

Arena Mapping Rover

Nazish Qamar*, Sivani Boyapati and B. Jaganatha Pandian

School of Electrical Engineering, Vellore Institute of Technology University,
Vellore-632014, Tamil Nadu, India; nazishqamar01@gmail.com,
sivani.boyapati@gmail.com, jaganathapandian@vit.ac.in

Abstract

Objectives: The study aims at designing an Arena Mapping Rover that has adequate functions to optimally map and analyze the shape of an enclosed space, which cannot be satisfactorily done by the human being or which may be too hazardous for the human being to explore. **Methods:** The rover is capable of achieving this by moving around the different parts inside the arena. It continuously retrieves data related to the surrounding of enclosure and simultaneously sends the data to the computer. **Findings:** The computer processes the acquired data and presents a two dimensional reconstruction of the arena environment. **Applications:** The autonomous rover consists of a firm metal structure, which makes use of two stepper motor for precise movement inside the arena. Three infrared proximity sensors, facing three different directions detect the boundary and obstacles in vicinity.

Keywords: Arduino, Dead Reckoning, Mapping, Matlab, Mobile Robot, Wireless Control

1. Introduction

Robotic industry is an ever booming industry. In industrial automation, home automation, mining exploration, construction, military combats, nuclear reactors, deep sea and ocean bed study, space missions to explore the surface of celestial bodies, robot plays a vital role. There have been many attempts in the past for localization as well as mapping the unknown environment simultaneously. Some approaches used probabilistic method for position estimation¹⁻⁴. Ultrasonic sensor, infrared sensor, laser scanner, stereo camera, electronic compass, etc, have been used in combination to solve this problem, but this leads to complex algorithms and consume more time⁵⁻⁸. Some approaches suggested the use of multiple robots in an unknown building to produce a floor map⁹⁻¹⁰. However, in narrow arena it is not sometimes feasible to deploy multiple robots. The aim of the proposed autonomous 2D mapping robot is to construct a floor plan of the surrounding lab environment on a wireless master PC by using the process of dead reckoning. This

allows predicting the nature of the location without putting any human life in danger. By designing and building the autonomous mapping robot, a huge step is taken towards the design of a fully functional exploration robot. The system has been designed and tested for the 2D mapping the autonomous mapping robot has been prototyped using Arduino Uno microcontroller board based on ATmega328. Rover consists of thin metal sheet platform used for attaching hardware to the robot, such as stepper motor, stepper motor driver module, bluetooth module, the power circuitry and wheels. Each of these components was interfaced through microcontroller using various drivers, modules and programs. While the robot is detecting & sending the location of the obstacle, the data is being sent continuously to Matlab software via Bluetooth Communication Protocol IEEE 802.15.1, for manipulation from the master PC. Figure 1 shows important functional blocks used in our robot.

The study also explains the 2D visualization software necessary for the display of data for mapping. In the end, the navigation algorithm used to control the movements

*Author for correspondence

of robot is explained. The idea of this rover develops from the need for the solution of two very similar problems in real time.

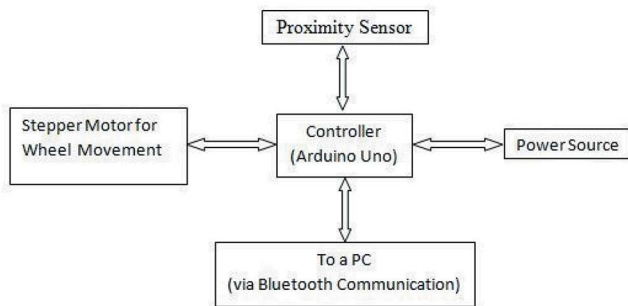


Figure 1. Main functional blocks.

The first one is the inefficiency of human being in carrying out exploration in unknown, unpredictable and potentially hazardous environment, which provide challenge for the navigation, as in case of environment having harmful radioactivity, terrorist activity or environment with blurred vision due to fog, mist, smoke, rain, hailstorm, etc. This problem becomes larger when added up with the incapability of human beings in entering into certain very minute and narrow environments and topographical structures or practically inaccessible surfaces like ocean bed, celestial bodies in space and so on. The second phase of the same problem is faced while using mobile robots in indoor environment such as hospitals, industries, corporate offices, etc., which involves complexity in carrying objects from one place to another within an enclosed environment. The proposed project idea of arena mapping rover using an IR sensor arrangement is both compact and extremely low in cost.

