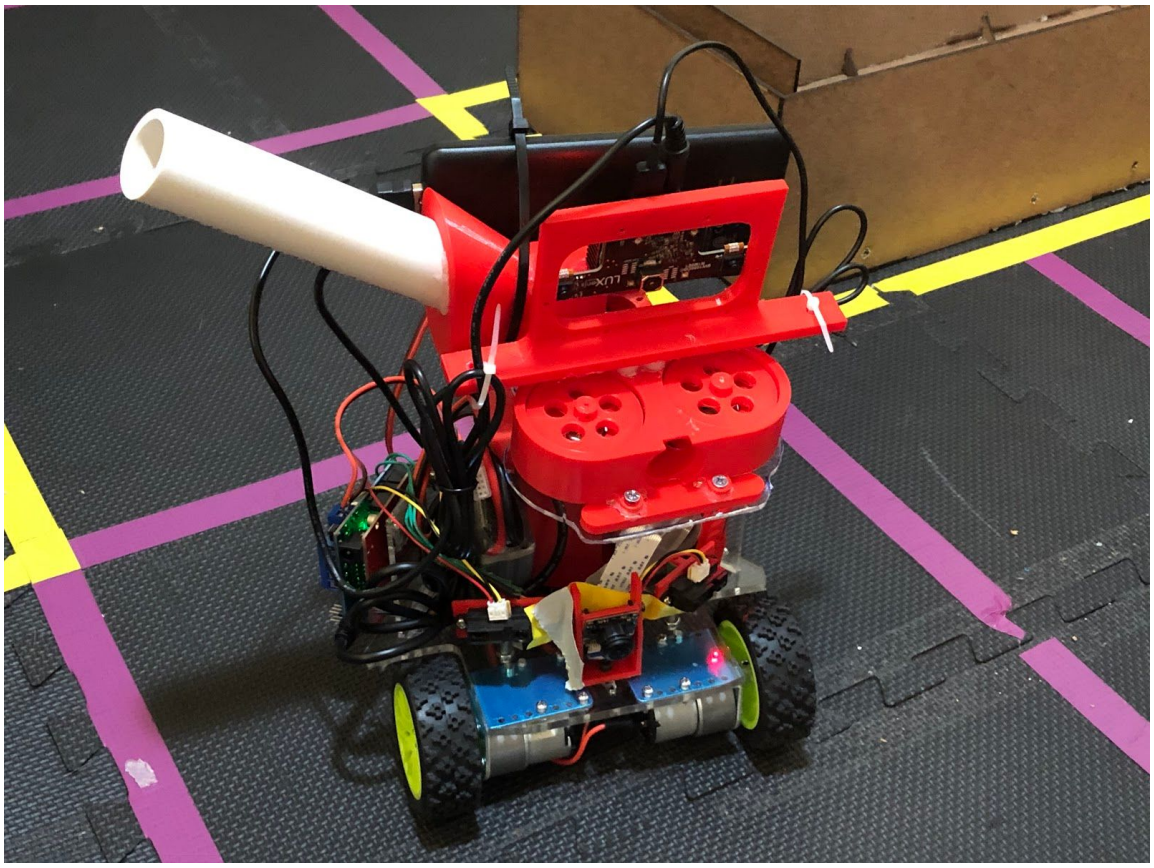
Design Report

MCEN 4115/5115 Mechatronics
05 May 2020



Team Master Blaster:

Cameron Casby

Graham Dean

Thomas Fugikawa

Riley Kenyon

Mitchell Scott

Professor Derek Reamon

Video https://drive.google.com/drive/folders/1dYghc2jL2LiEYdKZ0Kt69O3eU1Y9w5EW?usp=sharing

Github https://github.com/MithellScott/Mechatronics

DESIGN OVERVIEW

The RAMBO robot achieves its unique functions by integrating computer vision, targeting, and navigation capabilities with a flywheel firing mechanism. This allows its complex functions to be broken up logically between subsystems with communication coordinated by its microcontrollers and a microprocessor, two Arduinos and a Raspberry Pi. Course navigation is achieved with a Raspberry Pi and Pi Camera to detect the yellow lane boundaries by using a machine learning model. The heading is determined through classification, and the Raspberry Pi sends motor actuation information to an Arduino that controls the two driven wheels. This computer vision navigation system is supplemented by the use of IR sensors to provide additional information about the robot's surroundings. The robot follows the "right-hand" rule, always turning right in order to navigate its way through the course.

The Raspberry Pi is also used to receive information from a Luxonis camera for target recognition. This scheme takes advantage of the intrinsic benefits of both types of controllers. Communication between the Arduino and Raspberry Pi is accomplished with a serial connection over USB. An 11.1 volt 3S Li-Poly battery provides power to the Arduino and motor shields, whereas the power is supplied to the Raspberry Pi through a 5V regulated battery bank. The Luxonis camera is mounted inline with the firing mechanism such that the center of its field of view is colocated with the projectile's anticipated trajectory.

The robot navigates the course defined by the yellow tape while the Luxonis camera scans for targets. If a target is not detected by the Luxonis vision system, the Raspberry Pi evaluates the machine learning model with the current location given by the Raspberry Pi camera and determines the desired heading. Once a target is encountered, the state is shifted from navigation to targeting. The drive motors are used to reposition the robot such that the target is within the desired field of view of the Luxonis camera. The state then shifts from targeting to firing, in which the Raspberry Pi sends the firing signal to the Arduino, triggering the firing sequence.

