Obstacle Detection and Map Building with a Rotating Ultrasonic Range Sensor using Bayesian Combination

S. Kodagoda*, E.A.S.M. Hemachandra*, P.G. Jayasekara*, R.L. Peiris+, A.C. De Silva+ and
Rohan Munasinghe* (Corresponding author)
*Department of Electronic and Telecommunication Engineering , + Department of Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
Email: *{sk02,easmh02,pgj02,rohan}@ent.mrt.ac.lk , +{roshan02,anjula02}@elect.mrt.ac.lk

Abstract— **Map building and obstacle detection are basic tasks in mobile robot navigation. In this paper we present a probability based solution for map building which utilizes time of flight information from an ultrasonic sensor. In contrast to most of the available map building systems, this uses a single ultrasonic sensor, mounted on a rotating shaft. Discrete sonar observations taken at regular time intervals, by rotating the shaft in small step angles are incrementally merged into partial planes to produce a realistic representation of environment that is amenable to sonar localization. Readings of close proximity are averaged to suppress the transient errors and the results are further refined by using the sonar probability model to fill an occupancy grid and Bayesian combination for the purpose of data fusion. Experimental results obtained in a real indoor environment are presented to verify the validity of our algorithm.**

***Keywords*—Ultrasonic Range Sensor, Map building, Localization, Data Fusion, Bayesian Combination**

I. INTRODUCTION

Recently, mobile robots have grown in popularity in the fields of industrial automation, service, security and surveillance etc. For autonomous operation of a mobile robot, intelligent sensor systems are needed to provide information to recognize the surrounding environment, localization and map building [3]. These are important in the sense of not needing a priori knowledge of the environment, and through an exploration, the robot is able to make an internal representation of its work environment. Various sensors can be used such as vision, ultrasonic sensors, laser or infrared range finders. Ultrasonic is possibly the most common sensor on commercial robots operating indoors and on research robots due to their low cost and simplicity [3].

In most of the applications, ultrasonic sensors are used to measure time-of-flight (TOF) of the echo pulse to obtain the distance, given the speed of sound in air at ambient temperature. It provides simple but accurate range information between sensor and reflector. Despite this simplicity and accurate range measurement, use of ultrasonic transducers in map building is prone to few shortcomings. First is due to the acoustic beam width of the transducer. The width of the centermost lobe is often 30^0 or 60^0 which reduces the angular resolution of a single measurement. The width of the beam is also responsible for a problem known as foreshortening. That is due to the width of the sound cone if the surface is not perpendicular to the sonar one side of the cone will reach the object first and return a range first [1]. Second problem is multiple specular reflections which occur when the wave form hits a surface at an acute angle and the wave bounces away from the transducer [1]. If the reflected signal is bounced off from other objects until some energy is returned to the transducer it makes matters worse with a time of flight incompatible with the true relative range. The third problem is Crosstalk which arises when a ring of transducers are used. In this case some specularly reflected sound from an Ultrasonic sensor might wind up getting received by a completely different Ultrasonic [1]. The receiving Ultrasonic is unable to tell the difference between the sound generated by itself or its peers.

The errors resulted by sensor shortcomings causes an increasing difference between the robot's presumed and actual position. This position error must be reduced by a localizer to avoid potentially costly complications [5]. For this reason the localization algorithm happens to be the heart of any positioning application. Localization algorithms fall into two broad categories: ionic and feature based. Ionic algorithms appear to be more the more popular in practice, in part because they usually use an occupancy grid [2]. Occupancy grids are a mechanism for fusing sensor data into a world model or map. Data fusion is done either following an algorithm provided by a formal theory of evidence, either Bayesian or Dempster-Shafer, or by a popular quasi-evidential method known as Histogrammic in Motion Mapping (HIMM) [2]. Since occupancy grids fuse sensor data, the resulting map does not contain as much sensor noise.

The main contribution in this paper has been the development of a robust map building system to be used in a mobile platform to locate itself in an environment. A robot using this system is equipped with only one Ultrasonic sensor unlike other popular mobile robots which use arrays of Ultrasonic sensors (Nomad200 robot has a ring of 16 ultrasonic sensors). Since it makes use of a single rotating sonar transducer it is free of shortcomings such as crosstalk mentioned above. But data fusion is carried out using Bayesian combination to refine the readings corrupted by other inherent errors of the sensor. The following sections will outline the system that has been developed and the results that have been achieved are attached at the end to show the success of our system.

