

Obstacle Avoidance for a Mobile Exploration Robot With Onboard Embedded Ultrasonic Range Sensor

Jnaneshwar Das Mrinal Kalakrishnan Sharad Nagappa

PES Institute of Technology, Bangalore

Abstract

This paper deals with a low cost solution to obstacle avoidance for a mobile robot using just a single ultrasonic sensor. It allows the robot to navigate smoothly in an unknown environment, avoiding collisions, without having to stop in front of obstacles. The obstacle avoidance process is made up of three distinct stages - the mapping algorithm, the core obstacle avoidance algorithm, and the steering algorithm. The mapping algorithm takes the raw ultrasonic sensor readings and processes them to create higher resolution maps from the wide-angle ultrasonic sensor. The obstacle avoidance algorithm is based on the potential field theory which considers the robot to be a test charge that is repelled by all the obstacles around it, and which moves in the direction of the resultant of the forces acting on it. An algorithm which steers a mobile robot based on the differential drive system is also discussed. All these algorithms have been implemented and tested on our mobile robot ROAMER, and have been proved to work in practice.

1 Introduction

Obstacle avoidance is a primary requirement for any autonomous mobile robot. The robot acquires information about its surroundings through sensors mounted on the robot. Various types of sensors can be used for obstacle avoidance, as detailed below.

1.1 Types of Sensors

- Bump sensors, which are switches activated when the robot touches an obstacle. This is a simple, inexpensive method of obstacle detection, but operates only on contact, which makes it useful only for slow moving robots.
- Infrared proximity detectors, which detect the presence of an object in front of the sensor. They consist of a combination of an infrared light emitting device and an infrared light sensor. These sensors are merely proximity detectors, they cannot determine the range of the obstacle in front. The range of these sensors is also limited to a maximum of 80cm.
- Ultrasonic range sensors, which determine the range of the object in front of it. They work by sending a short burst of ultrasonic waves, and measuring the time taken for the echo to be received. They have a wide beam angle, typically 30° . These sensors have ranges of upto 6m.
- Laser range finders, which work on the same principle as ultrasonic range sensors, except that they use LASER instead of ultrasound. Laser range finders have a range of upto 30m, and are very accurate, having an angular resolution of upto 0.25° . However, they are very expensive compared to other sensors.

Of these, the ultrasonic range sensor was found most suitable for our requirement because of its low cost and ranging capability.

1.2 Obstacle Avoidance Algorithms

All mobile robots feature some kind of obstacle avoidance. The algorithm could be as simple as stopping if an obstacle is detected in front of it. This is common in track-following robots, where the robot follows a predetermined track. Another solution is wall-following, in which the robot moves alongside a wall at a fixed distance, following the contour of the wall till it reaches its destination. In the case of an exploration robot, the environment is unknown and obstacle avoidance has to be dynamic in nature. Our approach is an adaptation of the potential field method [3] or VFF method [2] in which the robot is considered as a test charge which is repelled by obstacles which are considered as like charges. Section 2 explains our mapping methods and an algorithm to obtain accurate data from the wide angle ultrasonic range sensor. Our potential field method adaptation is discussed in Section 3. Steering control for the differential drive robot based on the output of the potential field algorithm is described in Section 4.

The described algorithms have been implemented and tested on our test model, ROAMER (ROAMER, an Obstacle Avoiding Mobile Exploration Robot), which is discussed in Section 5.

2 Mapping

An ultrasonic range sensor detects the range of an object in front of it by sending a short burst of ultrasonic waves, and measuring the time taken for the echo to be received. These sensors typically has a wide beam angle of 15° to 30° . As a result, the cone gets wider with increasing distance from the transmitter. This increases the uncertainty in the spatial dimensions of the obstacle.

Most mobile robots which use ultrasonic range finders for obstacle avoidance have 12 or 24 sensors in a ring around the robot, to cover all directions. This is not very cost-effective, and moreover, all of them cannot be triggered at once due to crosstalk and interference problems. A maximum of 4 sensors which are at 90° to each other can be triggered at a time. Our solution was to use a single sensor mounted on a stepper motor to pan the sensor across 180° . This can be expanded to 4 sensors mounted at 90° to each other on the stepper motor for maximum scanning speed and full coverage.

Such a setup also provides the flexibility to perform fine scanning of an area for a more accurate map of the environment, with an angular resolution of the minimum step size of the stepper motor, as opposed to the 30° resolution of the sensor.

