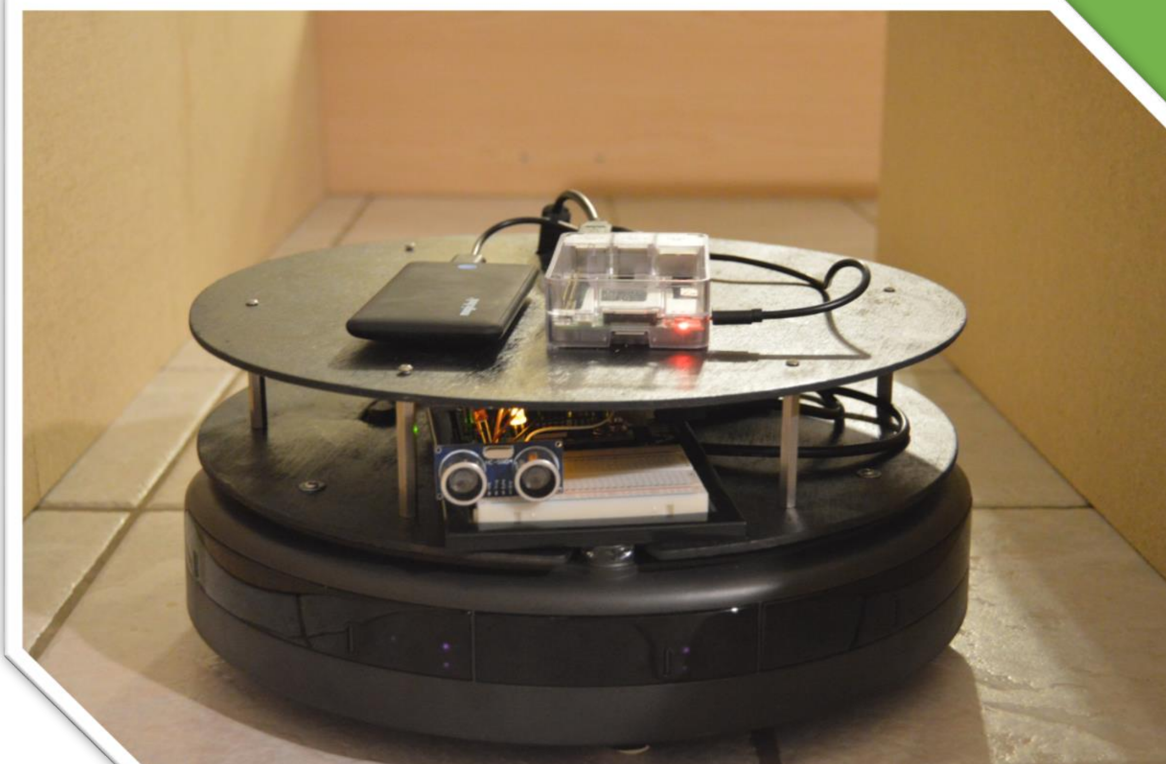
2017

Create 2 Report



Hamilton Chevez - Milad Ebrahimi

4/30/2017

Contents

INTRODUCTION

For the end of semester project, we picked to work with an iRobot Create 2 [1]. It is a modified Roomba made for STEM purposes. It has all the sensors as any other Roomba, bumper sensor, single proximity sensor, and cliff detection, except for no vacuum. The extra storage is meant for housing any extra components we wished to use. Due to previous project failures with drones we set out to make the minimal viable project we could with the 1 month time frame we had left of the semester. First, we explored the Open Interface api to understand what sensor data was available for us to work with and how to make it move. To make our Create 2 be autonomous, we knew at a certain point we would have to attach a mini computer or microcontroller to it instead of having it tethered to our laptops constantly. For this aspect of the project we picked a Raspberry Pi 3 [2]. The final stretch of the project revolved around using OpenCV[3] with a camera and the process of reading serial data from sensors both on board any that of an Arduino we used.

MOTIVATION

The initial intention of our semester project involved working with a Parrot A.R. Drone 2.0.[4]. The project goal was to make it fly autonomously given certain coordinates considering it had a gps dongle for detecting position coordinates. Upon doing so, if it detected any humans it would take pictures, and using the Twitter API, create new image tweets of the people it saw. This could prove useful in disaster relief. Unfortunately, our initial endeavours were never realized.

After many unsuccessful attempts of trying to make the drone fly, we went back to our project mentor, Tauhid, to ask what else we could do instead for our semester project. He introduced one of his current projects related to his research. He was attempting to make two Create 2 robots traverse an enclosed perimeter autonomously. Each one was equipped with an Arduino [5] running a series of commands while detecting triggers of the bumper sensor. The objective of his current project was to have both Create 2 traverse the space, but to avoid making contact no matter how close they got to each other. We saw a glimmer of hope we figured that getting a Create 2 to do what we wanted would be easier and less risky than trying a 3rd Parrot Drone. From then on, we went to purchase parts and started planning the project.

