CSCE 274 Section 2 Fall 2018 – Project 1 – John Esco, Adam Kenvin, Changxuan Yao

Description:

Our program consists of three .py files. The first file is the connection interface. This contains the basic functions used to communicate with the iCreate 2. It can write and read as well as open and close the connection. The second file is the interface.py. This contains most of the base functionality used in the project file. The interface is used to set the robot's mode, drive, and read data. Some of the data reading functions include: reading button presses, bump and wheel drop sensors, cliff, angle, distance reads, and reading the six light bump sensors. Added functionality for project four includes reading data from the dock using the IRCharOmni() method. The third file is project4.py. This is the main file for wall following and dock finding/navigating, and it imports both the connection interface and the robot interface. It first uses the connection interface to close the connection and then open it to establish a connection to iCreate 2. Then, it uses the robot interface to start iCreate 2 and set it to passive and full mode. After that, there is an infinite while loop that breaks when button is press, and thus proceed to performing the drive. The file includes the declaration of multiple gains. The gains include a kp, kd, setpoint, and a wallGain. The wallGain is essentially the distance in which a wall is detected that initiates a turn. The wallGain is compared to the front left light bump to determine when a wall is being approached so that a counter-clockwise turn can be initiated. For the main wall following behavior, the program checks the value of the right light bump sensor. This is compared to the setpoint and an error is found. The error is then used in combination with a PD controler to directly affect the speed. If the robot is too close to the wall, then the right wheel is sped up to increase the distance. If the robot is too far from a wall, then the left wheel is sped up to decrease the distance. During this process, if the iCreate 2 senses a dock with the IRCharOmni() method, it will first turn 90 degrees counterclockwise towards the dock, and enter a loop in which it carries out specific actions required for it to reach the dock. In this while loop, the robot will turn clockwise, turn counterclockwise, or drive straight depending on the dock sensor data. Once it reaches the dock (bumpers return true or charging sensors return true), it will stop, play a song, and exit the program.

Evaluation:

Our Wall following behavior effectively uses a PD controller to steer the robot to maintain a setpoint with the right light bump sensor value of 250. When it encounters a corner with a wall in front of it, it is able to turn and continue follow the wall. When it encounters a corner without the wall it turns right around the corner so that it is once again able to follow the wall. There is minor oscillation while running and at some angles the robot will bump into the wall. However, bump sensors have been accounted for and will slightly turn in this case so the robot can return to wall following. This behavior starts and stops on the clean button press. To obtain our gains we used trial and error. First we obtained the sensor value and adjusted the gains until the output would not exceed the range for velocity. Our dock tracking codes work decently well, and will result in iCreate 2 reaching the dock every run. However, it is not consistently charging when it reaches the dock.

Allocation of effort:

The project was worked on evenly by all group members. Some key contributions include:
John Esco: Added omni character reading to interface. Helped write docking code
Adam Kenvin: Helped with report and tuning the docking code
Changxuan Yao: Wrote the report and helped write the docking code