2.1 Fine Scanning Algorithm

A single range reading from the sensor implies that the area in a 30° sector in front of it with a radius of the range value contains no obstacles. A reading of a range of 1m is shown in Figure 2.1. This reading covers angles -15° to $+15^\circ$. Beyond the range value, there may be obstacles somewhere along the arc, but it is not certain at which positions the obstacles are present.

Figure 2.2 shows the next sensor reading, which is obtained after rotating the sensor through a certain angle, say 5° . This reading, with a range value of 1.2m covers angles -10° to $+20^\circ$. After obtaining this reading, we are certain that there is no obstacle between these angles for a range of 1.2m. Thus, in the overlapping region, the larger distance value is chosen over the smaller one.

If a subsequent overlapped reading is smaller than the existing value, then we have to consider the ‘age’ of the existing information. If the existing large range value is very recent, then the overlapped area is taken to have the larger range, because it can be assumed that no object moved into the intersecting area between the two scans. If the existing large range value

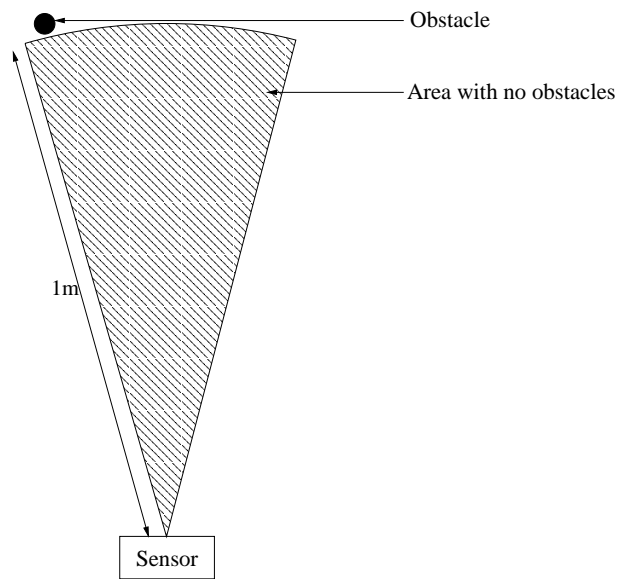


Figure 2.1: A single reading from the ultrasonic range finder

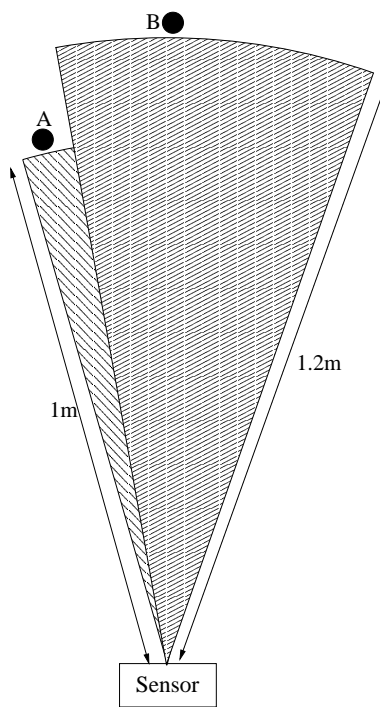


Figure 2.2: The second reading from the ultrasonic range finder overlapping the first

is old, then the smaller value is taken, because the robot may have moved within that time, or a moving obstacle may have appeared.

Using this technique, the accuracy of the ultrasonic range information can be greatly improved upto the step size of the stepper motor. However, there are some cases where the information is still not accurate. Consider the case shown in Figure 2.2. The gap between obstacles A and B can never be detected by the sensor at that distance. Once the robot moves closer to these obstacles, the gap can be detected.

3 Obstacle Avoidance

Information about the ranges of obstacles around the robot obtained from the mapping technique is stored in a table which is continuously updated. This range information is then used to calculate a destination direction which the robot must move towards. This calculation is performed using our obstacle avoidance algorithm based on the potential field method [3].

In the potential field method, the robot is considered as a test negative charge. All the obstacles are considered to be negative charges which repel the robot, and the target is assigned a positive charge which attracts the robot. The forces of all these charges on the robot are calculated in accordance with Coulumb's law, i.e. the force is proportional to the charge and inversely proportional to the square of the distance.

$$F \propto \frac{q}{r^2} \quad (3.1)$$

The robot is then steered in the direction of the resultant of all the forces added up. This results in motion of the robot similar to a charge moving in a potential field – avoiding like charges and moving towards the unlike charge.

