

Cooperative Mapping of an Unknown Space using
Autonomous Robots with Low-Level Sensors

Vissuta Jiwariyavej
Nimit Dhulekar

CS – 54: Introduction to Robotics

12 March 2009

Table of Contents

1. Abstract.....	3
2. Introduction	3
3. Techniques and Methods.....	4
4. Results and Discussion.....	5
5. Conclusion	6
6. References	6

1. Abstract

This project dealt with using 2 iRobot Creates to map an unknown area using only the low-level sensors present on the robots. Once the space is mapped, we display a graph of the traversed area. To help us read the sensor data from the Creates, we used the IRobot_Create module provided by RoboRealm, a robotic vision software application. The sensor information that can be retrieved from the Creates includes distance, angle, obstacle collision, cliff drops, IR bytes etc. We have mainly concentrated on the distance, angle and obstacle collision sensor readings.

Our approach consists of a client-server technology where one laptop acting as the client records information from one of the Creates and sends this information to the other laptop acting as the server. The server laptop then records information from the other Create, combines the traversal maps for the Creates and displays a graph.

2. Introduction

The iRobot Create is a hobbyist robot manufactured by iRobot that is based on the Roomba platform. It was introduced in 2007. In place of the vacuum hardware of the Roomba, the Create includes a cargo bay which houses a 25 pin port that can be used for digital and analog input and output. The Create is a complete robot development kit that allows programming new robot behaviours without knowledge of low-level coding and mechanical assembly. The Create's Open Interface provides a set of commands to control the motion and behaviour of the robot including driving, turning, playing songs, changing the state of the LEDs, sensing walls etc.

One of the significant advantages of using the iRobot Create is the ease of attaching external hardware and electronic devices. For the purpose of this project we used the Element Direct Bluetooth Adapter Module (BAM). The BAM enables wireless control of the iRobot Create robot. The BAM connects to the cargo bay port of the iRobot Create without any extra cabling. The BAM provides a virtual serial port connection between the iRobot Create and the laptop. The BAM also exposes the Create's programmable IO, making it easy to connect additional hardware.

RoboRealm is a powerful robotic vision software application for use in computer vision, image processing, and robot vision tasks. RoboRealm provides a simple user interface to control the iRobot Create in the form of the IRobot_Create module. The IRobot_Create module is a GUI designed by RoboRealm to read and control the sensor values on the Create.

The Server API provided by RoboRealm allows remotely controlling the RoboRealm application. In this project, RoboRealm functions as a server for serial communication between the BAM and the laptop. The RoboRealm package also provides a library containing API routines to manipulate variables and parameters. This manipulation capability has been leveraged to read the sensor data stored in variables and accordingly change the behaviour of the Create.

3. Techniques and Methods

The basic underlying technology used for this project was multilayer client-server architecture. It is multilayer since there is communication between the 2 laptops as well as communication between the program running on the laptop and the RoboRealm API Server which provides a virtual serial connection to the BAM on the Create. The steps involved in the process are the following:

- 1) The client laptop connects to RoboRealm API Server which in turn is connected to the BAM on the Create. This enables wireless communication between the laptop/RoboRealm and the BAM.
- 2) The client program sets the motion parameters for the Create. The Create will “drive” according to these parameters. These parameters include the velocities of the Create’s left and right motors. Initially the range of these velocities was set from 0-255 where 0 and 255 are the maximum forward and backward velocities respectively. These values are just a relative indication of speed and are not exact units. Other range values can also be set. The median of the range is the stopping velocity. Thus in this case it was 128. To move forward at a controlled rate, the velocity was set to 148. Increasing this number towards 255 or decreasing this number towards 0 increases the velocity thereby introducing more drift.
- 3) When a collision is detected using the bump sensors on the Create, a turn motion is initiated. The velocities of the motors are set to the stopping velocity and then the velocity of the left motor is set 148 and that of the right motor to 108. This enables on the spot turning. The angle sensor is now used to get a relative measure of the angle through which the Create turned. This value is not in degrees or radians. Thus tests were conducted to verify a value which gives an approximate turn of 90 degrees. This is one of the problems that were faced while working with the Creates that the encoders don’t always return exact values. The distance traveled from the initial point or the point of last collision is recorded.
- 4) For the purpose of this project we limit the time one robot moves to 3 obstacle collisions. Once this limit is reached, the information is sent over the network to the server laptop.
- 5) The server laptop now initiates the same process as outlined above in steps 1 though 4 and repeats the process with the other Create until a maximum of 3 obstacle collisions.

- 6) The information recorded from both the Creates is now collated on the server laptop and a graph is generated.

4. Results and Discussion

Initially the coding was done in C. But the RoboRealm API provided for C language was not able to read live sensor data and would return the values from parameters at initialization time. Thus the coding was redone in Java. On communication with Steven Gentner of RoboRealm, it was found that there was a thread configuration issue. This was not investigated further since the Java API worked reliably.

Another issue that arose was the limited range of Bluetooth. Although the BAM advertises a range of 300 feet, it was discovered that the laptop would lose control of the Create beyond 60 feet. Thus the need for more reliable communication arose and wireless communication was adopted instead of using only Bluetooth. The wireless communication was used for the client-server architecture between laptops while Bluetooth was still used for communication between the laptop and the Create. This had the limitation that a laptop would have to be in the vicinity of the Create but multiple Creates could be anywhere in the range of the wireless network.

The advantage of using the iRobot Create is that it is an inexpensive robot developed for robotic and educational purposes. The downside of the Create is that the sensors are not completely reliable. The wheel encoders are susceptible to drift and slip errors. The angle data is also dependent on the wheel encoders and thus has the same problems. These errors can propagate and completely disrupt the path of the Create. To overcome these errors, it was planned to integrate the Devantech Magnetic Compass which measures absolute direction with a 0.1 degree resolution and a 3 to 4 degree of accuracy. This compass is particularly designed for use in robots as an aid to navigation.

The Devantech Magnetic Compass could not arrive in time thus the backup plan was to leverage the wall signal sensor on the Create. The value returned by the wall signal varies in the range 0-4095. But this too was plagued by the problems as on the wheel encoders. The wall signal was used to reorient the robot onto the path when it would drift. This was found to be extremely hard since the robot would never orient according to the specified angle. There were 3 reasons for this:

- 1) The time lag between the commands sent to the Create and the execution of the commands by the Create. This lag would cause the Create to move more than the specified angle.
- 2) The turning angle cannot be set with a fine resolution and precision.
- 3) Also since the chassis of the Create is a replicate the Roomba chassis, the wall signal sensor is located on the right front side. Thus this can not

be used for frontal approach to the wall or moving along the wall with the left side in closer proximity to the wall.

5. Conclusion

The main goal of being able to get multiple robots to move around an unknown space using only low-level sensors, and report back information was achieved. One of the goals we had set out to achieve was reliable motion which could not be accomplished in the time frame as the Devantech Magnetic Compass did not arrive in time. The integration of the robot with IR sensors on the sides would allow better navigation in an enclosed space. Also the IR sensor values can be used to detect open spaces on the sides for mapping purposes. For the client-server architecture, an improvement would be to use a blocking listen on both the server and client side. The server and client can then record the information from the robots at the same time and whichever one finishes first would simply block waiting for communication from the other.

6. References

- 1) iRobot Create Owner's Guide
- 2) BAM for the iRobot Create Owner's Manual
- 3) iRobot Create Open Interface Manual
- 4) RoboRealm www.roborealm.com
- 5) Devantech Magnetic Compass <http://www.acroname.com/robotics/parts/R117-COMPASS.html>