II. OVERALL SYSTEM

There are several unique aspects to this system. Firstly the system uses a single sonar sensor (unlike the typical array of sensors) to scan a near 360^0 area achieved through the use of a stepper motor which is used to rotate the mounted sensor to different orientations. The second striking factor is the use of the Sonar probability model and the accompanying Bayesian combination method for data fusion and resolution improvement.

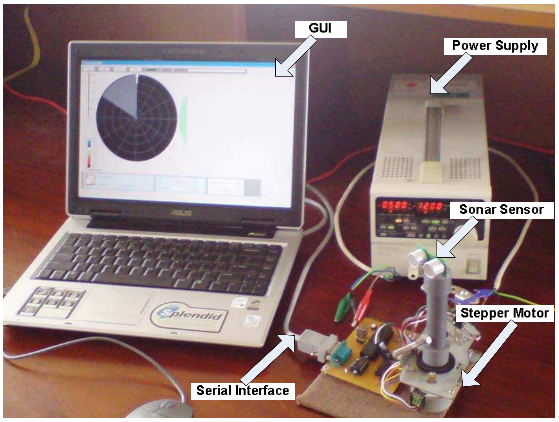

Figure 1. Overall System configuration

A. *Hardware Subsystem*

The hardware system of the map builder consists of three sub-components: Ultrasonic Range sensor, Stepper motor driver and Serial communication system. The microcontroller works as the central processing system and coordinates these sub-components.

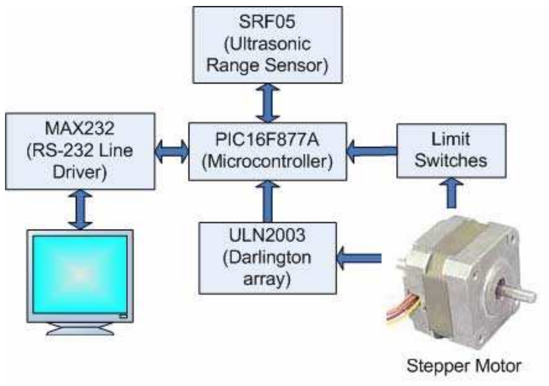

Figure 2. System Block Diagram

1) *Ultrasonic Range Sensor*

SRF05 Ultrasonic range sensor unit is configured to operate in 'Mode-1' in which a separate trigger and echo pins are utilized. These pins are connected to PORT A of the microcontroller unit and it is responsible for triggering and measuring the time of flight of the echo. The trigger pulse should have a minimum duration of 10µS, for stable operation and an internal ultrasound burst of 8 cycles at 40 kHz is sent. At the end of this process the module sets the echo pin and this state is preserved until an echo signal is received or sensors dynamic range is exceeded. At the reception of an echo, the echo pin is cleared by the module and the microcontroller detects it and stops the echo timer.

2) *Stepper Motor Driver*

The stepper motor is driven using the control signals initiated from PORT D of the microcontroller. The control signals are fed to a Darlington array (packaged inside ULN2003 IC) which in turn drives the stepper motor at 12V. Utilization of the Darlington mechanism prevents drawing large currents from the microcontroller at high loads. The resolution of the stepper motor is increased to 1.5^0 by utilizing a gear wheel with a gear reduction of 1:5. Readings are taken at each step location by momentarily stopping the shaft and four such readings are averaged to obtain a single reading for each 6^0. This helps the system to suppress any transient errors which may have occurred. The shaft on which the sensor is mounted sweeps a total angle of 300^0. The range measurements of the excluded area (60^0) are automatically covered by two 30^0 arcs at the two ends of the sweep. Two limit switches are used to detect the extreme ends of the sweep. The signals from the limit switches are interfaced to the PORT B of the microcontroller. PORT B interrupts are used to switch the direction of rotation upon receiving signals from the limit switches.

3) *PC Interfacing*

The program running on the PC sends commands via the RS232 interface which directs the microcontroller to switch between different modes of operation. Also the collected data on Time of flight is sent to the PC program via the RS232 link. The inbuilt USART(Universal Synchronous Asynchronous Receiver Transmitter) port of the PIC16F877A microcontroller is used for this purpose.

B. Software Subsystem

The software is responsible for the accurate representation of the scanned environment based on the information received from the microcontroller. This will handle the map display and the data fusion algorithm.

