AI and Robotics

Project

iRobot Create

By

Abhijit Gaikwad

Sanil Gandhi

Manoj Gandhi

The problem presented was to configure the iRobot Create series Robot to traverse the corridor of the Computer Science Department from Dr Kaners Cabin to Roslyns Office to collect the cookies. Then to carry the said cookies to Dr Shoaff's Office. The problem presents a closed corridor from the starting point till the Final destination. Thus in this scenario to localize itself and the control its movement the iRobot create can make use of the wall sensor. Tracing the wall makes the robot move rather efficiently and in a controlled motion. The Wall sensor allows sensing the presence of doors in the corridor and thus this helps with the realization of the desired destination.

The lacuna in this approach was that the wall sensors with the iRobot Create were not always reliable and hence we could not base our decision to stop at the destination solely on that. Thus we decided to use distances as a back up to help us with localization if we miss to count a door due to sensors, which allows the robot to stop at the approximate desired location.

The Secondary Target's put forth were to achieve the goal with the shortest time (To race with the other robots and ensure that the Robot in question reached the destination first.) and to compensate for a single interference by someone to any bumper sensor at any point. Thus it is imperative that the speed be tuned to the best possible value, such that it does not cause the robot to lose its localizations. We performed extensive tests to improve the motion of the robot to get the most speed and stability.

We used Python as the language of implementation for this project. Python allowed us easy access to serial communication. This made writing to device and reading from device convenient.

We used the following –

- Python 2.4.4 - http://www.python.org/download/releases/2.4.4/ (www.python.org)
- Pyserial – 2.4 for windows – www.sourceforge.net
- Pywin32 – (win32all package) – www.sourceforge.net
- pyCreate Package. - http://www.cs.hmc.edu/~dodds/erdos/pyCreate.zip

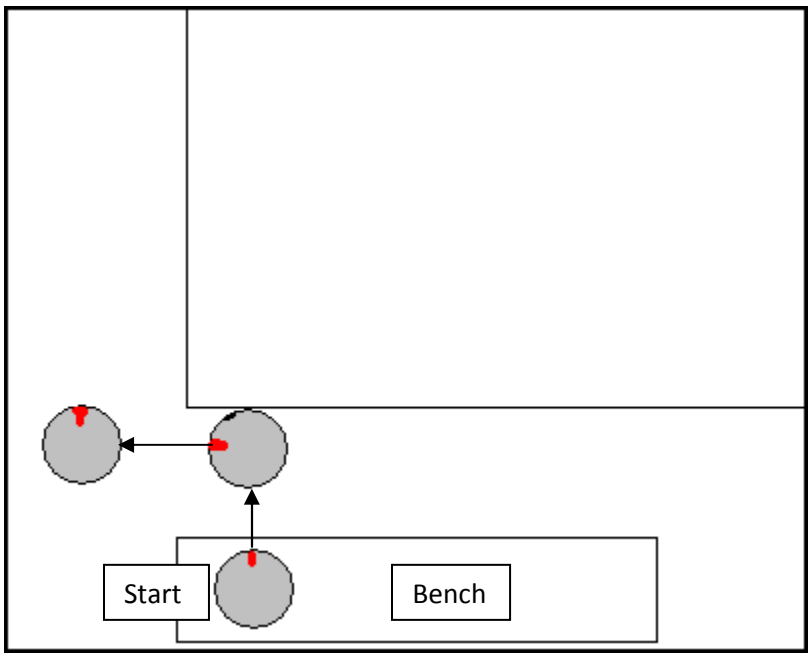
Comparison of Algorithms:

In addition to the Front, left and right bumpers, our algorithm used the Wall sensor and distance as opposed to our colleagues made use of angles, distance and wall sensor. We used the wall sensor to trace wall and used the same sensor to count doors. In addition we used distance in conjunction with the door sensors to make the decision to sopt at Roslyn's or at Dr. Shoaff's office. This ensured fast speed and accuracy, since we had a double check in place to base our decisions on.

