Faculty of Engineering & Technology – Smart Village
Mechanical Engineering Program

Mechatronics

Course Code: ME 359

Auto Turret

Authored by:

Essam Mohsen Essam	18107768
Omar Sherif Rabie	18107108
Joseph Raafat Attia	18107274
Amr Ashraf Abdelmawgoud	18107684

Submitted to
Dr. Abdelrahman Zaghloul
Eng. Mostafa Masoud

Table of Contents

Abstract 4

Chapter 1 Introduction 5

Background 6

Review 7

Primary Research 7

Secondary Research 8

Solution Preview 9

Decision Criteria 15

Chapter 2 Modelling 16

Face Detection with Camera 17

Locking and Tracking 20

Trajectory Analysis 21

Payload Delivery 28

Mechanical Design and Analysis 31

Design Criteria 31

Considerations and Limitations 31

Horizontal Movement 31

Tripod........ 32

Vertical Movement 34

Rollers 36

Motor Mount 37

Magazine and Bolt Mechanism........ 38

Chapter 3 Manufacture and Test 41

Manufacture 42

3D Printing and Material Selection 42

Assembly........ 42

Bill of Materials 44

Testing and Improvement 44

Mechanical Failures and Fixes 44

Electrical Failures and Fixes 48

Chapter 4 Conclusion 50

Conclusion 51

References 51

Appendix 53

Concepts 53

Software 54
Code 54

Figure 1-1: Illustration of laser guidance method of operation 9
Figure 1-2: Simplified illustration of rail operation principle 12
Figure 1-3: Drawing of a torsion onager, a close relative of the ballista with the same operational principle 13
Figure 1-4: Diagram of torsion spring in a torsion ballista illustration component and method of operation 13
Figure 2-1: Diagram showing different types of Haar features 18
Figure 2-2: Example showing the creation of an integral image from a test grayscale image..... 18
Figure 2-3: Example explaining the method of obtaining the sum of a rectangular set of pixels 19
Figure 2-4: Tracking system block diagram 20
Figure 2-5: Graph illustrating the possible ballistic trajectories at equally incremented elevations illustrating the envelope (done with MATLAB) 23
Figure 2-6: Graphical comparison between a linear drag and drag-less trajectory (done with MATLAB) 25
Figure 2-7: Velocity-time graph for a quadratic trajectory (done with ode45 in MATLAB) 26
Figure 2-8: Graphical comparison between quadratic drag and drag-less trajectory, note the large decrease in maximum height and range (done with ode45 and cumtrapz in MATLAB)............ 27
Figure 2-9: Trajectory plot for a quadratic trajectory, with higher launch speed to exaggerate asymmetry (done with ode45 in MATALB) 27
Figure 2-10: 3D model of FS5106B, initially used for horizontal motion 31
Figure 2-11: Side view of servo mount with the marked joints 32
Figure 2-12: Orthographic view of pod at different design stages 34
Figure 2-13: Orthographic view of simulation illustrating displacement distribution 34
Figure 2-14: Top view of SG90 horn vertical servo mount... 35
Figure 2-15: Top view of FS5106B horn on horizontal servo mount.. 35
Figure 2-16: Orthographic view of vertical servo mount assembly illustrating its motion.......... 35
Figure 2-17: Orthographic view of vertical and horizontal assembly servo mount 36
Figure 2-18: Orthographic view of toy DC motor 36
Figure 2-19: Photo of toy DC motors used for projectile delivery 36
Figure 2-20: Top and Orthographic view of rollers used in projectile propulsion 37
Figure 2-21: Motor mount assembly front view 37
Figure 2-22: Motor mount assembly orthographic view 37
Figure 2-24: Side view of magazine and mechanism design illustrating locations of attachment38
Figure 2-23: Orthographic view of SG90 servo motor used for bolt mechanism 38
Figure 2-25: Side view of bolt mechanism assembly, with arrows illustrating slider crank mechanism 38
Figure 2-26: Orthographic view of magazine design 39
Figure 2-27: Top view illustrating close up of motor mount location... 39
Figure 2-28: Orthographic view of complete design assembly 40

Abstract

This report is for an autonomous turret project, it uses a USB Webcam connected to a laptop, which applies a face detection algorithm on camera live feed. This algorithm is applied by running a python face detection code that utilizes OpenCV library as well as others accompanying it, then the face coordinates are communicated to an Arduino controlling a system of motors that steer the camera around to home on the target's face. This, combined with a dual motor dart propulsion system, constitutes this project make-up, as the system launches a dart at the target once it is close enough to the centre of the screen i.e. The aiming point. The concept of this project was inspired by the advances in modern weaponry, as more and more countries acquire smart weapon systems such as loitering munitions and robotic weapons systems and drones. The auto-turret can function as a defence system in wartime against aircraft and ground targets albeit with modifications to the ammunition and input target specifications. It can also function in peacetime as security in public events for protection against recorded criminals as well as a surveillance system if disarmed. This will save manpower and reduce security costs on vast scales, as it works 24/7 continuously, disregarding the need for maintenance, and doesn't require any of the costs that traditional security requires. In addition to that, there is far less security risk as it is much more difficult to infiltrate such security systems than human ones, and much easier to monitor and respond to when malfunction occurs. The servo motors were picked for their precise positioning and internal feedback, while the DC motor propulsion system was chosen for its light weight, low power, high speed requirements, since the dart projectiles are low in mass as well as its cost-effectiveness.

Chapter 1 Introduction

Background

The existence of turret defence systems stretches back to ancient times. The term's definition is broad, going far back as one of the oldest examples of siege warfare, the siege of Troy, as semi-mythic as it is. Ever since the dawn of civilization, people and nations have been looking for efficient ways to defend themselves from aggressors and continually improve them. As necessity is the mother of innovation, the need for protection gave birth to many innovations, one of which is the watchtower. It covers vast areas that can be monitored and protected from minor incursions by a few men utilizing their height advantage. This concept has survived centuries of changing aspects in warfare; the Syracusans defended their city from the Romans using such towers, so did the Crusaders of Acre more than a thousand years later. There was not much advancement in the technology, other than improvements in firepower, such as the addition of torsion catapults and the later introduction of gunpowder and firearms. Until recently, improving the weapon's destructive capability rested purely on its firepower. Lately, the wave of automation has reached turrets, with the introduction of weapons such as the Close-in Weapons System (CIWS) by the United States in 1973 [1], Israel in 2007 with the Roeh-Yoreh [2] as well as many other countries. The focus was shifted from increasing firepower to taking the human element out of the loop, thus decreasing the response time, which is vital for defending against high-speed attackers as well as making it possible to eliminate enemy munitions mid-flight. This shift has spread and is spreading faster now than ever to other countries. This project aims to innovate, design, and implement a non-lethal prototype sentry gun that is autonomous in some respect, offering target tracking, trajectory prediction, and independent decision-making on the feasibility of scoring a hit. As offensive weapons technologies advance so must defensive weapons technology to maintain the balance in the ongoing worldwide arms race thus we must aim to make progress in autonomous weapons field. This project is a manifestation of this progress, as much as it is a prototype, it is also a step in the right direction. [1]

Figure 1-1: Photo of CIWS in operation during a training exercise

Review

Before committing to this project or making any decisions with respect to it, a search was conducted on the internet, to see how those that did similar projects went about it. As the system is meant to be small scaled, non-lethal to the point of being a toy, advanced weapon systems in possession of modern armed forces were avoided, with more focus on other engineers' and enthusiasts' DIY projects, which squarely sits in the vision on what the project should resemble. Thus, it was agreed that such projects should be the primary object of inspection, with the secondary object being more advanced large-scale systems.

