DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF NEBRASKA, LINCOLN

Capture The Flag

Steven Brown

Michael Fay

W. Willie Wells

May 7, 2015

CONTENTS

1 INTRODUCTION

Embedded processors are prevalent in modern society throughout the world. Robotics is a field that could not advance without embedded processors and has subsequently advanced the design of embedded systems. Competitions showcasing robots are held worldwide by government, industrial, and academic institutions. One such competition was held at the University of Nebraska, Lincoln. The competition consisted of building a robot that would traverse a modified capture the flag-type environment faster than the opposing robot. Electrical tape was placed down the middle of a white board as a line following option for the competing robots. Wooden walls that were at least 3 inches high were placed on the edges of the white board such that they allowed narrow entrances on either side. Two of these environments were placed end to end. In addition to these features, each team drew a card with a random placement of a block. A wooden block was then placed in the indicated area on that teams side of the course. Each team consisted of two or three members. The design we employed involved using simple code with minimal states while using line following. The robot design had a weight limit of 500g and a dimension limit of 22 by 14 by 14 centimeters. Additional component limits included: 4 large servos, 1 small servo, 2 bump sensors, 4 infrared sensors and 1 range finder. Our robot's name is 2CPO. There is a featured video with 2CPO as well as a picture, Figure 7.1. The next section describes both the software and hardware design process that produced 2CPO. Characterization of the sensors used follows that section. Section 4 includes an evaluation of our performance. The fifth section contains other related information. Project conclusions are next, which is followed by all referenced figures.

2 TECHNICAL DESIGN

In order to successfully complete the challenge one needs to quickly and accurately detect objects and react appropriately to objects in the robots field of view. The two primary design components of the robot include physical and software construction and analysis.

2.1 Software Design

Throughout the competition we underwent multiple designs and strategies to detect, move, and relocate the center line. We defined a successful mission or course run to have the following basic requirements meant: Object detection, avoidance, and line relocation. This definition of success directly dictated the design approach of each software revision.

2.1.1 Initial Software approach

The original design of the software was to use the range finder to locate individual objects in the field of view of the robot. After a object was detected we would then track the object(s) through the frame ensuring that object position, count and direction would never been unknown after initial configuration and detection. While the object was being tracked, our software would allow for relatively basic Simultaneous Mapping and Localization of our robot.

2.1.2 Revision 2.0

Prior to actually implementing the above software design, a team member mentioned a simple logic flow that did not use the range finder. While the team went on break, a team member modified the existing block avoidance code from checkpoint 2 to support this design. The logic involved turning one direction for amount of time and then the opposite direction if the respective side's bumper sensor was hit with the direction's reversed for each bumper sensors. The bumper sensors were to be connected as interrupts.

2.1.3 Final Revision

During the next day's testing at some point this design was thrown to the wind for some unknown reason and 2CPO was thus crippled software wise but was able to finish the course during that day's round of testing. Only one bumper sensor was used in the software, the reason for this still baffles the uninformed.

2.1.4 Software Design Limitation

The software was designed to be minimal to minimize processing time of 2CPO but this did not have any benefit for 2CPO during the competition.

2.2 Hardware Design

Throughout this project, one need to way the benefits of power versus agility. As our group found out through testing, more servos does equal more power however, the cost of such upgrades

2.2.1 Design

A four wheel design was pursued at the outset of this assignment. The four wheel design was unable to turn that well and thus would not be able to avoid obstacles. Front wheel drive was then attempted but soon abandoned. Two bumper sensors were placed perpendicular to the direction of motion. Tongue depressors and electrical tape were used to mount the bumper sensors. Infrared sensors were mounted near the front rolling ball and below the bump sensor mounting support. The infrared sensors were used for line following.

2.2.2 Design flaws

The competition robot was too light and was thus pushed around during the first match. Prior to the second match a bulky, rubber guarded cell phone was taped to the bottom of 2CPO in the hope that the added weight would help in being able to push opponents around. If the bumper sensors had been mounted at a slight angle, 2CPO's performance may have been better. In comparison to the rest of the robot's at the competition, 2CPO was the longest robot present, the least compact. This was due to the protruding bumper sensors and rear mounted servos. At some point in the testing the software was modified making one of the bumper sensors sterile.