As we understand it, our colleagues used the angle's to count the number of doors. This is a good approach for generalized movement, but not the best for the problem put forth. The Robot's sensors were not very accurate and at times there seemed to be a very small variation in the readings for a door being present and being absent. Thus solely relying on the angle's for counting door could prove to be a problem if it missed to read a door. If multiple obstacles cause the front bumper to be detected the angle movement of the robot also could lead to false positives and may cause the robot to loose its localization.
We also thought it would be a good idea to avoid going under the bench as this consumes a lot of time and is not the most efficient. In the robot race scenario it would definitely prove to be a drawback.

Algorithm For iRobot Create Project :

Part 1: Under the bench to Dr. Kaner Office

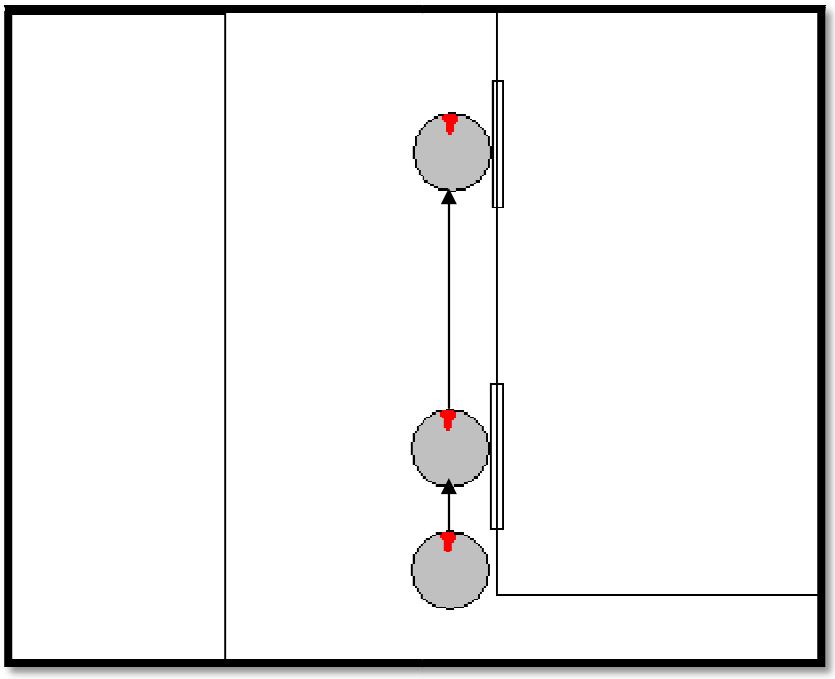


- Move forward till front bumper is sensed to detect the forward wall.
- Turn left 90^0 .
- Move forward distance 800mm = limit.
  a) While distance < limit repeat.
  b) Read Sensors.
  c) Calculate distance, presence of left, right or forward bump.
  d) If left bump then turn right
     If right bump then turn left
     If front bump - Stop, Come back at 10^0angle, move ahead at 5^0angle.
     No bump detected and if the wall present follow the wall.
     Else if wall is missing and distance greater than 500 turn right 90^0.

The iRobot Create has a wall sensor only on the right. Thus the first thing to do is reach the right wall. From our starting position we just need to move forward till the front bump is felt. Once we reach the wall we can simply turn left and trace the wall, till the absence of the wall is felt or the