Primary Research

Several creators on the internet have tried and succeeded in making systems similar to this one, one such as example that will be examined in this section, is a system that utilizes a similar servo-based tracking system with an Arduino microcontroller, with a pneumatic shooting system. The system exclusively targets human faces using a webcam and a python code running using OpenCV to detect faces, the face coordinates are then communicated to the Arduino microcontroller, using Pyserial , as an error relative the screen centre, upon which the decision making is done. Servos in each of elevation and pan, move in opposite to the error in fixed step sizes i.e., working in a negative feedback system to aim at the target and track it as it moves. The criterion for shooting is when the error is lesser than a certain threshold value, each separately defined in the Arduino code. the camera system is stationary, being at a fixed orientation throughout the entire process, with the propulsion system on the turret being the only moving part in the project. The pneumatic shooting system utilizes a compressor and a receiver system, where air pressure is accumulated over time and released in a short time span to produce a force pulse that launches the nerf dart. Another example of such a project is a nerf turret that use motor friction as a propulsion method, with a manual aiming system using the mousepad as the aiming input, such that the servo motors follow the cursor's motion about the area designated by the program. The creator creates this interface using a python program that translates the mouse motion into an error which is communicated to the Arduino through an app, for which an android version exists as well, so that the aiming can be done by touchscreen as well as a mouse cursor. [2] [3] Most other projects of the similar type and calibre done previously don't differ much in operational structure; one system acquires a target, one maintains the target, then other dispatches it, all integrated together as one mechatronics system. The electrical, mechanical, as well as control system design are done together in parallel, integrated together, thus the trial-and-error phase of the project was vastly shorter than if it had been done otherwise.

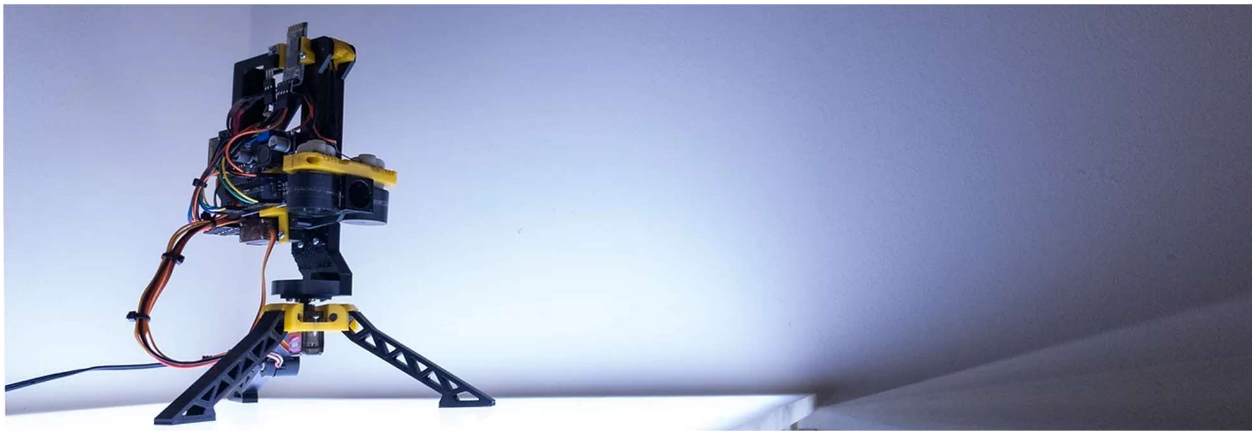


Figure 1-2: The Nerf turret done by littlefrenchkev

Secondary Research

Several countries have autonomous weapon systems that are intelligent, with independent target identification, tracking, and dispatch capability. The earliest example of such is the United States Navy Phalanx CIWS, that uses a radar-based target acquisition and tracking system, capable of shooting down live enemy targets as well as munitions such as cruise missiles on sea while countering mortar & artillery shells on land. These systems follow the same blueprint; (at least) one subsystem for acquiring a target, one for tracking, usually combined with the aiming system to track fast moving targets of the order Mach ~1 allowing for the compensation for impact time and lead angle, followed by a subsystem for dispatching the target, it is rather the pattern than not that such subsystem are integrated on several levels as this provides the optimum performance that is absolutely critical to such fields. This is a similar blueprint to the one followed by the creators mentioned in the primary research section, where the target tracking and aiming system are not necessarily integrated together. The CIWS utilizes a multi-sensor target acquisition subsystem, where operators can scan the field visually and identify targets prior to engagement with the added aid of a K_u-band fire control radar system for both acquisition and tracking, noting that both need not use the same sensor type or system. For example, the radar system utilizes two antennas working in tandem, one antenna is for search and acquisition, scanning a wide area, being precise enough to detect a target but not enough to continuously provide high quality information on it, that is the mission of the latter antenna, scanning a much smaller area in exchange for providing high quality positional information (bearing, range, range rate, altitude, velocity) with which it fulfils its mission of tracking the target. The tracking system is also aided by a forward-looking infrared sensor that can detect and continuously provides feedback on targets' heat signature location. This allows tracking of targets that may be less visible on the radar or infrared sections of the electromagnetic spectrum using sensor fusion algorithms. This system, while enjoying high levels of autonomy, must subscribe to human oversight to prevent friendly fire incidents and prevent collateral damage. The target dispatch system has consistently been the infamous 20 mm M61 Vulcan Gatling autocannon, the same gun the A-10 Warthog is armed with. This combines high firepower, a heavy round, independent and precise tracking, which maximizes the destructive capabilities of the weapons system. As of April 2017, Raytheon tested a novel electric gun that allows the Vulcan gun to unload at a variable rate for the aim of ammunition conservation. [4] [5] [6]

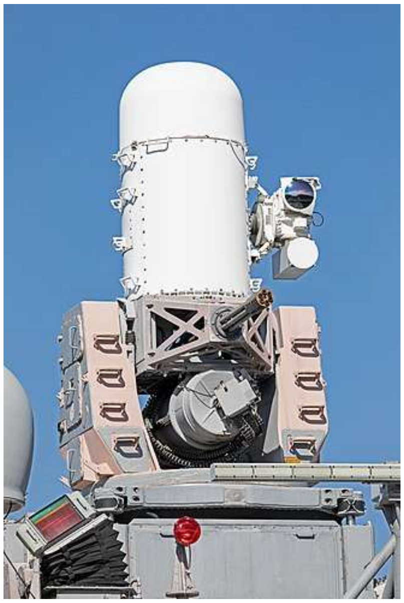


Figure 1-3: CIWS Block 1B

Solution Preview

Target Acquisition System (TAS)

Several TASs were suggested by team members: -

Laser Guidance

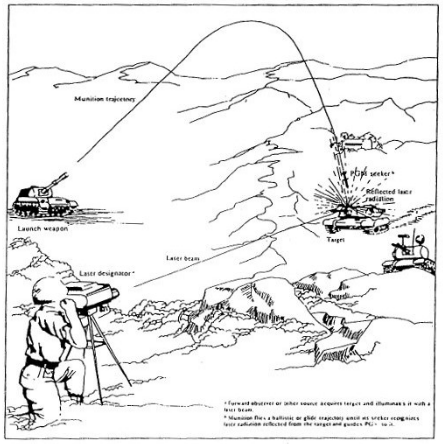

Munition trajectory
PGM seeker*
Reflected laser radiation
Launch weapon
Target
Laser beam
Laser designator*
* Forward observer or other source acquires target and illuminates it with a laser beam.
* Munition flies a ballistic or glide trajectory until its seeker recognizes laser radiation reflected from the target and guides PGM to it.

