EEL 5666 IMDL

Summer 2004

Final Report

Instructor: Dr Arroyo, Dr Schwartz

Student: Tsang-Wei (Dan) Huang

Date: 08/02/2004

Table of Content

Introduction

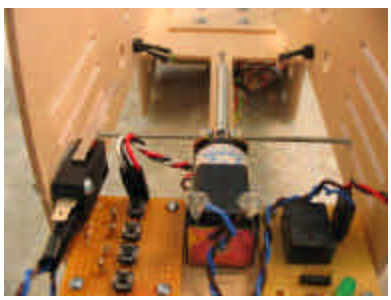
The original idea is to make a robot that can play basketball. Similar projects were done by other IMDL students before but there's room for improvement. The robot should be able to aim at the basket at any place within reasonable range. It should be able to calculate the distance and line up with the basket automatically, and then launch the ball.

This paper will present the major components of the system, both mechanical and electrical, and the experimental data and result. At the end of the paper some current limitation and future improvement will be discussed.

Integrated System

The system consists of the following components:

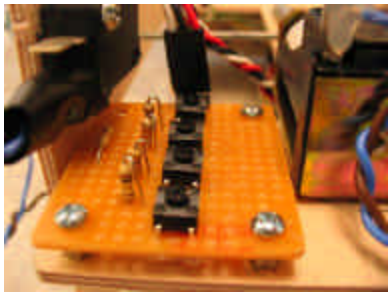
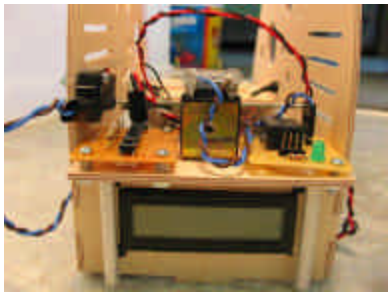
- Sonar Range Finder – Two of the Daventech SRF04 are used to measure distance and to line up with the target. They are mounted on the front corners of the robot. More details will be presented in later sections.
- Shooting Arm – The shooting force is provided by two coil springs as shown in the figures. When the arm is pulled back, it makes contact with a switch and then the A/D system uses the voltage reading on the switch to decide if the shooting arm is pulled and needs to be locked. The locking mechanism consists of a solenoid with a rod attached to it horizontally, a spring to pull the rod back to release the shooting arm, and two hooks on the shooting arm. The system can be seen in these figures.

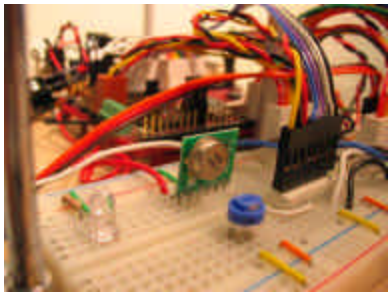
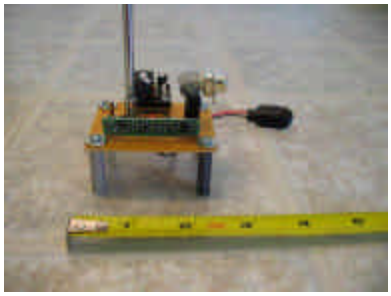

- Force Control – To control how far to shoot the ball, a metal rod is inserted on both side of the shooting arm with shock absorbers. The shooting arm also has a form in from of it to absorb the impact.



- Input and Output – There are four buttons for basic user input. The system uses A/D with a basic circuit that produces different voltages when different buttons are pressed. The output is a 16x2 LCD display with LED backlight. It uses 4-bit mode to save pins.