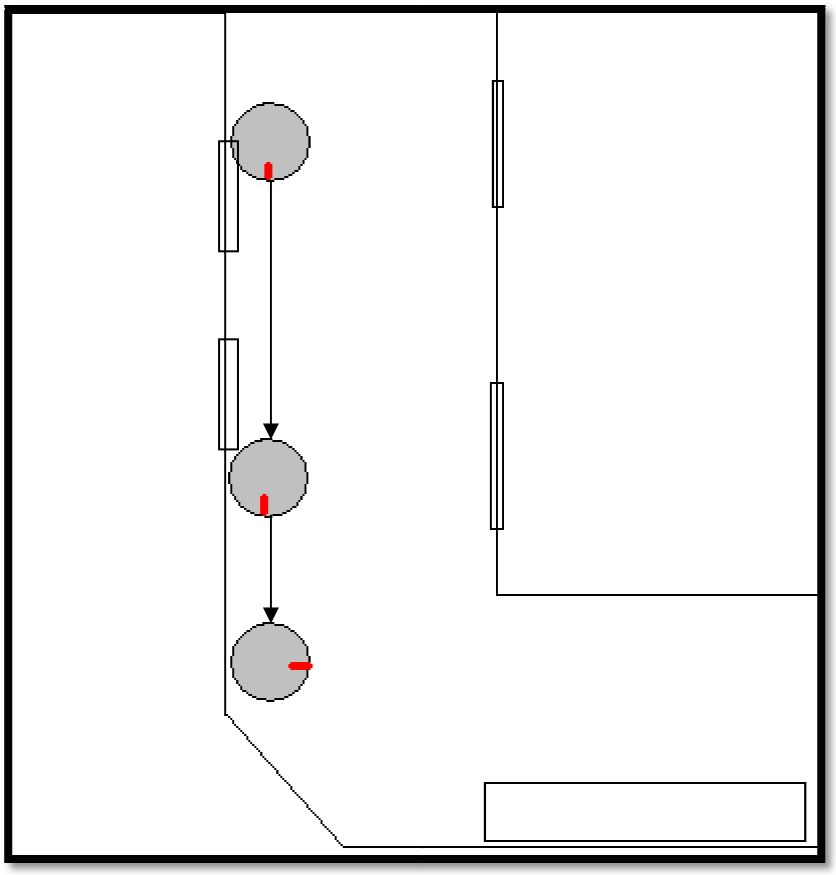
estimated distance is covered. In that case we turn right by 90^0 towards the next corridor. All the while the robot is configured to detect and compensate for the left, right and front bumps.

Part 2: Dr. Kaner to Roslyn Office.



- While distance < (limit = 7750) Repeat.
- Go straight.
- Read sensor.
- Calculate and update distance.
- Detect the presence of Left, Right or Front Bumper.
- If left bump go right 60^0.
  If right bump go left 60^0.
  If front bump Stop, Come back at 10^0 angle, move ahead at 5^0angle.
  No bump detected follow the wall if available.
  If no wall available curve right and go to the wall.
- If a higher value for wall sensor is seen and depending on distance estimate the presence of a door.
- On second door stop, turn right towards the door and move ahead till the front bump and beep for cookies.
- If second door is not read (or missed Due to sensor glitch), Stop on the backup – Which is the prescribed distance and turn into door and beep for cookies.

