**XFM Vending Machine**

Joseph Olin  
Michael Roark  
Branden Wagner

C458 Project Paper

May 2, 2017

### Abstract

This project designs and implements a robotic system that acts as a bar waiter. We use a robotic arm mounted on a mobile robot to bring consumables to customers who are sitting at the bar counter. The robot decides the delivery of consumables based on a color-coded card that each customer presents to the robot. We accomplish a robot control software architecture utilizing Arduino controlled sensors and effectors. Our project demonstrates how a familiar task, such as tending bar, can be handled by a low cost intelligent robot.

### Introduction

Attending customers at a counter is a repetitive task. First, the waiter/waitress must ask the customer if they desire a consumable. Then, the waiter must retrieve the consumable from where it is stored. Once the consumable is retrieved, the waiter makes a pass over the length of the counter, giving customers the order they desired. One notable model for attending customers can be found at Brazeiros in Louisville, Kentucky.

At Brazeiros, customers are given a color-coded coaster, with one side being red and other side being green. Based on the color of the coaster the customer shows, the customer may or may not be continually served freshly prepared meat. Our project proposes to take the repetitive task of attending customers at a counter and make that task the responsibility of an intelligent robot rather than a human.

The idea is a human will load the robot’s “magazine” with consumables (such as mint patties). Customers will each have a double-sided card with a certain color (e.g. blue) on one side and a different color (e.g. green) on the other side. Similar to how other restaurants such as Brazeiros operate, one color will indicate the customer wants an order and the other color will indicate the customer is content and does not desire an order.

The robot will make cycles through the counter, ignoring unoccupied spaces and customers who use the color-coded card to indicate they don’t desire an order. The robot’s goal will be to deliver items only to customers who desire an order as quickly as possible via the attached arm.

# System Details

### Environment

The environment consists solely of the counter, which is simulated with a long and thin table. There is a line for the robot to follow and there are cards for the customers to mark their order and red cards to mark the ends of the path where the robot will reverse direction.

The storage and loading area is on the robot’s deck. Sitting there is a plastic storage magazine, which will contain up to eight mint patties.

The bot travel area is the space on the counter where the bot moves. The space stretches from one end of the counter to the other end. The bot will move forward from the far end of the counter following a predetermined, taped path. As the bot follows the path it reads the green color cards the customers have laid on the counter alongside the tape, dropping off mint patties in the appropriate places. Once the bot reaches a red card it traverses back over the taped path. We also used blue cards to serve as false orders which the robot was intended to ignore.

The customer area is the area closer to the edge of the counter where customers will display their color-coded card to the bot, and where the bot will also place the customer’s order. Due to the scope and time constraints of this project a static number of six customers was chosen.

### Sensors

The sensors needed for the bot to perform the aforementioned tasks are: one IR reflectance sensor and a Red Green Blue (RGB) color sensor.

The Infrared (IR) reflectance sensor is for 1) allowing the bot to follow the taped navigation path and 2) for notifying the bot when it has reached the loading area (where the tape would end).

The color sensor is used to notify the bot what color card the customer is displaying, thus notifying the bot if it needs to give the customer a consumable on the return trip.

### Performance Matrices

Three primary performance metrics we used in this project are: the number of color cards the bot accurately detects, the number of successful deliveries the arm makes once it knows a delivery needs to be made, and the speed at which the arm delivers consumables.

**Observational Data of Robot Performance According to Performance Matrices**

**2 Green Cards, 1 Blue Card**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Passes** | **Missed color readings** | **Missed arm deliveries** | **Total color readings** | **Total arm deliveries** | **Color reading fail rate** | **Arm delivery fail rate** |
| 30 | 5 | 5 | 137 | 68 | 3.7% | 7.3% |

**4 Green Cards, 2 Blue Cards (3 sleeve failures)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Passes** | **Missed color readings** | **Missed arm deliveries** | **Total color readings** | **Total arm deliveries** | **Color reading fail rate** | **Arm delivery fail rate** |
| 30 | 25 | 29 | 221 | 135 | 11.3% | 21.4% |

It was observed that the robot’s performance degraded significantly in the last several passes, which correlated with the fact that the robot’s servo motors got increasingly warm as it continued to operate. Most noticeably, the arm appeared to drop more consumables as the gripper motor got warmer.

**4 Green Cards, 2 Blue Cards (Color Sensor Only)**

The following data was recorded to demonstrate further optimizations performed upon the color sensor, without any operation of the arm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Passes** | **Missed color readings** | **Missed arm deliveries** | **Total color readings (approximation)** | **Total arm deliveries** | **Color reading fail rate** | **Arm delivery fail rate** |
| 30 | 2 | N/A | 210 | N/A | 0.95% | N/A |

**Arm Delivery Speed Optimization**

Originally, the arm took approximately 14.6 seconds to deliver a consumable. After further breaking down the movements the arm makes, unnecessary movements were cut out of the delivery process. This optimization improved the arm’s delivery time from 14.6 seconds per delivery to approximately 9.3 seconds.