Figure 1-4: Illustration of laser guidance method of operation

The review covered the consideration of laser-guided munitions; it piqued the team's interest, as it seemed cost-effective to simply use a laser pointer and have a sensor or camera detect the dot's position where the laser meets the target. This position vector can then calculate the trajectory required for a hit and output to the gun carriage's required elevation and latitude angles. This method has the advantage of being versatile, as a laser can be pointed from a user placed virtually anywhere, thus making target designation simple and easy. However, there comes the question of how the dot produced from the laser can be detected; it was suggested to use a webcam and a laptop for image processing to detect the dot; however, there are several limitations, such as image resolution laser, dot size, and accuracy. A camera with too low a resolution would not show the dot placement; this effect becomes much more visible as the dot becomes farther and farther away. The other challenge faced by this method is that an algorithm is needed that can reliably detect bright dots or laser markers, which is very difficult due to the dot's resemblance to many everyday light effects, such as light reflection from glasses, windows, screens, any dot brighter than its background is a potential error of target misidentification. It was deemed unreliable, and in addition to that, the product would not be in the envisioned aim, not to mention it would only produce a 2D image of the object (only θ and φ coordinates would be retrievable with no 3D position data, i.e., range).

Rangefinder with Camera

As a modification of the laser guidance system, attaching a laser on the gun carriage was proposed so that the laser light ray axis would be parallel to the barrel's axis, as a marker of the barrel orientation, similar to the modules on several existing firearms. It was further postulated that, whether the laser can be replaced with a rangefinder. Coupled with a camera, such a system can fulfil the requirements in regard to target acquisition, as the radial distance-to-target can be measured, and combined with the camera inputs and, can be used to calculate all 3 coordinates required to relate the target point in space to the ballistics calculator that aims the barrel correctly. Already used in tanks and other direct fire systems, this system may be the ideal method for the subject application, it provides all the coordinates as well as being relatively simple as it only requires the camera to home on-target. That is only in theory unfortunately, as the market in Egypt for rangefinders or proximity sensors is small, at least for those within budget. Most of the sensors found in the search, had ranges of less than 1 metre, which did not fit the initial project criteria of ~10m. This idea was not within the team's logistical abilities; upon realizing this, the idea had to be modified to be feasible for a small-scale project.

Camera with Computer Vision

This system is a modification of the previous idea, it was decided to let go of the range finder instead attempt to utilize a camera aided by a computer vision face detection algorithm. Although this system is incapable of recognizing non-human objects as targets, this loss of generality is well made up for by its specificity, precisely its ability to focus on humans' faces as targets. The computer vision algorithm detects faces and puts a box around them, which would be the target for the Projectile Delivery System (PDS). This system was preferred opposite to laser markers as faces have large dimensions, thus there is wider tolerance for error (bullet drop would be less of a problem). For the range criteria (~10 m), this removes the need for a range finder, as the need for elevation compensation required would be miniscule. The underlying issues with this method, as stated previously, are that it cannot be generalized to multiple target types, yet even with an algorithm well capable of detecting faces, all that is needed to avoid is to turn around or a change of lighting. It can be easily tricked or avoided altogether, however this was not a good reason to let go off this idea, as no human is realistically expected to stay still in a certain pose while in range.

Target Tracking and Lock System (TTLS)

In order to track the target and maintain a lock on it, there needs to be continuous feedback from the camera to the motors responsible for the motion of the gun carriage. The camera cannot measure the error in a physically meaningful form, such as angles or distances, it can only read an error in pixels. This is satisfactory in terms of locking on the projectile, however it makes the system unable to hit smaller, faster targets, as such targets require a lead angle to adjust for the distance travelled by the target during the projectile's time-of-flight. Taking this constraint into account, there would be no point for a single battery system to track a target it cannot hit. Thus, the planned tracking system wouldn't suit fast-tracking, the focus was rather shifted on slower-moving targets- in plain technical terms, this design has no need for high-speed motors, requires precise positioning, & continuous feedback. This is precisely what servo motors are, thus the choice was made that servo motors are to be used.

Ballistic Aiming System (BAS)

Classic Trajectory Calculation

In a classic exterior ballistics problem, the user is given target coordinates and is demanded to determine the possibility of a hit using the system given to them, which varies widely as there are many systems currently in use, from anti-aircraft guns to ICBMs. Each system is treated differently according to its specifics, such as its range, area of use, etc.

In this case, the team consulted Ballistics: Theory and Design of Guns and Ammunition by Donald E. Carlucci & Sidney S. Jacobson. The trajectory was treated as that of a point particle then went on to see the effect of air resistance, depending on the payload of choice. This system is satisfactory in terms of accuracy, as it allows at least one trajectory to be calculated for a target, given the target's position in space, which is practical for systems that have sensors capable of acquiring target coordinates.

Bullet Drop Compensation

This approach is the one more commonly used in cases where target distance is difficult to discern with meaningful accuracy, such as in sniper situations prior to the commercial prevalence of rangefinders. Even nowadays it is not always safe to use a range finder on sniper missions. However as long as some estimate of the range can be made, or if range is constrained as is the case here, there is still hope for a hit given that the target is large enough. For small ranges, or high enough speeds, ballistic trajectories can be approximated as straight lines, with the minute deviations from the actual trajectory due to gravity and air resistance being termed as 'bullet drop'. If the target's size is larger than the bullet drops at the target range, then the bullet will still hit the target although it has dropped off course by a small deviation. While this has major range and accuracy drawbacks, it greatly simplifies the aiming system, as there is no need for target coordinates, all that is needed to land a clear shot is a clear line of sight from the shooter to the target. It gives the benefit of being able to make-do with just one camera, making it a simpler and more cost-effective system

Projectile Delivery System (PDS)

Railgun Electromagnetic Propulsion

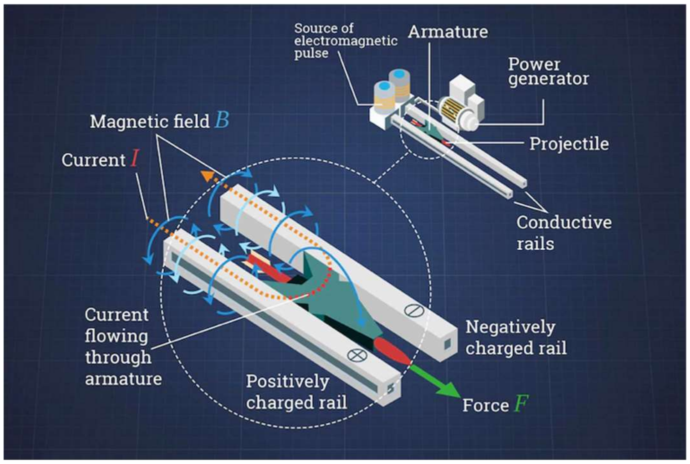


Figure 1-5: Simplified illustration of rail operation principle

Initially, many thoughts and ideas were presented as to how a project like this ought to be implemented. The vision was to make a severely downscaled railgun, envisioned to be capable of (~10 m) maximum range. It would be supplied by a capacitor bank consisting of several 1F 5V capacitors in parallel. This method works using the Lorentz Force, where a charged particle experiences a force when travelling across a magnetic field, wherein in this case, the charged particle would be electrons in the material, its motion constituting a current: -

$$\vec{\boldsymbol{F}}_{Lorentz} = q\left(\vec{\boldsymbol{B}} \times \vec{\boldsymbol{v}}\right) = I(\vec{\boldsymbol{B}} \times \vec{\boldsymbol{l}}) \qquad \{1.1\}$$

Where the same current will produce the magnetic field, through Ampere's Law: -

$$\oint \vec{\boldsymbol{B}} \cdot d\vec{\boldsymbol{l}} = \mu_0 I \qquad \{1.2\}$$

After Simulink simulations showed that there is a possibility that the armature could weld to the rails instead of launching, the probability of which could not be gauged due to limitations in simulation software, not to mention the problem of eroding rails that occurs regularly in already existing railguns, the issue of disposable rails made the idea expensive and risky for too little reward. The idea was deemed unworthy and later scrapped in favour of others. The Simulink model was constructed with help from a paper published by the US Navy. [7]

