CPE 470/670 Autonomous Mobile Robots

Lab 2 Corner Escape

Team 6

Omid Tutakhil,
Sandeep Mathew,
Jin Jiang

Hardware and Software

We used the Lego Mindstorms NXT kit for building our own robot. The kit came up with instructions on building a two wheeled robot. However since the problem we had to solve was a that of corner escape on a flat surface we decided the it would be best to use an alternate design and we also wanted more stability than a two wheel design. Our design was inspired by castor bot present in lego nxtprograms website. We noticed that castor bot has 3 wheels and the third wheel created more friction when the robot takes a turn. Therefore we came up with a modified version of castor bot with four wheels. We found the instructions for building the castor bot here : http://www.nxtprograms.com/castor_bot/steps.html . Our sensor design was inspired by the earlier handy board sensor design. We installed the touch sensors in the front of the robot similar to the earlier handy board design.

We used the brix cc IDE for building and testing our programs. We used the concept of metasensing and randomness to make the robot to take a random turn if there has been many bumps in a given time frame. We built the capability of metasensing and randomness to the obstacle avoidance program developed in the previous lab. Our program in the previous lab was designed to be modular and data driven so we had to make very few changes to the program when we had to replace the old robot. Because of the data driven approach of our program we were able to quickly test our program with different values of motor speed and wait times. We found out that if the robot got stuck in a very sharp corner, the textbook algorithm does not work very well. Therefore we wanted to add meta sensing capability to the random turn behaviour. However if we take more random turns we are going to get negative points, so we decided to use only a simple counter. Therefore if there are three random turns in a row we make our robot go backward for a much longer time so that we escape out of the sharp corner.

Problems and Solutions

During the process of designing and programming the robot the team encountered several problems that had to be overcome. During the building phase of the robot the team didn't have any major issues, the only issue that the team had was coming up with the design of the robot. After exploring the premade designs from the book inside of the kit the team came to the conclusion that the robots in the book were not going to be adequate for what the team had in mind. After looking on the internet the team decided to go with a modified version of the castor bot. After the initial build of the castor bot the only major modification was adding the bumpers to the robot.

The main problems that were encountered came from actually programming and testing the robot. Problems were the the bump counter, spacial reasoning, and the speed of the bot. The issue of the bump counter was coming up with a good time length that the robot would hold onto the number of bumps before resetting the counter to zero. Originally the timer was set to 4 seconds after a few trials it was found out that holding on to the number of bumps for more than 4 seconds was too long and would cause the robot to turn at incorrect times. After a few tests the team decided to use a timer of 1.5 seconds so that if the robot was stuck in a corner it would hit within the appropriate time and execute the corner escape.

The second issue of the spacial reasoning was caused by the fact that if the robot had issues with escaping the corner that it was stuck in, say when it tried to turn it would hit another wall during the escape, another counter would be incremented called the random counter, which counted the number of escapes it tried to do within 2 seconds. If the counter reached 3 within the two second time limit the robot will reverse for a longer amount of time and then do a random turn.

The third issues was the speed of the robot. If the robot was traveling at full speed the robot would make run into objects faster and the turns that the robot would make where to sharp. If the robot speed was too slow it would make very small turns and the hit counter could reset trapping us in the corner for a long time. To fix this issue the team experimented with different speeds until the speed was good enough to make it to the end of the path and back in a timely manner but at the same time not make any unnecessary random turns.

Unsolved problems

The touch sensor provided by lego nxt kit can only sense effectively if the obstacle hits the sensor heads on. Our robot will not work in cases where the obstacle is too short and makes an angle with the touch sensor. In such scenarios the robot may get stuck.