Part 3: Roslyn to Dr. Kaner

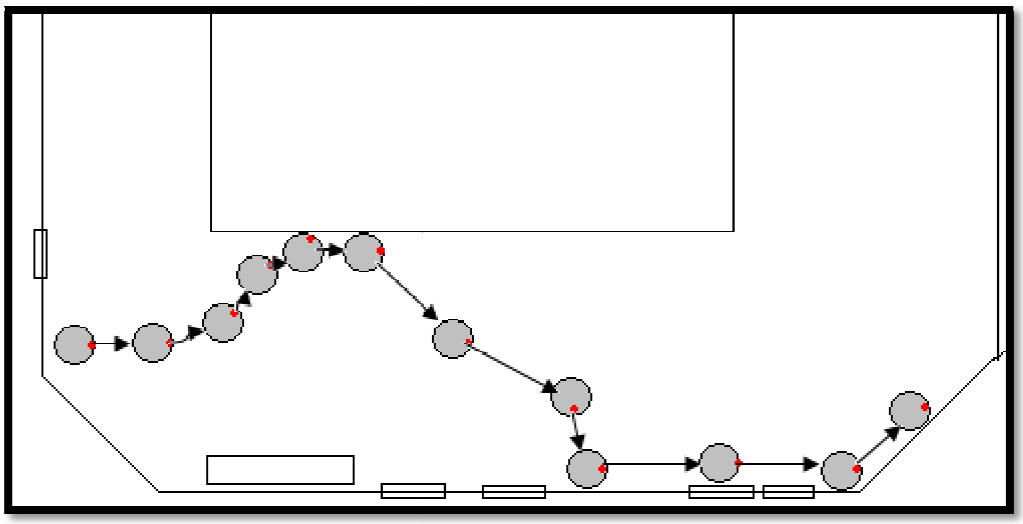


- Turn around 180^0 and go forward to reach the opposite wall i.e. until the front bump is sensed.
- Turn left at 90^0.
- While distance < limit follow the wall if available.
- If wall is not available curve right towards the wall.
- When the prescribed distance is covered turn left at 90^0.

The Robot is then programmed to go ahead into the corridor with slightly curving towards it's right to detect the presence of the wall. Once the presence of the wall is felt, the robot will use it's wall sensor to trace the wall and follow it. On the way the robot will use the wall sensor reading and approximate distances to count the number of doors. Roslyns room is the second door form there. Once it detects the second door, it turns into the door and beeps for cookies. In the event that the robot does not successfully count the door, based on approximate distance from the start corridor till Roslyn's door it will halt and beep for cookies. This corridor does not have any obstacles and has the least number of doors with the robot wall sensor on the same side and thus can be taken advantage of by tuning the robot to traverse the wall at the maximum possible

safest speed where it does not loose it's reading of the wall but can gain considerable time. Thus helps the robot make good time while not losing its location on the map. Similarly to return form Roslyn's cabin till Dr. Kaner's office, the most reliable and fastest will be to trace the wall. But since the sensor is only on the Right, we should move towards the opposite wall. Thus we can again exploit the straight nature of the corridor with no obstacles and trace the wall with the maximum possible safest speed. Here since we need to go ahead till Dr. Shoaff's office and if we continue to trace the wall, the bench creates an obstacle. Since it will be inefficient to traverse through the bench, it is advisable to avoid the bench in its entirety. Thus we use the approximate distance to the corridor and take a left turn at 90degrees.

Part 4: Dr. Kaner To Dr. Shoaff Office.



- A] Till Bench.
  a) Till bench.
  While distance > limit =1350.
  Go straight.
  Check for bumper and adjust direction accordingly.

  In this step the robot simply travels straight at a relatively slow speed to ensure that is does not deviate in its path sue to higher speed.

- B] Bench to wall.
  a) While distance > 1350 && < 5000.
  Turn left towards the wall.
  On left bump turn right 90^0.

Curve back into the wall.

Once we have reached the distance till the beanch, we need to ensure that the robot does not come in contact with the bench, thus we curve left to find the left wall. The iRobot Create does not have a left wall sensor. Thus we gradually curve into the left wall and compensate the direction on the left bump, to ensure that we trace the left wall without the availability of the sensor. This is somewhat slow, but it is a better and efficient approach than navigating from under the bench.

- C] From left wall to right wall from distance 5000 to 10000.
  Gradually curve towards the right wall to find the right wall.
  Scan for and compensate for bumper values as detected.

  Once we have cleared the Bench area. The remainder of the corridor does not have any obstacles. Thus the faster approach will be to trace the wall. Thus we gradually turn towards the right to find the right wall of the corridor. We however try and avoid the first two doors in that corridor, since doors will reduce the speed.

- D] From right wall to Dr. Shoaff's office. distance > 10000
  a) Follow the right wall using the wall sensor.
  b) If wall not available curve into the right wall to find the right wall.
  c) If distance > 17000 and detect a door stop and beep for delivery of cookies.
  d) Detect and compensate for bumper as required.
  e) If door not detected stop on the back-up (distance) and beep.

Once we find the right wall, we use the wall tracing and thus traverse the wall at a faster pace. Here we use a combination of distance and wall sensor to detect Dr. Shoaff's door to deliver the cookies. If the door is not detected the robot will stop in the approximate distance to the door.