1.1 Mapping the Plot

The self-navigating mobile robot plots a 2D Map of an arena by using the process of dead reckoning. The arena may also have irregular boundaries with some obstacles inside it. The aim of rover is to avoid the collision with these objects, choose the optimal path for covering the whole arena and plot the obstacles simultaneously. For this purpose, the whole arena is assumed to be divided into parallel horizontal and vertical grids. The robot follows the predefined horizontal grids starting from one boundary to the opposite boundary. The robot takes equal steps one by one and during each step, it detects the presence of the obstacle. If an obstacle comes in its

way, the robot tries to avoid collision by moving along the boundary of the obstacle and storing the deviation from the predefined path. The robot moves with the help of two stepper motors, which are connected to its wheels and proximity IR sensors detect the obstacles coming up in its path. The stepper motors are excited by giving electric pulses and hence moved according to the availability of the free space to turn the rover.

Once the obstacle is crossed the rover tries to come back to the predefined grid on which it was moving before the collision with the help of deviation data, which was stored while moving down. Stepper motors of desired specifications are used along with 12V lead acid batteries or stable voltage power supply to provide sufficient power to the motors. The interfacing of PC software with the hardware of the project model is done using Arduino Uno. The rover communicates to the master PC via Bluetooth HC-05 wireless technology. The plot of arena is obtained by interfacing the Arduino with the matlab software. The project has been divided into four major areas in order to achieve the targeted functionality.

1.2 Signal Acquisition and Testing

Initially, the testing of the IR sensors was required for sending signals to the master PC. This was done in two phases. In the first phase, when the rover was physically connected to the master PC with Arduino board via USB cable, the obstacles were brought near to all three sensors individually. The output voltage V_{out} given by the sensors changed when the obstacle was brought near. The changing values were fed into the analog input pins A0 for front sensor, A1 for right sensor and A2 for the left sensor of the Arduino board. If the change in value crossed a certain threshold value, it indicated the presence of obstacle in its path. This test was successfully performed.

The next phase of this test was to send the sensor output wirelessly, when there was no physical connection between the Arduino board and the master the PC. For this purpose, the bluetooth HC-05 module was used. The test was successfully conducted and the Arduino board was able to send the obstacle data to the master PC successfully. The data was sent to the matlab, where the sensor values were continuously analysed.

1.3 Construction of the Rover

The second stage was the construction of the rover. The rover is made up of hard mechanical structure over which

the components including the drivers, sensor modules, battery and wires were mounted.

A thin and strong metal sheet was chosen for this purpose. The thin platform over which the components were mounted was of dimension 6 inch by 9 inch. A rigid metal support was also given between both motors to reduce the vibration due to the stepping action of the stepper motor. The motors were perpendicularly attached to the metal sheet on the both sides, such that the motor bodies come inside the metal case and shafts of the motors point outwards on the both sides. The wheels were fixed to the shaft of the motors tightly using thin Teflon tape to avoid the slip of the wheels over the shaft. The motion of the rover was produced by exciting stepper motors. IR sensors were used for the detection of the boundary and obstacles. These all in turn were run with the help of 12V battery supply. Their working voltage was 12-24V. The accurate position and the fixed 1.8 degree step angle was an extraordinary feature of stepper motor, which was utilised for the calculations. The step angle accuracy was $\pm 5\%$. An accuracy of 3-5% of a step ensured precise positioning and repeatability of movement and the error produced in every possible combination was non-cumulative from one step to the next. The wheels were 7.5cm in diameter. The choice of stepper motor for the movement was central to the whole design and algorithm. Small size and lightweight was one demanding feature of the rover. The NEMA 17 standard, 47 mm long stepper motor was used. This allowed two stepper motors to be fixed opposite to each other just below a 6-inch wide rover platform. The weight of each motor was just 365 gram, hence, reducing the weight of whole rover. In fact the stepper motors were the heaviest component on the rover platform.