MAIN IDEA

Tauhid used an Arduino to control the Create 2, in our case we wished to control it with a Raspberry Pi since it would let us do “more”. As we began the project, we learned that it was possible to power the raspberry pi directly from the Create 2. However, in order to avoid damaging the raspberry pi a logic level shifter was needed to be created in order to lower the 5v TX line (Create 2 current out) to 3.3v to work with the Pi. The iRobot company offers documentation that shows two possible ways of creating your own logic level shifter with all the necessary parts. Since, none of us knew anything about electronics nor did we have the time to learn we kept searching for alternative solutions.

Furthermore, we found another viable solution but not the simplest on the Raspberry Pi Stack Exchange portal. It is possible to use a USB Car Charger to regulate the voltage from the Create 2 to the Raspberry Pi. This can be accomplished by connecting the leads from the Create 2 where the charger would meet the car port. A USB to microusb type b cable would then be used with the charger and connected to the Raspberry Pi. Since this required, soldering we opted for the most expensive but simplest solution. Using the Create USB cable that came with the Create 2. The solution this cable would provide became our sole reason for just purchasing the Create 2.

To move the Create 2 we used BreezyCreate2 [6], a python library. The library provided various functions for moving and turning the Create 2. For rotation and linear velocity changes, a value between -500 and 500 would be used with appropriate functions. Where -500 meant to rotate, counterclockwise, or move backward at max velocity. The positive extreme created a similar performance for the Create 2, regarding clockwise rotational and forward velocity. Our initial belief was that there would be a one to one mapping of the values to velocity, but that was not the case. We observed that values within certain range did not affect the velocity of the Create 2. For our project, we set the velocity to have a value of 30, for rotation, since any value between 21 and 29 created no noticeable difference, and a value of 50 for linear velocity.

Once we achieved the connection of the Raspberry Pi via the supplied Create 2 interface to USB cable and installed the necessary python libraries, we started formulating what environment our robot would traverse. For the purpose of achieving some basic path planning we created a maze that had only one entry and not exit point. The objective of the robot would be for it to reach the dead end of the maze. The simplest viable but most boring solution using the built-in sensors would be to perform the wall hugging algorithm. Where the Create 2 would bump into a wall, reverse, rotate, move forward, and bump into the wall again, and repeat. This algorithm would work if our maze had an exit point only.

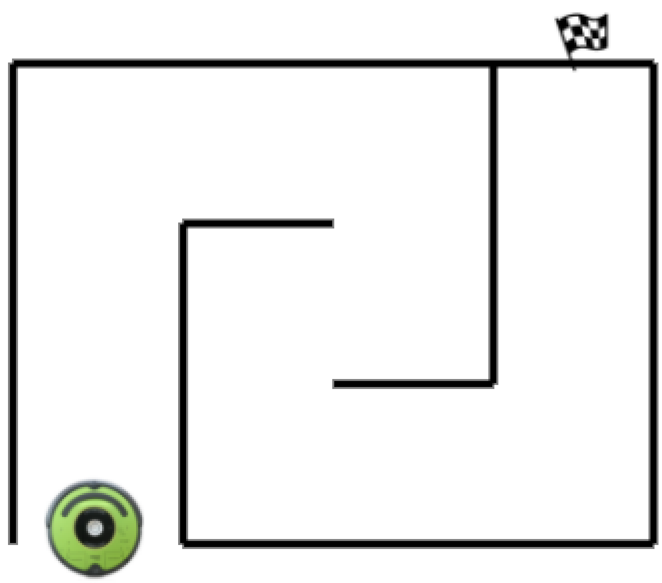


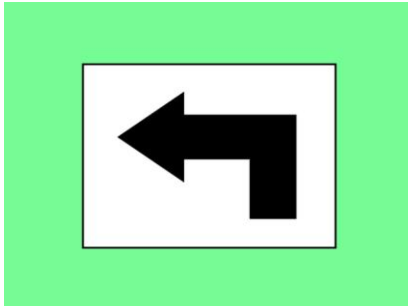
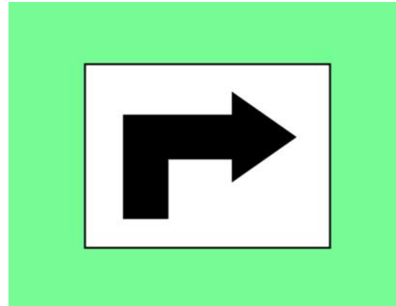
Figure 1

Sometime later we decided to not do the wall hugging algorithm, but instead we decided to attempt the traversal of the path while avoiding the walls. In order to avoid crashing in the walls of our path, we decided to use ultrasonic proximity sensors. We had used one in one in our previous locomotion class lab so we had some experience with it. Again, due to time constraints we opted for the easiest solution. Instead of learning how to do threading with python, and have one thread constantly polling the ultrasonic sensor to get the distance of any obstacle in front of the robot, we used an arduino to do that role. Essentially, we created our second thread for our project through hardware.

