# FINAL REPORT

*Canek Acosta*

Arturito

EEL 5666 IMDL

Dr. Schwartz and Dr. Arroyo

Mike Pridgen and Thomas Vermeer

# TABLE OF CONTENTS

# Abstract

Overall, this course has been an immense learning experience. Additionally, this is one of the first hands-on experiences I've had since high school. Working on Arturito has forced me to pull together everything I've learned in Computer Science, Math, Physics, and even Psychology to even be able to plan and design him. I had a great time tinkering away with my robot to get it to do what I wanted.

I intend to work with my strong point, Computer Science, and create a wonderfully effective omnidirectional precision artillery robot to play beer pong. Although I originally wanted to make him an all terrain robot, I was dissuaded due to my lack of experience with mechanical engineering. Thus, I will tackle problems manageable by careful programming and clever solutions.

Using the AVR Cam, I wish to do way more than blob detection, though originally that will be my goal. Eventually, I hope to be able to use it to override most other sensors once it be- comes reliable enough in things like edge detection. After much trouble, the precision offered by the launching mechanism was not good enough to play beer pong, though it had plenty of kick. Ultimately, it became a human hunting artiller fire robot. Using what I learn in this class, I will create another robot for my senior project at CISE and have it be like no other senior project done before.

## Introduction

Being a member of a fraternity, I find myself losing at what is mathematically, a quite simple game of projectile motion. This is predominantly due to my lack of sensation and actuation with digital precision. After some research online, and consulting the TAs various times, I have settled on a general design for my holonomic artillery robot. Throughout this paper we will cover the different aspects of what I plan to include in my robot, in addition as to how they directly affect the success of its goals. My objective is to have my artillery robot, Arturito, be capable of locating and firing at targets based on color.

# MAIN BODY

## Integrated Systems

The way I intend to structure Arturito is on a octagonal platform sandwich with four holonomic omniwheels driven by a DC gearhead motor each. On top of the sandwich lie the infared sensors along with the launching mechanism. Additionally, the PVR board, along with the motor drivers will be placed in the middle with the sonar and the camera facing towards the front. The robot itself is an attempt at a simple and small yet complete robot so that I can later work improve it's AI for my senior project.

# Mobile Platform

The mobile platform will be octagonal to simplify the angle responses to different situations in reference to its holonomic movement scheme. It seems to me that this will work to accomplish it's goal better since it will spin about its center to locate a target every once in a while, so it won't knock anything over or bump into anything. The electronics will certainly be crammed, but with the idea of biological engineering that empty space is wasted space, minimizing that waste would be best

# Actuations

In regards to actuation, four DC gearhead motors will be the way the robot moves about, controlled by motor drivers. Due to the selected wheels (Omniwheels), this robot will be holonomic. Each of these DC motors is controlled by its own motor driver in order to be able to fully control the direction and speed of each. This will give me the total control that I need to make a completely self-directed movement scheme with obstacle avoidance. The launching device will function a lot like a pitching machine used  for baseball or tennis. Driven by a DC motors controlled by a speed controller, the speed for launching can be varied to perform a variety of ranges.

## Sensors

The different sensors that will be employed are Infared, Sonar , and the AVR Cam . Infared will be used primarily for environmental awareness whereas the rest will be used in either locating a target or determining the aspects for the artillery fire.

Regarding the sonar, I am experiencing incredibly precise results for long ranges but very poor results for close ranges, that I have mapped a range to. This is mostly because it will be used in my launching mechanism's firing algorithm. Nonetheless the Maxbotix EV3 will certainly serve it's purpose as the rangefinder for determining launch speed for the launching device. Proper calibration upon initialization takes place with an object at about fourteen inches away.

Within 3-4 feet = 240 - 404

Within 5-6 feet = 400-600

Within 7-8 = 600-800

Within 8-9 = 800-1000

Within 9-11 = 1000-1500

Within 12 -18 = 1500 - 2300

Supposedly, this sensor can even see as far as 22 feet, though I won't be needing such values since my launching mechanism is unlikely to shoot that far.

Infared, will be the key component of collision avoidance. I have been able to make a note of the curve of values for certain ranges and made some progress in applying it to my more advanced collision avoidance schemes which attempt to have the robot translate away from object that may be at the sides. Thus far, here is my scheme:

Very Close = 3150 - 3600

Getting Close = 3600 - 3150

Medium = 2400 - 3150

Getting Far = 2200 - 2400

Far % % = 0 - 2200

Due to the odd decrease towards 3150 when getting very close Getting Close will have to be the point at which decision are made regarding proximity.

Furthermore, the AVR cam is up and running and was the most difficult to set up. Due to the software included with it, I have been able to test all the different functions that it is capable of as and apply it towards my goal. Using several two-letter commands, it is capable of capturing whole images, tracking colors, and setting colors to track. It is set to track my orange flag on boot in order to be able to find its targets.

The current implementation waits until the camera detects something of the preprogrammed color within a certain virtual box in the center of vision corresponding to where the sonar would be able to echolocate.

## Behaviors

The different behaviors that will be implemented will be, Collison Avoidance, Exploration, Target Acquisition, and Artillery Fire. Collision Avoidance will be done constantly with the help of infared, while using the accelerometer to detect a bump the infared may have missed. Exploration will involve wall following using a smoothing algorithm through fuzzy logic. I have created a table of values that will be able to use the ratio of motors to move at whatever angle I need to within the unit circle. Additionally speed will be regulated by the sensor allocated to a certain range of that unit circle. Target Acquisition will be done based on blob detection and sonar ranging. Artillery Fire will be computer internally using the Target Acquisition information to accurately fire a pong ball towards a target.