1.4 Establishing a Communication Network

This stage involved the setting up of a wireless communication network between the rover and the master PC to transmit control signal as well as receive back mapping data to the PC. Bluetooth communication protocol system was used for the bidirectional communication. HC-05 bluetooth module was used for its capability for small range environment. The results were satisfactory. The various advantages bluetooth communication over other methods of wireless communication are as follows:

- It does not depend on Line-of-Sight unlike Radio Frequency Based Communication Systems.
- It has a Range up to 30 Meters.
- It comes Pre-Installed on All the Latest Phones and Laptops, which in Our Case is used in Master PC. In this way it does not require a Separate Receiver Module to Access the Data.

1.5 Plotting and Mapping

The final stage involved interfacing of simulation software matlab with the bluetooth module HC-05, which is connected to the Arduino UNO. This stage also involved coordinate correction by taking into account the necessary rotation and linear motion with respect to a reference coordinate system.

2. Algorithms

For understanding the algorithm, following points are to be kept in the mind. The whole arena is divided into horizontal and vertical grids. Each intersection point of the vertical and horizontal grids has a unique (x, y) coordinate value.

- The rover can move in positive (x, y) coordinate, i.e. first quadrant only.
- The starting point of rover is always taken as coordinate (0, 0) and the other intersections points are given values with respect to that.
- Each coordinate is of length 10cm. This means two points can be minimum 10cm away from each other. Hence this is the resolution for plotting.
- Each turn of wheel on stepper motor produces a displacement of 2mm. Hence, this is the resolution for the movement of rover.

2.1 Rover Movement Algorithm

The prototype rover has been programmed to execute a set of commands in a sequential manner based on the following algorithm. The motive is to explore the whole arena for plotting by dead reckoning.

- When the power supply of the rover is switched on, the rover waits for 10sec for the user to place it parallel

to the boundary at the (0, 0) coordinate of the arena. It is assumed that the rover will start from corner only.

- The rover starts moving towards positive x direction after 10 seconds. The rover checks whether the right IR Sensor is detecting the lower boundary or not. This check confirms that the rover is going to move in correct direction.
- The rover keeps on updating the stored value of the present (x, y) coordinate as it moves along the x direction.
- The rover keeps on moving until any obstacle is detected by the front sensor. The rover stores this value as the length of the arena along x direction for future reference. This value will help in differentiating objects from the boundary.
- Now, rover sends this data to PC wirelessly and this data will be used for the plotting purpose in the end.
- If the object is detected before the midway between two coordinate, the value of just crossed (x, y) coordinate is sent to the PC for plotting. If the object (boundary in the case of first grid) is more than 5cm away from the just crossed coordinate, then the next (x, y) coordinate value will be sent for the plotting.
- After sending values to the PC, rover initiates the procedure for u-turn, which ultimately leads the rover to next horizontal grid 10cm above the present grid.
- For taking u-turn, rover first goes back, by reversing its motion of wheel on the same grid to the previously detected coordinate. After this, rover turns left by 90 degree. It moves straight towards upward (i.e., positive y axis in this case) up to 10 cm. Then again rover takes left turn, thus completing the u-turn.
- Now the rover keeps on moving on the second grid in negative x direction until the obstacle or again boundary is detected and in this case rover sends the value of (x, y) coordinate and completes the u-turn by turning in right direction twice in the same way as before. Thus, now facing towards positive x direction, but on the third grid.
- Rover keeps on repeating the same procedure until the whole arena is covered.
- A different action is taken when the obstacle is detected by the front sensor before reaching the boundary. This case is recognized by comparing the present x value with the maximum value of x stored at first grid.

- In this case, the rover recognizes that the detected obstacle is actually an object, which is inside the arena. Hence, instead of taking u-turn to come to next grid, rover turns towards negative y direction, along the boundary of the obstacle (i.e. downwards).
- The rover keeps on moving downwards until the left/right sensor keeps on detecting the boundary of the obstacle.
- When the left/right sensor stops detecting the boundary of the object, the rover again takes a turn of 90 degree toward the direction in which it was moving before meeting the obstacle.
- The rover keeps on moving in this direction until it crosses the lower boundary of the obstacle. Once the lower boundary of the obstacle is crossed, rover again turns towards the positive y direction (i.e. upwards) until it comes to the predefined grid on which it was moving before meeting the obstacle in its path. After reaching that grid the rover finally starts moving towards the same direction in which it was moving before meeting the obstacle.
- During this whole time the rover keeps on sending the data from the left/right sensors to the PC to plot the object.
- In this way figuring out the between actual arena boundary and obstacle boundary. Rover moves inside the arena till the termination condition is achieved.
- The termination will happen when the rover reaches either of sides of the arena in horizontal direction, which is indicated by the front sensor and at the same time rover detects object on either of the left or right side of the rover.

