**MOBILE ROBOTS**

Naveen M.George┼

┼ Department of Mechatronics Engineering

SRM University

Chennai, India

e-mail: naveen\_mg@srmuniv.edu.in

*Abstract***- The aim of the project is to build a collection of wheeled robots that are capable of communicating to each other and mapping a maze. The advantage of the system is its capability to map the given area in the minimum amount of time by splitting the work between two or more bots. The design and implementation of multiple mapping robots is undertaken using Digital Magnetic Compass, Ultrasonic Sensor and Arduino. The designed robots uses a metric, world centric approach for mapping algorithm. Robots follow the wall while continuously sending its co-ordinates to the base station. Base station or map monitor has PC with NRF module link connected with mobile robots and map is plotted on a GUI. The proposed approach is a simple and low cost robotic application to solve SLAM problem.**

Keywords- SLAM, Ultrasonic sensor, Arduino, MATLAB.

# Introduction

We all know programming just one mobile robot with artificial intelligence is hard. So adding more robots and having them exhibit a collective behavior can increase the difficultly level exponentially. This is what makes such an intelligence a hot topic in the world of robotics today. The project can be dealt with under various stages each more difficult than the previous.

Robust feature detection is one of the crucial problems of simultaneous localization and map building (SLAM) of mobile robot. Because the detected feature is used as a component to construct a feature-based map and also as a landmark to localize a mobile robot. If failing to detect features or obtaining inaccurate information of feature position, the robot should excessively depend on the odometry data to estimate its pose. However, unboundness property of the odometry data makes the estimation divergent. Therefore, to perform the SLAM successfully, feature detection should give accurate feature information as much as possible.

Role of sensing system is to detect the presence of objects and measure their positions. The objects can be neighbouring robots, obstacles and target. For interaction of the multiple robots, communication between robots is important to carry out specific task where one robot delivers orders or updates to other robots [1], [2], [3]. Multiple robots can be sent into an unknown building to produce a floor map [4]. Research is going on in the area of different mapping techniques over the time. Representing the geometry of the environment should have high accuracy [5]. A metric approach of mapping is one that determines the geometric properties of the environment. This representation is very useful, but is sensitive to noise. On the other hand topological approach is one that determines the relationships of locations of interest in the environment. World-centric mapping represents the map relative to some fixed coordinate system, while robot-centric mapping represents the map relative to the robot. [2].

Simultaneous localization and mapping determines the location or pose of a robot and construct map of an unknown environment at the same time. Robot path and map are both unknown. There are various related works on localization, mapping, or both (SLAM). Most of this work focuses on exploring an information space of environment[6],[7],[8],[9]. An autonomous mobile robot must recognize its position and find the path for itself. Some of the SLAM mobile robot in the indoor environment uses digital magnetic compass and ultrasonic sensors. Multiple sensor technique makes use of ultrasonic sensor, infrared sensor, laser scanner, stereo camera, and electronic compass to solve SLAM problem[10], [11],[12], [13]. Robot need to plan a motion path through the environment to navigate without colliding with obstacles. Data obtained by ultrasonic range measurements is used to detect and avoid obstacles in environment [14], [15].

1. THE HARDWARE DESIGN

The hardware assembly and specifications of mapping robot is explained in this section. Fig 1 shows the robot and its different modules. Fig 2 shows basic architecture block diagram of the robot. The robot uses Arduino Mega board as central processor and other input and output devices along with communication module and power supply.



Fig. 1. Mapping Robot: Actual Assembly

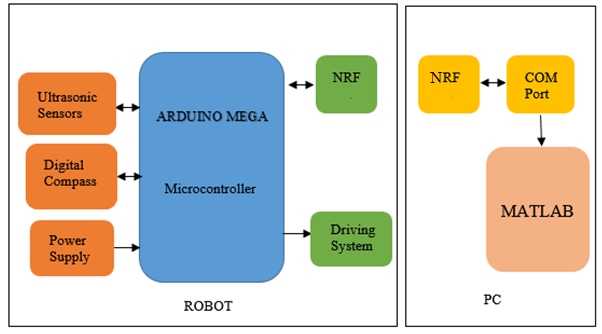
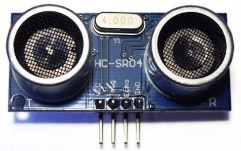