There are two modes of Operation available in the Map Building system aimed at giving the user added functionality. These are,

- Full scan or Global Mode: This allows the system to scan the entire area (300^0) that is limited by the limit switches. In this mode the area can be entirely viewed (360^0) using the probability Model.
- Local Scan Mode: This allows the system to scan a specific area (limited number of degrees) if a user needs to scan a particular segment.

The software has been written using Visual C# .NET 2005 and carries out serial communication with the PIC Microcontroller, the conversion of the echo parameters into distance information, the calculation of the probability model and also the display of the up-to-date map. The following sections explain the two methods in detail.

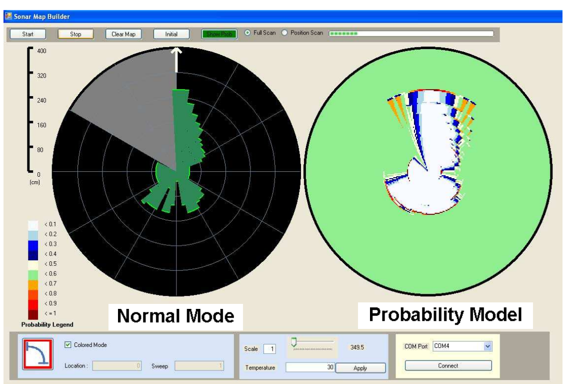

Figure 3. Map Builder GUI

III. MAP BUILDING

There are two methods available in the system for the purpose of map building.

A. Basic Map Building Method

The basic Map building functionality of the developed system involves the conversion of the echo time statistics sent by the Unit to the PC over the Serial Port in to distances and plotting them in a Radar like display. This basic functionality involves several aspects.

- Receiving the echo parameters from unit and converting them into corresponding distances.
- Storing them in an array of necessary size where each element corresponds to the most up-to-date distance for that specific angle.
- Drawing an arc with a radius representing the obstacle distance for each of the up-to-date readings every 2s so that the GUI effectively represents the distance values in the up-to-date array.

B. Probability Method

The more advanced functionality of our system is the calculation of a probability model [2] in accordance with the Sonar sensor in order to provide a more realistic as well as a combined (fused) representation of the received distance parameters. The probability model that has been implemented is based on the sonar sensor model derived by other researchers in an effort to allocate probabilities based on a sonar reading. This model is explained below in order to better convey the implementation of our system.

1) Sonar Sensor Model

In this model the Sensors' range area is divided into three regions based on the delay (i.e. the alleged distance of the obstacle) [2]. The Whole area is divided into a grid (whose number of elements will decide the resolution of the Mapping model) and each of the squares is assigned a probability based on its position with relative to the sensor reading.

The probability of occupation will be calculated differently based on which region it occupies. The probability value that this model gives is P(s/H) or the probability that the sonar would return a value given the grid element was really occupied. The three regions will have different equations based on which the probability of occupation is calculated as discussed in section B(ii).

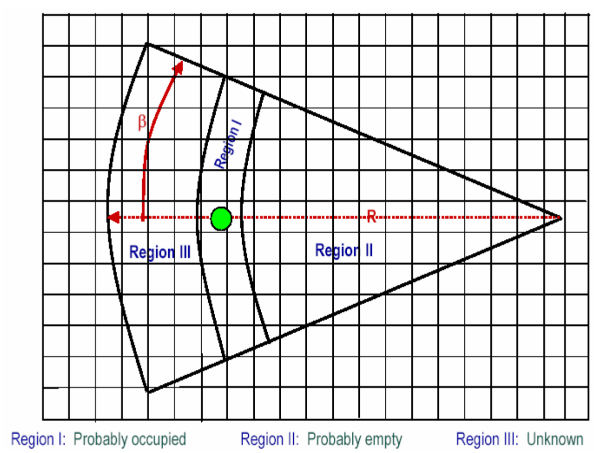

Figure 4. Sensor Model regions

2) Probability Algorithm

The grid which covers the entire possible sweep area is filled as the reading for each angle comes from the Micro- controller. The overlapping regions can be combined using any of number methods such as Bayesian, Dempster-Shafer and HIMM (Histogramic In Motion Mapping) [2]. Of these the Bayesian method has been employed to combine the different readings in our system. The implementation has been achieved as follows.

The system will create several arrays for the following purposes and populate them with initial values.