Paired DC Motor Frictional Propulsion

It was proposed to use Nerf darts as ammunition and launch it using the friction and the relative between the dart and the two motor shafts. There would be a magazine containing 4 Nerf darts, each dart being fed using gravity and a servo mechanism. The magazine would be situated vertically above the barrel, each dart would fall out in sequence. In order for the payload delivery to be commenced, a servo mechanism feeds the dart forward where it picks up speed from the motor’s friction, thus being launched out of the barrel. One major drawback of this propulsion method is that it produces relatively low launch velocities, thus shorter ranges are reached by the system. The corresponding equation for the launch force produced by this system would be the following: -

$$F_{motor} = -b(\dot{x} - v_{motor}) \qquad \{1.3\}$$

Torsion Ballista

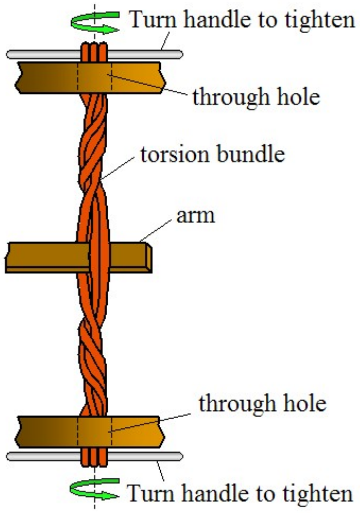
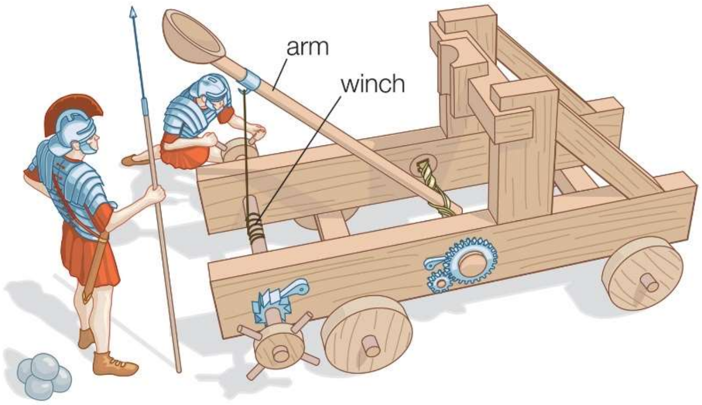


Figure 1-7: Diagram of torsion spring in a torsion ballista illustration component and method of operation



© 2010 Encyclopædia Britannica, Inc.

Figure 1-6: Drawing of a torsion onager, a close relative of the ballista with the same operational principle

This method is an upgrade of the original tensioned ballista, although it is certainly an outdated method, it is simple and efficient, providing a simple, purely mechanical system. Technically, the tension ballista isn’t much more than an oversized bow, storing potential elastic energy and releasing much faster than it was stored, resulting in a higher power than could be achieved by hand, only limited by the mechanical strength of the load-bearing material. The advent of the torsion ballista doesn’t change the principle of action; energy is still stored in and released from a spring. The real change is in the switch from tension to torsion, which is revolutionary, as the energy is released in rotational form, so achieving higher launch speeds is not only done by increasing stored energy, but it can also be done by simply increasing the length of the moment arm. This may come at the expense of torque produced, but the torque need only accelerate the projectile, its purpose almost ceases after the initial jerking motion. All seems well in theory, yet where this system provided simplicity, it gave rise to complication elsewhere, there came the issue of how to reload the projectile, how to restore energy to the torsion spring, all of which were possible on the condition that they be manual, as was always the case. This was a major dealbreaker, as this lay completely out the project scope, there was no fast response, no automation in this approach. The same reasons the ballistae, catapults, and trebuchets of old, have been far outdated and unused, have persisted to keep them so. Since the force in this system comes from a spring, it is none other than the force predicted by Robert Hooke, famously named Hooke's Law: -

$$F_{catapult} = -ks \qquad \{1.4\}$$

Mechanical Design and Structure

The team members unanimously agreed that the gun carriage system would be mounted onto a tripod. Tripods provide stability for gun placements with minimum number of supports, as long as the centre of mass is within the triangle between the three points on the rod that meet the floor, then the system is statically stable. The recoil force, estimated to be less than ~10% of the total project's weight, along with the expectation that the gun was expected to fire with equal performance in all directions led to its' neglect during initial static analysis of the tripod. Having the tripod act similarly in all directions requires geometrical symmetry in both the positioning and design of each of the tripod legs; each one had to be identical to the other, which is the case for most tripods in military and civilian use.

Decision Criteria

In the conclusion of the brainstorming sessions, certain issues were converged on, such as a tripod configuration as well as the servo tracking mechanism, as options in their respects were quite narrow. It mattered more the method executing them than it did which systems would be used to fill these roles. Both TAS and PDS proposals were each evaluated independently on a scale of 1 to 5, on criteria of feasibility, cost-effectiveness, reliable operation, range and simplicity, among other things. The choice of BAS would later be based on decisions taken in regard to the former, especially TAS as they are integrated together.

Table 1: PDS Rating table

	PDS		
Criteria	Railgun	Motor Friction	Ballista
Feasibility	1	4	2
Affordability	1	4	3
Reliability	1	3	3
Generality	4	2	4
Range	4	2	3
Simplicity	2	3	2
Score	2.2	3.0	2.8

Table 2: TAS Rating table

	TAS		
Criteria	Laser Guidance	Rangefinder and Camera	Camera
Feasibility	2	1	3
Affordability	3	1	4
Reliability	1	3	4
Generality	5	3	1
Range	3	4	3
Simplicity	3	3	4
Score	2.8	2.5	3.2

Team members agreed upon individual system rating scores for each criterion, then the arithmetic averages of each system were taken, systems with the highest average would be the most suitable. As shown in tables 1 & 2, the chosen PDS system would be the Dual Motor system paired with the Face-tracking Camera system. Since there are only two BASs that can be used, one requiring target coordinates, one that doesn’t, since the camera system only gives a 2D image, it was evident that the way to go was bullet-drop compensation as the BAS of choice.

Chapter 2 Modelling

Face Detection with Camera

- In this section, the choice of method of action of the computer vision (CV) system will be thoroughly examined

In the search for appropriate CV methods, which for the team was an uncharted field that none of the members had any experience with. Thus, a lot of time was spent researching the different subfields of CV and mainly focused on face detection as that was the objective. Several options were available, first of which was MATLAB's CV Toolbox, which was readily available, and had widespread community support, so it was expected that any issues would have readily present fixes in the community. However, there was one downside, which is that we needed to have MATLAB on every laptop computer that we test it on, this would have not been a problem if it wasn't for the fact that the setup for MATLAB is 18 GB and the program itself is 27 GB. It is very inconvenient to have such a big program on every laptop we use, it would severely demobilize us during work and testing, not to mention that MATLAB requires high processing power to run and when it runs, it seldom does it quickly enough on small-processing power devices. It was considered an unfortunate but necessary burden to bear, until one team member, the one responsible for the TAS, discovered that we can use Python OpenCV library. This dramatically changed things, as all that was needed was the code file for the face detection only, which was small in size and could run on VS Code. It was much easier to obtain set up for and install on team devices, as it does not need a license and is smaller in size, never exceeding 1 GB, with the code file and libraries being an entire order of magnitude lower. This yielded greater mobility as it can be tested on any laptop provided, moreover, have a reliable internet connection any time.