Fig. 2. Mapping Robot Block Diagram

The mapping robot has Arduino MEGA development board which consists of Atmel’s ATmega2560 microcontroller with other electronic components which can be programmed using the software. The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Robot has two kind of sensors used for navigation. Fig 3 (b) and (c) shows digital compass HMC5883L and ultrasonic distance sensor HC - SR04 respectively. HMC5883L is 3-Axis Digital Compass IC. The I2C serial bus allows for easy interface. It enables 1 to 2 Degree Compass Heading Accuracy. Working range of Ultrasonic ranging module HC - SR04 is 2cm to 400cm with accuracy of 3mm. Output voltage from sensor is corresponding to the detection distance from sensor to an object. Robot has two DC geared motors for motion control. Two caster wheels are attached to front and back end of robot for support. Driving system of robot allows it to move forward, backward and rotate clockwise or anticlockwise. Communication between robot and PC is achieved using NRF. NRF module is a module designed for transparent wireless serial connection setup. This robot has 7.4 Volts battery for powering of driving system and Arduino.

(a) (b)

(c) (d)

Fig. 3. Hardware contains of robot (a) Arduino Mega micro-controller (b) Digital compass - HMC5883L (c) Ultrasonic distance sensor - HC-SR04 (d) NRF module .

Base station or has PC with NRF link connected with mobile robot. PC has MATLAB software with Arduino driver to communicate with PC’s COM port. Live coordinates send by mobile robot receives by PC and map is plotted on MATLAB’s graph. Fig 4 shows mechanical layout assembly of designed robot.

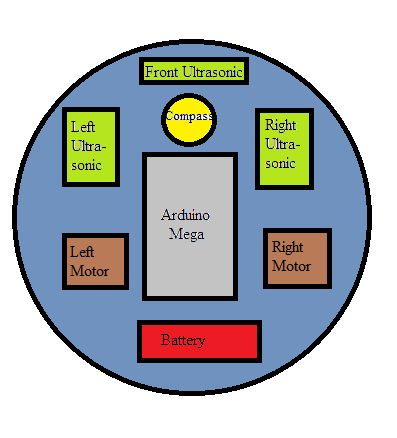


Fig. 4. Mapping Robot: Mechanical Assembly

# Stages

1. Obstacle Avoider

A basic obstacle avoider was created to understand and check the primary working and functions of the ultrasonic sensor, servo and motor driver. The robot was able to travel around without any collision with Arduino code alone.

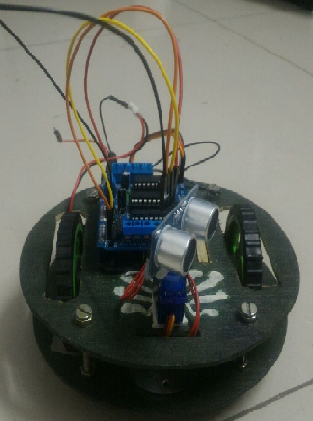


Fig 5.Obstacle avoider

1. Remote controlled robot

The functioning of the RF module was tested and range and usage determined by converting the obstacle avoider into a remote controlled robot directly through the PC.



Fig 6.Reciever communication with PC

1. Sonar

The capability of ultrasonic sensor to map an area is determined by making a sonar.



Fig 7.Sonar

1. Real-time plotting of a stationary robot

A real time map around a stationary robot is plotted with an ultrasonic sensor and servo motor.

