**Object Mapping and Detection Using Ultrasonic Range Finding with DE2 Bots**

Philip Balê

Caitlyn Caggia

Eydan Fishel

Can Kabuloğlu

ECE 2031, Digital Design Lab

Section GTL

Georgia Institute of Technology

School of Electrical and Computer Engineering

Submitted

April 29, 2016

# **Abstract**

This paper presents a method to detect and locate objects using ultrasonic rangefinding with a DE2Bot. The purpose of this project was to improve the sonar mapping capability of the DE2Bots. The robot navigated through an arena, represented as a 5 x 4 grid, with three to six objects randomly placed outside of the center aisle. First the existing functionality of the DE2Bot was explored, including movement, ultrasonic range finding, and radio communication. After testing an initial depth-first search algorithm, a faster algorithm was created to keep runs under 60 seconds. Other supportive assembly subroutines were developed to move the robot along a clear path through the arena, detect and record objects and their locations, and transmit data containing object count and coordinates. In the final demonstration, the three trial runs produced a final score of 3597 points. In the first two trials, partial credit was awarded for object count and location. The correct object count and coordinates were reported in the final trial, which was considered a perfect run. All three trials earned time bonuses and avoided deductions by colliding with walls or objects. Additionally, an updated Gantt Chart is included to provide a project plan and timeline.

**Object Mapping and Detection Using Ultrasonic Range Finding with DE2 Bots**

# **Introduction**

This document discusses a method of sonar mapping using a DE2Bot to count and locate objects in a walled rectangular arena. The team first learned existing features of the DE2Bot, including how to move the robot, how to measure the distance of objects with the ultrasonic sensors, and how to relay coordinates via radio signal. Next, the team focused on developing a search algorithm to navigate the arena and find objects.

The object search algorithm first checked for objects along the main aisle, then checked branches where the robot can see the wall from the main aisle. Additional code was developed to move the robot along a clear path through the arena, store object locations, and transmit coordinates containing object locations.

In the final demonstration, the team’s cumulative score from all three trials was 3597. The first two trials earned partial credit for object count and location. The third trial was a perfect run, with the correct object count and coordinates reported. All three trials earned time bonuses and avoided deductions by colliding with walls or objects.

# **General Methodology**

The objective of this project was to program a DE2Bot to detect and locate three to six randomly placed objects on the field, and send the number of objects and their locations to a central computer in 60 seconds or less. Efficiency was a priority to earn bonus points for remaining time. The robot avoided hitting walls or objects, so no points were deducted from the trial run scores.

The team developed and tested code to:

* Move the robot to specific locations on the field without hitting objects or walls.
* Detect nearby objects using the ultrasonic sensor.
* Record the location of detected objects in memory.
* Determine the robot’s next location using a search algorithm.
* Transmit number of objects and object locations via radio signal.

## **Moving on the Field**

First, the team needed to move the DE2Bot on the field accurately, keep track of the DE2Bot’s position, and avoid collisions with objects or walls. Since all objects were placed within a grid, the DE2Bot only needed to move in straight lines or 90-degree turns. To navigate the arena, the team developed code to:

* Turn left and right at 90o angles.
* Move forward and backwards two feet (the length of one tile on the grid).
* Track the DE2Bot’s location inside the arena.

Each step was independently coded in assembly as subroutines and tested on the field for accuracy; later these subroutines were worked into the main search code. The robot’s location was determined by its predicted location rather than its actual location.

Odometry was used to accurately execute turns. The mechanics of the DE2Bot introduced odometry error when the robot moved forward, backward or turned. The error increased with the length of distance traveled, so the team chose to have the robot move only one tile at a time and rotate no more than 90 degrees at a time. Movement error was also reduced by selecting a wheel velocity high enough to make the wheels move, but low enough that inertia had minimal effects. The team also chose a search algorithm which minimized movement to reduce errors related to robot movement.

## **Detecting Objects**

The DE2Bot checked on the right then the left to see if an object is present in a nearby tile. Sonar sensors 0 (left side) and 5 (right side) were used to collect the distance data in millimeters, which was converted to feet and saved as variable called object distance. Object distance could be positive or negative, depending on the orientation of the robot and which sensor was being used. If an object was less than 2 ft. away (or 610 millimeters) in a given direction, then an object was one tile away. If the robot detected a wall, the search algorithm recognized that a detected surface out of bounds, and no location was recorded. Since the robot only checked for objects immediately to the left or right of its current tile, all surfaces were perpendicular to the sensor, which reduced error in sonar measurements.

## **Recording Object Locations**

Every time the robot detected a new object, it noted the robot’s x-position, y-position, and the object’s distance from the robot. The object distance was determined by the sonar readings and could be a positive or negative value.