3 Sensor Characteristics

For the project to successfully be completed one needs to fully understand and numerical characterize each sensor. This is required to get the most accurate sensor reads and movements.

3.1 Infrared Sensor

The two infrared sensors used had a threshold value of 750. This value was used in the code to facilitate line following. The infrared performed well during the competition. They stayed where they were mounted and resulted in zero complications. Actual characterization of these sensors has not changed since project one. As a result, only the diagrams are included. For more information on these characterizations please refer to project one report.

3.2 Proximity and Ambient Light Sensor

The frequency the range finder was read at for the checkpoint was 781.25 kHz. The range finder was not used in the competition. Tests for interactions or interference between the Range Finder sensor and three other conditions: another Range finder sensor, an IR line following sensor and changes in ambient light. The range finder to be tested was configured using the settings in the provided figure and held approximately 6 cm from a test block. A second range finding sensor was then configured on a second Arduino and placed against the test block directed back at the main range finding sensor. Data was continuously captured during this process and no noticeable changes in reading were observed. This first process led to the conclusion that there is no noticeable interaction between two range finding sensors using the settings shown. The same experiment was run using an IR line following sensor and no noticeable interactions were observed with that type sensor either. Finally, the range finding sensor was tested for ambient light interactions by using a similar process. The range finding sensor was configured using the settings in the provided figure, with a test block approximately 6 cm away with both distance and ambient light reading being recorded continuously through

the serial print functionality. In general, the recorded distance did increase the readings by approximately 150 when the testing environment was shaded from any ambient light. The decision to not incorporate the range finding sensor in the final robot design was influences by the findings of ambient light portion of the testing. The figures provided illustrate the testing setup in addition to the readings obtained in the testing.

3.3 Bump Sensor Sensor

The bumper sensors were connected to interrupt pins and served to help implement obstacle avoidance or more accurately navigation around encountered obstacles. Boolean global variables were used to indicate that an interrupt had happened. The inactivation of one of the bumper sensors provided endless frustration and assisted with poor competition performance. Their is not a reason for its deactivation.

4 Performance Evaluation

The performance of 2CPO the Sunday before the competition was unexpected. When 2CPO's performance was brought to an expected level, testing that day was discontinued until the day of the competition. On competition day 2CPO did not perform as expected and was eliminated in the first match (by the team who would eventually win the entire competition). Weight difference was an issue for 2CPO. The opposing robot was able to push 2CPO around. The bumper sensors did not respond when 2CPO was forced into corners. A single flag was all that 2CPO was able to capture before losing the match. If 2CPO's weight had been closer to the weight limit, robot fights would have easily gone in 2CPO's favor. A sleeker design would have enabled 2CPO to perform tighter turns and speed up overall progress. A wheel fell off during the second competition day's match during the round in which 2CPO captured its only flag.

5 Conclusion

A properly weighted robot with a weight near the limit had a higher probability of winning confrontations but if the two competing robots never met the faster robot would capture the opponents flag first. The use of a range finder did not significantly improve or hinder a robot's chance of capturing the opponents flag first. Using bump sensors and simple turning logic without the use of infrared sensors was another option that resulted in plenty of captured flags. Robots including 2CPO often caught the edges of blocks or the edges of the central door (gate). Not enough consideration was put into this potential event. One may design a beautiful algorithm but then when tested in a real application it may fail. Testing is crucial to all design processes. The first design in most cases is not the same as the end product for the majority of products developed in the fields of Electrical Engineering, Computer Engineering, and Computer Science. Final and first designs may even vary considerably. Though one attempts to account for all cases, unaccounted for events may occur during testing.

6 Figures

The following pages include figures of sensors and robot construction.

