

Team Five: Erin Keith & Qandeel Sajid
Lab 4
Date: 10/8/12

Introduction

The goal of this lab is to design a robot for a harvesting contest. There are two parts to this contest. The first part is food collecting in which the robot wanders around a designated area counting the distributed food. The collecting occurs for sixty seconds and then the robot starts searching for home. Robots are judged based on how much food that can collect and how long it takes to find home.

Description

The food pieces the robot is meant to collect are RF transponders, and color pieces of paper. Each group was provided with a RF ID Sensor and a Color Sensor. They were also provided with two ultrasonic sensors to prevent the robot from falling off the edge of the area.

The robot was designed with RF ID Sensor half an inch above the ground. This was due to the sensor's inability to sense the transponder more than half an inch away. The Color sensor was placed about two inches away from the ground. Both of these sensors were placed in the back of the robot. The two ultrasonic sensors were placed in the front of the robot as the "eyes" of the robot. Both of the angles were placed about thirty degrees from the center facing opposite direction.

The program of the robot was designed so the robot can easily switch between the food collecting and home searching phase. To do this the group used a Boolean that would only be activated as soon as sixty seconds have passed. When the sixty seconds have passed, the Boolean causes the robot to stop searching for transponders and colored paper and start searching for home with a higher speed.

Difficulties Encountered

Some of the difficulties in this program dealt with using the sensors, learning how to use the parallelism function in NXC, placing the "eyes" of the robot correctly and forming the good searching algorithm. The sensors were a bit difficult to work with since they needed to be placed at the correct angle and distance to be able to sense correctly. The RF Sensor could only sense the transponder when it is directly above it and the maximum distance between the sensor and the transponder could only be about half an inch.

The problem with the Colored Sensor dealt having to test with the distance and the angle of the sensor. For example, if the sensor were placed too close to the ground, the sensor's light would cause the robot to confuse black with white. The ability to sense black was important for the robot to correctly find its home.

The last problem with the robot occurred with using the penalization function in NXC. Originally, the group believed they could run the processes for food collecting in parallel, stop it, and then run the processes for home searching in parallel. However, the way the group assumed the `stopalltasks()` function worked was incorrect since it caused all processes to stop including the main. To deal with this the students started to use a Boolean to switch between the food collecting and the home searching.

The robot's eye needed to be placed at the correct level so it could detect the lower dark blue wall of the area without accidentally mistaking the dark blue as black and considering the wall its home. Originally, the robot had its eyes placed in the center, both pointing forward. This caused the robot to be glitchy in the corners since it could not detect the wall needed to be directly ahead for it to sense correctly. To fix this, the eyes were placed at a distance so the robot would not get too close to the walls.

The group had originally thought about using some sort of searching algorithm (like searching from one line to the other in multiple lines) but the uncertainties in the contest proved that this method would be useless. The correct this the group decided to just use randomization.

Discussion of Unsolved Problems

There were no unsolved problems.

Results

Our robot performed better than expected.

Conclusion

In conclusion, testing with the sensors and the robot's speed helped to design it to do as well as it did.

Appendix

See our website for the source code.