### System Functionalities

* Bot will move forward along the counter, following taped path using the line follower sensor
* Bot will recognize colored cards using the RGB sensor
* Bot will move back along the counter, following the taped path
* Arm will grab and hold on to a consumable
* Arm will unload a consumable onto the order card

### Related Research

Our approach was to define the problem first, then design the robot. As the problem was refined, we also needed to alter the design. Some designs solving a similar problem include Zexuan, et al (2015), and Casavela (2012). Our design intends to combine movement of a wheeled base with the delivery capabilities of an arm into an autonomous delivery system for small solid objects.

We decided to use a color coding system for ordering based on observations of the system in use at Brazeiros. Their system uses a green card to start service and a red card to stop service. (Brazeiros 2017)

Path planning is a difficult obstacle to overcome for autonomous robots. Zexuan, et al, (2015) shows an example the procedures needed to plan transportation along a set path. Our project plans to use a combination of physical aids and programming to overcome obstacles in the path. We intend to use a fixed and relatively small path traversed in two directions.

Ju (2016) explains a strategy for manipulating small round objects using positive air pressure. In this project they found that having a more flexible solution added efficiency and gave unexpected side benefits. We were unable to add a similar system, but were able to make a flexible platform which could be extended with this technology.

Casavela (2012) explains the challenges he overcame in using C++ to program an Arduino. We were able to draw many lessons to design the necessary programs to drive our systems. Casavela’s paper also gave several insights on how to engineer robotic systems and properly interface them.

**Development Timeline**

**Physical Development**

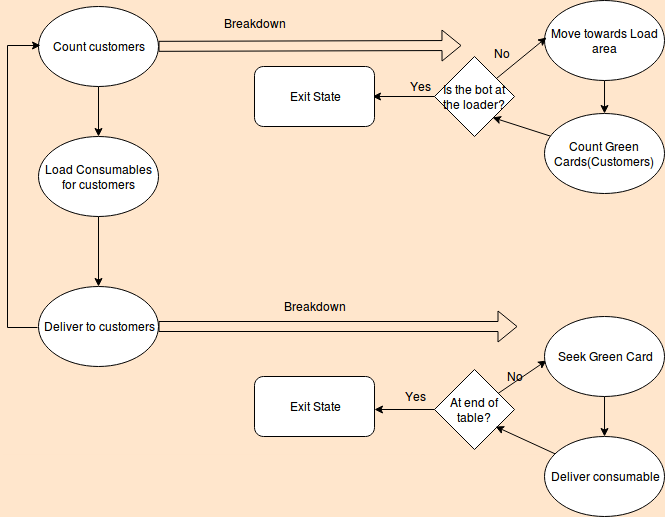
|  |  |  |
| --- | --- | --- |
| **Task** | **Approximate Date Completed (2017)** | **Description** |
| Assemble arm | March 3 | The arm came in a kit and had to be assembled |
| Test wheel motors | March 14 | The robot motors had to be tested to ensure all were working properly |
| Setup rover platform | March 14 | A sturdier, plywood platform was required on top of the chassis so the arm could be mounted |
| Mount arm on platform | March 14 | The arm was secured with screws to the wood platform |
| Wire everything together | March 21 | The wired circuit for getting power to all the servos was completed |
| Mount color sensor | March 21 | The color sensor was mounted to the underside of the chassis using screws and metal brackets |
| Mount line follower sensor | April 7 | The line follower was mounted to the front underside of the chassis using screws, metal brackets, and hot glue |

**Software Development**

|  |  |  |
| --- | --- | --- |
| **Task** | **Approximate Date Completed (2017)** | **Description** |
| Arm movement | March 9 | Write code to enable the arm to grab consumables |
| Bot navigation | March 9 | Write code to make the bot drive around via its wheels |
| Move bot and operate arm | March 14 | Write code to ensure the bot can both operate the arm and drive around |
| Tune color sensor | April 7 | Tune the sensor inputs to get more accurate sensor output. Then, map the sensor output to RGB values |
| Integrate the 3 components | April 14 | Write code that ensures the robot can operate the arm as desired, drive on the table, and read colors |
| Tune line follower sensor | April 28 | Tune the line follower sensor so the robot follows the taped line on the table |
| Integrate all 4 components | April 30 | Ensure the robot can drive on the table (following the line), operate the arm, and read colors |

Been

**State Transition Diagram**



### Division of Labor

**Version Control Manager** – Branden Wagner

**Bot Navigation** – Michael Roark

**Bot Color Reading** – Joseph Olin

**Bot Loading and Unloading** – Branden Wagner and Michael Roark

**Presentation Preparer** – Joseph Olin and Branden Wagner

### References

Brazeiros . (2017). Meats. Retrieved February 16, 2017, from http://www.brazeiros.com/menu/meats/

Casavela, S. (2012). C++ PROGRAM FOR DRIVING OF AN AGRICOL ROBOT. Annals Of The University Of Petrosani Mechanical Engineering, 1411-19.

Ju, B. (2016). Robotic gripper can throw darts, balls - with no arm motion | Cornell Chronicle. Retrieved February 16, 2017, from http://www.news.cornell.edu/stories/2012/02/robotic-gripper-can-now-throw-things

Zexuan, Z., Jun, X., Jian-Qiang, L., Fangxiao, W., & Qingfu, Z. (2015). Global path planning of wheeled robots using multi-objective memetic algorithms. Integrated Computer-Aided Engineering, 22(4), 387-404. doi:10.3233/ICA-150498