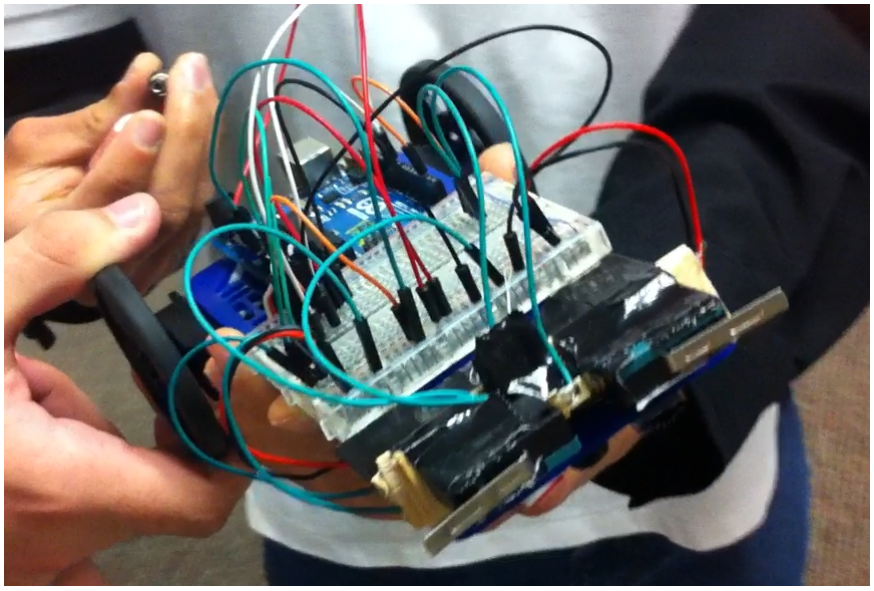

Figure 6.1: Competition 2CPO

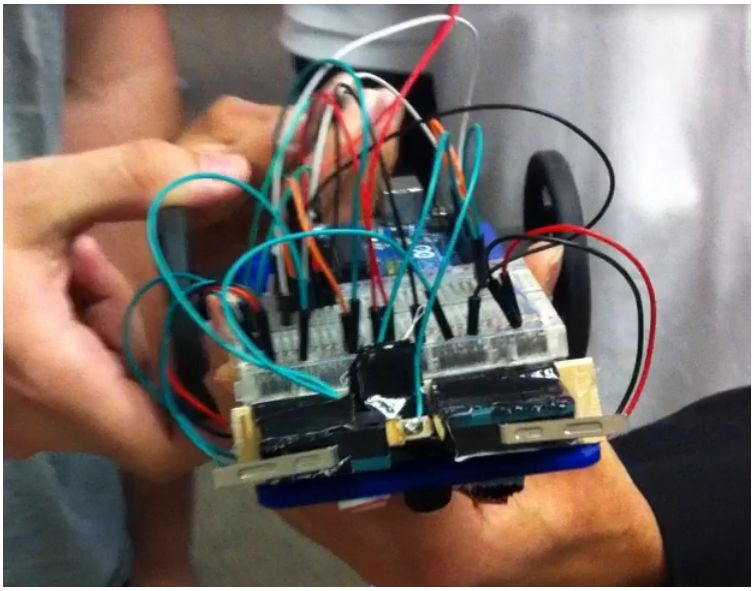

Figure 6.2: Competition 2CPO 2nd View

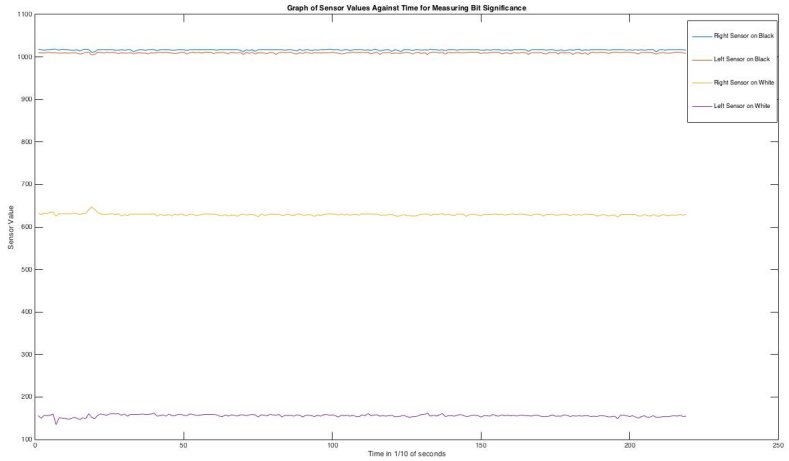

Figure 6.3: Sensor Values vs Time

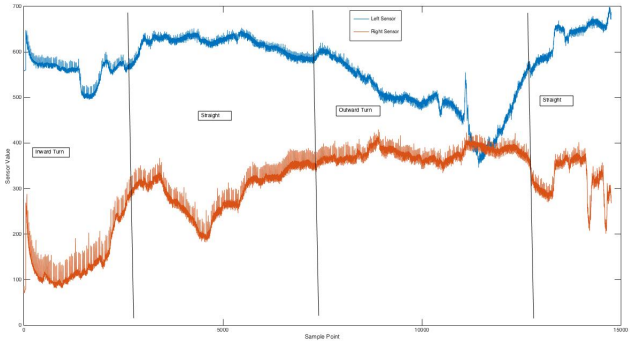

Figure 6.4: Sensor Values When Following a Line

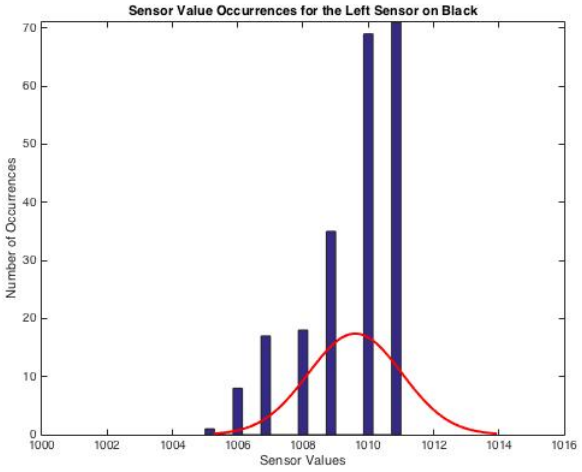

Figure 6.5: Left Sensor Over Black Value Distribution

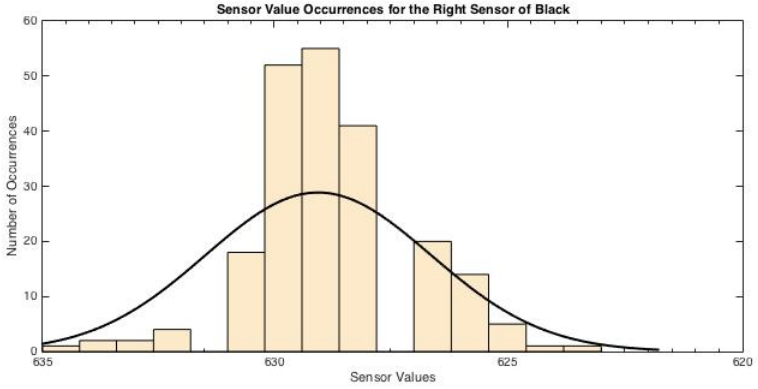

