Design, manufacturing, and testing of a fully automated, soccer-playing robot

ME 588 World Cup Final Report

Manan Patel

Christopher Larkin

Andrew Swanback

Jake Lengacher

Gabriella Giachini

Group: 5

Date: December 12th, 2022

TA: Jie Wang

Abstract

This work summarizes the development carried out of a robot able to autonomously seek out, acquire, aim, and launch a ball into a goal. The team settled on a control scheme using ultrasonic sensors for position and orientation control, as well as for detection of the ball. This combined with a mechanum drive train allowed the robot to operate entirely on a cartesian grid, simplifying the searching and launching logic. A manually primed, spring powered launcher was used for its high repeatability and power density, combined with a funnel based collector.

The system was largely successful in achieving these goals, successfully acquiring and firing the ball. The seeking algorithm worked well, and the launcher was extremely reliable and effective. The largest reoccurring challenge, and opportunity for future improvement is improving the stability of the -position control system, which performed acceptably, but due to interference between ultrasonic sensors, and noise in the system, was not as reliable as desired. Future work could include the use of more precise, higher end range sensors, such as LIDAR or infrared.

Introduction

The objective of this project is to build an autonomous robot that can find, capture, and shoot a ball into a goal. For the development of this robot, the entire system and work was separated into three subsystems: mechanical, electrical, and software.

The overall system was designed via separate designs for each function. Firstly, to find the ball and to position the robot, ultrasonic sensors were found to be the most accurate and efficient method. Secondly, for the movement of the robot, mecanum wheels were chosen to be able to move sideways along the score zones and eliminate the need for turning. Thirdly, to catch the ball, a funnel-like piece with a servo-actuated hook was used to guide the ball into the shooting mechanism. Finally, a simple spring-loaded mechanism was connected to a rod that pushes the ball out of the aimed funnel to score the goal.

Everything was controlled by an Arduino Mega. This microcontroller was chosen for its multiple input/output pins and to be able to easily pass generated code from the Arduino IDE into the robot as commands. Several PWM I/O pins were connected to two h-bridges, wired to the four motors, to control the direction of the mecanum wheels. Moreover, the hook that drives the ball into the funnel and the spring-loaded pin that shoots the ball outwards into the goal were actuated by servo motors, also connected to I/O pins. Finally, the rest of the digital pins were used by the ultrasonic sensors. Four out of the five sensors were placed by pairs at the back and side of the robot. The idea is to implement a position controller by gathering and averaging data from each pair of ultrasonic sensors and use it to adjust the speed of the motors and thus fine-tuning the overall desired position, direction, and movement of the robot. A battery of 6V was chosen to power the entire robot.

A finite state machine was developed to automate the actions of the robot (e.g.: move, find ball, capture ball, aim ball, and shoot ball) according to the inputs received from the ultrasonic sensors. Since a lot depended on the accuracy of the sensors, calibration was conducted, and median filters were chosen to attenuate outliers. The car continuously checked if it had rotated while driving using the ultrasonics on each side and adjusted the motor speeds accordingly to correct, which was outside of the FSM. The FSM was aware of the field only as an x-y coordinate system, with x and y values provided by the backwards and leftwards ultrasonic readings.

System Design

Mechanical Design

In order to simplify the necessary physical design of the system a pre-made drive train was selected. To implement the cartesian grid strategy discussed above, mecanum wheels were used, allowing side to side movement in addition to turning like a conventional tank drive. It was decided to localize the bulk of the electronics onto the drive train, and mount the launcher to the side of the frame. As a result, the weight of the batteries, H bridges, and controllers balance out the offboard weight of the launcher, allowing the drive base to run stably without tipping (an additional ball caster was included on the launcher to also prevent tipping). Figure 1 below shows this arrangement.

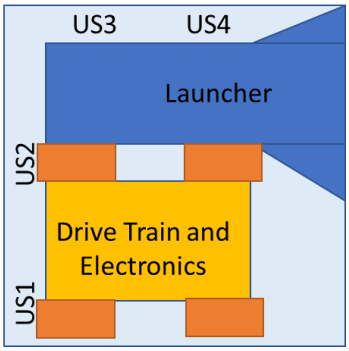


Figure 1: High Level Robot Space Claim. US stands for ultrasonic

More involved was the design of the launcher. A spring pin powered launcher was selected over other options due to the low number of moving parts in the system. The project statement required that the robot return to the starting area to be reset after each run, so a hand primed mechanism was tenable. This allowed a stronger spring to be used, without the need for a large enough motor and priming mechanism to draw it back. To fire, a servo was used to pull the catch on the mechanism and allow the spring to fire. Figure 2 below shows this operating principle.

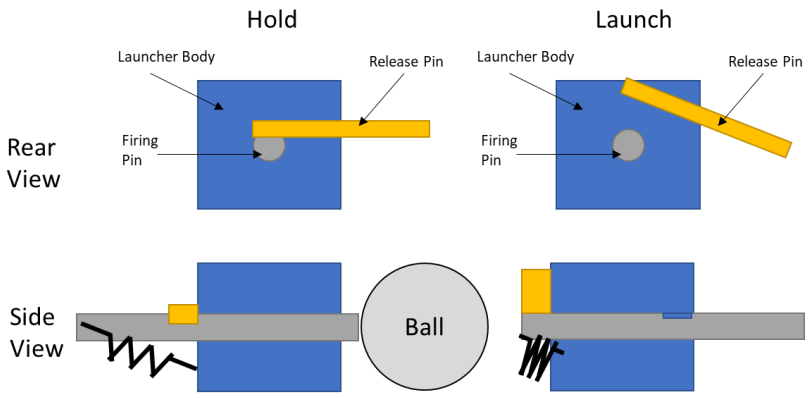


Figure 2: Launcher Working Mechanism

In order to ensure the ball went straight when fired, a barrel was added around the launcher ensuring that there was very little room for deviation in the initial position, ensuring extremely consistent, repeatable shooting.

The collection method selected was a simple funnel. The shape of the funnel was specifically designed to ensure that the ball would be carried into the center of the barrel, while maximizing the width of the collector, giving some margin for alignment error when collecting.

To ensure that the ball was able to reach the back of the launcher, and not be lost during movement, a simple retention arm was used to push the ball the remainder of the way to the back, shaped carefully to fold easily out of the way but also reach to the back of the launcher.

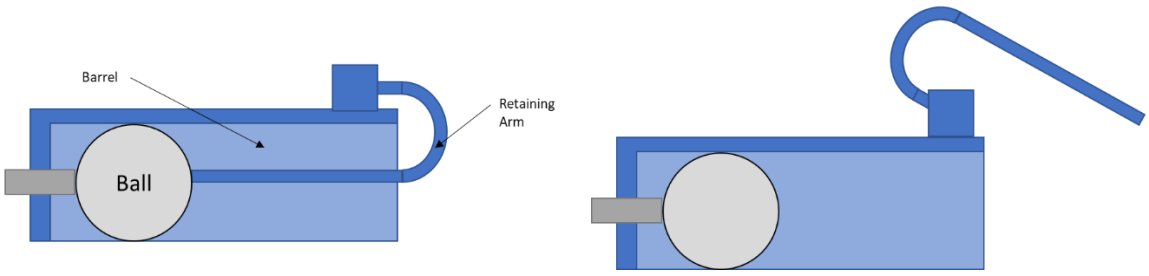


Figure 3: Collector Arm Positions

Finally, the ultrasonic sensors required by the control scheme were mounted on the edges of the frame to allow for positioning on the field, while the sensor for ball detection was mounted in the back of the barrel.

Electrical Design

To effectively implement the control scheme specified above, an electronic system was designed. The core of the design was an Arduino Mega, chosen for its large number of input and output ports. In addition, two batteries were used to supply the system: 6V for the drive motors, and 9V for the Arduino, which then supplied the sensors through its regulated 5V output.

The first type of connection made was the H bridges driving the motors. The H bridges used already had flyback capacitors installed, and thus only had to be given a supply voltage from the 6V battery, and the Enable and Input pins, wired to PWM and digital Arts on the Arduino. Figure 4 shows this connection.

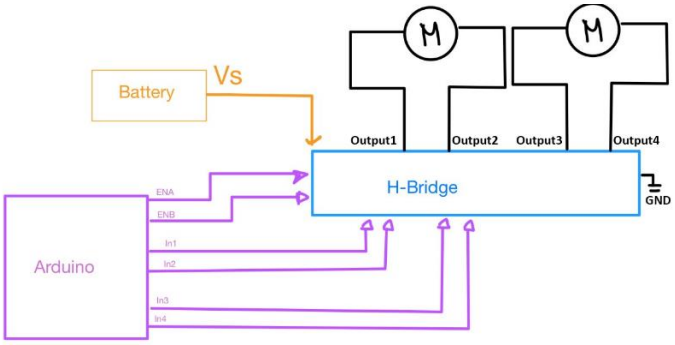


Figure 4: Circuit Diagram for Arduino + 1 H-bridge + 2 Motors + 6V Battery

Ultrasonic sensors were supplied using the Arduino's 5V regulated output, and interfaced using the I2C protocol, using a pair of digital input and output pins. Figure 5 below shows this connection.

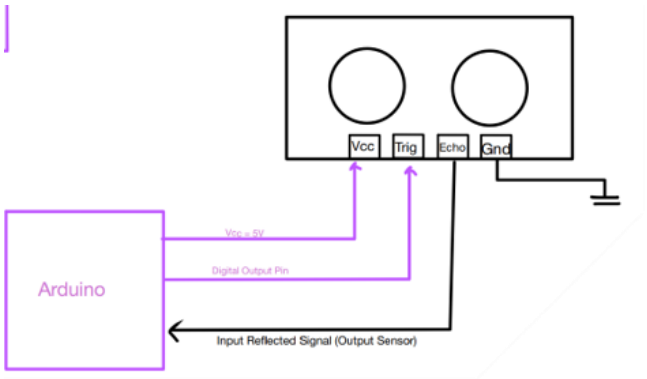


Figure 5: Circuit Diagram for Arduino + 1 Ultrasonic Sensor

The servos were powered by the 6V external battery and commanded simply using a PWM output from the Arduino. Figure 6 shows this connection. A complete pinout for the Arduino can be seen in Appendix A, including the relevant pin numbers for device.

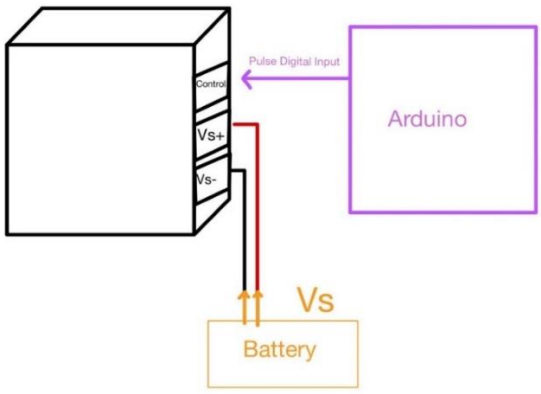


Figure 6: Circuit Diagram for Arduino + 1 Servo Motor + 6V Battery

Software Design

The software system used a finite state machine (FSM) to handle the high-level strategy. To keep the FSM manageably complex, the alignment problem was abstracted from it. Aligning the robot with the walls is the key to its localization, as then the ultrasonics could be averaged to create a simple x-y coordinate system so the robot can use closed-loop control to always be precisely where desired. The alignment problem simplifies the FSM to only supervising a few functions: going to a coordinate, firing, and resetting the firing pin, and using and resetting the collection hook. The state machine strategy is detailed in Table 1 and Figure 4 below.

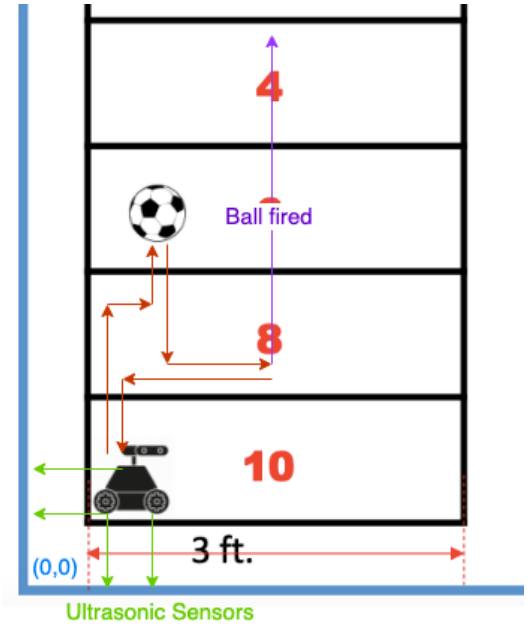


Figure 7: Searching and firing strategy

Table 1. State Machine high level strategy.

State	Output
0. halt() + calibrate()	Button press -> 2
1. forward()	at BL corner of 6 -> 3
2. right()	ball found -> 5, not found -> 4
3. left(slower)	ball found -> 5, not found -> 10
4. forward(slow)	ball acquired -> 6, assume we can't lose it
5. backward/forward()	correct zone -> 7
6. left()	getX() < CENTER -> 8, getX() == CENTER -> 9
7. right()	getX() > CENTER -> 7, getX() == CENTER -> 9
8. fire()	delay(500) -> 10
9. left()	at wall -> 11
10. backward()	at start -> 1

The alignment of the robot was the most difficult part. The same scheme was implemented for the left side ultrasonics and the back side, averaging them, and using other data validation schemes. On each side, the two sensors were subtracted to get a difference, and from that an angle that the robot was off from square. This angle was fed into an adjustment term in the motor code to slightly speed up or slow down certain motors to return it to square. Another term was added to keep the absolute positioning correct, so that going forward would not result in a left/right positioning error.

Results

Since many functionalities of the robot relied greatly on the ultrasonic sensors, calibration was performed to see accuracy and if any adjustments/corrections needed to be done. The following figure shows the calibration data of one of the sensors tested. It compares the real distance from the sensors versus the measured distance by the sensor.

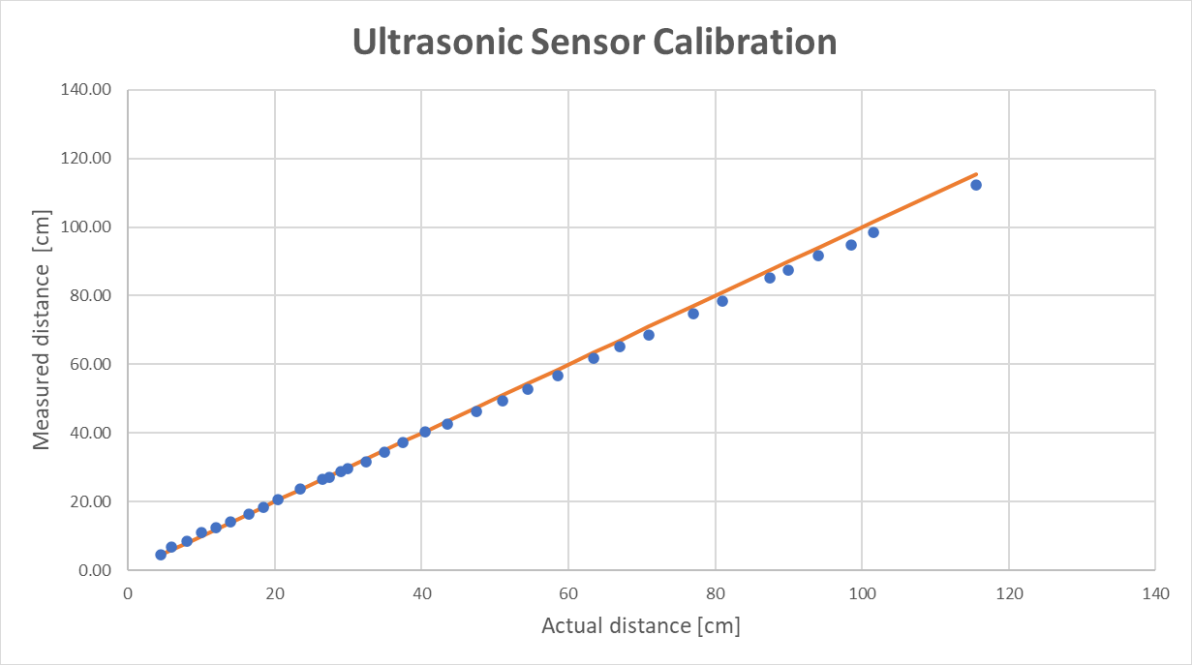


Figure 8: Calibration data of 1 ultrasonic sensor

Figure 4 shows how the actual distance is almost proportional by 1 to the measured distance. At small lengths, the amounts are almost exact. However, as the distance increases an error of around 2.5cm appears. For that reason, the team decided to operate the ultrasonic sensors within acceptable ranges. For example, the inner sensor to find the ball, would work in a 0 to 60 cm range. Anything farther the code would ignore as it would possibly be incorrect, or the sensor would no be able to differentiate between the ball and wall.

Testing the sensors also showed that output fluctuates a lot and has many outliers. For that reason, a median filter was used, and logic was implemented to deal with outliers from the averaged data of the ultrasonic pairs.

Discussion

The robot built was successful in accomplishing the objectives:

The position controller used to adjust the position, direction and movement of the robot was very effective. The team managed to have the robot be exactly where it needed to be and move in a very stable manner. Unfortunately, the control scheme depending heavily on the battery of the robot and as it was depleted, its functionality decreased and with it the position controller failed due to inability to make up for motors failing.

The method to find the ball was very successful. The team was afraid that the tunnel would hinder the ultrasonic and yield bad data. However, the sensor was placed and adjusted inside the tunnel until it only yielded information when objects where right in front of it. Moreover, logic was used to only accept objects within 24 inches of the sensor. Even with these precautions many times the sensor did not find the ball despite it being in front of the funnel. Critically, the FSM dealt with this by constantly switching from the "looking for ball" to the "capturing ball" state, forcing the robot to sweep while also getting closer to the ball until finally capturing it.

Aiming the ball worked well most of the time. The ultrasonic sensors placed on the back and side of the robot were used to move the robot to the farthest score zone and place it exactly in front and in line with the goal. The position controller was extremely useful in this case, however, the robot sometimes failed to be perfectly aimed and other times took a long time to find the right position (in some cases, it never found it, and it kept on shifting in place).

The spring-loaded mechanism chosen to shoot the ball was excellent. The force applied on the ball was strong enough to prevent unwanted spinning (that would move the ball sideways instead of maintaining a straight trajectory) and overcome any friction from the tunnel.

Overall, the robot managed to fulfill all its goals autonomously. However, the team did face a lot of issues. The biggest constraint was not having enough power for the entire robot. A 6V battery was chosen since that was the necessary power needed to drive the wheels. However, the team forgot to think how the rest of the electronic systems would be powered. The first solution found was a 9V battery wired to a small breadboard, but after lots of testing, it was fried. A barrel plug connector was then utilized, but the Arduino barrel port was malfunctioning too (this might be because instead of buying a real Arduino mega, a cheaper version was found). Finally, at the last minute the ME375 robot battery was used. This was the only

available solution that worked; however, it was not fully charged, and it unbalanced the robot (as it was very heavy).

Another big issue the team had to deal with was wiring the output of the h-bridge incorrectly to the back wheel motors. This caused a lot of issues since the team could not figure out why the robot would not drive straight and adjust itself as commanded. After a lot debugging (e.g. print statements, testing the wheels individually…) the bad wiring was found out, and fixed.

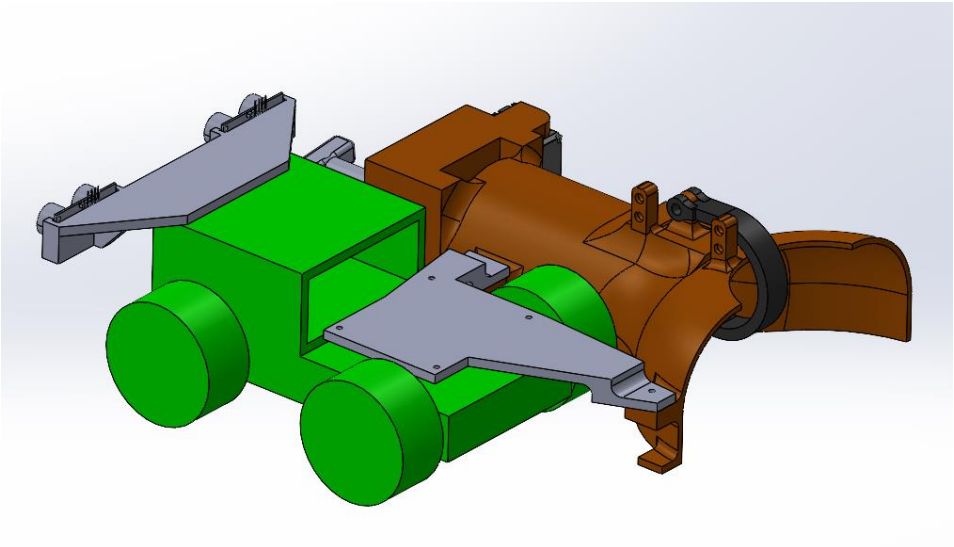
Conclusion

This design was successful in its goal of developing a system able to detect, acquire, aim, and fire a ball on a field. The system's collection and firing subsystems worked well and consistently. The source of the most problems came from inaccuracies with the ultrasonic sensors and lack of power. Noise from ultrasonic readings forced us to enforce tight cutoffs on different readings, making our system overall less robust to different scenarios. Additionally, every state was reliant upon an ultrasonic sensor whether it was smoothing out driving, finding the ball, or aiming the goal. All actions' reliability was further reduced by low power issues. Motors would receive less power, and therefore did not correct mistakes as quickly. For future iterations of the vehicle, higher quality sensors and higher capacity batteries would be a worthwhile investment to help both collecting the ball and moving the robot. Making these changes would have the greatest impact on the robot's operations, for the smallest investment.

Appendix

TYPE	PIN	DEVICE	PURPOSE	COLOR		OUTPUT1	MOTOR1	
PWM	2	H-Bridge	ENA Signal H-Bridge (speed for motor 1)	GREEN		OUTPUT2	MOTOR1	
	3		ENB Signal H-Bridge (speed for motor 2)	BLACK		OUTPUT3	MOTOR2	
	4		ENA Signal H-Bridge (speed for motor 3)	BROWN		OUTPUT4	MOTOR2	H BRIDGE1
	5		ENB Signal H-Bridge (speed for motor 4)	BLUE				
	31	H-Bridge	In1 Signal H-Brige (direction for motor 1	YELLOW		OUTPUT1	MOTOR4	
	33		In2 Signal H-Brige (direction for motor 1	ORANGE		OUTPUT2	MOTOR4	
	35		In3 Signal H-Brige (direction for motor 2	RED		OUTPUT3	MOTOR3	
	37		In4 Signal H-Brige (direction for motor 2	BROWN		OUTPUT4	MOTOR3	H BRIDGE2
	39		In1 Signal H-Brige (direction for motor 3	GRAY				
	41		In2 Signal H-Brige (direction for motor 3	PURPLE				
	43		In3 Signal H-Brige (direction for motor 4	BLACK				
	45		In4 Signal H-Brige (direction for motor 4	WHITE				
DIGITAL	22		Input Trigger Signal (sensor 1)					
	26		Input Trigger Signal (sensor 2)					
	51		Input Trigger Signal (sensor 3)					
	48		Input Trigger Signal (sensor 4)					
	50	Ultrasonic	Input Trigger Signal (sensor 5)					
	24	Sensors	Output Reflected Signal (sensor 1)					
	28		Output Reflected Signal (sensor 2)					
	53		Output Reflected Signal (sensor 3)					
	46		Output Reflected Signal (sensor 4)					
	52		Output Reflected Signal (sensor 5)					
	47	Servo	Digital Control Input (servo motor 1)					
	49	Motors	Digital Control Input (servo motor 2)					

Appendix A: I/O Pin Out for all electrical components



Appendix B: Isometric view of CAD