2.2 Error Correction Movement Algorithm

An extra pulse is given to the left wheel of the rover to a turn the rover a little right after each step while moving in forward direction. This is because after each step, error is found to be 2mm difference between the distance of the both right and left tires with respect to the reference, which is at the starting point of each 10cm step. This means that the left wheel is always 2mm behind the right wheel. Hence, an extra pulse which further moves left wheel 2mm is given after each step.

2.3 Mapping Algorithm in Matlab

The steps used by the rover for continuously sending the data of the obstacle to the PC when it is moving inside the arena are.

- The rover send the (x, y) coordinates of the obstacle to the matlab via bluetooth after each step.
- The obstacle data is stored in a 2D matrix in the matlab, with each element at 'x' column and 'y' row of matrix representing a (x, y) coordinate of the arena plot.
- Once the termination condition is achieved, the 2D map is plotted using this 2D Matrix. The algorithm is best described by the flow chart shown in the Figure 2.

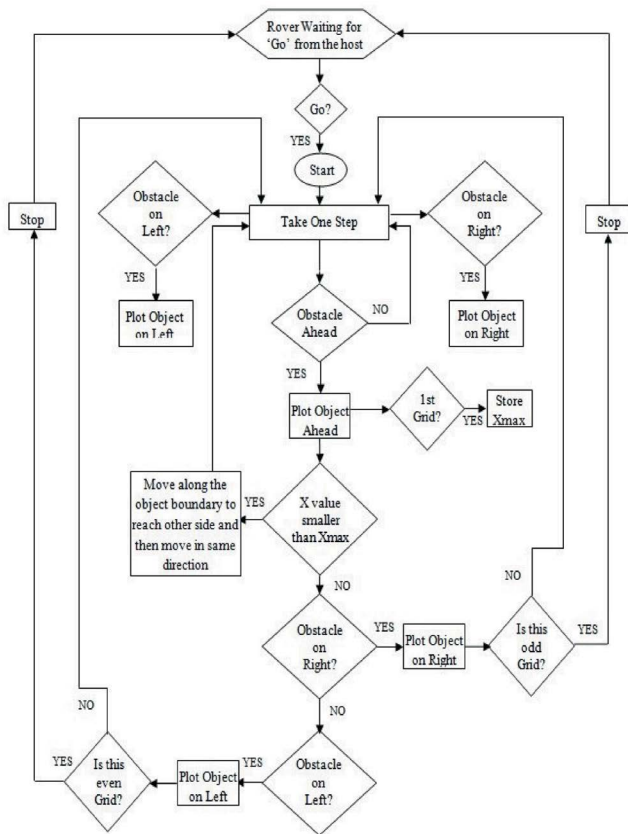


Figure 2. Algorithm flow chart.

3. Prototype Model

3.1 IR Proximity Sensors

There are three proximity IR sensors as shown in Figure 3, which detect the obstacles in all possible direction while exploring the arena. The data from these sensors is sent by the Arduino Uno to the PC. The sensors continuously

send small output voltage V_{out} to the analog input pins of the Arduino board. However, when an obstacle is detected in the proximity, which in our case has been calibrated approximately 25 mm for front sensor and 50 mm for the left and right sensors, a high V_{out} is sent to the analog input pins of the Arduino Uno. This sudden change in the values of the V_{out} is interpreted as the presence of some obstacle in the path of the rover or the presence of obstacle on the either sides of the rover.

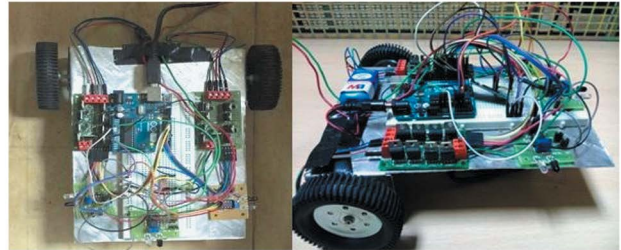


Figure 3. Top view and side view of the rover.