Figure 6.6: Right Sensor Over Black Value Distribution

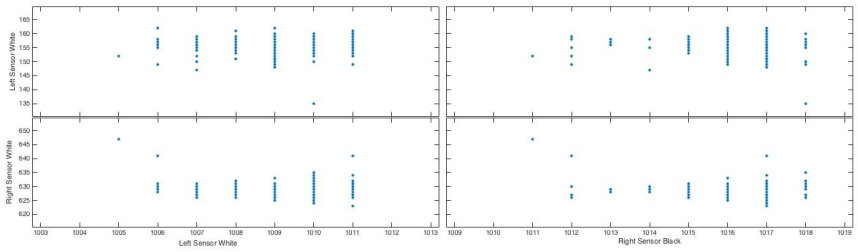

Figure 6.7: Sensor Value Distribution

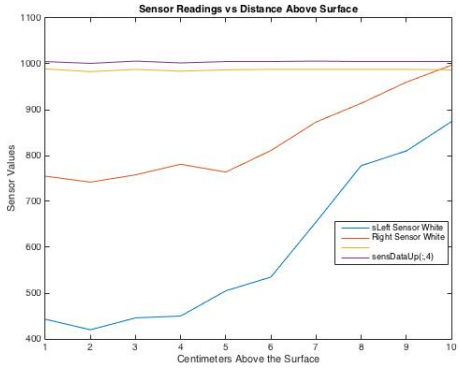

Figure 6.8: Sensor Data While Increasing Distance

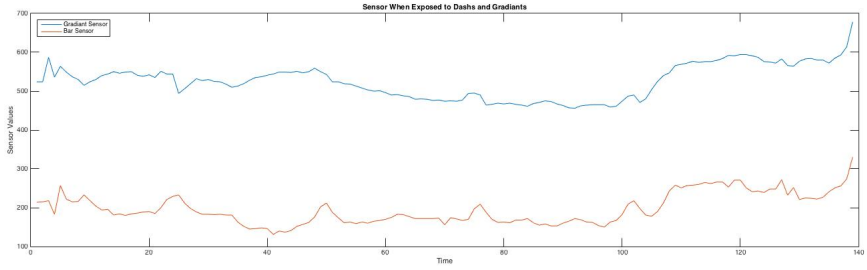

Figure 6.9: Sensor Data Over Varying Surfaces

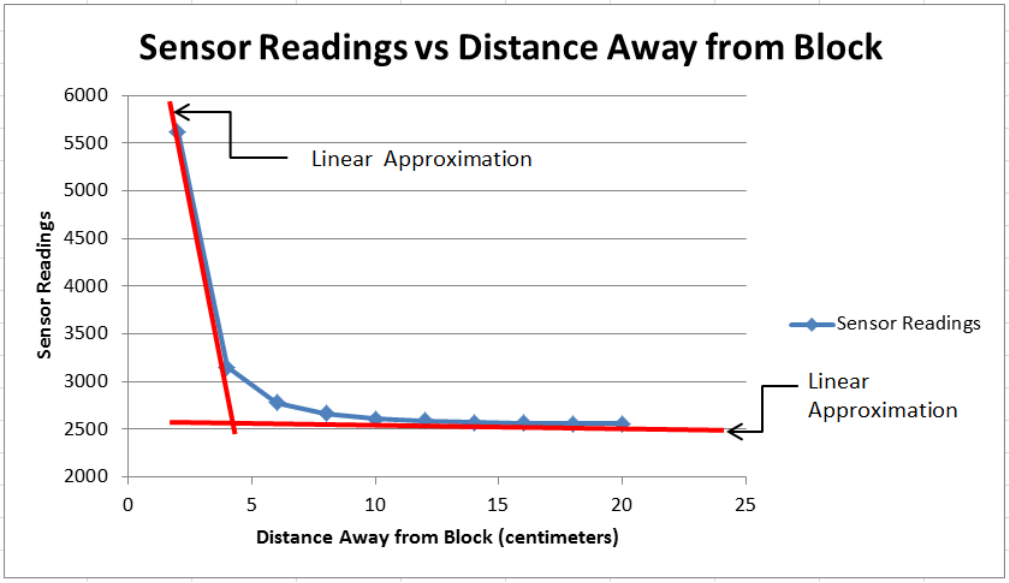

Figure 6.10: Range Sensor Distance Approximation

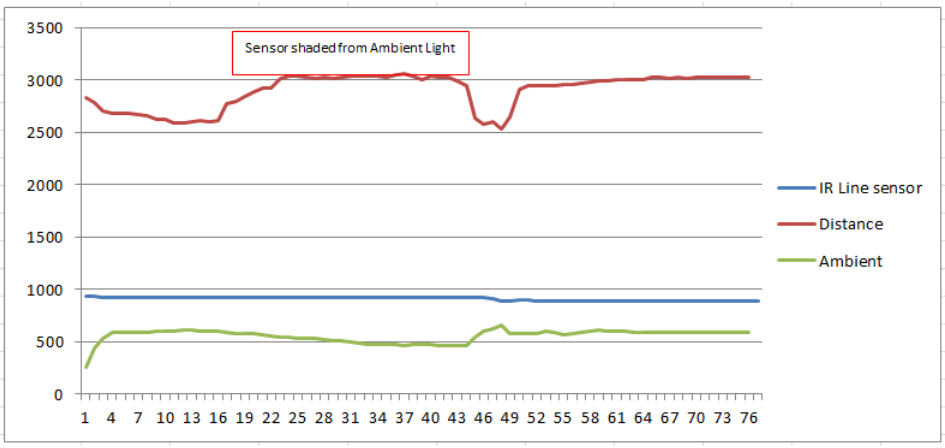

Figure 6.11: Range Sensor External Noise