During the firing sequence, the feeder mechanism is retracted, the flywheels are accelerated and gain momentum, and the projectile is ultimately forced into the flywheel cage which propels it towards the target. Once the target has been knocked over, navigation is resumed in order to traverse the remainder of the course. By performing these operations in sequence, the robot is able to systematically find and eliminate targets within any course with the proper lane boundaries regardless of its starting position.

SYSTEM BLOCK DIAGRAM

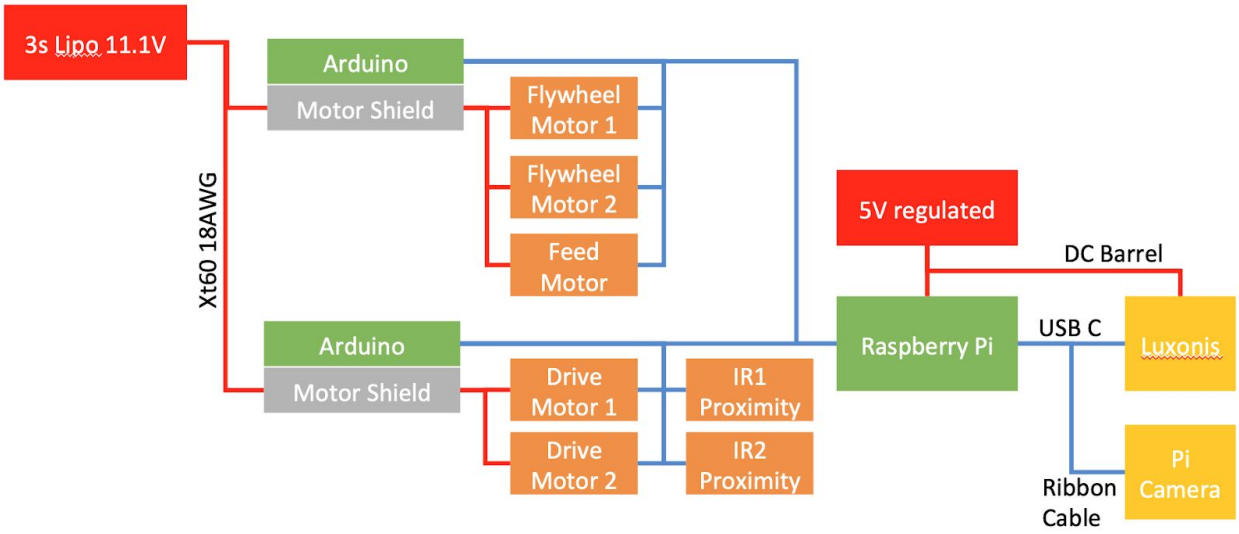


Figure 1: Communication schematic block diagram

SPECIFICATIONS

Software Ideology

The processing of the robot is accomplished on the Raspberry Pi with auxiliary support from the microcontrollers to perform general sensor acquisition and motor allocation. The purpose of the Raspberry Pi includes image processing and computer vision to return the centroid of a target, motion planning via a machine learning model to determine a heading based off the images from the Raspberry Pi camera, and communicating state information with the microcontrollers.

In order to achieve proper functionality, the robot was segmented into upper and lower systems that encompass functionality for the firing and navigation, respectively. This provided a logical breakup of functional responsibilities that corresponded well with the physical breakup of the robots subsystems. To run the functionality concurrently, multithreading is used in Python to operate both navigating and targeting simultaneously. The two operations share common flags to determine actions in each thread, the global flag used to communicate is the detection of a target which disables navigation while the targeting procedure occurs.

Originally planned was the use of ROS, robot operating system, to publish and subscribe to messages describing the states of the system. Due to time constraints and python environment issues, multithreading replaced the communication between navigating and targeting.

Vision System

Overview

The vision system is responsible for detecting the robot's environment. This includes the boundaries of the course as well as the targets. These two functions have been divided between navigation and targeting subsystems. The vision system consists of a Luxonis camera, a Raspberry Pi camera, and two IR proximity sensors as well as their mounting hardware. The Raspberry Pi camera and the IR sensors are used for navigation, so they are fixed to the chassis. The camera points forward to detect the environment directly in the path of the robot, and the sensors point diagonally forward to aid in collision avoidance with the course walls.

The Luxonis camera is used for targeting, and is mounted directly above the firing mechanism so that it remains in line with the trajectory of the projectile. The team designed a custom, 3D printed mounting bracket for the Raspberry Pi Camera in order to provide built-in adjustability and convenient pointing modifications during calibration and testing. The IR sensors were mounted on 3D printed mounting plates that allowed for pointing adjustments in the horizontal plane.