## Experimental Layout

I was able to run modularize many components of my robot and piece it together one part at a time. Starting with sensing and actuation I was able to check what each input and output would mean. Following with the implementation of my camera, and finally the launching mechanism. Each of these were combined piece by piece to finally make up Arturito.

# Closing

## Conclusion

Arturito has certainly been my greatest engineering achievement. I have learned so much thanks to the great TAs and professors. Between projecticle motion, and some programming I knew beforehand, everything else I have applied was learned on the fly for the very purpose of this class. The AVR Cam was certainly the most difficult aspect of completing this robot. I aim to fine tune the process of computer vision along with incorporate more artificial intelligence in the future and have it serve as my senior project for Computer Science.

## Documentation


Tom Nally, "Projectile Motion in 3D Space", [Webpage], Available at HTTP:

http://babek.info/libertybasicfiles/lbnews/nl130/proj3d.htm

John S. Palmisano, "PROGRAMMING - COMPUTER VISION TUTORIAL", [Webpage], (2005-2009), Available at HTTP:http://www.societyofrobots.com/programming_computer_vision_tutorial.shtml

John S. Palmisano, "OMNI-WHEEL ROBOT - FUZZY", [Webpage], (2005-2009), Available at HTTP:

http://www.societyofrobots.com/robot_omni_wheel.shtml#holonomics3


# APPENDICE

```
void main(void)
{
	xmegaInit();						//setup XMega
	delayInit();						//setup delay functions
	ServoCInit();						//setup PORTC Servos
	ServoDInit();						//setup PORTD Servos
	ADCAInit();					//setup PORTA analong readings
	lcdInit();						//setup LCD on PORTK
	lcdString("    Arturito    ");	//display "Arturito" on top line (Line 0) of LCD
	lcdGoto(1,0);			//move LCD cursor to the second line (Line 1) of LCD
	lcdString(" Beer Pong Champ");	//display "Beer Pong Champ" on second line
	PORTQ_DIR |= 0x01;					//set Q0 (LED) as output
	int motorL;
	int motorR;
	int motorB;
	int motorT;
```

```
int avgX = 0;
int avgY = 0;
int infaredM;
int infaredL;
int infaredR;
int sonar;
int leftSensor;
int rightSensor;
int oldavgX;
int oldavgY;
int upperLX;
int upperLY;
int lowerRX;
int lowerRY;
unsigned int *tracking;
delay_ms(2000);
lcdGoto(0,0);
lcdString("Camera :      ");
lcdGoto(1,0);
lcdString("                    ");
lcdGoto(1,0);
USARTInit();
delay_ms(200);
lcdString(SendCommand("DT\r"));
delay_ms(2000);
lcdString(SendCommand("ET\r"));
while(1)
{
	infaredM = ADCA1();			//move ServoC0 to current position
	infaredL = ADCA3();
	infaredR = ADCA2();
	sonar = ADCA0();
	leftSensor = (infaredM + infaredL)/2;
	rightSensor = (infaredM + infaredR)/2;
	oldavgX = avgX;
	oldavgY = avgY;
	*tracking = ListenForTracking();
	upperLX = tracking[3];
	upperLY = tracking[4];
	lowerRX = tracking[5];
	lowerRY = tracking[6];
	avgX = (upperLX + lowerRX)/2;
	avgY = (upperLY + lowerRY)/2;
	int avg = (avgX + avgY)/2;
	lcdGoto(1,0);
	lcdString("                ");
```

```
        lcdGoto(1,0);
        if( oldavgX == avgX && oldavgY == avgY)    // if not tracking color, explore
        {
        lcdGoto(0,0);
        lcdString("Camera :      ");
        lcdGoto(1,0);
        lcdString("searching...");
        if(leftSensor <3000 && rightSensor <3000)
        {
        MotorD2(-50);
        MotorD3(-50);
        MotorD1(0);
        MotorD0(0);
        }
        else if(infaredM >= 3100)
        {
        MotorD2(-50);
        MotorD0(-50);
        MotorD1(-50);
        MotorD3(50);
        }
        else if(infaredL >= 3100 && infaredR < 3100)
        {
        MotorD2(-50);
        MotorD3(-50);
        MotorD1(-50);
        MotorD0(50);
        }
        else if(infaredR >= 3100 && infaredL < 3100)
        {
        MotorD2(-50);
        MotorD3(-50);
        MotorD1(50);
        MotorD0(-50);
        }
        }

        else                                        // if tracking color
        {
        lcdGoto(0,0);
        lcdString("Camera :      ");
        lcdInt(avgX);
    if(sonar <= 400)
        {
        lcdGoto(0,0);
        lcdString("Sonar :      ");
```

```
        while(sonar <= 400)
        {
        sonar = ADCA0();
        lcdGoto(1,0);
        lcdInt(sonar);
        MotorD2(50);
        MotorD3(50);
        MotorD1(0);
        MotorD0(0);
        }
        MotorD2(0);
        MotorD3(0);
        MotorD1(0);
        MotorD0(0);
        lcdGoto(1,0);
        lcdString("FIRING!");
        Launch();
        }
        else if(sonar >= 500)
        {
        lcdGoto(0,0);
        lcdString("Sonar :      ");
        while(sonar >= 500)
        {
        sonar = ADCA0();
        lcdGoto(1,0);
        lcdInt(sonar);
        MotorD2(-50);
        MotorD3(-50);
        MotorD1(0);
        MotorD0(0);
        }
        MotorD2(0);
        MotorD3(0);
        MotorD1(0);
        MotorD0(0);
        lcdGoto(1,0);
        lcdString("FIRING!");
        Launch();
        }
        }
        delay_ms(100);                                  //delay 100ms
}
```