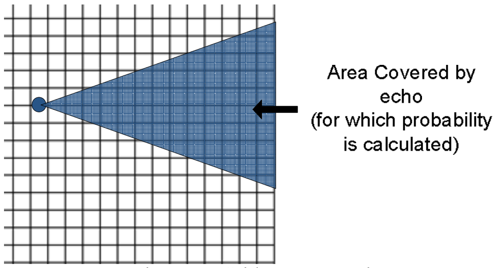

Figure 5. Grid Representation

- Distance Array: 1-D array keeping the distances calculated for the latest sweep. An element is updated every time a reading is taken and used to decide the regions to which each relevant grid element falls, i.e. which equation is used to calculate the probability of occupation.

$$Distance = \{d_k\} \quad k = 1 \ldots m$$
$$where\ m = \text{No of scan steps}$$

- Grid-Angles: Contains the angles for each grid element with respect to a reference direction and is used in deciding whether a grid element is within the sonar beam width for a particular echo. This is filled at the start and will remain constant throughout (while only being referenced).

$$GridAngles = \{a_{ij}\} \quad i = 1..n,\ j = 1..n$$

- Grid-Length Array: This will contain the distance of each grid element from the center grid element. This is also calculated at the initial time and not updated thereafter.

$$GridLength = \{r_{ij}\} \quad i = 1..n,\ j = 1..n$$

- Angle-Inside Array: This keeps track of which element falls inside the current reading beam width ($\pm 30^0$). This is overwritten for each sonar reading and will contain a '1' (within the beam-width) or '0' (outside the beam-width).

$$AngleInside = \{u_{ij}\} \quad i = 1..n,\ j = 1..n$$

- Grid-Probability Array: This keeps the calculated probability of occupation (the fused value) and is updated every time a sonar echo result is received. This is initialized with all values at a probability of 0.5.

The overall functionality of the algorithm which calculates the Probability of Occupancy of the covered area is shown below. The calculations are carried out as follows.

- Where the Delay for the k^{th} reading is t_k
- Speed of sound V
- R is the Range of the Sonar sensor
- β is the Beam width of the Sonar Sensor
- θ_k is the angle (with respect to reference) at which the sonar was facing at the time of the reading
- MaxOccupied is the Maximum likelihood of a grid element of being occupied.

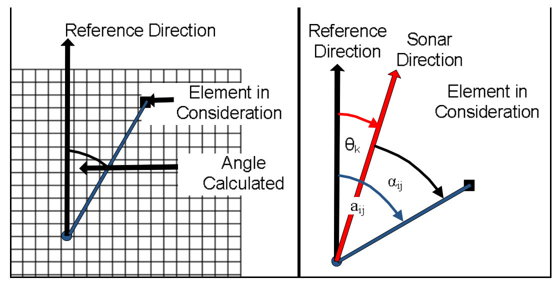

Figure 6. Reference Directions and Angles

$$d_k = {V \times t_k}/2 \qquad \alpha_{ij} = (a_{ij} - \theta_k)$$

Where % error is p,

For the grid elements that have $u_{ij} = 1$

Equations for the different grid Elements are [2]

$\underline{d_k \times (1-p) < r_{ij} < d_k \times (1+p)}$

$$P_{ij}\left(s/H\right) = \frac{\left(\frac{R - r_{ij}}{R}\right) + \left(\frac{\beta - \alpha_{ij}}{\beta}\right)}{2} \times Max_{Occupied} \qquad (1)$$

$\underline{r_{ij} < d_k \times (1-p)}$

$$P_{ij}\left(s/H\right) = 1 - \frac{\left(\frac{R - r_{ij}}{R}\right) + \left(\frac{\beta - \alpha_{ij}}{\beta}\right)}{2} \qquad (2)$$

$\underline{d_k \times (1+p) < r_{ij}}$

$$P_{ij}(Occupied) = Unknown \qquad (3)$$

As explained earlier, since this probability is P(s/H) the next step is to convert it into P(H/s) which is the probability of the grid element being occupied given that there is a sensor reading at d_k distance.

$$P_{ij}\left(H/s\right) = \frac{P_{ij}\left(s/H\right)P(H)}{P_{ij}\left(s/H\right)P(H) + P_{ij}\left(s/H\right)P(\overline{H})}$$

$$=\frac{P_{ij}\left(s/H\right)\times 0.5}{0.5\left[P_{ij}\left(s/H\right)+\left[1-P_{ij}\left(s/H\right)\right]\right]}$$

$$P_{ij}\left(H/s\right)=\underline{\underline{P_{ij}\left(s/H\right)}} \quad (4)$$