OpenCV is a python open-source library used for CV applications, as its namesake suggests [8]. It has several image processing functions, that, in combination with several other libraries such as dlib and Face_recognition, can be used to create a program for face identification and can put a square around the detected face. In addition to this advantage, OpenCV also uses less RAM and is faster as it is written in C/C++. The algorithm detects faces in two steps: -

1. Capture

The camera captures live feed from the environment, at a fixed frame rate so that each frame is processed by the algorithm. For this purpose, a camera with a high resolution and framerate would be important to ensure good performance from the system. Initial testing was done on in-built laptop webcams, as well as on a mobile webcam owned by one team member prior to the commencement of this project. The online market for webcams, while expensive, it was the most guaranteed and reliable than the local market where cheaper and lower quality webcams were offered. Due to limited budget constraints, there had to be compromise and consideration for both options in the search for an appropriate webcam. It was agreed that the budget allocated for the camera would not exceed 300 EGP, and that a minimum resolution requirement of 1080p would be suitable for the task at hand. Three options were found, a 4K camera for 800 EGP, a 1080p camera for 250 EGP, and a 720p camera for 50 EGP. The last option was of extremely low hardware quality, its operational life was expected to be very short, thus it was discarded from the selection first, as well as the 800 EGP due to it exceeding the project budget. This left only one option for the camera, which was the 1080p option, suiting the minimum requirement while staying just below the budget limitation with the added benefit of being lightweight making it the final camera choice.

2. Extraction

Once the image is captured, the search for facial features commences. The algorithm in use is the Viola-Jones algorithm [9], this algorithm has gained publicity in the world of CV as it has wildly eased the detection of facial features on many devices worldwide. It works through the detection of Haar-like features, which essentially means it separates the image's pixels into regions, calculates the pixel intensities and their sum over each region, then calculates the difference between these sums. Then the features of the image can be detected by comparison with the following figure.

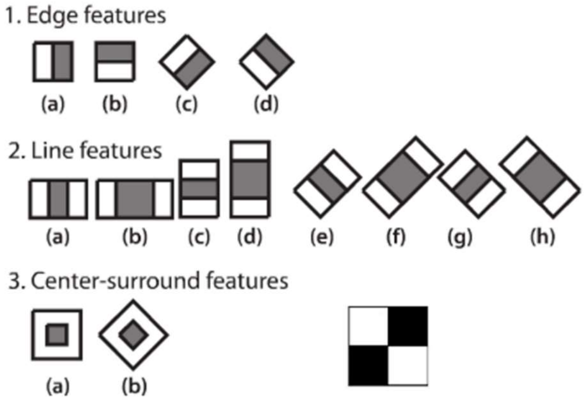
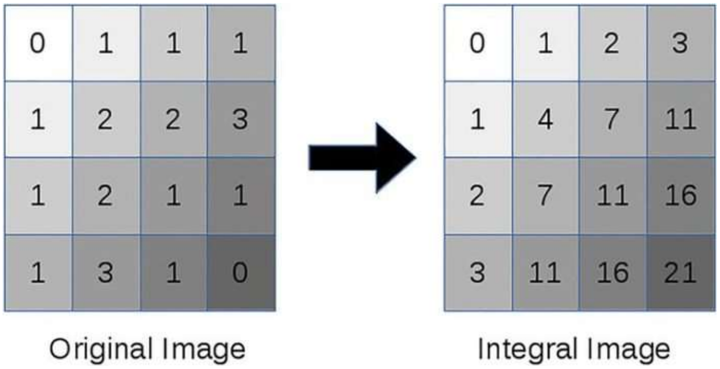


Figure 2-1: Diagram showing different types of Haar features

This can be done repeatedly by classifiers, in a process termed 'cascade classification' in which the inputs of each classifier are the outputs of its predecessor. The first step is to convert the coloured image into a grey scale image, so that the intensity would be only one number rather than the original 3 (RGB); then the Haar wavelets are calculated and identified by the algorithm as described in the opening paragraph and sorted into their types, as shown in the figure, where each type is useful for identifying its namesake. The second step is creating an integral image, one in which every pixel is the sum of all pixels above and left to it, including itself.



Original Image

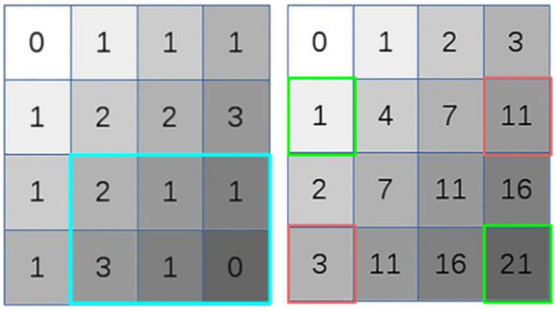
0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0

Integral Image

0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

Figure 2-2: Example showing the creation of an integral image from a test grayscale image

This allows the calculations to be cut short, as the operations need only be done on the edges of the square, as shown in the figure. Therefore, for a square image of n pixels, the time complexity would be decreased from O (n^2) to O (1), which makes a huge difference when n is of a high order of magnitude. [10] [11]



0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0

0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

Figure 2-3: Example explaining the method of obtaining the sum of a rectangular set of pixels

Using traditional algorithm: -

$$2 + 1 + 1 + 3 + 1 + 0 = 8$$

Using integral image method: -

$$21 + 1 - 11 - 8 = 8$$

The same result was obtained in 2 steps rather then 4, in fact the result will be obtained in 2 steps regardless of the number of pixels, since every set of pixels is rectangle, and all rectangles have 4 corners.

Before the ML (machine learning) algorithm, AdaBoost (Adaptive Boost) which greatly reduces the complexity of our algorithm yet another time by selecting the most important features of the ones identified throughout the image for face detection can be introduced, the reader must first be familiar with the concept of Cascade Classifiers. It can be deduced from its namesake that it classifies, which isn't quite novel given the subject of this section. It does this by being trained beforehand on databases of positive images (images which it has been told that there is a face for sure) and negative images (images in which there was no face for sure), after the training is done, the algorithm is ready to be tested on live images where it partitions and runs the classification algorithm on each part before moving on to the next one. This is where the cascading part comes in, the classifier doesn't forget about the previous section, rather than that, the outputs are given to the next classifier, thus the 'cascade'. At this point, it has been mentioned several times that there are multiple classifiers, the question that shortens the run time for the algorithm when answered is, are these classifiers all equal? The answer to that question is not quite simple, some perform well, others not so well. This may seem quite obvious and too simple to be mentioned, however this is precisely what is needed; all classifiers are classified as either weak or strong, and their weights are distributed according to that criterion. AdaBoost acts as a supervisor and judge over all the classifiers, judging which are the strong and the weak, assigning each their weight accordingly, so that the strong classifiers get more say in the final output than the weak classifiers [12]. This system accelerates the learning, training, as well as the response time of the classification process, thus optimizing the face detection system.

Locking and Tracking

- In this section, the subject of discussion will be the method with which target position on-screen is found, as well as the manner in which the target will be tracked by the TTLS

After the face has been detected, a box is placed around it on the screen, for tracking purposes the target will be the point at the centre of this box, this is the point where the gun camera must maintain its aim. If the camera were an ironsight or a sniper scope, then the objective is to keep the target at the centre of the scope i.e., the centre of the screen. Thus, the objective of the targeting system has been more precisely defined by identifying the desired position. It is the difference between the box centre and screen centre that will be the error that the system will attempt to minimize. This allows the building of the following block diagram for the system: -