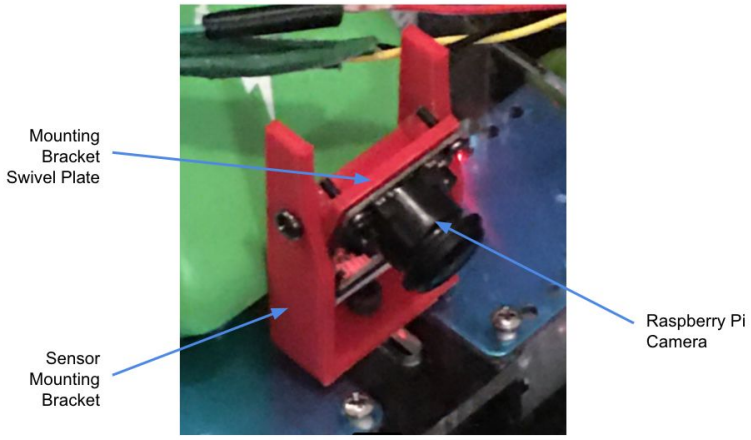


Figure 2: 3D printed mounting brackets allow for the sensors to be mounted and provide built-in pitch and yaw pointing adjustability

Navigation Functionality

The navigation system collects images of the course in front of the robot with a Raspberry Pi Camera. It then uses a deep learning model to predict a heading for the robot. This process is "memoryless," so it only makes decisions based on what the camera sees at the moment and does not depend on previous measurements, actions, or positions. This is implemented using a lookup table on the Arduino that associates each heading, rounded to the nearest 45 degrees,

with a motor allocation. Shown in table 1 are output options which are chosen based on trained camera inputs.

Bearing	Left Motor (% of maximum)	Right Motor (% of maximum)
-90°	-100%	100%
-45°	50%	100%
0°	100%	100%
45°	100%	50%
90°	100%	-100%

Table 1: Table of motor speeds depending on headings sent to the Arduino.

After choosing an output, the motor shield sends the appropriate signals to the driving wheels in order to achieve the desired motion. The robot is trained to follow a “right-hand” rule: when an intersection is detected, it always chooses to follow the wall on its right side. By following this rule, we can ensure that the entirety of the course is navigated systematically. The robot modifies the nominal speeds based on measurements from the IR sensors, slowing down appropriate wheels to better detect and avoid walls beyond the field of view of the navigation camera.

Design Decisions

As an alternative, the team considered using computer vision color segmentation to determine the current state of the robot from tape lines and make movement decisions based on what kind of intersection or other region the robot occupies. Image segmentation required much more computation and dealing with noise from the background. However, the team decided to use deep learning because it did not require creating a complex decision-making process from the ground up, and the team could gain experience with machine learning models. Some of the challenges with the deep learning model was collecting the right data so the model would make valid predictions. The first few generations of the model had issues where it would not turn down some of the more narrow alleys and it had trouble turning around after dead ends. The solution to these problems was to capture more sample images from these scenarios so the model would better predict the best action. The final model was trained on three different sets of images, in total 550 images. The biggest issue with the deep learning approach was the computing power. The camera and model cycle time ranged from 3 to 5 seconds which is not fast enough to keep up with where the bot actually is, so predictions would be made for where the bot used to be 3

seconds ago. A solution for this is to build a lighter model that can make faster predictions, with the trade-off being less robust predictions.

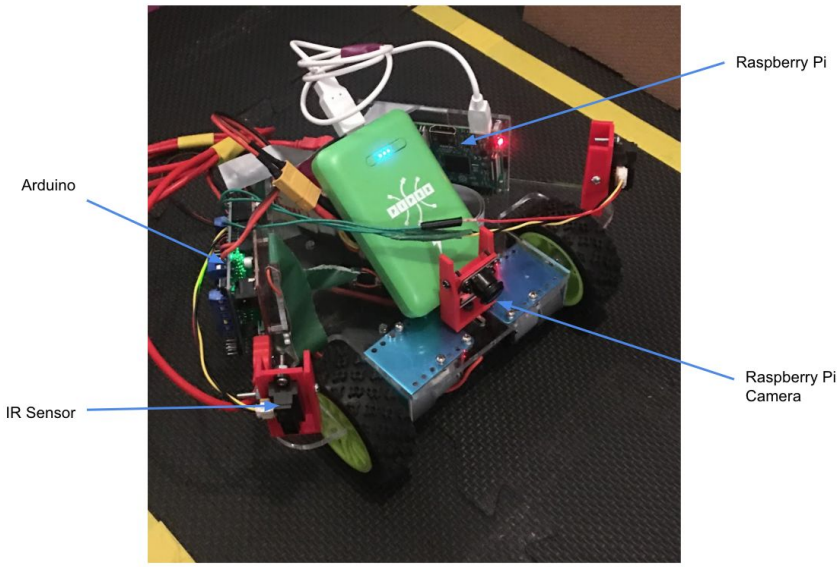


Figure 3: The navigation system consists of two IR sensors, a Raspberry Pi Camera, an Arduino, and a Raspberry Pi

Targeting Functionality

Targeting is achieved through object recognition by a Luxonis camera. This camera is mounted just above the “barrel” of the firing mechanism with a 3D printed mounting bracket. The targeting is achieved by using computer vision with masking based on the color of the target. Once a target is determined and the bounding box of the image is larger than a set threshold (to avoid false positives or detecting the target when knocked over), the system recursively evaluates images retrieved from the Luxonis to determine whether the target is near the center of the image. If the target lies off center in the field of view, the drive motors are actuated to rotate the chassis and re-orient the camera. Once the center is within a desirable range, the firing sequence commences and the Luxonis re-evaluates to see if the target was hit. If it was a success and there is not a target detected, navigation re-initializes, otherwise the firing sequence repeats.