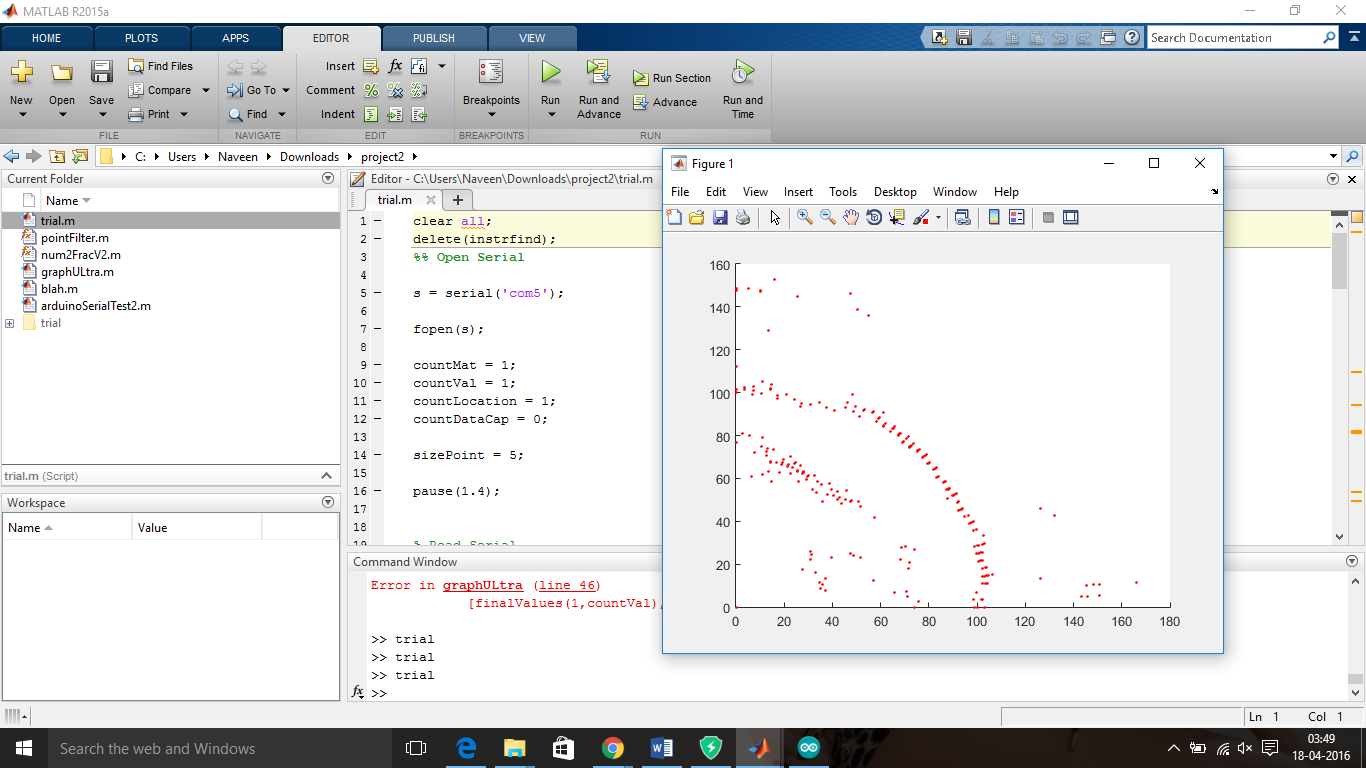


Fig 8. Real time map

1. Communication

The algorithm for each robot is different so that they follow different paths. The first robot runs on basic obstacle avoidance, second on left wall following and third on right wall following. Various other algorithms can also be used on each of the robots.



Fig.9 SLAM

Communication between the robot and the PC is set up and the robot is placed into the map to map the entire maze up to its capability. The time taken to plot the entire maze is recorded.

Later on it was decided to plot the compass values only during turns so that the compass readings do not interfere with NRF module and to get straight lines between each turns.