Where

$P(H)$ = Unconditional probabilty of being Occupied = 0.5

$P(\overline{H})$ = Unconditional probabilty of being Unoccupied = 0.5

$$P_{ij}\left(s/\overline{H}\right)=1-P_{ij}\left(s/H\right) \quad (5)$$

After P(H/s) is calculated for each relevant grid element, the Prior probability (the element P_{ij}) is combined using the Bayesian method. This method is used to fuse the other sonar observations with the new probabilities. At the first update all the grid elements are initialized with the a priori probability of being occupied (i.e. 0.5).

$$P_{ij}\left(H/s_1,s_2\ldots s_n\right)=\frac{P_{ij}\left(s_1,s_2\ldots s_n/H\right)P(H)}{P_{ij}\left(s_1,s_2\ldots s_n/H\right)P(H)+P_{ij}\left(s_1,s_2\ldots s_n/\overline{H}\right)P(\overline{H})}$$

$$P_{ij}\left(H/s_1..s_n\right)=\underline{\underline{\frac{P_{ij}\left(s_n/H\right)P_{ij}\left(H/s_{n-1}\right)}{P_{ij}\left(s_n/H\right)P_{ij}\left(H/s_{n-1}\right)+P_{ij}\left(s_n/\overline{H}\right)P_{ij}\left(\overline{H}/s_{n-1}\right)}}} \quad (6)$$

The simplified equation means that the fused probability could be calculated using the previous probability value maintained in the array and the newly calculated value. This is what has been implemented in the system. This method has been implemented with the hope of achieving data fusion from the overlapping readings, thereby achieving improved resolution [6].

IV. RESULTS AND DISCUSSION

The system was used in several contexts to gauge the performance of both the basic mode as well as the probability model. The testing was carried out in an indoor environment. In all the situations we were able to obtain satisfactory results, especially with the probability model. We were able to successfully demonstrate that the probability model was capable of improving the system‘s resolution as well as accuracy. The major outcomes are outlined below.

A. *Major Outcomes*

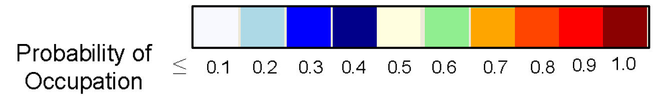

Figure 7. Probability Color code

1) *Full Environmental Scan: Small Square Environment*

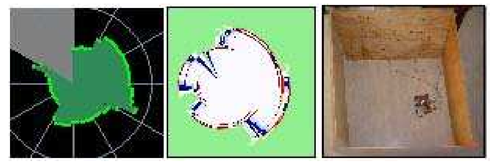

Figure 8. i.) Normal Mode ii.) Probability Model iii.) Actual Setup

Both modes produced a slightly distorted image (justifiable given the very large beam width of 60^0) of the environment. Though the normal mode shows the corners in a clearer manner, the Probability model shows a more complete representation of the area.

2) *Local Environmental Scan: Edge Environment*

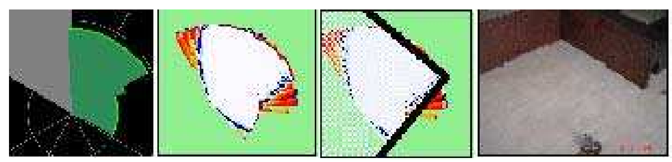

Figure 9. i.) Normal Mode ii.) Probability Model iii.) Edge Emphasized iv.) Actual Setup

When the corner was further away from the platform the shape tended to be better defined with the Probability model showing a better definition of the edge.

3) *Local Environmental Scan: Small Obstacle*

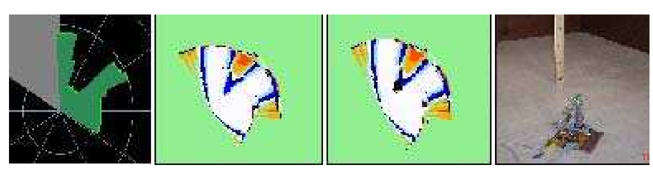

Figure 10. i.) Normal Mode ii.) Probability Model iii.) Object Emphasized iv.) Actual Setup

The Probability model's performance was far superior to that of the normal mode where the very small sized object was shown as a very large obstacle due mainly to the large beam width. Due to the sensor fusion this has been almost entirely removed in the probability model. Thus it can be seen that the probability model greatly improves the resolution of the system.