Figure 4: The resulting bounding box from image processing of the target using a mask based on the color of the target

Design Decisions

These sensors were chosen for their form factor, ease of integration, reliability, and availability. The team considered using a pixy camera for target detection due to its high level of hard-coded object recognition functions. However, the Luxonis camera was eventually selected because it has a useful combination of built in object recognition functionality while still allowing for a degree of customization that was extremely beneficial. Although ultimately classical computer vision techniques were utilized on the Raspberry Pi, a preliminary object recognition model was created using tensorflow for the Luxonis. If time allowed, the model could be refined and exceed the performance of processing on the Raspberry Pi due to the on-board resources of the Luxonis and the Myriad X chip.

Figure 5: Luxonis camera (left) and Raspberry Pi camera (right)

Firing Subsystem

Overview

The Firing Subsystem is centered around a modified version of the flywheel launch mechanism used in the ‘Rival Jupiter” nerf gun designed by Nerf community member ‘Out of Darts.’ This firing mechanism launches small, spherical, foam balls by passing them between the small space between two rotating flywheels. In order to feed the balls into the flywheel mechanism, our team designed a custom feeder mechanism which also allows for storage of the balls in a PVC magazine. The motion of the flywheels and the feeder mechanism are driven by DC motors which are controlled by an Arduino and motor driver. 3D printing was selected as the fabrication method for all components due to the relative ease of creating complex geometries and the anticipated low loading conditions of the parts.

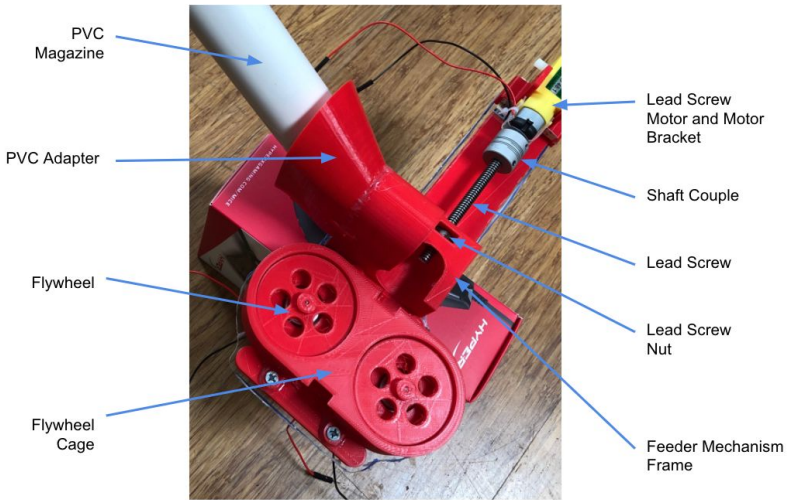


Figure 6: Image detailing the components of the firing mechanism

Flywheel Launch Mechanism

The flywheel launch mechanism utilizes two DC motors to rotate the two flywheels which are housed in the flywheel cage. These two DC motors are connected to channel A of the motor driver. The positive and negative leads of one motor are switched so that the motors run in opposite directions and achieve the required flywheel rotation. The space between these two flywheels is slightly smaller than the diameter of the deformable foam balls. Therefore, when a ball is forced into the space between the flywheels, the rotating flywheels “grip” the ball and propel it rapidly out of the flywheel cage.

Feeder Mechanism

The feeder mechanism utilizes a DC motor, a lead screw, a hex nut, and 3D printed structural components in order to create a custom linear actuator which feeds the balls into the flywheel launch mechanism. The feeder mechanism frame is not only the primary structural component of the system, but serves as a “chamber” for the projectiles, holds the hex nut in a stationary position, and provides a sliding surface along which the motor bracket can translate. The motor is supported by the motor bracket such that the shaft of the motor is coaxial with the central axis of the hex nut. The shaft of the motor is connected by a shaft couple to the lead screw which is then threaded through the hex nut. As the motor rotates, the lead screw is threaded through the hex nut, which is held stationary by the feeder mechanism frame. This causes the lead screw, the motor, and motor bracket to translate forward and backward achieving the desired linear motion.

The feeder mechanism frame has an adapter for a gravity-fed PVC pipe “magazine” which stores the balls prior to firing. When not in use, the lead screw is stored in the forward position. This serves as a physical barrier which prevents the balls from rolling into the chamber.

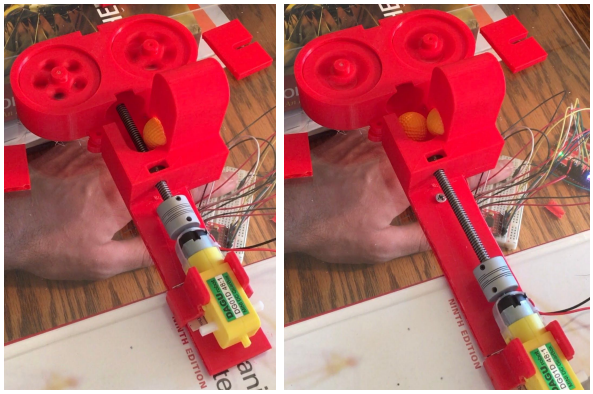


Figure 7: The lead screw of the feeder mechanism prevents projectiles from entering the chamber when in the forward position (left) when the lead screw is retracted a projectile drops into the chamber (right).