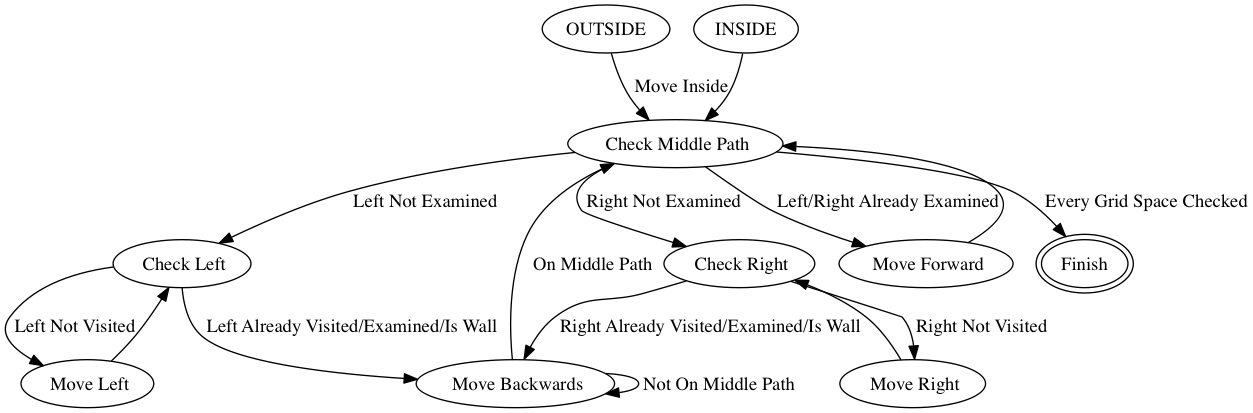
There were four cases for determining where the object was located:

* If the robot was on the center aisle,
  + The y-value of the robot was the same as the y-value as the object.
  + The x-value of the robot plus the object distance was the x-value of the object.
* If the robot was on a branch,
  + The y-value of the robot plus the object distance was the y-value of the object.
  + The x-value of the robot was the same as the x-value of the object.

Location data was kept directly in the DE2Bot’s memory as 12 predetermined memory locations, consisting of six sets of x- and y-coordinates. The new object’s position was then compared to existing object locations. If a match was found, that object was already recorded, and the search would continue. Otherwise, the new coordinates were immediately reported, and object count was incremented by one. The new object would be stored in an “empty” memory location, which was the first location that still had (0,0) stored. Object count was sent when six objects were found, or when the search pattern was finished.

## **Determining the Robot’s Next Position**

A flowchart is depicted in Figure 1 to illustrate the search process. To begin the search, the robot started two feet outside of the arena, facing the inside, and moved forward into first tile. The robot scanned the right and left sides with sonar to detect objects that could be one or two tiles away. The robot advanced forward a tile and repeated this process until the entire center aisle had been checked. Then the robot proceeded to check branches where the main wall was detected from the center aisle. If both the left and right side were clear, then the right side would be checked first, and the robot would continue immediately to the left side without stopping to rotate toward the back of the arena.

**Figure 1.** Flowchart illustrating the robot’s decision making process to find objects, record new objects, and determine where to move next.

In each branch, the robot looked both above and below its current location to search for objects. If the robot was on a branch and could see that two or three adjacent tiles were clear, the robot would not check that side again. Additionally, the team determined that all objects could be found by only examining each side twice. If the robot had already checked one side twice, it would not check that side again. If at any point the robot detected six objects, the search ended early.

## **Transmitting Object Count and Locations**

Code to send data from the DE2Bot to the central computer was provided with the project details. Coordinates were sent each time a new object was found. Object count was sent when six objects have been detected or when the search was completed.