Ranger Sensor Configuration Code

```
#include <Wire.h>
#define VCNL4000_ADDRESS 0x13  // 0x26 write, 0x27 read
// VCNL4000 Register Map
#define COMMAND_0 0x80  // starts measurements, relays data ready info
#define PRODUCT_ID 0x81  // product ID/revision ID, should read 0x11
#define IR_CURRENT 0x83  // sets IR current in steps of 10mA 0-200mA
#define AMBIENT_PARAMETER 0x84  // Configures ambient light measures
#define AMBIENT_RESULT_MSB 0x85  // high byte of ambient light measure
#define AMBIENT_RESULT_LSB 0x86  // low byte of ambient light measure
#define PROXIMITY_RESULT_MSB 0x87  // High byte of proximity measure
#define PROXIMITY_RESULT_LSB 0x88  // low byte of proximity measure
#define PROXIMITY_FREQ 0x89  // Proximity IR test signal frequency, 0-3// 781.25 kHz
#define PROXIMITY_MOD 0x8A  // proximity modulator timing
```

Figure 6.12: Initialization of Range Sensor

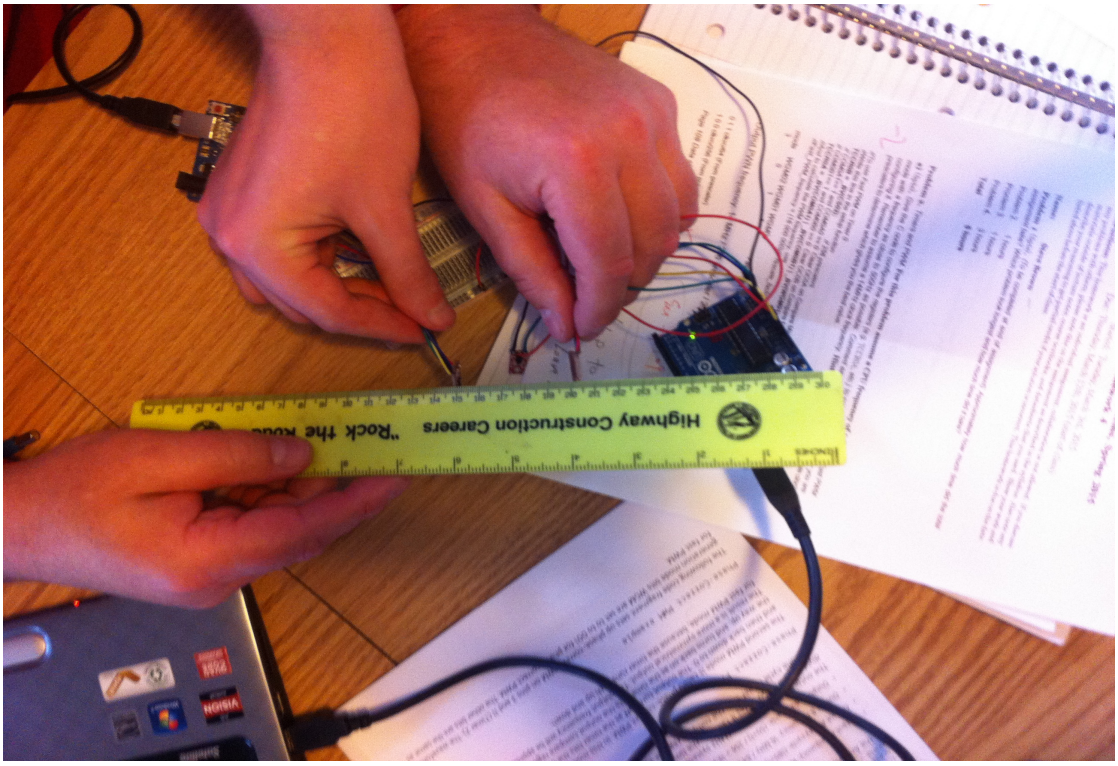

Figure 6.13: Testing Process for range sensor