3.2 Linear and Turning Movement

The linear movement of the rover is achieved by a pair of 12V high torque stepper motors, which are controlled by dual H-bridge motor driver. The control signals for the movement of the motors are sent by the microcontroller in the form of combination of four electric pulses to each stepper motor. The stepper motors are given pulses in the full drive sequence. When both the stepper motors get the pulses in opposite combination, that is, the pulses sequence for one stepper motor is AB, BC, CD, AD (as shown in Figure 4) and the pulses sequence for other motor is AD, CD, BC, AB (as shown in Figure 5), then the robot is moving in forward direction. The need to give opposite set of pulses for both individual stepper motors lies in the fact that the motor shafts are in opposite direction to each other. Hence, rotation in opposite direction produces the cumulative effect of robot going in forward direction:

The turning movement is obtained in one direction when the stepper motors are given same set of pulses. It produces an effect of moving both wheels in opposite direction with respect to each other. Hence, collectively producing an effect of rover turning right or left inside the arena¹¹.

3.3 Controller

The microcontroller performs the most important functions. It integrates all the on-board equipment

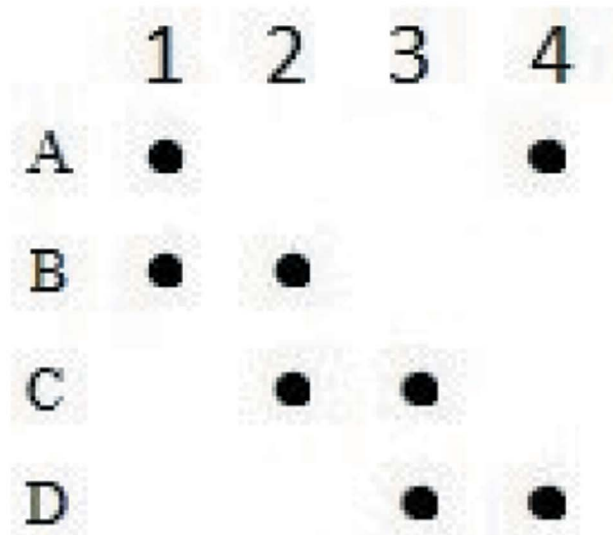


Figure 4. Excitation sequence for clockwise movement.

that the robot processes and manipulates. The robot is programmed to work autonomously without human intervention, hence taking decision according to the desired condition in the real time. The controller computes the distance from objects and changes the direction of the robot according to availability of free space. It processes and manipulates the data and sends it wirelessly to the host PC. The Arduino Uno board based on the ATmega328 has been utilized for its low cost and reliable computing capabilities.

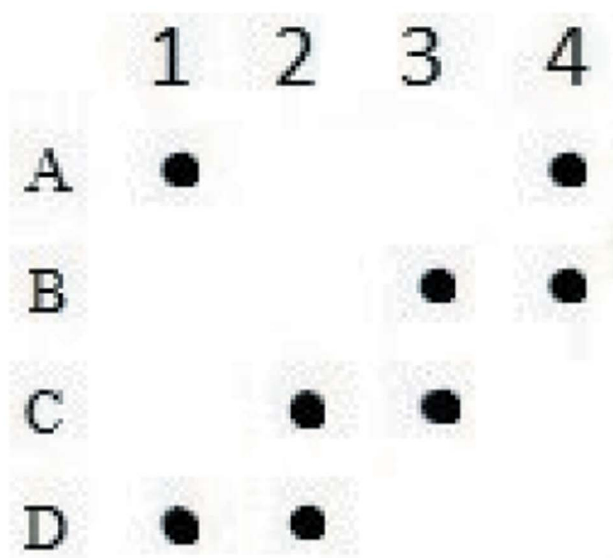


Figure 5. Excitation sequence for anti-clockwise movement.

3.4 Communication

The bluetooth HC-05 module that is mounted on the rover helps in establishing a wireless communication medium between the Arduino Uno based rover and host PC. The data from bluetooth is processed in matlab. It uses in-built function from bluetooth library to receive and send the data from host PC to the rover and vice versa. The model has been tested for range of 10m inside lab environment.

