Carnegie Mellon University

Final Report

Frisbee Thrower

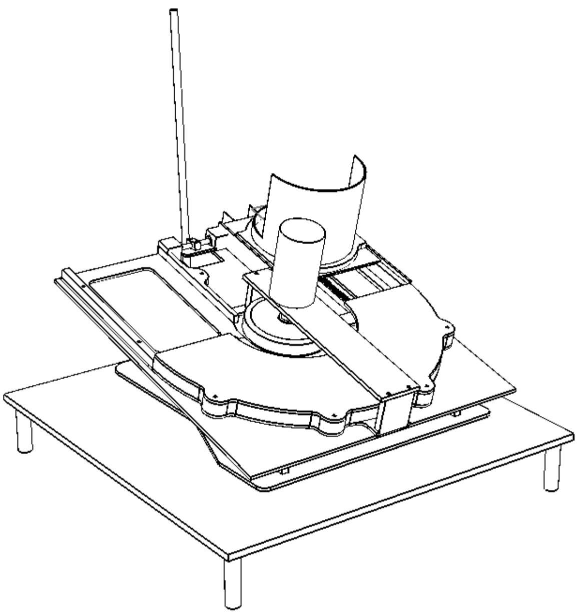
Creators:
Yannick Poffet
Jeff Quackenbush
Evan Walden
Jackie Yang

Instructor:
Dr. John Dolan

A report submitted in fulfilment of the requirements of 24-778 Mechatronic Design

presented by

My Disc in a Box
Department of Mechanical Engineering



Abstract

A robotic Frisbee launcher was designed and manufactured to fire Frisbees accurately through targets 10 feet away. To meet requirements specified in Carnegie Mellon Unviersitys Mechatronic Design class, a launching concept was chosen to favor consistent and accurate launching. A system was then designed to incorporate camera vision for target detection, two degrees of aiming freedom, a robust hopper and feeder system to trigger launch, and a microcontroller for system control. After appropriate analysis, each subsystem was modelled using Computer Aided Design (CAD). For Frisbee launching a spinning flywheel was created in conjunction with a 180° track. A CMUcam4 camera system detected the targets and enabled active control of the positioning systems. Vertical positioning (pitch) was achieved using a lead while rotational positioning (yaw) was achieved using a VEX Robotics gearing system. Ultimately, accurate Frisbee launching was achieved, course requirements were met, and the teams Frisbee launcher won the final competition.

Contents

Chapter 1

Project Overview

1.1 Goal

The objective of this project is to satisfy the requirements set forth by the Carnegie Mellon University course Mechatronic Design. The course objectives were to design a robot capable of launching miniature frisbees a distance of 10 feet at a series of targets at varying heights in two different tests to measure performance. Currently there are a vast number of people around the globe who are incapable of throwing a Frisbee through a small target accurately. Many of these poor throws could have been prevented with the use of My Disc in a Box.

Chapter 2

Design Requirements

2.1 Explicit Requirements

Specifications outlined in the class project description as well as our own analysis helped us determine device requirements. Requirements developed pertained to form, power, cost, and safety. They also included the specifications of the specific tests that our device would have to pass. The form restriction was that the device must fit within a cube with dimensions of $2' \times 2' \times 2'$. The device must use a dedicated power supply. Overall cost must be under \$600 if reimbursement is desired. For safety, the machine must not damage anything with which it interacts.

Additional requirements were determined based on the tests. The Frisbee launcher was required to fire at targets $10'$ away that could be placed anywhere within an area $3'$ high and $4.5'$ wide extending from the ground. From this it was determined that launcher must be able to rotate by at least $25.36°$. The launcher must also be able travel a vertical angle of $16.7°$. A representation of the target area is shown in Figure 2.1.

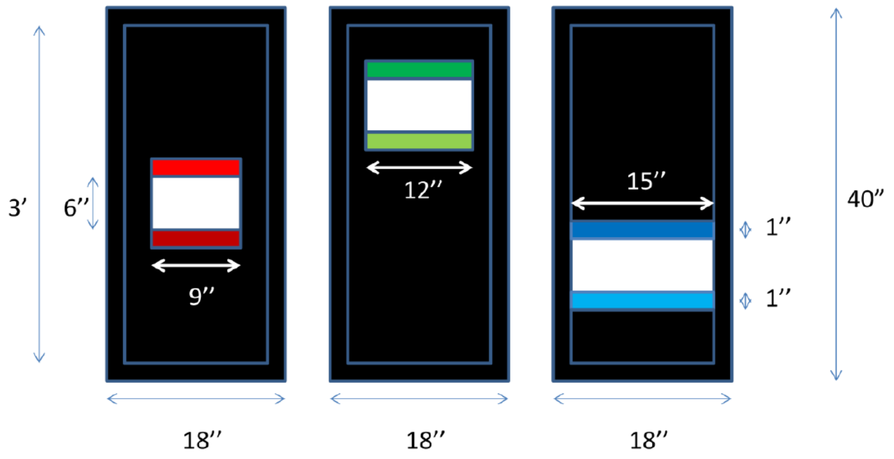


FIGURE 2.1: Target specification

The description of the first test is as follows. The device was given 15 seconds to aim at a target and then 20 seconds to fire 3 projectiles through it. Four trials were conducted. In the first two trials the target was a blue $15''$ wide by $2''$ tall target and an accuracy of $5/6$ was required to pass. In the second two trials the target was a green $12''$ wide by $2''$ tall target and an accuracy of $3/6$ was required to pass. Since vertical positioning is slower than rotational positioning this meant the machine must be able to travel $16.7°$ vertically in 15 seconds. The machine was also required to fire three projectiles within 20 seconds, constraining our feeding mechanism.