To read the data that the Arduino was reading from its Ultrasonic Sensor, the Raspberry Pi was connected to it via a serial usb-cable. The Arduino sent through a stream the sensor data and the Raspberry Pi read the content of the stream and the value would later be casted to a float.

With a way of determining the proximity of the robot to walls, we began designing our decision making algorithm. In summary, the algorithm performs three actions. It detects if there is a wall near the robot, rotates counter clockwise approximately 90 degrees, checks if there is a wall, if not then it moves forward else it rotates 180 degrees clockwise. At this point it will check again if there is a wall, if there isn't then the loop starts from the top of execution but if there is then the robot was reached the dead end of the maze. At this checkout of the project, we began to continue pushing the limits of what we could do with the Create 2 with the given time limit.

The next phase of the project involved enhancing the autonomous ability of our Create 2 by adding a camera to the project and using OpenCV to detect images. The Images we wanted to capture and process via OpenCV those of arrows. Hence, during the execution of the program when the Create 2 reached a wall it would proceed to detect what arrow was on the wall and rotate 90 degrees counter clockwise if it was an arrow pointing to the left, and vice versa.

This aspect of the project was never fully realized. Our main stumbling block came from our lack of experience with OpenCV library. We found various online resources and tutorials that taught us much. Some actually had achieved similar objectives what we had in mind for our project. However, they required using a regular USB webcam. When we used one with the Raspberry Pi, the Pi proved to be too slow for reading frames at an adequate rate. In order to circumvent this issue, we opted for using the raspberry pi camera module that is attached directly to the Pi's motherboard.

The challenge began here, the library used for capturing video frames did not return the frame in a type that would work with the tutorials we found online. After many trial and errors, we finally made progress with one Python script [7]. Unfortunately, upon the detection of the arrows that we had designed for our project, somehow its detection caused our python program to crash. Surely it was an error in the OpenCV portion of the script, but due to time constraints we did not pursue the debugging of the code and removed that feature completely for the project.

FUTURE DIRECTIONS

With regards to future plans for our project, we really would like to have the OpenCV functionality. On top of previously stated features, we believe that ability to detect edges may also help us navigate corners better. If the Create 2 is close to a corner it needs to rotate about, then by used an edge detection filter, we could approximate how far the corner is with respect to the frame of the Create 2 and the camera. Another plan, involves the use of more ultrasonic proximity sensors with the Create 2. By adding 2 or 3 ultrasonic proximity sensors at perpendicular axis to each other we would like to attempt to design a vornoi mapping of the path as it is traversed. What we would attempt to do is essentially try to mimic the capabilities of a Lidar sensors with only ultrasonic sensors.

The final plant for the project would be along simultaneous localization and mapping of the robot as it traverses any complex environment. If the previous experiment with the ultrasonic sensors is a success, then we think we could create a mapping of any environment by creating a graph like representation. The create 2 would start by rotating 360 degrees and creating a mapping of how close objects are to it. That would be the starting position and node. After, it would pick the point furthest from it, previously recorded while rotating, and move towards it for 3 seconds or when an obstacle comes in contact with the Create 2. Since the ultrasonic sensors are mounted on top of the Create 2, the bumper sensors would be used to circumvent the height requirement. It's possible the ultrasonic sensor may not detect objects below it but the bumper sensor would. Upon stopping, the Create 2 would create a new node, and give a weight value to the distance between the previous node and the new node. The cycle would then repeat, with the Create 2 rotating another 360 degrees. There exists the possibility that the Create 2 would return to the area of one of the previous nodes. If this were to happen and ideally we wish it would, then the final stage of this aspect of the project would to do A*, depth first, and breadth first searches for leaf nodes and explore different paths.

BOOKMARK

[1] http://www.irobot.com/About-iRobot/STEM/Create-2.aspx

[2] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[3] http://opencv.org/

[4] https://www.parrot.com/us/drones/parrot-ardrone-20-elite-edition

[5] https://www.arduino.cc/

[6] https://github.com/simondlevy/BreezyCreate2

[7] https://robosifo.wordpress.com/2015/11/21/weekly-report-11/