## **Project Management**

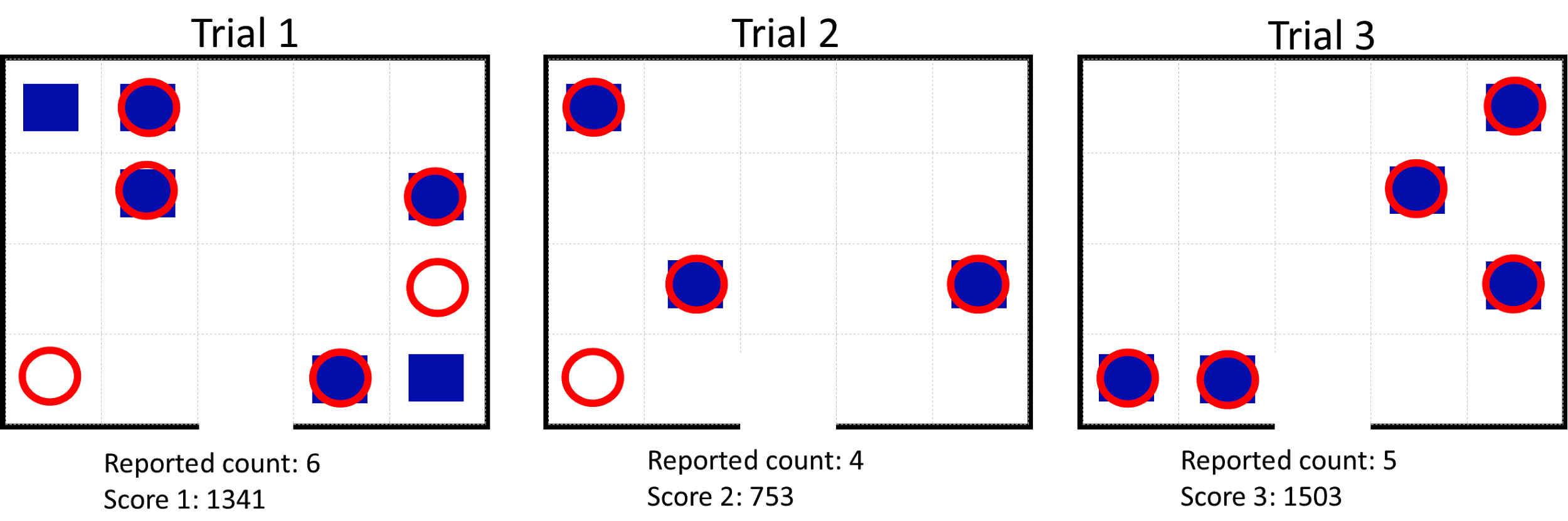
Appendix A contains an updated Gantt Chart outlining the intermediate tasks to complete the project and related reports and presentations. Major tasks included completing the final presentation, preparing for the demonstration, and writing the final report. Changes to the original Gantt Chart include:

* Longer time than expected needed to perfect robot movement, which caused a delay in radio communication tasks.
* Additional task of developing a new search algorithm, as the original search algorithm didn’t meet time requirements.
* Shuffling which person was assigned to each task to accommodate logistics and team strengths more effectively.

A Project Logbook has been supplied in Appendix B to break down in detail the team’s schedule and short-term goals.

# **Technical Results**

The team’s cumulative score from all three trials was 3597. The first trial had a correct object count and 4/6 correctly reported locations. Partial credit was awarded for the remaining two locations. The second trial received partial credit for overcounting by one, and full credit for reporting object coordinates. The final trial was a perfect run, with the correct object count and all coordinates reported correctly. All three trials benefited from timing bonuses, and there were no deductions for hitting objects or walls. The diagrams in Figure 2 show the object placement, reported locations, and points earned for each trial.



**Figure 2.** Diagram showing the actual object placement as blue boxes and reported locations as red circles for each trail. Object count and points earned for each run is labeled below each trial diagram.

In trials 1 and 2, the robot falsely detected an object in the bottom left corner at (1,1). This error was a result of movement inaccuracy, as the robot considered itself in tile (1,2), while it was nearly in tile (1,1). Thus, since the wall was closer than anticipated, the robot inaccurately detected the wall and counted it as an object located at (1,1). This error was generally avoided in practice runs by carefully placing the robot in the correct starting position and using a fully charged robot for consistent motor output.

# **Conclusions**

The DE2Bot was able to detect objects in the arena in almost every arrangement with at least partial accuracy. The robot achieved everything the team proposed. The robot:

* Moved to specific locations on the field without hitting objects or walls.
* Detected nearby objects using the ultrasonic sensor.
* Recorded the location of detected objects in memory.
* Determined its next location using a search algorithm.
* Transmitted the number of objects and object locations via radio signal.

Some improvements that the team could have added would have included removing the pause every time the robot moved forward one tile and have the robot scan for objects while moving. The team could have also perfected the movement function of the robot. This would make sure the robot did not drift off into another tile instead of going straight. Another way to account for movement error would be to use odometry rather than hard-coding the objects’ locations.

The team was able to complete their goals thanks to an effective contingency plan. The Gantt chart provided realistic goals. The Gantt chart also kept the team focused on which tasks to prioritize. Some adjustments were made to the timeline as some tasks proved to take more time than predicted, like testing movement functions. Movement accuracy was one of the most significant challenges the team faced. The robot kept drifting in one direction and sometimes went into another tile. This occasionally led to the robot falsely detecting the wall as an object. This proved to be the case in trials 1 and 2 in the final demonstrations, costing the team 700 points.

**Appendix A: Updated Gantt Chart**

**Appendix B: Project Logbook**