The description of the second test is as follows. The device was given 15 seconds to aim at a target. Then the device had 40 seconds to fire three Frisbees, shooting at at least two different targets. This was done four times in four trials. In each trial a red $12''$ wide by $2''$ tall target was paired with the blue and green targets already mentioned. The red target was worth 20 points, the green target was worth 4 points, and the blue target was worth 2 points. A minimum of 24 points was required to pass the test.

2.2 Coolness Factors

Several coolness factors for our project include: distance shooting, all mechanical engineering team, human targeting, and an innovative hopper design. While projects were only required to shoot through a target at a distance of $10'$, our device was capable of shooting through targets at $25'$. Our device was also capable of shooting Frisbees at humans $25'$ away, reaching them at chest height. Additionally, most groups had a mix of electrical and mechanical engineering talent, while our group was purely mechanical engineers. We still were able to produce the competition winning robot. Lastly, our hopper design, was very different from most groups and worked very well with minimal jamming or other problems.

Chapter 3

Design Concepts and Rationale

3.1 Concept Overview

The goal of this project was to develop a Frisbee Thrower that accurately and consistently launched Frisbees through targets $10'$ away. To accomplish this, the mechanism had to be able to locate the target, aim and position the launcher, and launch the disc consistently. Four different launching techniques were considered and compared in a Pugh Chart as shown in Figure 3.1. The techniques were evaluated based on expected accuracy, consistency, simplicity, controllability, and disc velocity. The chosen technique was a launcher that uses a rotating flywheel to propel the discs. It was believed that this technique would result in the best accuracy and be the simplest to produce. The firing apparatus was mounted on an adjustable platform that was aimed using motors controlled by an Aruduino MEGA microcontroller. A camera on the front of the mechanism was used for targeting. A hopper fed the discs to the launching mechanism one at a time.

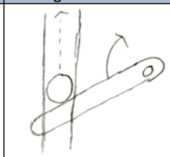
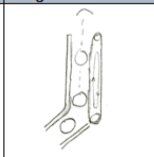
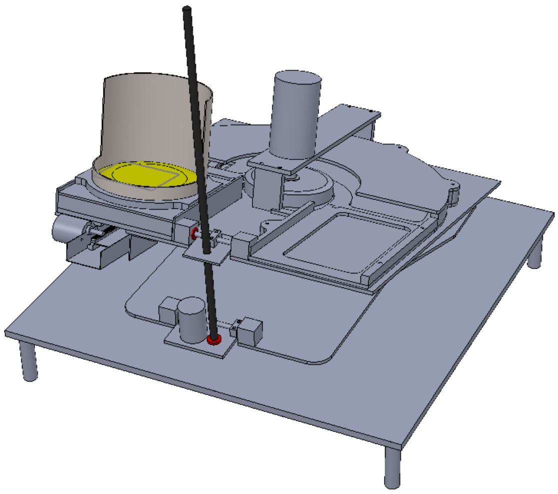
Pugh Chart					
Design type		Rotating Fly wheel Launcher	Rotating Arm Without Track	Rotating Arm With Track	Straight Belt Launcher
Sketch					
Criteria	Weight				
Accuracy	3	1	0	1	0
Simplicity	2	1	1	0	1
Controlabillity	2	1	-1	0	1
Disc Velocity	1	0	1	1	0
Consistency	3	1	-1	1	0
Net Score		10	-2	7	4

Figure 3.1: Pugh chart

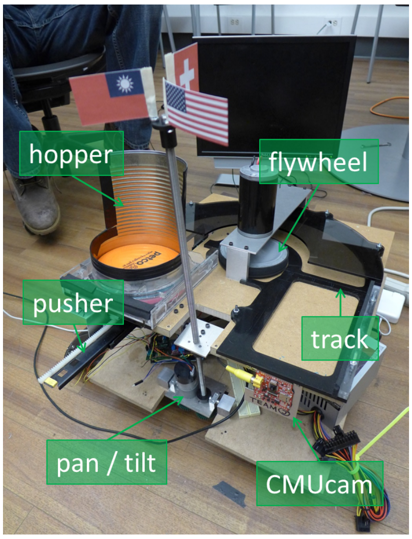
Given the constraints and requirements our team has developed a design with a variety of actuation and sensing capabilities to enable it to achieve these goals. Figure 3.2 shows the overall system with all of the components, which include: Frisbee hopper/loading system, the launching system, the rotational mechanism, elevating mechanism, and the sensing device (CMUcam4). Other components which are not pictured in the CAD include the Arduino board which was the microcontroller subsystem, and the power supply subsystem.

3.2 Systems Detail

The flywheel subsystem was mounted to the top plate of the launcher and actuated by a DC motor. The actuating motor was originally designed to be under the top plate. However, this constrained



(a) CAD design model



(b) Actual robot

FIGURE 3.2: System overview

the minimum vertical (pitch) angle the device could achieve. In the final device (Figure 3.2(b)) the flywheel motor was attached to a mounting structure above the plate. An acrylic track was also attached to the plate and positioned relative to the flywheel. The Frisbees travel along the 180° track during firing. The final subsystem seen on the top plate is the hopper and feeder mechanism. A rack and pinion mechanism with a triangular wedge pushed the Frisbee onto the track and into the flywheel when firing was desired.

The top plate was attached to another lower plate. One side was attached by hinges and the other was attached using the vertical positioning subsystem - a lead screw with nuts attached to pin joints. In order to reach the edges of the shooting range the bottom plate rotated relative to the base platform. This rotation was done using an RC motor and a VEX Robotics gearing device. This was chosen due to the ease of controlling an RC motor and the small panning distance required - 25.36 degrees. The vertical tilting was done by rotation of the lead screw using a DC motor. This was positioned as far from the hinge as possible so less force was required for lifting.

All of the electronics including the power supply were placed on the lower plate. The CMUcam4 was mounted to the top plate. This meant that the camera moved during both panning and tilting. This enabled us to do active feedback control using the camera. Both the vertical and rotational positioning systems would adjust themselves until the camera detected that the target was in the middle of the camera (with some predefined offset). Using this method we did not have to store position data or calibrate the device at the beginning of testing. When the camera detected that the machine was aimed accurately, the stepper motor actuated rack and pinion pusher was triggered. A limit switch detected when the pusher had travelled far enough for firing and reversed the direction of stepper motor.