Firing Sequence

When the “Fire” signal is received from the targeting system, the firing mechanism goes through the following sequence in order to launch a projectile:

1. Feeder motor turns on in counter-clockwise direction
    a. Lead screw retracts
    b. Ball drops from the magazine into the chamber and is prevented from rolling forward by an angled tab on the bottom of the chamber

2. Flywheel motors turn on
3. Feeder motor turns on in clockwise direction
    a. Lead screw returns to forward position, pushing the chambered ball over the angled retention tab and between the two flywheels
    b. Ball is forcefully propelled out of the flywheel cage by the rotating flywheels

Design Decisions

As alternatives to the motor driven actuator, the team considered an off the shelf linear actuator, and a solenoid driven actuator. Some of these other preliminary designs can be seen in the appendix. After performing a trade study, we found that the decreased cost and complexity of the motor driven actuator made it preferable to the other options. However, the major downside of this choice is the length of the firing mechanism. Our team recommends that for future iterations of this design, options for decreasing size are investigated more completely. This could include offsetting the motor from the axis of the lead screw and transferring torque from the motor shaft with belts or a gear train.

Our team also ran into problems when loading multiple balls into the magazine. At first, the channel for the balls to drop into the chamber was slightly too small. This caused the seam of the balls to occasionally get caught against the 3D printed material. We were able to load and shoot multiple balls but would periodically run into this issue. In order to attempt to mitigate this, we widened the channel. However, after widening this channel, the ball was not restricted enough during the lead screw cycling process. The torque from the rotating lead screw now rotates the ball up and out of the chamber if a second ball is pressing the first ball against the screw. We then attempted to make a cover to prevent this, but it caused the ball to wedge itself between the cover and the lead screw, stalling the motor. As such we were unable to test with multiple targets without hand loading a ball after each shot.

Our team recommends that the feeding channel be redesigned so that the lead screw sits closer to the center of the ball rather than below it. We believe that this will make the ball less likely to rotate upward and out of the chamber when the magazine is fully loaded.

Firing Mechanism Mount

Original Pan-Tilt Mechanism

The pan-tilt mechanism provides the vision system and projectile mechanism with a means to orient relative to the features within the “playing field”. Mounting the two aforementioned systems on the pan-tilt mechanism enables the robot to scan the area in search of targets, identify enemy position, and fire a projectile at the target. Keeping these two subsystems together will simplify the relative geometry differences and trajectory calculations necessary to

execute a successful shot. The system will be capable of panning 340° and tilting 15° above and below the horizon. The exact capabilities will be determined once the exact geometry of the system is set, but a stepper motor or servo motor will certainly be capable of achieving these ranges. Figure 8 shows the pan-tilt mechanism, composed entirely of off-the-shelf components and 3D printed parts (primarily mounting structures).

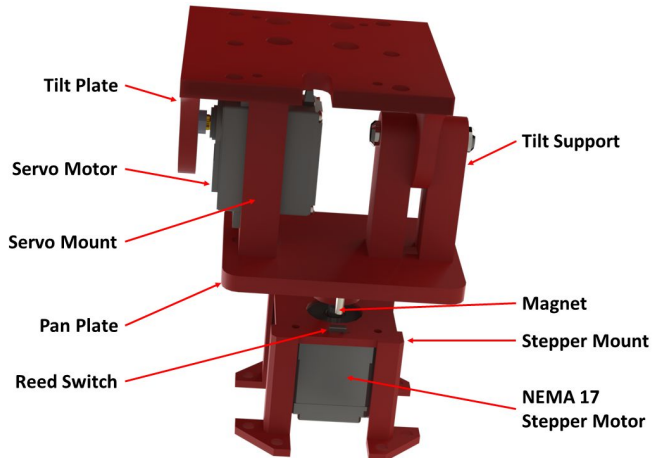


Figure 8: Pan-Tilt Mechanism

Following integration with the firing mechanism, a few problems were realized. Most significant of these was the heavier load and offset center of mass of the firing mechanism on the pan-tilt. This caused the stepper motor to work harder than intended to keep the pan-tilt in place, which led to overheating of the stepper shaft. This caused warping to occur at the pan plate-stepper shaft interface, leading to a significant “cant” in the remaining part of the pan tilt, and most critically, the firing mechanism. A few solutions were discussed, but due to a change in the course (namely the removal of an elevated, moving target), it was decided to remove the pan-tilt mechanism and replace it with a semi-fixed mount for the firing mechanism.

Modified Mount

The modified mount depicted in Figure 9 is fixed to the chassis and locks the firing mechanism into place. Due to some uncertainties, the team designed the mount such that the tilt angle could be manually adjusted outside of its normal operating conditions, to hone in on the ideal firing angle. Once the angle is set, the firing mechanism (and targeting camera) will always be geometrically-oriented the same in relation to the chassis and subsequent components. As a result of this change, however, the pan functionality was lost. This was solved by permitting the drive motors to make incremental changes to focus the firing mechanism at the center of mass of the target via slightly less fluid movements.

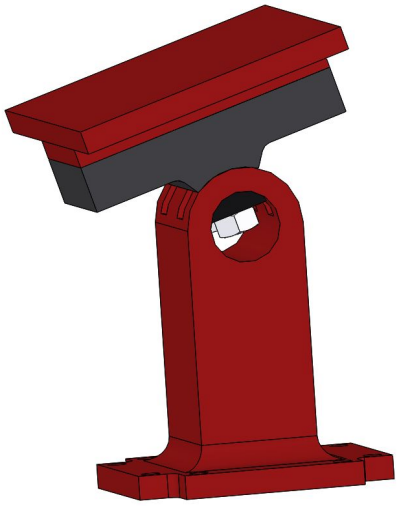


Figure 9: Firing Mechanism Mount

Chassis System

The chassis system provides mechanical interfaces for the various components of the system, as well as provides the necessary stiffness and structure to support the weight of the robot. This system consists of a triangular base plate and two vertical mounting panels made of acrylic. The base plate has rectangular cutouts that fit with tabs on the vertical panels in order to join the two together. By providing extra cutouts on the base, we have allowed for some adjustability of vertical panel position. The base plate also features mounting holes and cutouts for the wheels and their driving motors, as well as mounting holes for the navigation sensors.

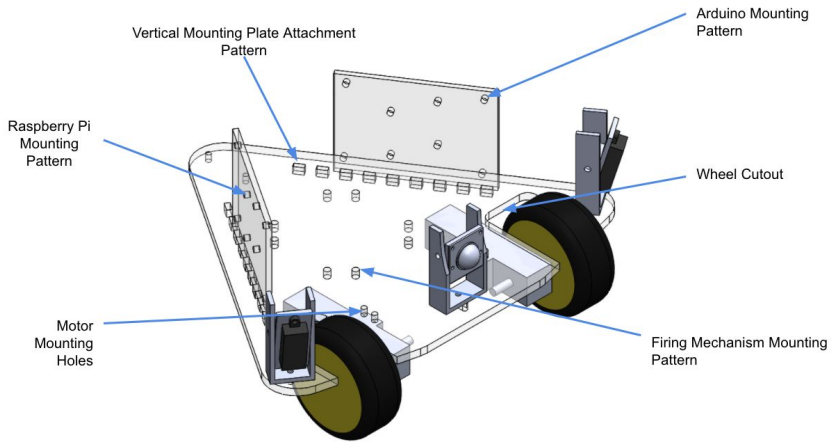


Figure 10: The Chassis system provides the structure required to support the weight of the other subsystems as well as provides their mechanical interfaces

The two vertical panels have hole patterns for mounting of the Arduinos and Raspberry Pi. By mounting these vertically, we can more efficiently utilize the mounting surface available on the base plate.

All components of these mounting plates were made by generating a CAD model and laser cutting the parts. This allowed for rapid manufacturing of these parts from inexpensive acrylic material in the hopes that iteration would be fairly simple as our design progressed. However, due to the impacts of the COVID-19 pandemic and the resulting lack of access to the ITLL laser cutters, we essentially had to freeze the design progression after our second iteration. This presented challenges as there are numerous changes that would have been made to the robot design to make it more streamlined, efficient, and organized.

Drive System

The robot is driven by two wheels, each connected to a DC motor. Stability is gained by the addition of a mount ball transfer that freely rotates. These two wheels, driven by DC motors that are powered by the 3S LiPo battery and controlled by an Arduino motor shield, provide forward and reverse motion, as well as vehicle rotation. Control of these DC motors is achieved by the navigation system.

The drive system not only provides the necessary mobility for course navigation, but also provides the pointing adjustments necessary to orient the firing mechanism towards a target when it has been identified by the targeting system.

DESIGN PROCESS

Our team followed an initial design process called the “7 ways” method in which each team member creates 7 different design concepts independently to inspire creativity. The general idea is that if you do not see the designs of other team members right away, you are less likely to be influenced by their designs, and the team is able to come up with a broader and more diverse range of design solutions.

While we did not create 7 concepts each, we aimed to develop around 3 overall design concepts per team member. After developing these independently, we met as a team and discussed our ideas, lumped similar concepts together, and weighed the benefits and drawbacks of each. At the end of this meeting we had downselected to one final design concept.

We broke up into smaller teams to divide the work among our team members. These teams took the lead on the more detailed design of their specific subsystem with input and help from the

other team members. In this way we were able to assign direct ownership of subsystems to a team member and each team member was able to obtain a design role within the project.

LESSONS LEARNED

This project has taught us many lessons both with respect to the application of the technical information that was presented during the course, as well as with respect to the challenges of integrating mechanical and electrical systems. This challenge was exacerbated by the fact that contact between team members was significantly limited due to the COVID-19 pandemic. This forced us to isolate the subsystems more than we would have liked during the design process. As a result, integration was not approached as early as it could have been. This demonstrated the importance of effective and efficient communication during the design of integrating systems and is a lesson that we will all take forward.

APPENDIX

System Requirements

Level 0:

0.a	The robot shall perform its functions autonomously
0.b	The total cost of the robot shall be less than $200
0.c	The robot shall have a maximum of two camera systems

Navigation System

Level 1:

N.1	The robot shall navigate the entirety of the shooting range

Level 2 (Derived):

N.2.a	The robot shall orient itself with respect to the course from within the starting areas
N.2.b	The robot shall not cross the road boundaries specified by 1 in. yellow tape
N.2.c	The robot shall detect intersections marked by 1 in. purple tape

Targeting system

Level 1:

T.1	The robot shall detect 9 x 3.5 in. red targets located throughout the shooting range

Level 2 (Derived):

T.2.a	The targeting system shall scan from -10 to 20 degrees in the horizontal plane (parallel to ground)
T.2.b	The targeting system shall scan from +30 to -30 degrees in the vertical plane (parallel to the robots axis of forward motion)
T.2.c	The targeting system shall track moving targets at a minimum speed of .03 fps (speed will increase for higher rounds)

Firing system

Level 1:

F.1The firing system shall shoot the 9 x 3.5 in. red targets located throughout the shooting range

Level 2 (Derived):

F.2.a	The projectile shall have a maximum density of 100 kg/m^3
F.2.b	The firing system shall have a maximum rate of fire of 1 round every 3 seconds
F.2.c	The projectile shall not travel higher than 6’ vertically when fired

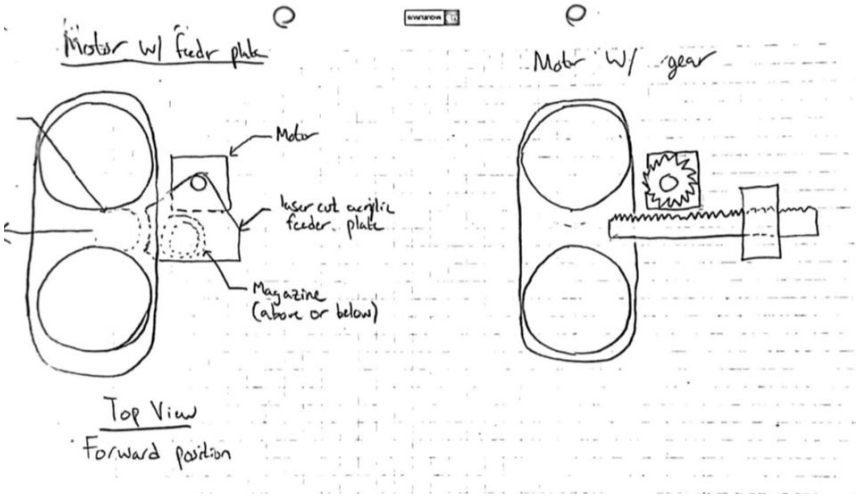
Mechanical

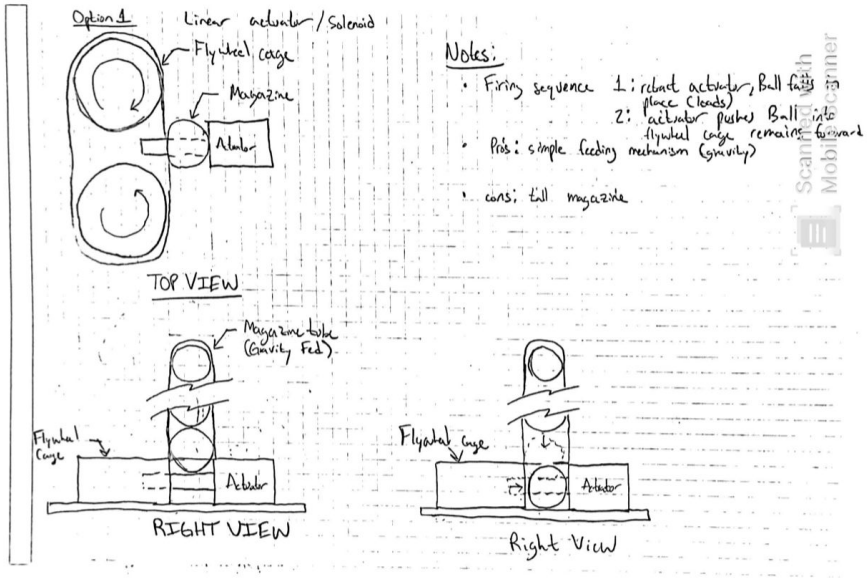
Level 1:

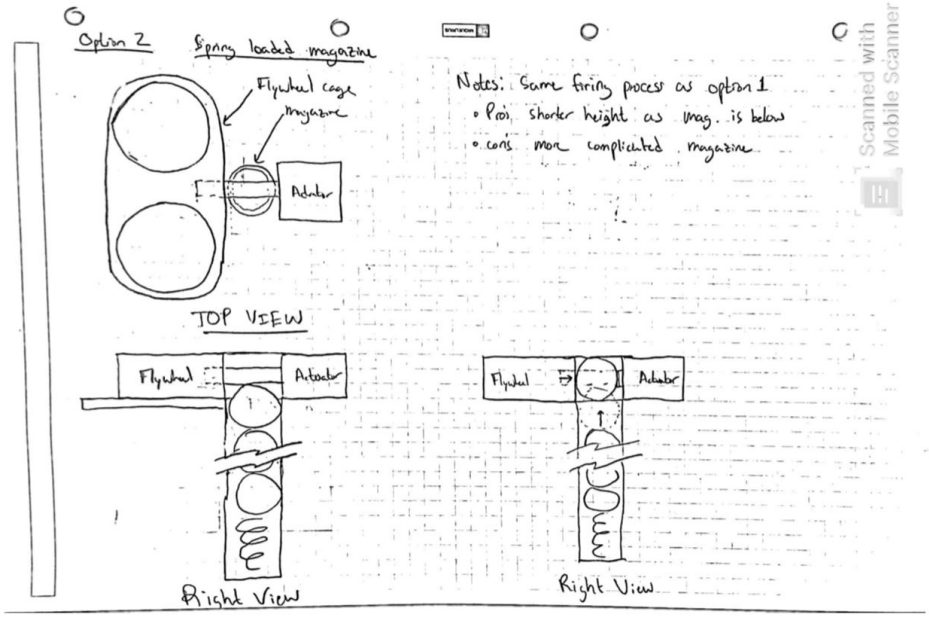
M.1 The robot shall fit in a maximum volume envelope of 14 x 14 x 17.5 in (L x W x H)

Preliminary designs

Preliminary rough designs for firing and feeding mechanisms:






Option 2
Spring loaded magazine
Flywheel cage
Magazine
Actuator
TOP VIEW
Flywheel
Actuator
Right View
Notes: Same firing process as option 1
• Pro: shorter height as mag. is below
• cons: more complicated magazine
Flywheel
Actuator
Right View

Mechatronics Project BOM

#	INITIALS	Component Name	Status	Assembly	Unit Price	Qty	Price	Spent	Vendor	ID	Link
1	CC	SunFounder Metal Gear Digital RC Servo Motor	Acquired	Pan Tilt	11.99	1	11.99	0.00	Amazon		https://www.amazon.com/SunFounder-Digital-Helicopter-Fix-Wing-Airplane/dp/B01KZKOIZC/ref=sr_1_3?dchild=1&keywords=servo+motor&qid=1587932011&sr=8-3
2	CC	Stepper motor - NEMA-17 size - 200 steps/rev, 12V 350mA	Pending	Pan Tilt	14.00	2	28.00	0.00	Adafruit	324	https://www.adafruit.com/product/324
3	RK	IR Proximity Sensor - Sharp GP2Y0A21YK	Purchased	Sensor	13.95	2	27.90	27.90	SparkFun	242	https://www.sparkfun.com/products/242
4	RK	Infrared Sensor Jumper Wire - 3-Pin JST	Purchased	Sensor	1.50	2	3.00	3.00	SparkFun	8733	https://www.sparkfun.com/products/8733
5	N/A	SparkFun Red Board	Acquired	Processing	x	2	x	0.00	ITLL		
6	N/A	Raspberry Pi 3 B	Acquired	Processing	x	1	x	0.00	ITLL		
7	TF	Wheels (Pack of 4)	Purchased	Chassis	10.99	1	10.99	10.99	Amazon		https://www.amazon.com/gp/product/B01HCWDDES
8	TF	12V Geared DC Motor	Acquired	Chassis	x	4	x	0.00	ITLL		
9	TF	Motor Bracket	Acquired	Chassis	x	4	x	0.00	ITLL		
10	MS	Raspberry Pi Camera	Acquired	Sensor	15.00	1	15.00	0.00	Amazon		
11	N/A	Luxonis Camera	Acquired	Sensor	x	1	x	0.00	Luxonis		
12	RK	Ball Caster Omni-Directional Metal	Purchased	Chassis	5.95	1	5.95	5.95	SparkFun	320	https://www.sparkfun.com/products/320
13	GD	Lead Screw & Nut	Purchased	Projectile	3.95	1	3.95	3.95	McMasterCarr		https://www.mcmaster.com/98935a703
14	GD	Shaft Couple	Purchased	Projectile	26.95	1	26.95	26.95	McMasterCarr		
15	**CC**	#4 Socketcap Screw	Acquired	Chassis	0.35	12	4.20	4.20	McMasterCarr		
16	GD	Acrylic	Purchased	Chassis	5.00	1	16.00	16.00	Colorado Plastics		
17	RK	DC Motor (130)	Purchased	Projectile	1.95	2	3.90	3.90	SparkFun	711	https://www.adafruit.com/product/711
18	RK	3S Lipo Batteries w/ XT60 connectors (Pack of 2)	Acquired	Power	29.99	1	29.99	0.00	Amazon		https://www.amazon.com/gp/product/B07Y67MKQB/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1
19	RK	XT-60 connectors (Pack of 6)	Acquired	Power	6.29	1	6.29	0.00	Amazon		https://www.amazon.com/gp/product/B01HENCZ7G/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1
20	RK	20AWG USB to DC 5.5x2.1mm Barrel Jack (Pack of 2)	Purchased	Power	5.78	1	5.78	5.78	Amazon		https://www.amazon.com/gp/product/B01MZ0FWSK/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1
21	RK	Heat Shrink Tube	Acquired	Power	7.99	1	7.99	0.00	Amazon		https://www.amazon.com/gp/product/B01MFA3OFA/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1
22	RK	18 AWG silicone electrical wire 2 conductor parallel wire	Acquired	Power	11.98	1	11.98	0.00	Amazon		https://www.amazon.com/gp/product/B07FMRDP87/ref=ppx_yo_dt_b_asin_title_o00_s01?ie=UTF8&psc=1
23	RK	Motor Drive Shield L293D	Acquired	Chassis	8.99	1	8.99	0.00	Amazon		https://www.amazon.com/gp/product/B00TMA4YSS/ref=ppx_yo_dt_b_asin_title_o00_s02?ie=UTF8&psc=1
24	RK	Tenergy Balance Charger for Li-Po	Acquired	Power	29.99	1	29.99	0.00	Amazon		https://www.amazon.com/gp/product/B004FGWY54/ref=ppx_yo_dt_b_asin_title_o00_s03?ie=UTF8&psc=1
25	RK	Nerf Rival 30 Round Refill Pack	Purchased	Projectile	11.83	1	11.83	11.83	Amazon		https://www.amazon.com/gp/product/B07DF73794/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1
26	RK	Geared DC (TT) motors	Acquired	Projectile	x	1	x	0.00			
27	RK	3D printed parts (rotary feeding mechanism)	Fabricated	Projectile	x						
28	TF	Wheel Mounting Hardware - 12mm hex to 6 mm (Pack of 2)	Purchased	Chassis	8.76	1	8.76	8.76	Pololu		https://www.pololu.com/product/2686