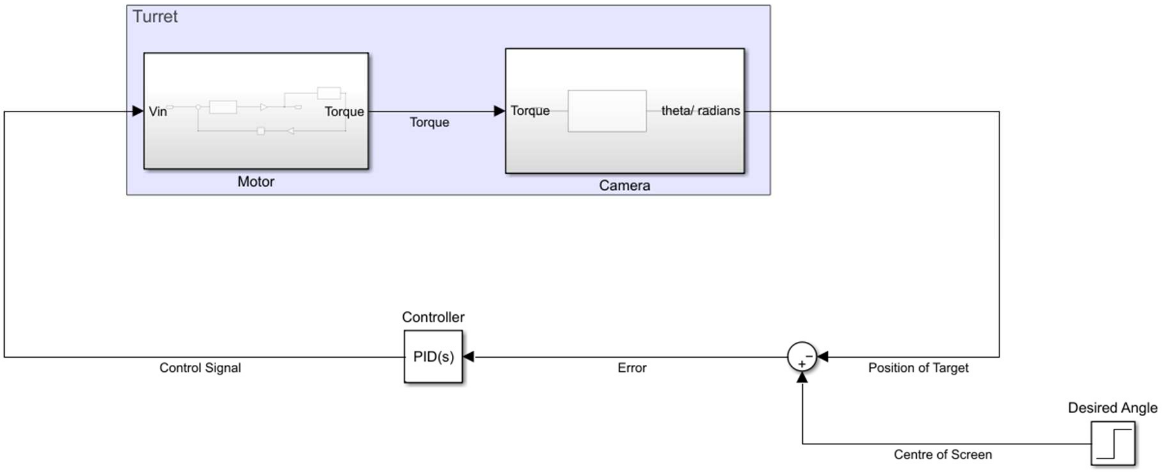


Figure 2-4: Tracking system block diagram

One issue that needs to be addressed in this report is the fact that the incoming position error is not in angular form, it is rather in pixel form. This however wont quite make a difference as both methods will guarantee the behaviour required from it. One other issue here is the fact that servos themselves have internal position feedback, it is essentially a DC motor with position feedback, so feedback system cannot be changed to another more suiting for specific needs, either regular DC motors is to be paired with PID control system or improvise the servo system to interact with the pixel error reading. Needless to mention, servos do not take instructions in terms of desired pixel, rather in desired angle, so there need be a combination of the precise and accurate positioning of the servo with an external feedback system that minimizes the pixel error. This can be done by making the servo rotate right or left (up or down in case of the elevation servo), in constant steps depending on the sign of the pixel error, which has in fact been done by a creator mentioned in the review. This system can be further improved to provide smoother responses, by varying the step size to be proportional to the error, or its cumulative sum, or its difference, in other words, while servos have internal feedback, it can still be overridden by moving it in steps and executing PID control on the step size. This will provide higher quality response as well as less vibrations on the load and the servo shaft. One downside to this system, however, is that it cannot quite be simulated as the original angle error system has been simulated, thus it is not suitable to use the benefit of the PID tuner app in Simulink, neither can trial-and-error be used in simulation, it must be tried and optimized during testing, which would not be time-efficient and could possibly damage components if incorrectly managed.

Trajectory Analysis

- In this section, the dart trajectory will be analysed with and without air resistance and calculate an appropriate formula for bullet-drop so that an effective range may be calculated.

In modelling any mechanical system in engineering, 1 principle is always of most usefulness, that of Newton's 2nd law (for a system of constant mass): -

$$\vec{\boldsymbol{F}}_{net} = m\vec{\boldsymbol{a}} \qquad \{2.1\}$$

Along with the definitions of acceleration and velocity: -

$$\vec{\boldsymbol{a}} := \frac{d}{dt}\vec{\boldsymbol{v}} = \frac{d^2}{dt^2}\vec{\boldsymbol{x}} \qquad \{2.2\}$$

With these equations and some initial conditions, the trajectory can be found as will be shown below with position vector and initial velocity $\overrightarrow{\boldsymbol{r_0}}$ & $\overrightarrow{\boldsymbol{v_0}}$

Initial conditions: -

$$\overrightarrow{\boldsymbol{r_0}} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \qquad \{2.3\}$$

$$\overrightarrow{\boldsymbol{v_0}} = v_0 \begin{pmatrix} \cos\varphi_0 \\ \sin\varphi_0 \end{pmatrix} \qquad \{2.4\}$$

a) Ballistic trajectory without air resistance

Applying {2.1} to projectile: -

$$\vec{\boldsymbol{F}}_{net} = m\vec{\boldsymbol{g}} = m\vec{\boldsymbol{a}} \qquad \{2.5\}$$

$$\vec{\boldsymbol{g}} = \begin{pmatrix} 0 \\ -9.81 \end{pmatrix} m\,s^{-2} \qquad \{2.6\}$$

In combination with {2.2}, gives the following 2nd order linear differential equation: -

$$\frac{d^2}{dt^2}\vec{\boldsymbol{r}} = \vec{\boldsymbol{g}} \qquad \{2.7\}$$

The solution of which is obtained by integrating twice with respect to time and plugging {2.3} & {2.4} for the integration constants: -

$$\vec{\boldsymbol{r}} = \frac{1}{2}\begin{pmatrix} 0 \\ -9.81 \end{pmatrix} t^2 + v_0 \begin{pmatrix} \cos\varphi_0 \\ \sin\varphi_0 \end{pmatrix} t + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{2.8}$$

Which is a vector equation that can be separated into 2 parametric equations, one for y and one for x: -

$$x(t) = v_0 t \cos\varphi_0 + x_0 \tag{2.9}$$

$$y(t) = \frac{-9.81}{2} t^2 + v_0 t \sin\varphi_0 + \ y_0 \tag{2.10}$$

It is this pair of equations that the rest of this section's drag-less analysis will ultimately rest upon.

This set of parametric equations can be merged into one explicit equation, by solving for t in {2.9} and substituting it in {2.10}. The equation can be further simplified by offsetting both x and y by their initial values: -

$$y \ - \ y_0 \rightarrow y$$
$$x - x_0 \rightarrow x$$

$$t = \frac{x}{v_0 \cos\varphi_0} \tag{2.11}$$

$$\therefore \ y(x) = x \tan\varphi_0 - \frac{g \sec^2\varphi_0}{2{v_0}^2} x^2 \tag{2.12}$$

This is a parabolic equation, with the following parameters: -

$$y_{vertex} = \frac{{v_0}^2}{2g} \sin^2\varphi_0 \tag{2.13}$$

$$x_{y=0} = \pm \frac{{v_0}^2}{g} \sin 2\varphi_0 \tag{2.14}$$

Which allows a graph of the projectile path to be plotted, given trajectory parameters $(v_0, \varphi_0, y_0\ , x_0)$.

It is immediately noted that all possible trajectories lie within a certain envelope, the equation for which can be derived by, and is the following: -

$$y = \frac{{v_0}^2}{2g} - \frac{gx^2}{2{v_0}^2} \tag{2.15}$$

Which also happens to be a parabola. Thus, any target with coordinates (x_t, y_t) that lie within the envelope, is a feasible target. It is also worthy of noting that every target within the envelope can be hit at 2 different elevation angles. [13] [14]