All of our subsystems worked very well. Our hopper was the most innovative subsystem because the pusher actually dragged the individual Frisbees from underneath the Frisbee stack. It contacted the inside of the front edge of the Frisbee instead of the back outer edge as most other hoppers did. This meant that the pushing device was out of the way of the stack of Frisbees when they fell into position. The main reason for the success of our Frisbee launcher was the appropriate selection of our flywheel motor. Our Frisbees fired with a faster velocity than most other groups, causing them to travel with a straight flight path. This helped with consistency of firing. The main issue with our launcher was the speed of the vertical positioning. This subsystem used a lead screw that we had found, and that had shallower pitch than was desirable. For this reason an power booster was necessary to help the motor achieve speeds that would meet vertical positioning speed requirements.

Chapter 4

Architecture

4.1 Functional Architecture

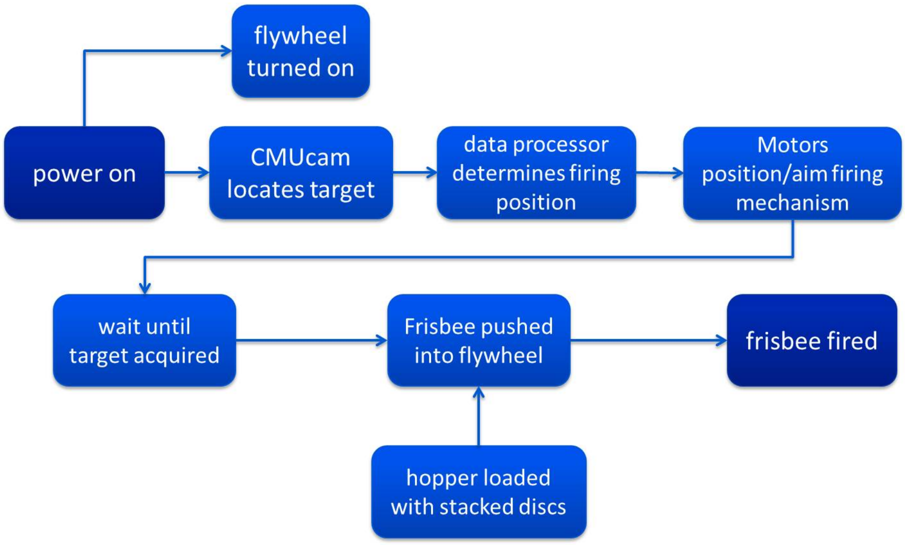


FIGURE 4.1: Functional architecture

The functional architecture of the Frisbee launcher is fairly simple and has only changed slightly from what we originally intended in the diagram above. The robot is turned on and detects switch configuration to determine what sequence to follow. The CMUcam locates the target and reports error based on how far off the robot is from being accurately positioned. The rotational and vertical positioning motors are activated and run until the error on position is very small. When the error is small for long enough the robot has detected that it is positioned accurately. It turns on the flywheel motor so it starts spinning. After waiting a short time to allow the flywheel to reach top speed, the feeder is activated and a Frisbee is pushed into the flywheel, firing it. One additional adjustment to this is that after the flywheel starts spinning we actually have the robot re-evaluate whether or not it is aimed correctly and reposition itself. This is because when the flywheel is turned on the robot sometimes rotates slightly.

4.2 Physical Architecture

Figure 4.2 shows the physical architecture of our system. Structural connections are shown by blue lines, the Frisbees path is green, actuation is purple, and data connection is pink. The structural frame houses the rotating base and the electronics. The rotating base is actuated by an RC servo