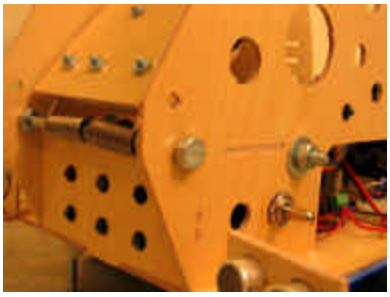



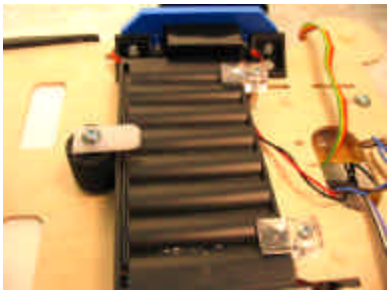
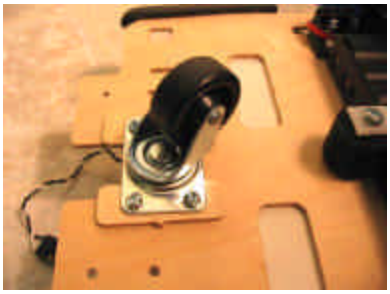
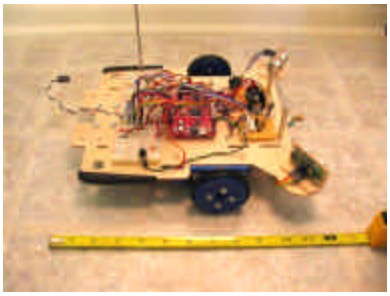
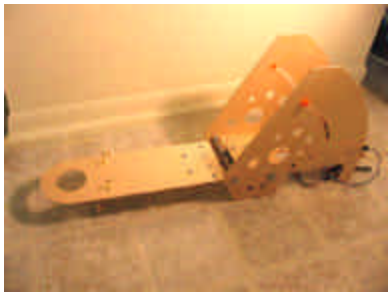
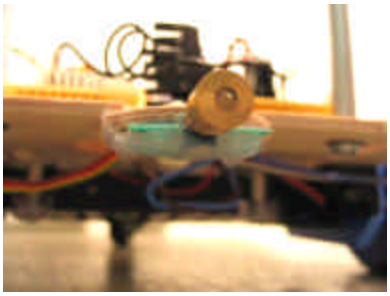
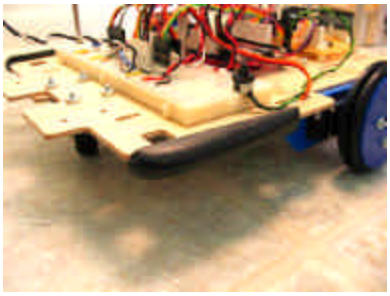
- Wireless Communication – A Laipac TLP434 315MHz wireless communication module is used to send out serial messages to another computer or robot. It can be used to send out messages for debug or display purposes.

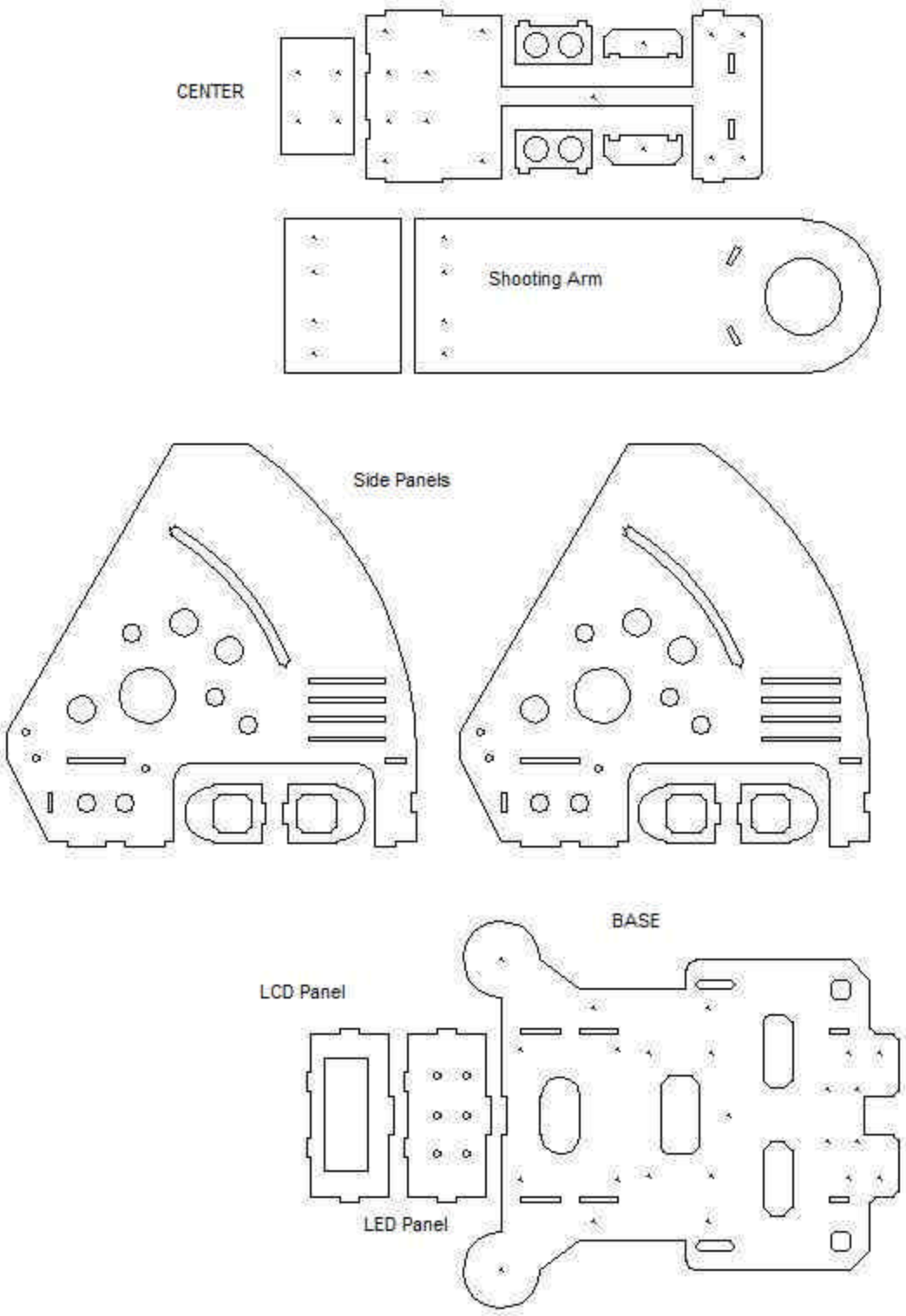
Mobile Platform

The platform has three main parts: The base, the side structures (2), and the shooting arm. There are also other small parts used to support or connect the components together. The base provides space for the MAVRIC-II board, the power board, sonar systems, and a breadboard for other connections. The front panel has 6 LEDs mounted and the LCD display is attached to the back panel. The two side panels support the shooting arm and have guiding tracks for the solenoid. The center piece provides support for the coil spring and has a hex screw underneath to provide extra support. Without it the robot can not hold the springs in place or the structures would be damaged easily.




This is the Autocad drawing for the robot.



Actuation

I use two GWS S03N servos for locomotion. The servos are pre-modified potentiometers to adjust the center position.

Specifications:

Size: 1.55 x 1.40 x 0.79in.

Weight: 1.48oz.

4.8-6v Speed: 0.23-.18 sec/60°

4.8-6v Torque: 47-56 oz-in.

The two servos use the regulated 5V output from a 7805 voltage regulator. A caster is installed on the back of the robot for balance and support.

For wheels, I modified the O-Ring wheels to have double rings to provide additional support and traction.

Originally I used code obtained from a website to generate PWM signals to control my servos. It gives me random problems where one of the servos will be out of control and just keep spinning forever with a constant speed. I've assumed it's the power regulator being overheated and not providing enough current for the servo. I later scratched the PWM module and re-wrote the whole system completely on my own to generate exact waveforms and precise timing. It's been working perfectly and the servos are running great without problem.

Sensors

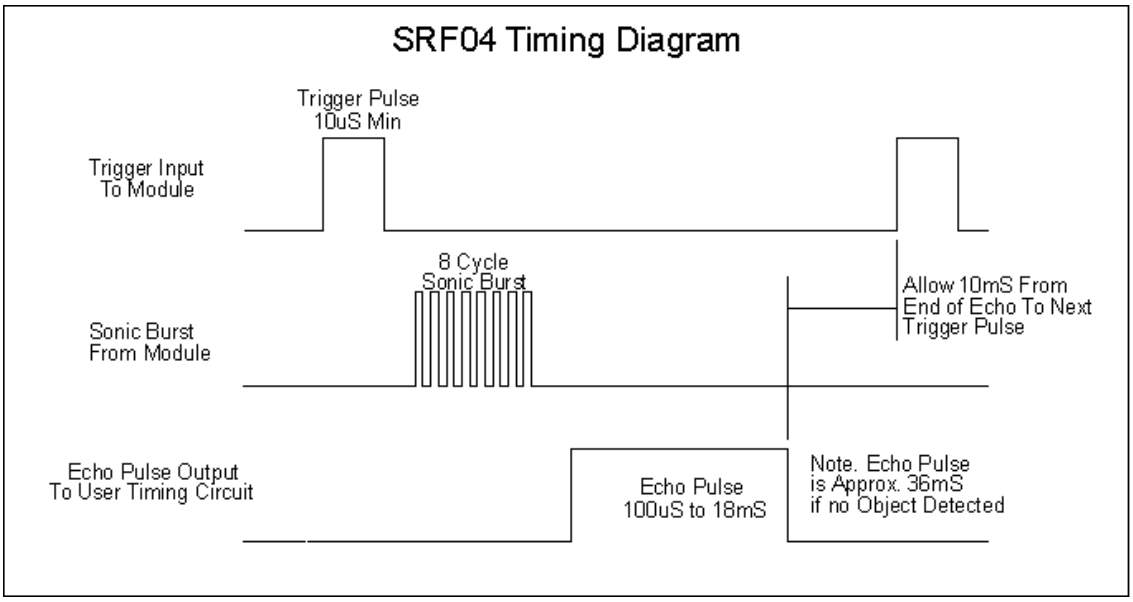
Sonar Range Finder – Devantech SRF04

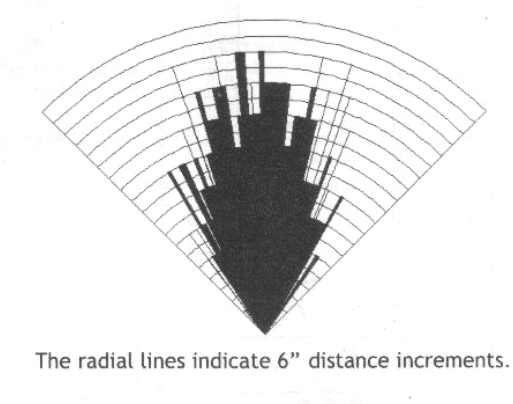
The special sensor system is the sonar range finders by Devantech. With one sonar, the robot can detect distance and obstacles. With two sonars and an angle between them, the robot can line up to a specific target.

Theory of Operation

The sensors transmits a pulse of ultrasonic waves that travels at the speed of sound, which can be calculated by V = 331.4 + 0.6 Tc m/s, where Tc is the temperature in Celsius. In room temperature, 25 degree Celsius, sound travels at approximately 346.5 meters per second, or 13.64 inches per millisecond, or 1.14 feet per second. The sensors pulses for a brief moment after the transmission to avoid echo. The sound reflects from any object, if there's any, in its path and is detected by the receiving module in the sensor. By calculating how much time is elapsed between the transmission and the echo, the distance can be determined.

Timing Diagram





The radial lines indicate 6" distance increments.

Software Implementation

The approach is suggested by Will. First a timer is setup to trigger an interrupt every 50 us. In the interrupt service routine, a state variable is checked and updated to create a state machine that acts according to the timing diagram above. There are four states:

- SRF_State_init: The initialization state, the sensor is started when the state variable is set to this state.
- SRF_State_wait: After the trigger input is set to high for one cycle (50 us), the system enters the wait state. It stays in this state until the output line is high.
- SRF_State_echo: The state when the echo line is high, every time the system enters the interrupt handler, it checks the echo line, if it's still high, a counter is incremented, once the echo line is low, the reading is completed and the counter contains some value proportional to the distance.
- SRF_State_idle: The state when the sensor is inactive.

SRF Sensor Application

To make use of the sensor, I place two sensors 6 inches apart from each other, and turn them toward each other, creating a cross-eyed vision. The sensors are triggered one after another, with at least 10 milliseconds in between to avoid picking up previous transmissions. Beam patterns overlap and if both sensors report a pair of matching values, the object can be assumed to have the same distance away from both sensors, which indicates it's at the center of the robot. A more detailed diagram is presented in the special sensor report.

Conclusion

When the two sensors are turned at a small angle (15 to 20 degrees), the beams overlap and the region extends to the maximum range. As the angle gets bigger to about 25 degrees, the inner boundaries became almost parallel. This is, in theory, the optimal arrangement since the error margin can be maintained at a fixed value, regardless of the distance. When the angle gets above 30 degrees, the boundaries intersect and create a zone in which objects can be detected by either sensor. Beyond this zone, neither sensor can detect any objects, even it is directly in front of the robot. This kind of arrangement isn't completely bad, because it can be used to do obstacle avoidance, since it has a wider view than the previous arrangements and can "see" more on the side of the robot. A better approach is to make the angles flexible and controllable by software, so the angle can be adjusted for different purposes.

Behaviors

The robot has the following behavior:

- Initialization: The robot boots up and initialize the system, then it asks the users to place it at a preset distance to take a trial shot and calibrate the sonar reading. It'll use this reading as the reference for later modules.
- Obstacle avoidance: Once a shot is made successfully, the robot wanders around and avoids obstacles by turning left or right with a random decision and a random duration. After a preset time the robot stops and asks the user to give it the ball.
- Lining up with the basket: After the ball is loaded, the robot sweeps 360 degrees trying to find the basket. There is a cylindrical object directly under the hoop to indicate the center of the hoop and to give the robot something to aim at. Once the robot thinks it has found the target, it'll stop and go to next step.

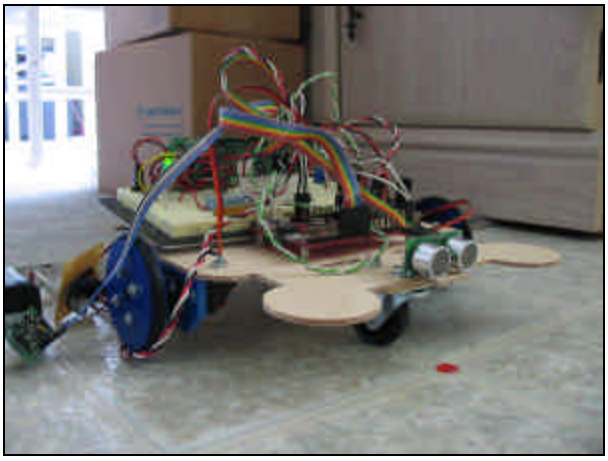
- Adjusting distance: After the robot finds the hoop (or somebody's foot that fools the poor robot), it'll determine whether it's too close or too far away from the basket and either move forward or backward to get to the "sweet spot". While moving it's possible to loose aim at the target if two servos don't run at the same speed, which is normal. So the robot will re-center itself once it's off the target by a certain threshold.
- Taking a shot: Once the robot reaches the correct distance and lines up with the basket, it'll release the lock and the shooting arm will launch the ball.

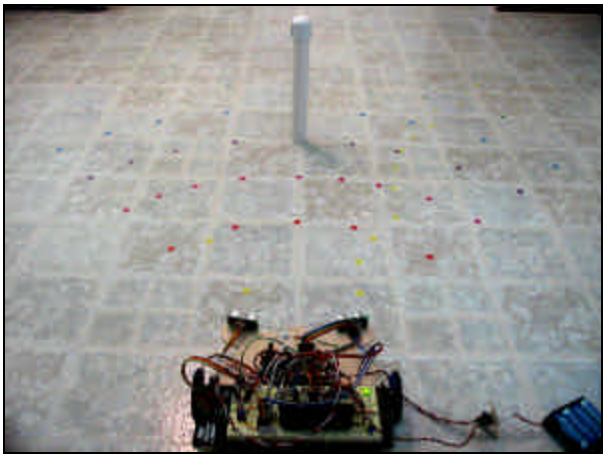
Experimental Layout and Results

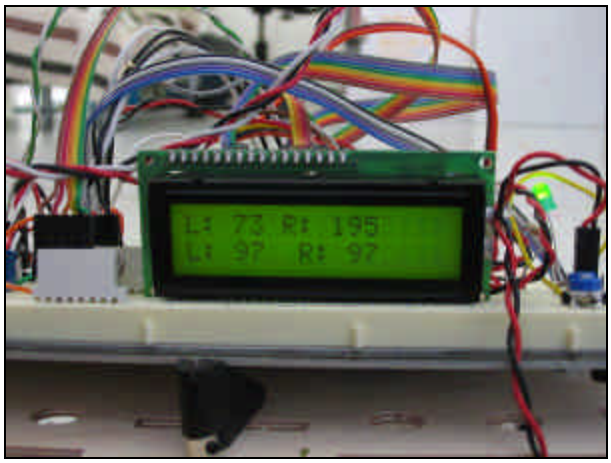
The following data are obtained from the SRF experiment.

Experiment Setup

The experiment setting is the dining area in my apartment. I used small stickers to mark certain important points. The object to be detected is a PVC pipe with one inch diameter. The following pictures show the experiment in progress.







Experiment Data

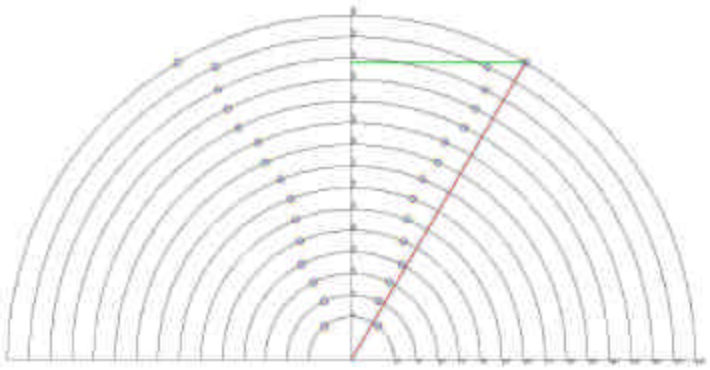
distance	reading	range	angle(rad)	angle(deg)
6	16	3.750	0.675	38.682
9	25	3.825	0.439	25.151
12	33	5.250	0.453	25.944
15	43	7.000	0.486	27.818
18	52	7.250	0.415	23.752
21	60	7.750	0.378	21.657
24	68	8.500	0.362	20.742
27	77	9.875	0.374	21.453
30	85	12.000	0.412	23.578
33	94	13.000	0.405	23.200
36	102	15.750	0.453	25.944
39	109	17.250	0.458	26.251
42	118	18.625	0.459	26.324
45	126	19.000	0.436	24.975
48	137	24.250	0.530	30.345

Distance: how far away the PVC pipe is placed from the origin – the red line in the diagram.

Reading: the reading from the SRF sensor for the specific distance. Details on how the reading is obtained will be presented in later sections.

Range: how far way from the last detectable spot to the center axis – the green line.

Angle: ASIN (range/distance), the angle specified by the red line and the center axis



In the diagram above, each line represents a 6” increment.

The readings from the table above yield an average of 25.7 degrees wide, cone shaped pattern. The beam pattern specified in the datasheet is 30 degrees, which isn’t too far away from the actual readings. One thing that was observed during the process, the furthest spots detectable by the sensor tend to change in each trial, sometimes they can be up to 6 inches apart when the object is placed more than 3 feet away. This really complicated the experiment because I couldn’t make up my mind as to where to mark the spot.

Conclusion

This is my first time attempting to build a robot. It was quite an experience. The original idea was too complicated to accomplish in summer so I had to change the objective to be more realistic. The most challenging part is the mechanical design and the shooting force control. Programming isn’t a big issue for me once I get familiar with the MCU and its sub-systems.

One limitation of the robot is that it can not distinguish the actual basket or something else that just happen to be within its shooting range and has a cylindrical shape. The sonar will be fooled

and think it's the actual basket. A possible solution is to create another way for the robot to find the basket. Light source, for example, can be used together with the existing system to minimize the false-positive rate.

Another area of improvement is to make the sonar mobile so the angle between them can be adjusted to meet different needs. As discussed in the sensor report, different angles can be used to accomplish different tasks, such as better obstacle avoidance or wider viewing range.

The most difficult and challenging improvement will be to figure out a way to control the force. As of now the shooting distance is adjustable, but needs to be fixed each time the robot runs. Because the force required to launch the nerfball is very powerful, to find a way to stop such force and be mobile at the same time can be very difficult.

The manual loading process can be improved too. Again, because of the force of the springs, it might take a very high-torque motor with gear train to produce enough force to pull the arm back.

Appendices

Program code included in a separate page.