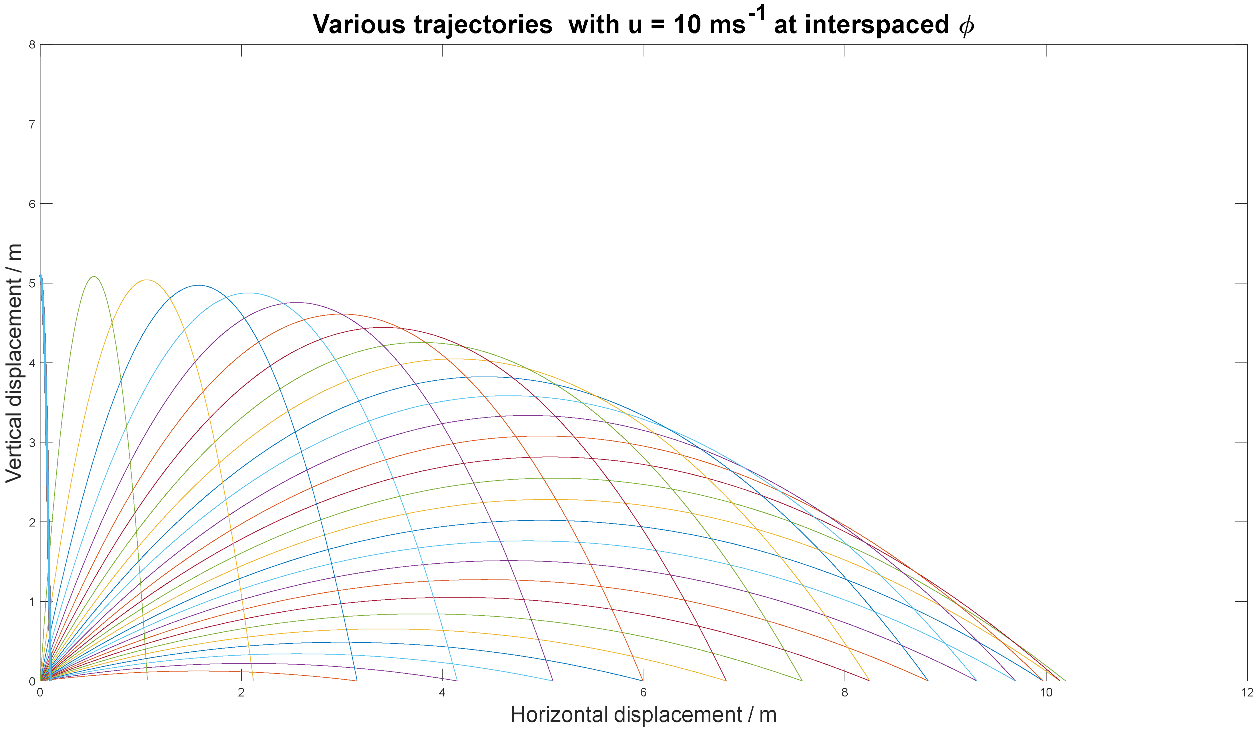


Figure 2-5: Graph illustrating the possible ballistic trajectories at equally incremented elevations illustrating the envelope (done with MATLAB)

b) Ballistic trajectory with air resistance

There are many types and causes of air resistance in ballistics, which vary with countless parameters, such as Reynolds Number (Re), temperature, pressure, etc. These cause the air resistance to be a problem too complicated for many to handle. An approximation would be handy in this situation, and indeed it is the main method used in most levels of application and research. That is the Taylor approximation, this method expands any function into a polynomial series, when applied to air resistance, 2 terms dominate the rest of the series, one linear and one quadratic: -

$$f_{drag} \approx f_{lin} + f_{quad} = -bv - cv^2 \qquad \{2.16\}$$

This simplifies the problem further by taking into account each at a time. In this scenario, it would be helpful to examine both terms and their relative strengths, in order to obtain the criteria of neglection for each.

$$\frac{f_{lin}}{f_{quad}} = \frac{b}{cv} \propto \frac{1}{v} \qquad \{2.17\}$$

It can readily be seen that quadratic drag becomes more and more pronounced as speed increases, therefore it is a must that its effect be considered at high speeds to ensure that experiment fits prediction. Whereas coefficients b and c are dependent on geometry of projectile as well as environmental factors [13]: -

$$b = \beta D \qquad \{2.18\}$$
$$\mathrm{c} = \gamma D^2 \qquad \{2.19\}$$

- β & γ are the environmental coefficients, for air: -

$$\beta = 1.6 \times 10^{-4} \ {}^{Ns}/_{m^2}$$
$$\gamma = 0.25 \ {}^{Ns^2}/_{m^4}$$

- D is the characteristic length

One final note on this, is that the term $\frac{f_{lin}}{f_{quad}}$ is related inversely to the Reynolds Number of the flow, such that when the Reynolds Number is high, the quadratic term overpowers the linear term and vice versa when it is low.

For this case, $D = \frac{4(\pi r^2)}{2\pi r} = 2r = D = 0.013\, m$
$\rightarrow b = 1.6 \times 10^{-4} \times 0.013 = \ 2.08 \times 10^{-6} \ {}^{Ns}/_{m}$
$\rightarrow c = 0.25 \times 0.013^2 = \ 4.225 \times 10^{-5} \ {}^{Ns^2}/_{m^2}$

At a launch speed estimate of 10 ${}^{m}/_{s} \rightarrow \frac{f_{lin}}{f_{quad}} = \frac{b}{cv} = \frac{2.08 \times 10^{-6}}{4.225 \times 10^{-5} \times 10} \sim 10^{-2}$

Which indicates, for this estimate, that the linear drag term can be neglected in favour of quadratic drag for ease in calculations. Nevertheless, linear drag analysis still carries some reward as it provides insight for how the trajectory will be affected by air resistance, while still being an easier problem to solve.

For linear drag: -

$$\vec{\boldsymbol{F}}_{net} = m\vec{\boldsymbol{g}} + \vec{\boldsymbol{f}}_{lin} = m\vec{\boldsymbol{g}} - b\dot{\vec{\boldsymbol{r}}} = m\ddot{\vec{\boldsymbol{r}}} \qquad \{2.20\}$$

The solution for this 2nd order linear differential equation is predictably more difficult than the previous case. However, there is no need not go through the solution here, as in the prior section, many ballistics textbooks and papers have obtained the solution countless times before [13]: -

$$x(t) = \frac{mv_0 \cos\varphi_0}{b}\left(1 - e^{-\frac{b}{m}t}\right) \qquad \{2.21\}$$

$$y(t) = \frac{m}{b}(v_0 \sin\varphi_0 - gt) + \left(\frac{m}{b}\right)^2 \left(g - e^{-\frac{b}{m}t}\left(g + \frac{b}{m}v_0 \sin\varphi_0\right)\right) \qquad \{2.22\}$$

These two equations show the parametric solution from which an explicit solution can be obtained in the same manner as previously done, before doing that however, it must be noted that while the domains for each function are the same, the ranges are not. This can be checked by calculating the limit for $x(t)$ & for $y(t)$: -

$$\lim_{t\to\infty} \frac{mv_0 \cos\varphi_0}{b}\left(1 - e^{-\frac{b}{m}t}\right) = \frac{mv_0 \cos\varphi_0}{b} = x_{max} \qquad \{2.23\}$$

$$\lim_{t\to\infty}\frac{m}{b}(v_0\sin\varphi_0 - gt) + \left(\frac{m}{b}\right)^2\left(g - e^{-\frac{b}{m}t}\left(g + \frac{b}{m}v_0\sin\varphi_0\right)\right) = -\infty \qquad \{2.24\}$$

From {2.23} & {2.24}, it is noted that while a closed form explicit solution for the trajectory exists, the domain for this function would be limited to: -

$$0 \le x \le \; x_{max} \qquad \{2.25\}$$

Where the fruitful piece of insight could be noted that this is the maximum range for striking ground targets. It is worth noting in comparison the striking difference between {2.20} & {2.10}, and that the former is non-parabolic and non-symmetric for that matter. The trajectory can no longer be represented by a parabola with a vertex and a focus, rather it becomes an irregular shape, which can be illustrated when compared with the prior case: -