4. Result

For the project demonstration rover explored a square shaped arena of dimension 200cm x 200cm. The dimension was decided taking into account the technical details about the most of the possible movements of the rover and need to plot a real world practical scenario. The linear resolution for the movement of the wheel is 2mm and the rover senses the obstacle at each 10 cm step. So, the plotting resolution is 10 cm. This means if the object is just a few cm less than 10 cm away distance, even in this situation it is plotted at a distance of 10cm away. Hence, the rover is able to make fairly reasonable estimate of the dimensions of the test object and its distance from the sensors while traversing through all four edges. In this way, boundary of the arena was also covered. Different plots obtained in different test cases are as follows:

4.1 Arena without any Object Inside

The rover maps a square arena of dimension 200cm x 200cm. Hence, the whole arena has a length of 20 steps in both x and y directions. The Figure 6 shows the top view of the arena. The Figure 7 is the corresponding 2D plot obtained in the matlab. The rover starts from one corner, as shown in Figure 6. This edge is assumed as coordinate (0, 0). The rover follows forward coordinates in x direction till the obstacle i.e. boundary in this case, is encountered. The rover, hence, proceed towards the coordinates (0, 0), (1, 0), (2, 0) and so on, until the coordinate (20, 0) in the case of this particular arena. When the boundary is detected by the front sensor the rover takes u-turn and starts moving towards the coordinates (20, 1), (19, 1), and so on till (0, 1), where again the opposite boundary is detected. In this way rover again takes the u-turn and keep on doing same thing until the termination condition is obtained, which in this case is the non-availability of the free space inside the arena. These all values of the

obstacle coordinates, which are stored in the 2D matrix in matlab are plotted as given in Figure 7.



Figure 6. Arena without any obstacle inside.

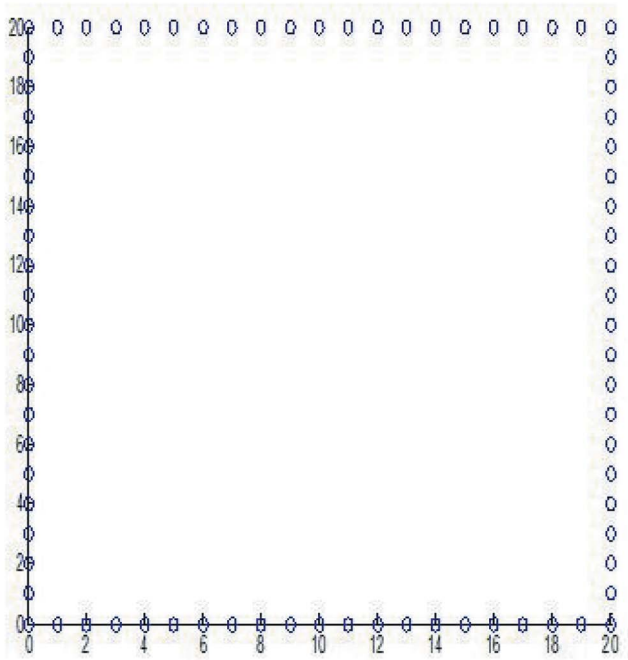


Figure 7. Plot of arena without obstacle inside.

4.2 An Arena with Rectangular Object Inside It

In order to test the capability of the rover to detect the objects inside it as well as the boundary, a rectangular

object of dimension 50cm x 40 cm is kept in the center of the arena. The rover moves in the same way except a different algorithm is followed when the obstacle, which is kept at the centre, is encountered inside the arena. Therefore, the object starts from one corner, which is taken as the origin that is (0, 0) and other coordinates of first quadrant in the positive 'x' and positive 'y' direction, comes after that. Hence, the rover first moves on the lowermost grid, that is (0, 0), (1, 0), (2, 0), up to the coordinate (20, 0). This time it stores the value before the first u-turn for comparison, as this value indicates the maximum length of arena in x direction.

Now the same procedure as described in previous case is followed, but when the obstacle is encountered in the centre of the arena, rover ensures that this is indeed the obstacle but not the end of arena in x direction by comparing the value stored while taking turn in first grid. Hence, the robot just turns toward negative y axis.

