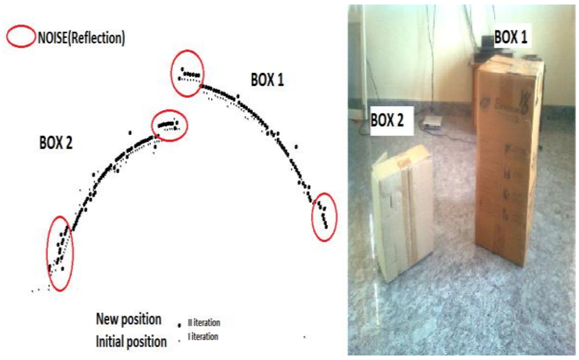

Fig. 4. Multiple sweeps of multiple objects.

Fig. 5. Without new object.

Fig. 6. New object recognized after the 1st sweep during the 2nd sweep.

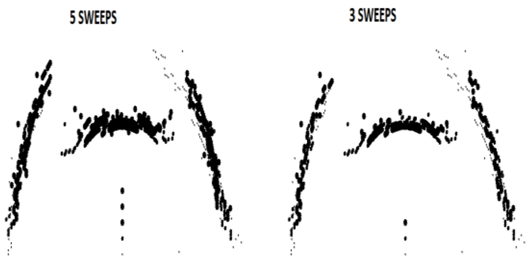

Fig. 7. Multiple sweeps adds more noise.

The mapping was tested by introducing a new object after the first sweep. In Fig. 5 the map is the one generated after the first sweep and Fig. 6, shows the map, after the new object has been introduced. We can see that the map is updated during the second sweep to include the newly introduced object. Thus, the map is updated in real time. In Fig. 7 we compare maps obtained after multiple sweeps for the same area and notice that there is no new information added but a lot more noise is present. Therefore the number of iterations must be of optimum value.

The proposed prototype can be used in a domestic environment having a very low visibility. Our algorithm can be used in unmanned aerial vehicles used for slam, robots involved in search and rescue operations for mapping collapsed buildings as well as mapping areas where there is zero visibility. The components used are also easily available and affordable. The prototype is economical and uses minimum hardware and software all of which is open source.

VII. Conclusion

In the work presented in this paper, Simultaneous Localization and Mapping, of an area, by an autonomous robot has been achieved using a sonar sensor. A simple algorithm has been developed to map in a domestic environment. This can easily be extended to larger environments. The novelty of the work is that the robot does not have to physically move over the entire area, to map it and recognize the obstacles. Further, the whole mapping has been achieved using a single sensor, as opposed to available strategies which often use multiple sensors. The number of iterations required for an effective mapping has been reduced. We can implement the algorithm and the hardware in various unmanned vehicles in environments where visibility is restricted as well as accessibility restricted environments since our algorithm is applicable to simple prototypes that are small in size. Our future work will concentrate on generating 3-dimensional maps using cameras along with a sonar sensor by further optimizing our algorithm.

References

[1] S. Riisgaard and M. R. Blas, *SLAM for dummies.*

[2] K. Cho, S. H. Baeg, and S. Park, "Natural terrain detection and SLAM using LiDAR for an UGV," in *Proc. 12th International Conference on Intelligent Autonomous Systems, IAS-12*, June 2012.

[3] S. Becker and N. Haala, "Grammar supported façade reconstruction from mobile iDAR mapping," in *Proc. ISPRS XXXVIII CMRT-09*, September 2009.

[4] K. Celik, S. J. Chung, S. M. Clausman, and A. K. Somani, "Monocular vision SLAM for indoor aerial vehicles," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, October 2009.

[5] R. Wagner, O. Birbach, and U. Frese, "Rapid development of manifold-based graph optimization systems for multi-sensor calibration and SLAM," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011*, September 2011.

Akshay Kumar Shastry was born on June 20, 1992. He studies in pre-final year and does his bachelor in Engineering, Electronics and Communication, K.S Institute of Technology, Bangalore, affiliated to Visveswaraya Technological University. He is interested in the field of robotics and their applications.

Sanjay Anand was born on August 24, 1992. He studies in pre-final year and does his bachelor in Engineering, Electronics and Communication, K.S Institute of Technology, Bangalore, affiliated to Visveswaraya Technological University. He is interested in the field of robotics and smart systems.

V. Chaitra was born on June 24, 1992. She studies in pre-final year and does her bachelor in Engineering, Electronics and Communication, K.S Institute of Technology Bangalore affiliated to Visveswaraya Technological University.

Uma Rao K. was born on 10th March 1963. She obtained her B.E in Electrical Engineering and M.E in Power systems from University Visvesvaraya College of Engineering, Bangalore and her Ph.D from Indian Institute of Science, Bangalore. She is currently Professor, department of Electrical & Electronics Engineering, R.V. College of Engineering, Bangalore. Her research interests include FACTS, Custom Power, Power Quality, renewable energy and technical education.