

LAB #2 CONTROLLING THE IROBOT CREATE

OBJECTIVE

- Get familiar with the iRobot Create mobile robot platform;
- Learn how to collect data from the Create and display them in MATLAB;
- Learn how to control the motion of a Create;

INTRODUCTION

I. iRobot Create

An iRobot Create is a Roomba robotic vacuum cleaner without the vacuum part. It is a fully autonomous mobile robot on its own. Details about how a Roomba (or a Create) works can be found in the following link: <http://electronics.howstuffworks.com/gadgets/home/robotic-vacuum.htm>.

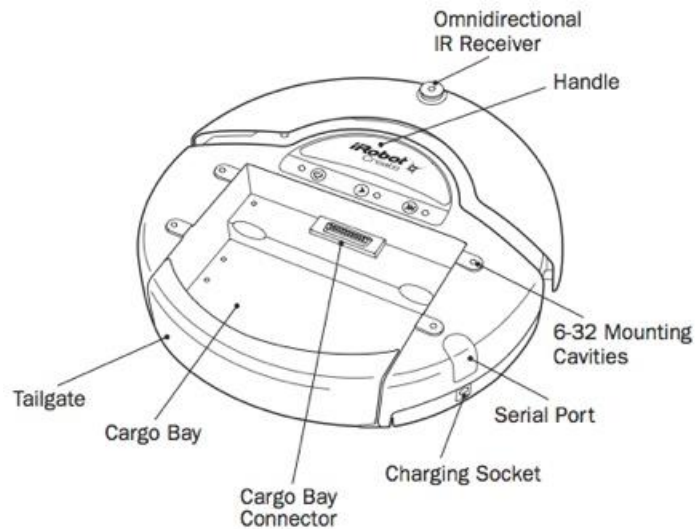


Fig.1. Layout of the iRobot Create

II. Create Open Interface

The Create Open Interface include a physical electronic interface and a set of communication protocols that allows you to read sensor outputs from the Create and to control its behavior. Details about the Create Open Interface can be found in the following link: http://www.irobot.com/filelibrary/pdfs/hrd/create/Create%20Open%20Interface_v2.pdf.

III. MATLAB Toolbox for the Create

Directly programing the Create using the communication protocols described in the Create Open Interface can be confusing sometimes because all commands are based on a series of numbers. To solve this issue, researchers at the US Naval Academy have developed a highly intuitive MATLAB toolbox for the iRobot Create. Basically they helped us to do what we did in Lab #1. With this toolbox, you can acquire sensor data and control the Create through calling different MATLAB functions. The toolbox and

its description can be downloaded from the following link:
<http://www.usna.edu/Users/weapsys/esposito/roomba.matlab/>

IV. The SMART Robot

During the experiment, you will be provided with a SMART (Shared Mobile Autonomous Robot for Teaching) robot which consist of a Create, a small laptop, a sensor interface board, a Microsoft Kinect, a set of 6 ultrasound range finders, and a 3D printed robot structure. The Create is connected with the laptop with a serial interface and the the MATLAB toolbox for the Create is pre-installed in the laptop.



Fig.2. The WVU SMART Robot

V. Warnings

1. Do not keep the Create power on unattended. The robot may drive itself off the table. Only turn on the robot power when you are ready for a test and turn it off right away when you are done with the test.
2. Flip the laptop up when you are driving the robot. Do not use the laptop as a bumper!
3. If your laptop needs an update when restarting, let it finish. Do not turn off the laptop in the middle of an update.

PROCEDURE

I. Initialize a Create and Collect Sensor Data in MATLAB

1. Initialize the Create with the 'RoombaInit' function provided in the Create toolbox;
2. Use the 'AllSensorsReadRoomba' function to read sensor data from the Create;
3. Locate the 'bump', and 'cliff' sensors on the robot and show how you can change the output values;
4. Log the voltage and current data for 200 steps and plot them in MATLAB;
5. Properly close the serial port with the 'delete' and 'clear' functions;
6. Show the instructor your results so far.

II. Control the Create

1. Turn on and off the LED with the 'SetLEDsRoomba' function;
2. Make a beeping sound with the 'BeepRoomba' function;
3. Program the robot to drive a 1m by 1m square and return to the original position and heading angle use the 'travelDist' and 'turnAngle' functions;
4. Program the robot to drive the same square path use the 'SetDriveWheelsCreate' function. **Hint:** you will need to check the robot travel distance and angle outputs constantly to decide what action to take next;
5. Implement some safety pre-cautions to the previous program so that the wheel motors will be turned off if the robot bumps into an object or any of the cliff sensors is activated;
6. Run a full test and plot the robot distance, angle, voltage, and current data over time;
7. Perform multiple runs with the same program and see if the results are always repeatable;
8. Perform more tests with different robot wheel speed settings;
7. Show the instructor your results so far.

DELIVERABLE

An abbreviated lab report of your experiments and answer the following questions:

1. How did the Create measure the distance and angle?
2. Does the robot always come back to the original position and heading angle? If not how much error was there?
3. Does the error have anything to do with the robot wheel speed? If so, explain why?
4. With the data you collected from the last set of tests, can you plot the robot x and y positions? Does it show a square? Does the result match the visual observation?
5. What is the typical power consumption of the robot?