It moves along the whole boundary of the obstacle until it reaches the same horizontal grid on which it was moving before encountering the obstacle. In this way the rover crosses the obstacle and comes back to its predefined grid on which it was supposed to move. The same procedure continues till the termination condition is achieved. The top view of the arena in this case along with object is given in the Figure 8 and the 2D plot is given in Figure 9.



Figure 8. Arena with rectangular object inside.

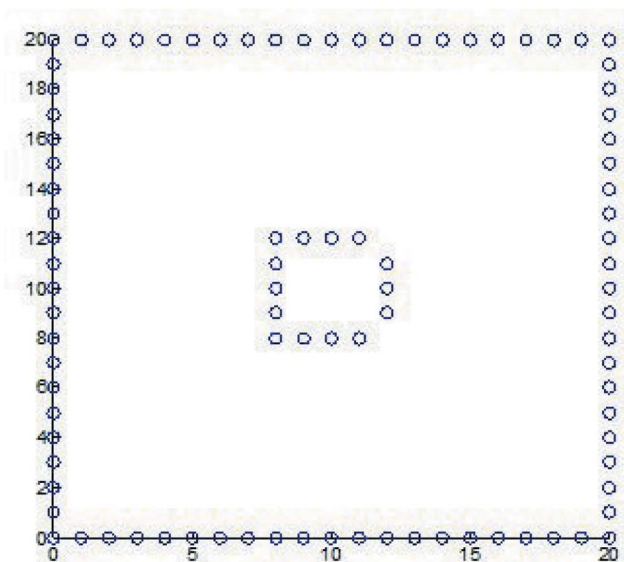


Figure 9. Plot of arena with rectangular object inside.

5. Conclusions

In this study, we have successfully made an autonomous arena mapping rover, which is capable of exploring an arena without human intervention, in order to achieve a 2D map of the boundary as well as the obstacles inside the arena. We have presented the scenarios where the rover was able to successfully map an enclosure when it is empty or when it has a rectangular object inside it. A complete standalone rover prototype with various features such as wireless communication system, robust design and precise control of its movement has been tested positively. The functionality can be expanded to build a 3-D map of the arena in the future and to plot an arena with irregular boundaries.

6. References

1. Castellanos JA, Montiel JMM, Neira J, Tardos JD. The SP map: a probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*. 1999; 15(5): 948–52. <https://doi.org/10.1109/70.795798>.
2. Position estimation for a mobile robot using vision and odometry [internet]. <https://ieeexplore.ieee.org/document/220052>. Date accessed: 12/05/1992.
3. Dissanayake MWMG, Newman P, Clark S, Durrant-Whyte HF, Csorba M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*. 2001; 17(3): 229–241. <https://doi.org/10.1109/70.938381>.
4. Mobile robot exploration and map-building with continuous localization [internet]. <https://ieeexplore.ieee.org/document/681416>. Date accessed: 20/05/1998.
5. Neural networks for sonar and infrared sensors fusion [internet]. <https://ieeexplore.ieee.org/document/859830>. Date accessed: 10/07/2000.
6. Simultaneous Localization and Map building using Vision Camera and Electrical Compass [internet]. <https://ieeexplore.ieee.org/document/4108637>. Date accessed: 18/10/2006.
7. A Novel Obstacle Detection Method Based on Monocular Camera and Laser Radar [internet]. <https://ieeexplore.ieee.org/document/7469005>. Date accessed: 12/12/2015.
8. Fusion of triangulated sonar plus infrared sensing for localization and mapping [internet]. <https://ieeexplore.ieee.org/document/7469005>. Date accessed: 07/2005.
9. Churavy C, Baker M, Mehta S, Pradhan I, Scheidegger N, Shanfelt S. Effective implementation of a mapping swarm of robots. *IEEE Potentials*. 2008; 27(4): 28–33. <https://doi.org/10.1109/MPOT.2008.924855>.
10. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping [internet]. <https://ieeexplore.ieee.org/document/844077>. Date accessed: 24/04/2000.
11. All about Stepper Motors [internet]. <https://learn.adafruit.com/all-about-stepper-motors>. Date accessed: 2015.