4) Local Environmental Scan: Two Obstacles at close proximity

Figure 11. i.) Normal Mode ii.) Probability Model iii.) Objects Emphasized iv.) Actual Setup

When two small objects were placed at close proximity the normal mode displayed two very large objects, whereas the probability model was able to accurately determine the small size of the objects as well as show the gap between them in an accurate manner.

Based on the above results the impact of sensor fusion is evident, especially in the B, C and D situations where the object size is most closely represented by the probability model. These successful results highlight the effectiveness of this method.

B. Comparison of the Basic and Probability method

Based on the results obtained for the two methods (normal and probability) and by analyzing the underlying concepts and available literature it is possible to carry out a comparison of the two methods.

- The basic method makes no consideration of the fact that the displayed arc is far smaller than the beam width of the sonar sensor. This means that even though the display recognizes and draws an object at a certain angle this object may in fact reside in a different arc. Additionally the size of the detected object will be greatly exaggerated in this method.

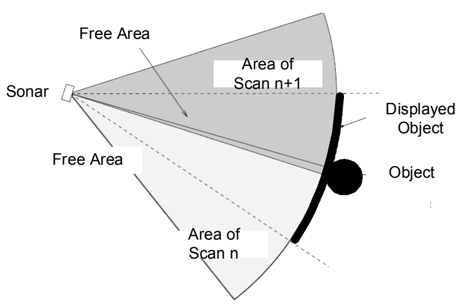

Figure 12. Actual and detected object sizes

- Though the step size of the stepper motor can be reduced in an apparent effort to improve the resolution of the reading, it is not possible to effectively improve the resolution of the system using the basic method. The only method of improving system resolution is to break the area into a grid with a larger number of elements in the Probability method, thus improving system resolution at the cost of added processing [6].
- Only the probability method is capable of fusing the different overlapping readings, thus making full use of all the observations which is highlighted by the superior resolution especially in C and D.
- The basic method will have an unchanging likelihood of an obstacle throughout the entire arc whereas the actual likelihood of occupancy is a probability which varies with the position [6].

V. CONCLUSION

We have been able to successfully develop a robust sonar based map building system for use in a mobile platform for the purpose of navigation and localization. This has been accomplished with the use of a single rotated sonar sensor with the use of a stepper motor and system performance and resolution has been significantly boosted with the implementation of sensor fusion with the use of Bayesian combination and the sensor model. This has been tested successfully in an internal environment and its resolution improvements and other advantages have been observed. Based on the obtained results we have been able to conclude that the probability method is capable of providing a superior performance in most circumstances.

The system designed has several attractive features which make it ideally suitable in a number of situations. This system is capable of scanning a near 360^0 area using a single sonar sensor making it far cheaper and less complex than most current systems. Thus this will also be free of most issues arising due to the use of multiple sonar sensors. In addition with the use of the probability model the system is capable of fusing together the readings from all the angles, thus representing a much more complete representation of the scanned environment. Thus the system is capable of achieving a significant improvement in the resolution of the system over the conventional method.

This system has been designed with the view of being used as a module in a mobile platform and would require additional modifications when being used in such a capacity. If the mobile platform is to be used for the purpose of building environmental maps, the probability model can still be used with the robot motion also being accounted for.

REFERENCES

[1] Robin R. Murphy , “Introduction to AI Robotics,” Prentice-Hall of India, New Delhi 2004, pp. 210–216.

[2] Robin R. Murphy , “Introduction to AI Robotics,” Prentice-Hall of India, New Delhi 2004, ch 11.

[3] Hyoung Jo Jeon and Byung Kook Kim, “Feature-based probabilistic map building using time and amplitude information of sonar in indoor environments”, Department of Electrical Engineering, Korea Advanced Institute of Science and Technology

[4] Vassilis Varveropoulos, “Robot Localization and Map Construction Using Sonar Data”

[5] Jinsuck Kim Roger, A. Pearce Nancy and M. Amato, “Feature-Based Localization using Scannable Visibility Sectors”, Department of Computer Science Texas A&M University.

[6] Craig L. Robinson, Daniel Block, Sean Brennan, Francesco Bullo, and Jorge Cortes, “Nonsmooth analysis and sonar-based implementation of distributed coordination algorithms”, General Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign