Cameron Ray

Phillip Voss

Yilei Shen

Zachary Kasper

# Code Overview

Our reactive approach followed along the lines of the subsumption architecture of Brooks. It did this, in that our code has layers between which global variables communicate with one another in order to process and react to data. One example of this is in the use of the bumper node to control the escape behavior. If the bumper node changes its state to PRESSED, on any of the three bumpers, this data is then used to command the motor to back the robot away from the object. According to the data the robot will move smartly in another direction and continue its execution of autonomous movement.

As far as the code is concerned, the layout was fairly simple. The main() method subscribed to each of the necessary components, as well as set up the required publisher. Each of the subscribers calls a callback method which documents the necessary information. The bumper subscriber simply sets a Boolean according to the bump data. The scan callback takes the array of 640 items and converts the array index to degrees. The scan callback also adds up the total amount of items that are either less than .45 or NaN. The lowest the laser scanner can reliably go is .45 meters, which is approximately 1.5 feet. The callback will then determine whether there are more obstacles on the right or left side and determine which way to turn accordingly. The subsumption architecture comes into play with the layers between which the global variables communicate. At the heart of this method is a conditional statement. The conditional goes through the hierarchy of the required items, starting with the bumper. When the bumper is pressed, the entire node is ended because the first item in the hierarchy is stopping the robot when the bumper is hit. The user input is handled automatically, so we don't need to initiate it, just stop the other layers from activating if there is player input. If neither of those things are active, the robot will check if it needs to avoid an obstacle. If the obstacle is symmetric (i.e. the robot is getting stuck), the robot will turn completely around to escape. If none of the three previous layers are activated, the robot will then start executing the autonomous driving, which consists of moving forward for one second then rotation at a random angle between -15 and 15 degrees. This loop continues until either the bumper is hit, or the node is stopped by the user.