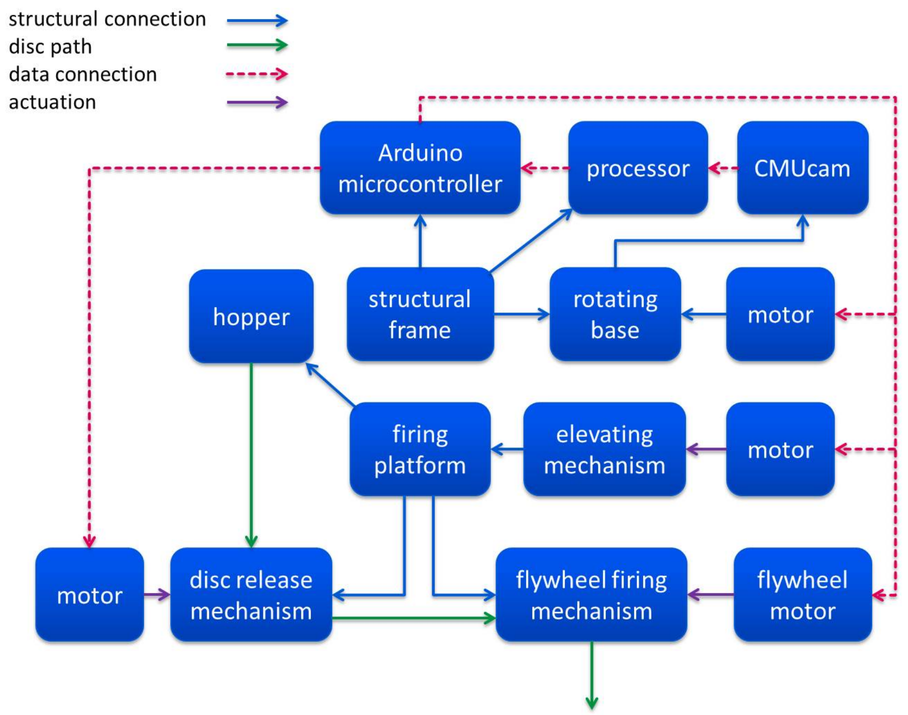


Figure 4.2: Physical architecture

motor. This RC servo motor is run by the digital pin out of the Arduino. The firing platform is attached to the rotating base and the elevating mechanism which is actuated by a DC motor. It is controlled with a motor driver board and three digital pins from the Arduino. To have more power, a power boost is connected to the 12V from the power supply and send 20V to the motor driver board. The hopper is also on the firing plate which the disc release mechanism attached below it. This is actuated by a stepper motor, controlled through a stepper driver board and two digital pins from the Arduino. Also implemented in this system is a limit switch that senses when the pusher has gone as far as it can pushing the Frisbee into the flywheel. This information is calling an interruption in the code and it is connected to a digital pin. The DC flywheel motor is controlled through a motor driver board to avoid high current change in the power supply and is controlled through one digital pin. The camera is mounted to the firing platform. The information is exchanged through two communication pins.

Chapter 5

Subsystem Descriptions

5.1 Mechanical

Our project is divided into the following sub-systems: sensing, rotating, elevating, loading and launching. The functional architecture shows how the subsystems interact with one another. Below is a description of each of the subsystems.

5.1.1 Sensing

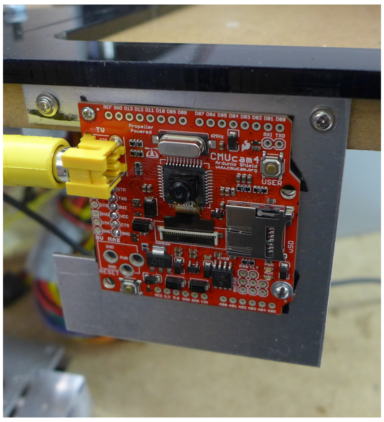


FIGURE 5.1: CMUcam4 for color tracking

The CMUcam is attached to the launching mechanism so that the camera moves and rotates with the system to apply accurate feedback control (Figure 5.1). We use CMUcam given its relative ease of use in interfacing with the Arduino MEGA microcontroller. It is capable of real time vision processing.

Once CMUcam sense the target position, it will control the elevating & rotating mechanism to move to aim the target. Using CMUcam we compute the horizontal and vertical error in our aiming. The rotational RC motor activates until horizontal error is minimized. The lead screw DC motor activates until vertical error is minimized. When error is sufficiently small for long enough the firing phase is activated.

5.1.2 Elevating

The elevating subsystem plays an integral part in our devices capability to address the accuracy constraints given in the project description. Using a small pitch lead screw we can control the launch

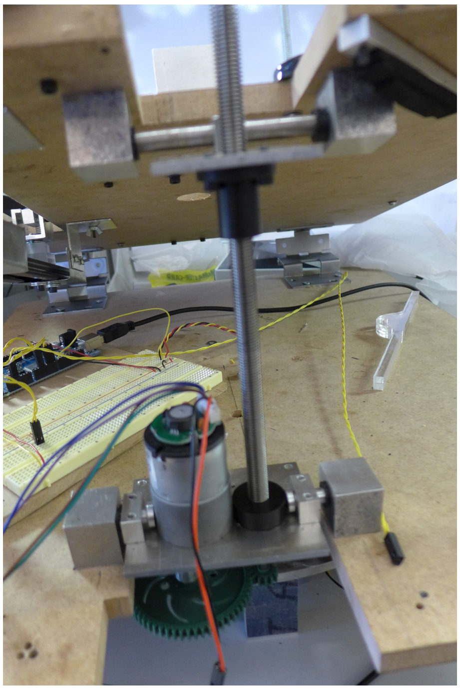
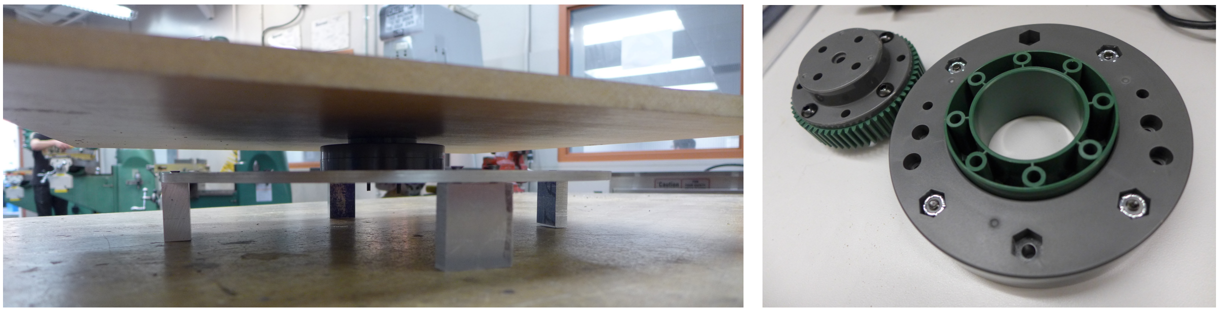


FIGURE 5.2: Elevating mechanism

angle with a large amount of precision. The vertical subsystem is shown in Figure 5.2. Our lead screw is attached by a nut on a pin joints to the top plate. This nut allows the top plate to travel up and down the lead screw. The pin joint was created by attaching mounting blocks with delrin bushings in them to the top and bottom plates. A steel rod passes through a mount on the nut. Bushings were chosen over ball bearings to cut costs. The lower plate is also attached using this type of pin joint. The lead screw fits into another delrin bushing pressed into the plate. The actuating DC motor is also attached to this plate so that it remains parallel with the lead screw. The motor and lead screw are then coupled together with two spur gears. We decided to attach the lead screw using pin joints that allow it to change angles so that the lead screw itself was not subjected to any torque. There was however some torque on the bushing it sat in causing binding. To overcome this issue a 1:1 gear ratio was used to couple the motor and lead screw instead of the 4:1 ratio originally intended. A DC motor was used here because it could be run at higher rpm than a stepper motor. This helped us overcome the small pitch angle of the lead screw and our inability to implement a 4:1 gear ratio. However, a power boost was necessary.

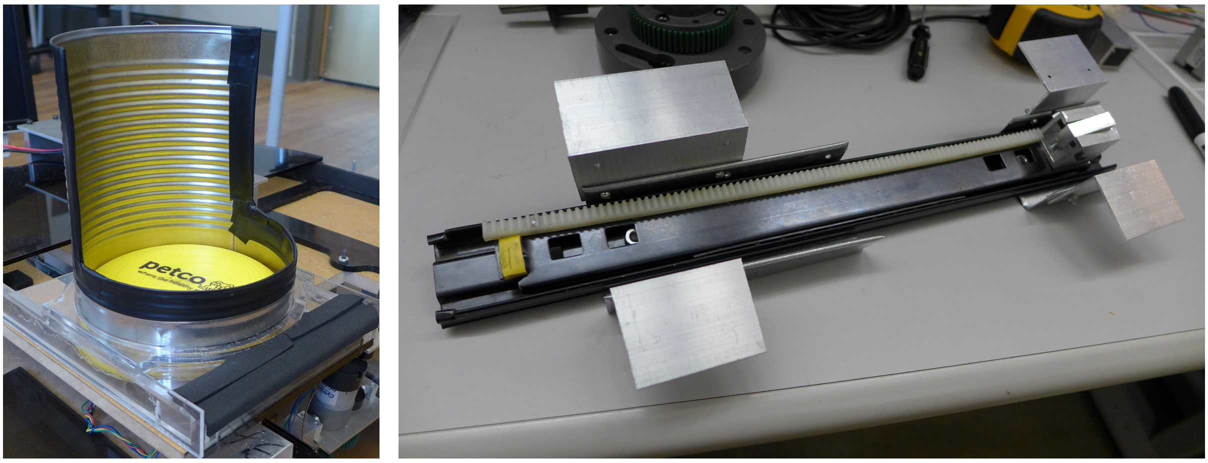
5.1.3 Rotating

We have the rotating subsystem outlined in Figure 5.3(a). The joint used is one purchased from VEX Robotics (Figure 5.3(b)). The joint works by attaching a motor to the larger diameter piece of the joint. A gear on the shaft of the motor then engages with the inner diameter piece of the joint. Rotating the motor and gear causes the inner diameter piece to rotate while the outer diameter piece stays fixed. For us, the larger diameter piece attaches to the base of our robot. The lower rotating plate then attaches to the smaller diameter piece.



(a) Rotational plates (b) VEX turntable joint

Figure 5.3: Rotating mechanism



(a) Hopper to store Frisbees (b) Frisbee pusher

Figure 5.4: Loading mechanism

An attempt was made to place the joint under the center of mass of the robot. This would help it overcome torque that could cause binding. The joint is however subject to torque that results from vertical positioning actuation. Though it is plastic it is strong enough to withstand these forces. The joint was actuated by an RC servo motor. The servo motor is easy to control and although it can only rotate 180°, and that rotation is divided by the gear ratio to determine rotation of the whole robot, it is still capable of rotating the robot 30° which is more than the 26° required.

5.1.4 Loading

Our hopper is shown in Figure 5.4(a) and the feeding mechanism is shown in Figure X. The core of the hopper is an acrylic plate with a Frisbee sized hole sitting on acrylic spacing walls. The spacing walls are sized so that only one Frisbee at a time can fit under the plate. To mount the Frisbee holding container, another acrylic plate was made with a slightly bigger hole concentric with the first plates hole. This was glued to the first plate with acrylic glue. It allows the Aluminum can container to be press fit onto the acrylic plates.

The feeding mechanism uses a linear desk drawer bearing with a pushing mechanism on it. The bearing is attaching to a rack that couples to a pinion on a stepper motor shaft. The stepper motor causes the pusher to slide back and forth. The pusher contacts the inner edge of the Frisbee through a slot in the upper plate. The pushing mechanism is a triangular wedge attaching to a small u-bracket with pin joint. A spring under the wedge allows it to be pressed down. This was designed so that the wedge could be compressed to allow the pusher to get back under the stack of Frisbees when it has finished feeding a Frisbee. However, in actuality the spring is strong and the Frisbees raise themselves a bit instead.

When the fire single is given to the stepper motor, it is run until the pusher mechanism contacts a limit switch. By this time the Frisbee it was dragging will have already made contact with the flywheel and launched. Once it contacts the limit switch the pusher reverses direction and travels back beneath the next Frisbee. When it has travelled a certain distance it reverses direction again, pushing the next Frisbee into the flywheel.

5.1.5 Launching

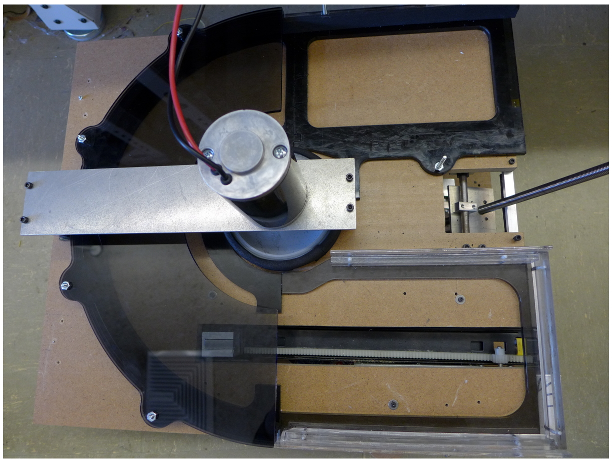


Figure 5.5: Launching mechanism

The launching subsystem is depicted in Figure 5.5. We have a spinning flywheel in the middle is powered by a large motor spinning at 5000 rpm. The motor is attached to plate mounted so as not to interfere with the track. Hand calculations were done to determine the necessary motor speed (Figure 5.6). They describe the relationship between motor speed, Frisbee angular velocity, and Frisbee linear velocity. Some video tests were conducted to determine desired Frisbee velocities. The motor is mounted over the flywheel and top plate as opposed to under the plate so that it would not limit the vertical travel distance.

An acrylic track was laser cut with walls dimensioned so that distance between the flywheel and walls was about 6″. This allowed the Frisbee fits snugly and will not vibrate or jostle around leading to inaccuracies. The flywheel and track walls were lined with neoprene foam. The foam compresses as the Frisbee is fired ensuring that it is enveloped by the wall and causing greater contact area and

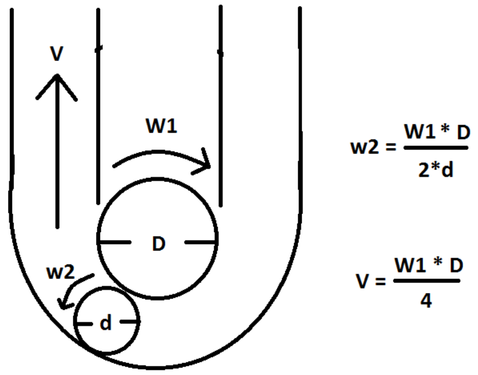


FIGURE 5.6: Velocity relationship

greater normal forces. This helped maintain a high friction force so that the Frisbee actually rotated in accordance with the determined equations. A cover was placed on the track to keep the Frisbee from rising and coming out of the track. This was necessary to constrain Frisbee motion.

To ensure that the flywheel did not wobble during firing, the other end of the shaft attaching to the motor and passing through the flywheel was inserted into an angular contact ball bearing. This prevented the shaft (and flywheel) from rotating. To allow a tight fit in the bearing an angular contact bearing was used that could withstand axial forces. This meant the bearing could actually support the weight of the motor and flywheel if necessary. Our launching system was one of the most consistent systems demoed and was a big factor in the robots ultimate victory.

5.2 Electronic

Figure 5.7 shows the layout between electronic devices.

Three switches are read once when the robot is turned on. This way we do not have to care about de-bouncing them or putting them on interrupts.

The limit switch is calling an interrupt because we need instantaneous response when the pusher hits the maximal position.

5.3 Software

A flowchart of Arduino software design is shown in Firgure 5.8.

First we start by checking for the existence of the color we want to track. We start with red, if it does not exist in the field we will go for green, and if not for blue. This way, if a color is not well calibrated, we are sure to track another one that work better.

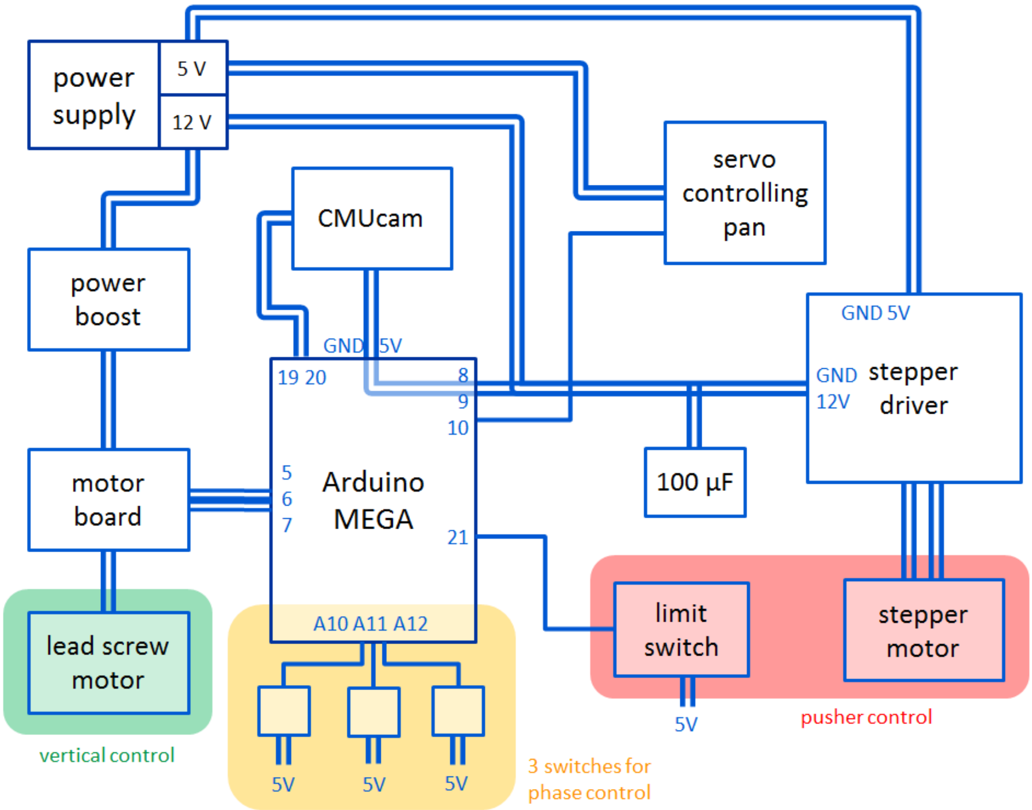


FIGURE 5.7: Electronic layout between devices

As the camera is moving with the robot, we are using direct feedback with a noise filter to have smoother data. Once we get close to the target, we start taking the mean of the error and go to the next phase only if it is under a threshold.

The first time we are in position, we just turn the flywheel on and go again for an aiming phase. This is done because the flywheel moves the robot when turned on. So it is required to have a good final aiming. On the last phase, the Frisbee pushing, there is a counter to limit the number of Frisbees shot in each target.

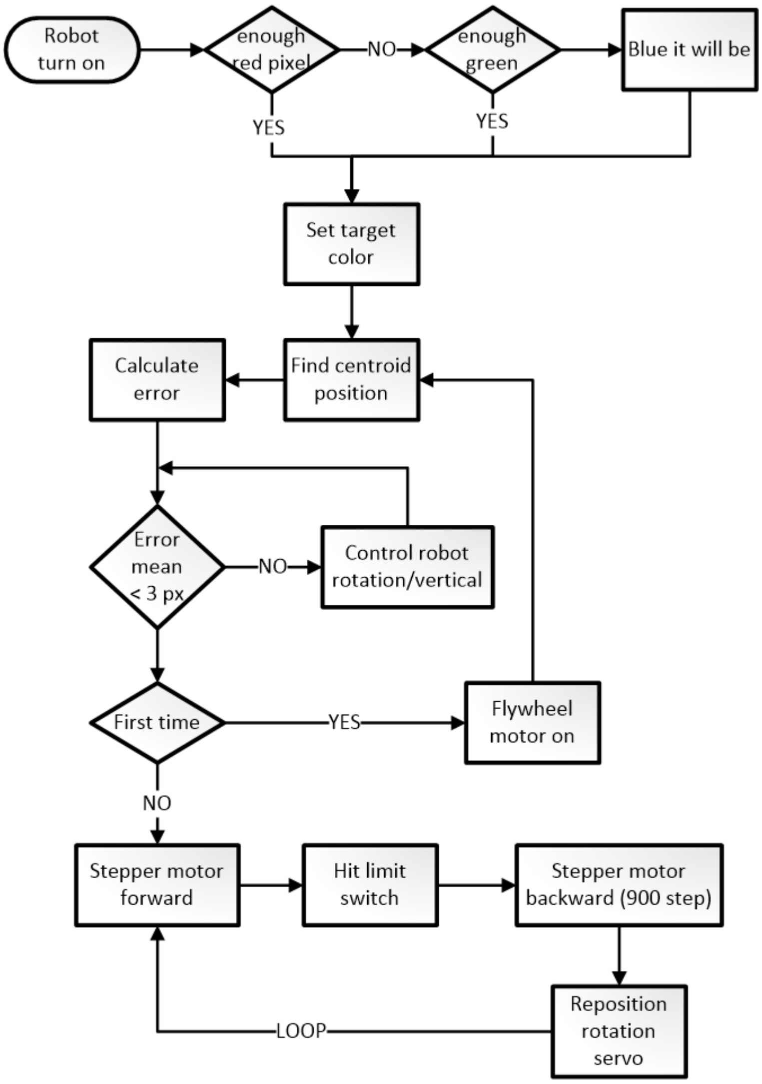


FIGURE 5.8: Flowchart of software design

Chapter 6

System Modeling, Development & Performance

6.1 System Modeling

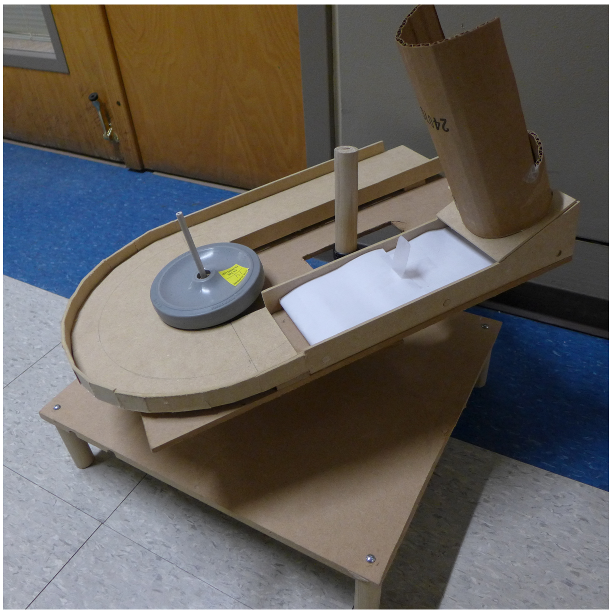


FIGURE 6.1: Design prototype

A mock up of our design was created using cardboard and other scrap materials (Figure 6.1). This helped us determine and evaluate the layout of the system. Afterwards a general CAD model of the system was created followed by detailed CAD of each subsystem assembled. One important calculation that was done was on necessary flywheel speed. This was already discussed in Chapter 5.

6.2 Development

Many problems came up during the fabrication and implementation of the robot. However, generally each theoretical subsystem worked as intended. There were both mechanical and electrical issues that we had to overcome. The electrical issues being the most difficult. Mechanically, one issue was getting the Frisbees to launch far enough. Our original choice of a rubber sheet to line the track was not supplying enough friction. Switching to neoprene foam solved this problem. Additionally the Frisbee kept flying off the top of the track. For this the acrylic cover was an extremely effective solution. Another issue that came up once our launcher fired with enough strength was that we could not fire low enough to hit the target when it was on the ground. To solve this problem we machined Aluminum spacers and placed them under the hinges (Figure 6.2). This lifted the back of the firing plate up allowing a shallower firing angle. Our biggest mechanical issue was the speed of the vertical positioning system. At first iteration the pin joint for the lower plate was not in line with the lead screw connection. This caused binding when the system was lowering so it travelled more slowly going down than up. As we added weight to the robot the problem got worse. The solution was to redesign the pin joint to be in line with the lead screw connection. This helped a bit but it still took a power boost and the elimination of our gear ratio to get the vertical system to move well.

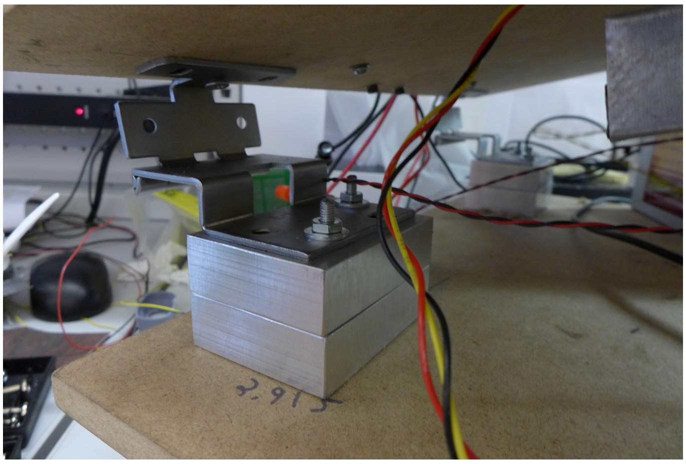


FIGURE 6.2: Aluminum spacers

There were also a number of electrical issues. A couple times after soldering together our final circuit board we would discover that it did not work and have to completely redo it. Additionally, after all the electronics were assembled, when the flywheel turned on it would cause the Arduino to shut off. It turns out not enough power was being supplied to the Arduino. The solution was to supply a 12V line to the Arduino instead of the original 5V.

6.3 Performance Evaluation

Our robot performed extremely well. Its worst performance came during the demo because at that time there were problems with the CMUcam. Target sensing was not very good during the demo, but we had to fire anyway. We still managed to meet the specifications of the project for the most part. Table 6.1 shows our demo results.

Our best performance came during the finals of the public competition when we had a chance to calibrate the robot better and when CMUcam was working. For this test we were allowed to fire 12 Frisbees (three per trial) at whatever targets we choose, with blue being worth 6 points, green being worth 9, and red being worth 12. We elected to fire all 12 Frisbees at the red target. In the finals we accurately fired through 11 out of the 12 red targets receiving 130 points out of the 144 possible (two

Table 6.1: Test performances

Test	**Our Results**	**Required Results**
Test 1 Blue Targets	4/6	5/6
Test 1 Green Targets	4/6	5/6
Test 2	28 points = Two Green Targets + One Red Target Hit	24 points

points were deducted for a slight human interference when it turned out we had not placed the robot down correctly). This performance outdid our expectations.

Our robots performance excelled in other ways as well. We were able to accurately fire through targets $25'$ away. Our hopper and feeder system very rarely got stuck. We were even able to fire at people enabling a fun game of catch.

Chapter 7

Conclusions & Future Work

7.1 Possible improvements

Overall the My Disc in a Box Frisbee launcher was very successful. There are a number of things that we would change and do differently if we did the project again. Weve discussed issues with the speed of our vertical positioning system. For this reason even with the power booster and other changes we made to the system, the time from turning on to launching a Frisbee was a bit longer than most other teams. Looking back on things we should have done more analysis on vertical positioning speed before we assembled the system. A lead screw with a much steeper pitch and fewer threads would have allowed for much faster positioning. It may have sacrificed some positioning resolution; however, the fine resolution we had was not necessary. Additionally the use of a ball screw instead of a lead screw could have helped avoid binding. A faster and more powerful motor would also be useful for this system. A general redesign of the system also may have improved speed. We could have used a four bar mechanism, a servo with gearbox directly on the pin joint, or a rack and pinion.

Another thing we should have done differently at least in terms of the scope of the class would have been to have completed our robot a few more days before the initial demo as we needed more time to debug. The week between the first demo and the public demo provided us with the time needed to correct the issues that it was having which allowed us to win the public competition.

If we were to create My Disc in a Box as a marketable product, we would need to use more professional machining techniques and fewer off-the-shelf components. We would use Aluminum platforms instead of wood. The platforms would be machined to a shape only necessary to support the structure and not square. Machined Aluminum would be done using a CNC machine. A linear bearing would have to be designed to replace the desk drawer bearing. Finally the excess lead screw would be cut off and the parts would be colored.

7.2 Conclusion

Overall our project was fairly simple and easy to put together. We were successfully able to overcome all of the problems that we ran into with the device, though some of which plagued us for over a week. Accurate analysis of our subsystems helped ensure that our robot did what we expected to. Ultimately My Disc in a Box was the competition winner for the class.

Chapter 8

Parts List

8.1 Cost & Description of Parts

Sub-system	Part	Unit Cost (USD)	Quantity	Total Cost (USD)	Description
Loading	Nylon Spur Gear Pinion	4.11	1	4.11	
	Nylon Spur Gear Rack	6.39	1	6.39	
	Stepper motor from class	30	1	30	drive pusher
	Limit switch	4.45	1	4.45	for pusher position control
Launching	3/4" adhesive backed neoprene foam	7.15	1	7.15	attach to track wall & flywheel to increase friction
	Acrylic Sheet 1/4" Thick, 12" X 24"	18.8	2	37.6	laser cut track
	Acrylic Sheet 3/16" Thick, 12" X 24"	15.62	2	31.24	laser cut track
	Aluminum rod	9.29	1	9.29	for flywheel assembly
	First CIM motor	24.99	1	24.99	drive flywheel
	Angular Contact Ball Bearings	5.95	3	17.85	hold flywheel in place
Rotating	Turntable Bearing Kit	19.99	1	19.99	major part of rotating
	Servo motor from class	30	1	30	drive turntable
Elevating	Lead Screw	24.58	1	24.58	
	Flange Nut	19.28	1	19.28	
	DC-DC Converter Module	16.99	1	16.99	power boost
	Motor Driver	27.95	1	27.95	control DC motor
	VEX Gear Kit	12.99	1	12.99	
	DC motor from class	30	1	30	drive lead screw
Structure	scrap material from shop	50	1	50	wood plates and some aluminum pieces
Sensing	CMUcam v4	99.95	1	99.95	in charge of vision
	Power Supply Unit	35.99	1	35.99	
	Arduino MEGA from class	58.95	1	58.95	microcontroller
	FTDI Cable 5V	17.95	1	17.95	for CMUcam debugging
			TOTAL	617.69	

FIGURE 8.1: Parts list