For the case of an exploration robot without a target, the algorithm has to be slightly modified. A threshold limit is set beyond which the charge assigned to the range is positive and attracts the robot. Thus, obstacles which are close by to the robot will repel it, and the free spaces attract the robot.

$$\vec{F}_\theta = -\frac{k}{|\vec{r}_\theta|^2} \cdot \hat{r}_\theta \quad (3.2)$$

Equation (3.2) is the basic equation representing the force exerted on the robot by a single obstacle at a distance r and an angle θ from the robot. k is a numerical constant selected to bring the force into the required range. The charge q is not present in this equation because different angles do not have different charge values. This equation is modified to make near ranges repel the robot and far ranges attract, as shown in Equation (3.3).

$$\vec{F}_\theta = k \left(\frac{1}{t^2} - \frac{1}{|\vec{r}_\theta|^2} \right) \cdot \hat{r}_\theta \quad (3.3)$$

Here, t is a threshold distance at which the obstacle neither attracts nor repels the robot, and ranges beyond the threshold distance are attractive. The total force on the robot is obtained by vectorial addition of all the individual forces.

$$F_{total}^{\vec{}} = \sum_{\theta=0}^{2\pi} \vec{F}_{\theta} \quad (3.4)$$

$$F_{total}^{\vec{}} = k \sum_{\theta=0}^{2\pi} \left(\frac{1}{t^2} - \frac{1}{|\vec{r}_{\theta}|^2} \right) \cdot \hat{r}_{\theta} \quad (3.5)$$

Equation (3.5) gives the expression for the resultant of all the forces acting on the robot. The robot is steered in this direction in order to avoid obstacles. The magnitude of the vector can be used to control the speed of the robot. This results in slow movement in a cluttered environment, and faster motion when there are less obstacles surrounding the robot.

4 Steering Control for a Differential Drive System

A mobile robot which uses differential drive has separate speed and direction control for the left and right sets of wheels. Such a system is very flexible for a mobile robot because it does not have a minimum turning radius, it can execute a turn around its own center. Arc turning is achieved by driving the left and right wheels at different speeds. Driving one set of wheels forward and the other set backward results in in-place turning.

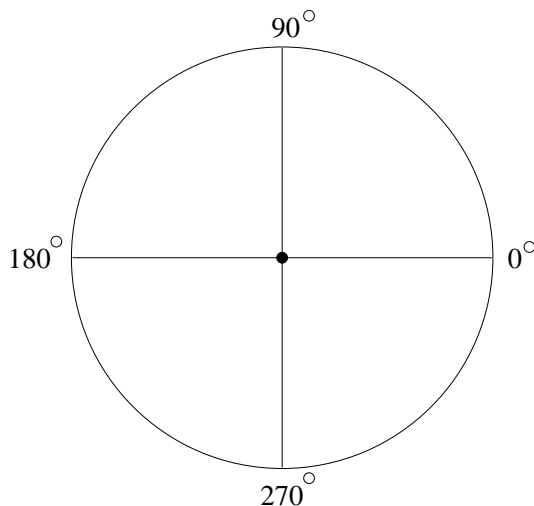


Figure 4.1: Unit circle in which all possible force vectors lie

An algorithm was designed to convert the force vector obtained from the obstacle avoidance algorithm into the left and right wheel speeds required

for the differential drive system. The magnitude and direction of the force vector is plotted as a point on a graph, after being normalized so that the maximum possible magnitude is 1. Thus, all possible force vectors will lie in a unit circle around the origin, as shown in Figure 4.1.

On this circle, a point at 90° indicates that the robot has to move straight, a point at 0° indicates that the robot has to turn right in-place, and at all points in between, it has to exhibit varying degrees of turning.

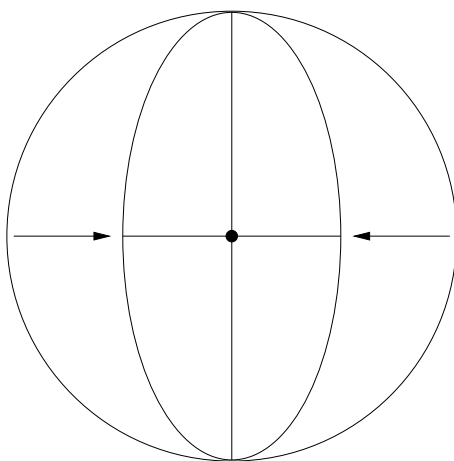


Figure 4.2: The unit circle shrunk into an ellipse

This unit circle is shrunk into an ellipse as shown in Figure 4.2, by dividing the x coordinate by a constant. This ensures that the wheel speeds of the robot are lower while turning and faster when moving straight. Turning at high speeds can result in slipping and inaccurate turns, this prevents it from happening.

Now the existing force vector is rotated clockwise by 45° , as indicated in Figure 4.3. The new coordinates of the vector ($newx, newy$) are then taken as the left and right wheel speeds respectively (after normalization). This results in the robot turning in the required direction based on the direction of the original force vector. This dynamic steering algorithm ensures that the robot doesn't have to stop in front of an obstacle to go around it, instead it navigates smoothly to the required direction. The force vector is updated every time new range information is obtained, which is approximately every 20 or 30 milliseconds, and the wheel speeds are also correspondingly modified every time. Modifications in the wheel speeds will not be abrupt, because every reading changes the range associated with only a narrow angle, resulting in a very small change in the resultant force vector. Since the wheel speeds vary smoothly according to the sensor information, and the speeds are updated almost in realtime, the robot follows a smooth course around the obstacles present in the environment, while still being able to respond immediately to changes in the environment like dynamic obstacles.

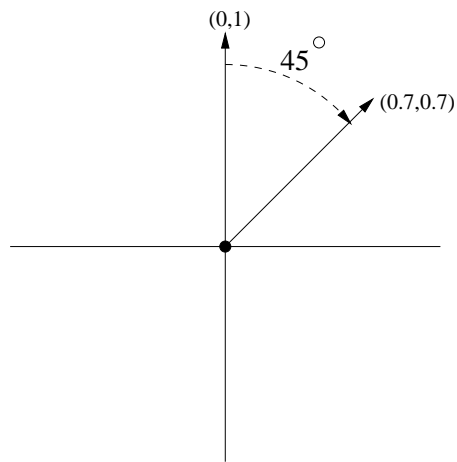


Figure 4.3: Rotation of the force vector

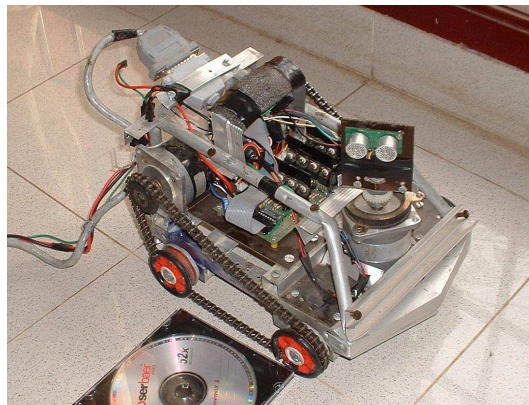


Figure 5.1: ROAMER - an Obstacle Avoiding Mobile Exploration Robot

5 ROAMER

ROAMER is the mobile robot we designed and constructed, and implemented the obstacle avoidance algorithms on. It uses a differential drive system for mobility, but uses tracks instead of wheels, like a tank. The drive motors used are stepper motors. It has a single ultrasonic range sensor mounted on a stepper motor in front which can pan across an angle of 180° in steps as small as 0.45° . The robot is controlled by a computer through the parallel port connected to the required driver circuits on the robot itself.

RTLlinux was chosen as the software platform because realtime functionality was required, especially to measure the sonar range timing, and to trigger the stepper motors. A graphical user interface was designed so that the range data can be viewed on the screen in a radar-like display to aid in debugging (shown in Figure 5.3). It also has the capability of providing

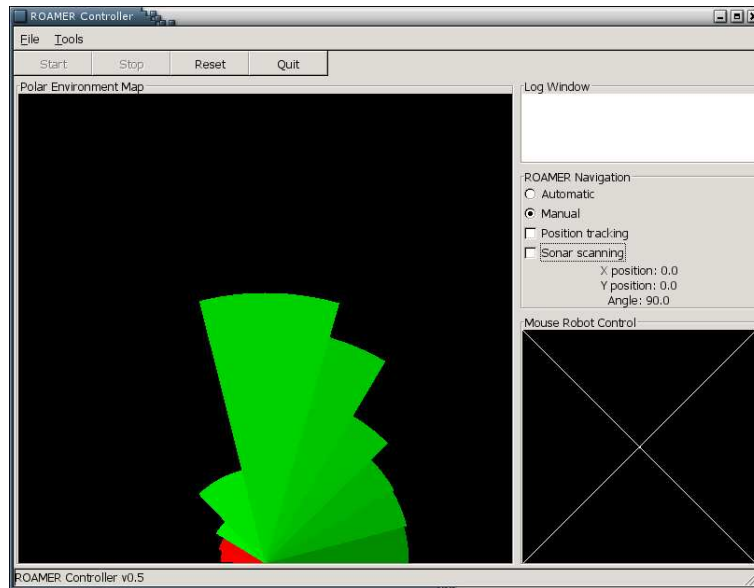


Figure 5.2: Screenshot of the ROAMER controller program. In the polar map shown, the red sectors represent obstacles which are very close by and below the threshold limit, and the green sectors represent ranges that are above the limit and thus attractive.

manual control for the robot, which makes it useful as a tele-operated vehicle as well.

The default interval at which ultrasonic range readings are taken is 15° , which provides 12 readings over 180° . Each reading overlaps the previous by 15° . This provides for fast scanning of the environment – an average of 500ms to scan 180° , and thus the robot can move at high speeds without danger of collision. If an area needs to be mapped more accurately, or in case of a highly cluttered environment, the step size is reduced and the maps obtained are of higher resolution, which can help in navigating precisely through smaller pathways.

A real-life test setup for the robot is shown in Figure 5.3, and the path of the robot in this setup is shown in Figure 5.4.

6 Conclusion

A low cost solution to obstacle avoidance for a mobile robot was designed using a single ultrasonic sensor panning an angle of 180° . An algorithm was designed to process and correct the ultrasonic range data to get higher resolution maps. The potential field method [3] was modified for the case of an exploration robot and used for obstacle avoidance. A steering control method was devised which allows for smooth navigation of the robot using

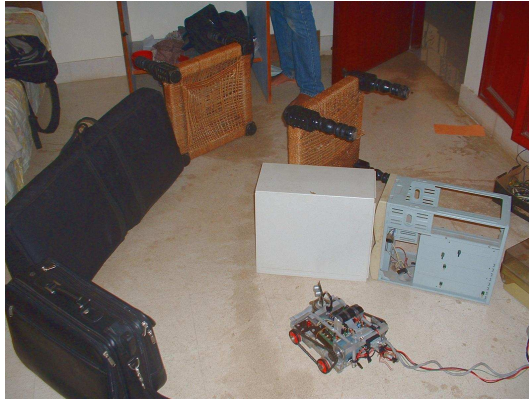


Figure 5.3: Test setup for ROAMER

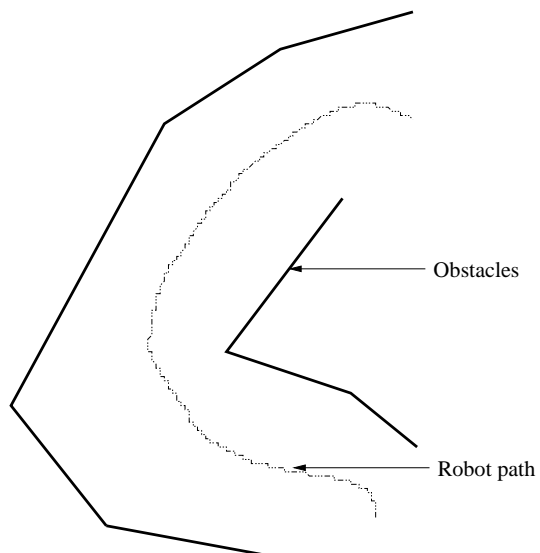


Figure 5.4: Path of ROAMER in the test run

the direction provided by the modified potential field method.

All these algorithms have been implemented on our robot ROAMER, and tested in real-world situations, and were found to be suitable for an exploration robot. The robot was tested upto a maximum speed of 10cm/sec, the speed being limited by the stepper motor drive mechanism. These algorithms can be adapted to any situation where autonomous navigation through a previously unknown environment is required, for example, robots that explore potentially hazardous territory, robots that defuse landmines automatically, or even household vacuum-cleaning robots that clean entire floors.

References

- [1] Borenstein, J., Everett, H.R., and Feng, L., 1996, "*Where am I? - Sensors and Methods for Mobile Robot Positioning*" University of Michigan.
- [2] Borenstein, J. and Koren, Y., 1989, "*Real-time Obstacle Avoidance for Fast Mobile Robots*" IEEE Transactions on Systems, Man and Cybernetics.
- [3] Krogh, B.H., 1984, "*A Generalized Potential Field Approach to Obstacle Avoidance Control.*" International Robotics Research Conference, Bethlehem, PA.