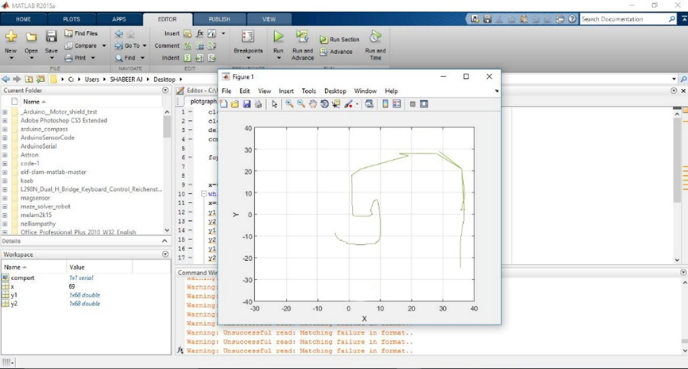


Fig.10 Map

1. Summing up of all the process

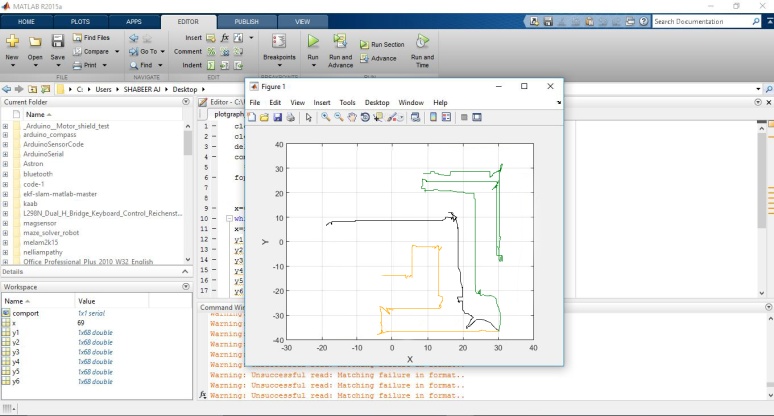


Fig.11 Final map

All the given concepts must be integrated into a single program to perform the required task. The SLAM alone itself is a high end program for the Arduino to handle hence most of the processing is undertaken within the computer while only the required type of signal is taken from the Arduino. For further accurate and advanced outputs a high end board such as the Raspberry pi is required along with better sensor inputs through LIDAR, Kinect or camera.

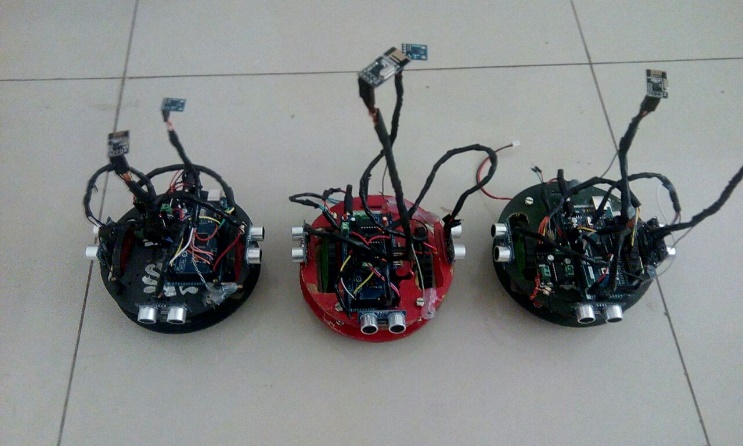


Fig.12 Final Robots

1. ALGORITHM

Robot navigation and mapping algorithm is implemented with a minimum level intelligence on maximum cost reduced robots. Motion path of robot is planed through environment to navigate without colliding with obstacles. We used concept of world centric approach for mapping along with a modified wall following algorithm for navigation.

When robot turns on first it read value of its heading degree using digital compass. Then it looks for front and side obstacle distance using ultrasonic sensors. Designed robot follows wall to navigate through environment. Robot moves forward when there is no obstacle or wall at front and a side wall is present. As robot turns it sends its co-ordinates to base station. Co-ordinates are updated using compass value.

If front wall is detected then robot 1 takes right turn, when left and right walls are not detected robot takes right turn, if left wall alone is not detected it takes a left turn and if walls are present on all three sides, the robot turns back. If front wall is detected then robot 2 takes left turn, when left and right walls are not detected robot takes left turn, if right wall alone is not detected it takes a right turn and if walls are present on all three sides, the robot turns back. Robot 3 follows a random obstacle avoidance path.

While in turning process robot uses speed control to make turn. After robot completes its 90 degree turn it stops and again read for both ultrasonic values. The compass value is updated to the computer just before making the turns. When robot turns on it assumes itself to pointing towards north with heading degree of 0. At each corner it takes turn of 90 degree. At each forward movement robot points towards one of the 4 directions. User need to select appropriate COM port of the PC and baud rate. Baud rate of 9600 is set at robot transmitter end and at base station receive end. Data received by NRF receiver appears at PC’s COM port. Then using MATLAB the compass values are plotted onto a graph. Robot sends co-ordinates in form of x, y, x, y… and is thus continuously updated. X-Y graph is plotted on MATLAB front panel.

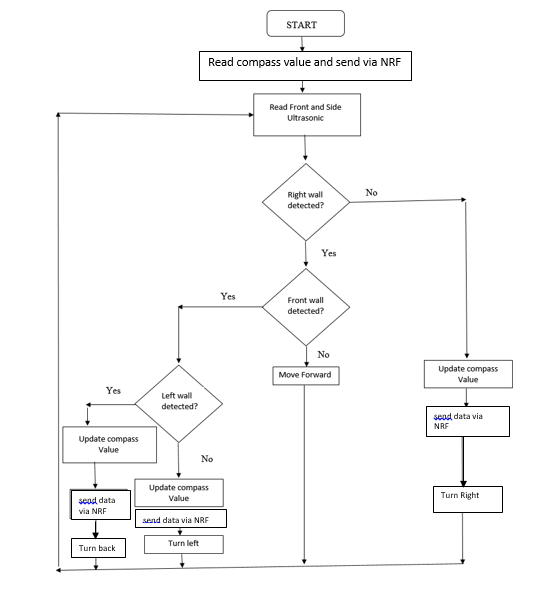


Fig.13 General Flowchart

Algorithm for NRF24l01 communication (TRANMITTER):

1. Start;

2. Define all the necessary header files and structures and variables;

3. Begin the radio;

4. Set the number of nodes or addresses required for radio communication for both transmitter and receiver;

5. Set the radio data speed and frequency that do not interfere with any other frequency around;

6. Set the MISO (Master In Slave Out) PIN of nrf24l01 as LOW initially;

7. Open the radio for writing or sending the data;

8. Obtain the data to be send to the receiver;

9. Set the MISO PIN as HIGH;

10. Send the data to the receiver using defined receiver address;

11. Set the MISO PIN back to LOW;

12. Stop

Algorithm for NRF24l01 communication (RECEIVER)-

1. Start;

2. Define all the necessary header files and structures and variables;

3. Begin the radio;

4. Set the number of nodes or addresses required for radio communication for both transmitter and receiver;

5. Set the radio data speed and frequency that do not interfere with any other frequency around;

6. Set the MISO (Master In Slave Out) PIN of nrf24l01 as LOW initially;

7. Set the radio to stop transmission mode and set as receiver mode;

8. Open the radio for reading the transmitted data;

9. Obtain the data send to the receiver using the unique addresses defined for different transmitters by delaying for each transmitter;

10. Print the obtained data to MATLAB;

11. Stop

1. CONCLUSION AND FUTURE WORK

We presented a distributed approach to mobile robot mapping and exploration. The system enables teams of robots to efﬁciently explore predetermined environment. The robots initially explore on their own, until they can communicate with other robots. They exchange sensor information with other robots, they estimate their relative locations using a magnetometer readings. During exploration, the robots update their predictive models based on observations in the environment. Our experiments indicate that this approach supports map merging decisions signiﬁcantly better than alternative techniques using single robot mapping. In order to overcome the risk of false-positive map matches, the robots actively verify location by communicating each other. If the robots meet at the meeting point, they know their relative locations and can combine their data into a shared map. Mapping and map merging uses a SLAM technique that models uncertainty by local probabilistic constraints. Shared maps are used to coordinate the robots and to estimate the location of other robots. Our mapping and exploration system was evaluated under some of the toughest real-world conditions yet imposed on a robotics project. Prior to the evaluation, the developer team was allowed to test their robots only in one half of the environment. During the evaluation runs, the robots had to rely on their own wireless network to exchange information. The robots successfully explored the environment in all four ofﬁcial evaluation runs. All maps generated during these runs were virtually identical, indicating the high accuracy and robustness of our system.

With the current hardware used not much additions can be made to the project but if the hardware components are changed there is large range of possibilities. With higher processing power better sensors and camera can be added to execute complete SLAM.



Fig.12 Multi-Robot Mapping

A quadcopter can be made to work as the base station instead of the PC. The base station can control and coordinate the actions of the wheeled robots. Once these robots are capable of such complete high level tasks they can be used for space explorations or on rescue missions.

A microprocessor such as raspberry pie is necessary to perform such complex tasks as computational power and processing speed is necessary for all the process. Range sensors such as LIDAR and cameras can be used for better and easier integration of SLAM.

Once it is capable of all the above tasks it can be made further into a SWARM system in which the robots has a very high level of intelligence and the possibilities are vast.

# REFERENCES

1. Dhiraj Arun Patil, Manish Y. Upadhye, F. S. Kazi, N. M. Singh, “Multi

Robot Communication And Target Tracking System With Controller

Design And Implementation Of SWARM Robot Using Arduino,” IEEE

International Conference on Industrial Instrumentation and Control (ICIC), Pune, India. May 28-30, 2015.

1. Emaad Mohamed H. Zahugi, Ahmed M. Shabani and Dr. T. V. Prasad,

“Libot: design of a low cost mobile robot for outdoor swarm robotics,”

IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, May 27-31, 2012.

1. Saxena Ankita, Satsangi C.S., Saxena Abhinav, “Collective collaboration for optimal path formation and goal hunting through swarm robot,” IEEE 5th International Conference on Confluence The Next Generation Information Technology Summit, 2014 , Pp. 309-312.
2. Churavy C., Baker M., Mehta S., Pradhan I., Scheidegger N., Shanfelt

S., Rarick R., Simon D., “Effective implementation of a mapping swarm of robots,” Potentials, IEEE, vol.27, no.4, pp.28,33, July-Aug. 2008.

1. S. Thrun, “Robotic mapping: A survey,” Exploring Artificial

Intelligence in the New Millenium, San Mateo, CA: Morgan Kaufmann, 2002.

1. F. Chenavier and J. Crowley, “Position estimation for a mobile robot using vision and odometry,” Proc. IEEE Int. Conf. Rob.Autom, Nice, France, 1992, pp. 2588-2593.
2. J. Castellanos, J. Montiel, J. Neira, and J. Tardos, “The SPmap: A probabilistic framework for simultaneous localization and mapping,”

IEEE Trans. Robot. Autom., vol. 15, no. 5, pp. 948-953, Oct. 1999.

1. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M.

Csorba, “A solution to the simultaneous localisation and map building (SLAM) problem,” IEEE Trans. Robot. Autom., vol. 17, no. 3, pp. 229-241, Jun. 2001.

1. B. Yamauchi, A. Schultz, and W. Adams, “Mobile robot exploration and map-building with continuous localization,” in Proc. IEEE Int.

Conf. Robot. Autom., 2002, pp. 3715-3720.

1. Mark Pupilli, Andrew Calway, “Real-Time Visual SLAM with Resilience to Erratic Motion,” Proc. Of the 2006 IEEE Computer

Society Conference on Computer Vision and Pattern Recognition, June 2006, Vol 1, pp.1244-1249.

1. Ho-duck Kim, Dae-Wook Kim, Kwee-Bo Sim, “Simultaneous Localization and Map building using Vision Camera and Electrical Compass,” SICE-ICASE International Joint Conference 2006, Korea, October, 2006.
2. Dirk Hahnel, Wolfram Burgard, Dieter Fox, and Sebastian Thrun, “An efficient fast slam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,” In proc. of The IEEE/RSJ Iint. Conf. On Intelligent Robots And Systems (IROS), 2003, pages 206-211.
3. Slawomir Grzonka, Christian Plagemann, Giorgio Grisetti, and Wolfram Burgard, “Look-ahead proposals for robust grid-based slam with raoblackwellized particle filters,” Int. J. Rob. Res, 28(2):191-200, 2009.
4. Silva, A. Menezes, P. Dias, J., “Avoiding obstacles using a connectionist network,” Intelligent Robots and Systems, 1997. IROS ’97., Proceedings of the 1997 IEEE/RSJ International Conference on, vol.3, no., pp.1236,1242 vol.3, 7-11 Sep 1997.