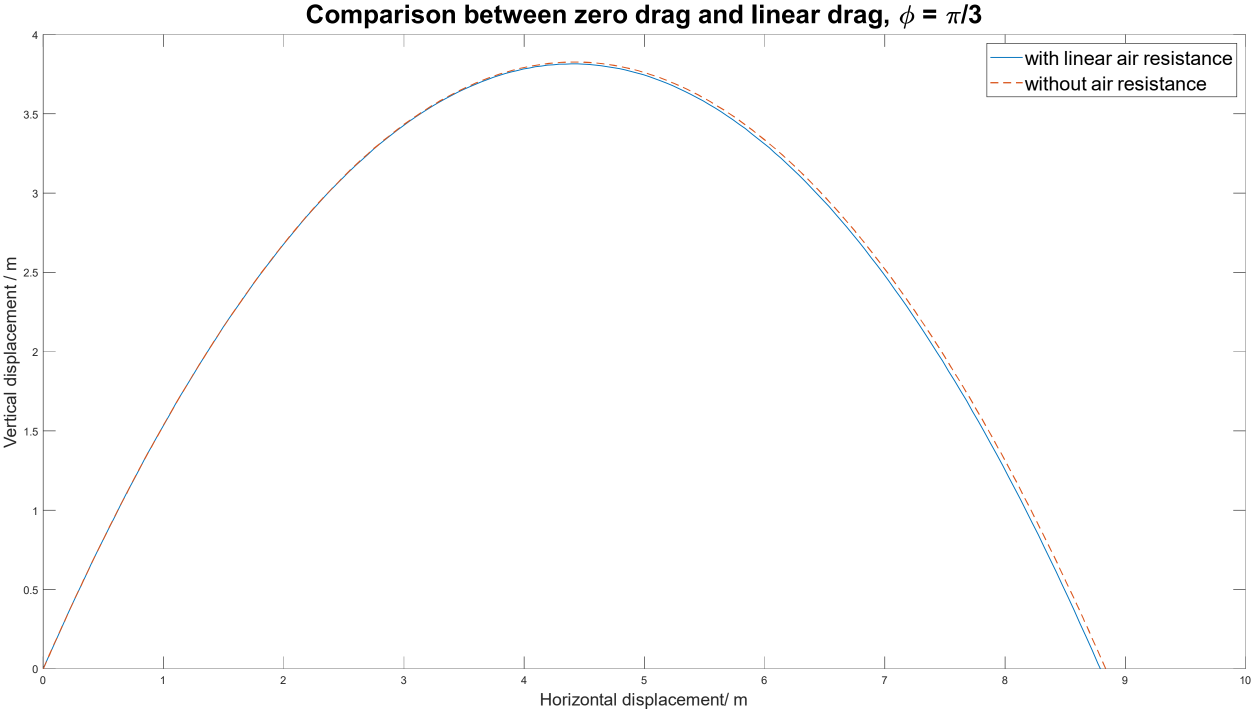


Figure 2-6: Graphical comparison between a linear drag and drag-less trajectory (done with MATLAB)

As expected, the linear drag term caused little deviation from the original trajectory, barely detectable by the naked eye, even after axes have been scaled to exaggerate the effect.

For quadratic drag: -

$$\vec{\boldsymbol{F}}_{net} = m\vec{\boldsymbol{g}} + \vec{\boldsymbol{f}}_{quad} = m\vec{\boldsymbol{g}} - c\left(\dot{\vec{\boldsymbol{r}}}\cdot\dot{\vec{\boldsymbol{r}}}\right)\hat{\boldsymbol{r}} = m\ddot{\vec{\boldsymbol{r}}} \qquad \{2.26\}$$

$$c\left(\dot{\vec{\boldsymbol{r}}}\cdot\dot{\vec{\boldsymbol{r}}}\right)\hat{\boldsymbol{r}} = c(\dot{r})^2\hat{\boldsymbol{r}} = (c\dot{r})\dot{\vec{\boldsymbol{r}}} \qquad \{2.27\}$$

$$\therefore m\ddot{\vec{\boldsymbol{r}}} = m\vec{\boldsymbol{g}} - (c\dot{r})\dot{\vec{\boldsymbol{r}}} \qquad \{2.28\}$$

Which gives us a 2nd order non-linear differential equation [13] [14], which features cross dependence due to the quadratic drag term, which makes the x and y components of the acceleration non-separable. It is due to this complication that no general analytical solution has been found for this equation, it must be solved numerically at time step intervals. This can be done with software such as Python, MS Excel, MATLAB, or Simulink which can be used to plot and study the trajectory. In this report, it will be solved using the ode45 function in MATLAB, in order to do this, the equation must be converted from 2nd order to 1st order form, which can be easily done by choosing the velocity as the state variable: -

$$\dot{\vec{\boldsymbol{v}}} = \vec{\boldsymbol{g}} - \left(\frac{c}{m}v\right)\vec{\boldsymbol{v}} \qquad \{2.29\}$$

$$\vec{\boldsymbol{v}} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \qquad \{2.30\}$$

Then these equations are converted into state equation form: -

$$\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \vec{\boldsymbol{v}} \qquad \{2.31\}$$

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} -z_1\sqrt{{z_1}^2 + {z_2}^2} \\ -9.81 - z_2\sqrt{{z_1}^2 + {z_2}^2} \end{pmatrix} \qquad \{2.32\}$$

It is in this form that ode45 can solve the equations in MATLAB using Runge-Kutta integration, allowing the velocity curve to be solved for, given initial conditions: -

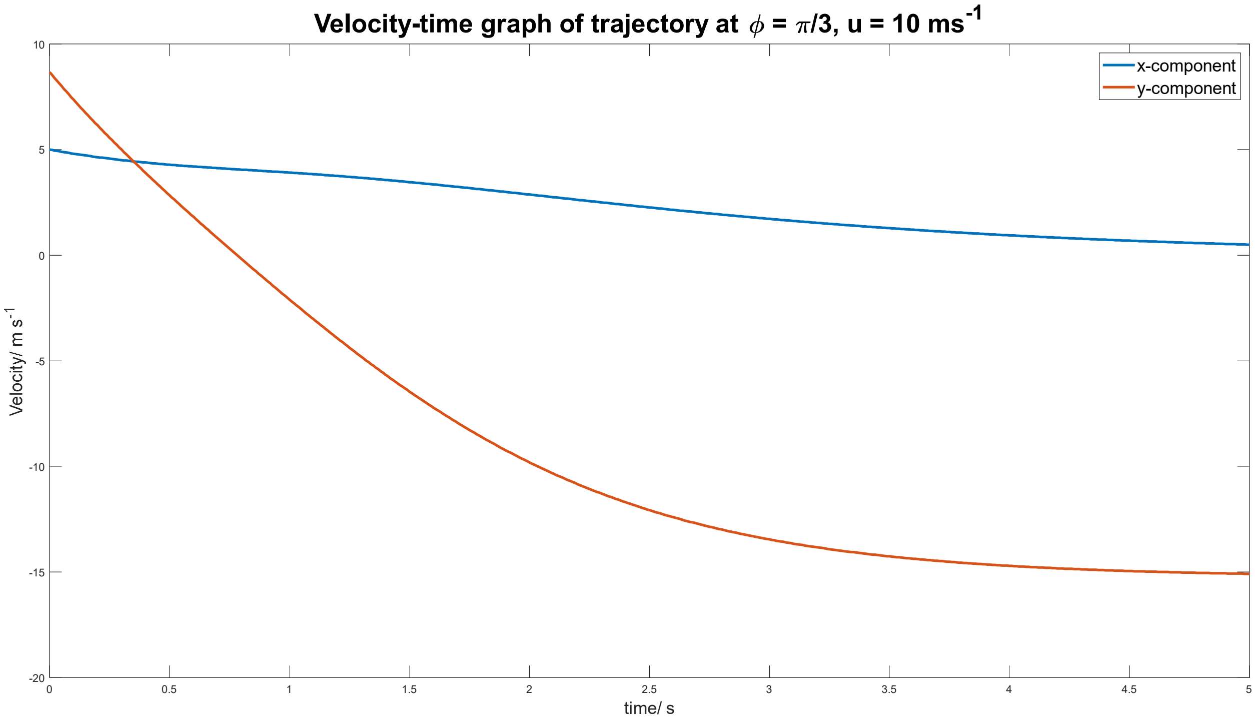


Figure 2-7: Velocity-time graph for a quadratic trajectory (done with ode45 in MATLAB)

These values are each obtained as vectors or 1D arrays in MATLAB, from the definition of velocity as the time derivative of displacement, the displacements in x and y can be obtained and plotted in comparison with the drag-less trajectory: -

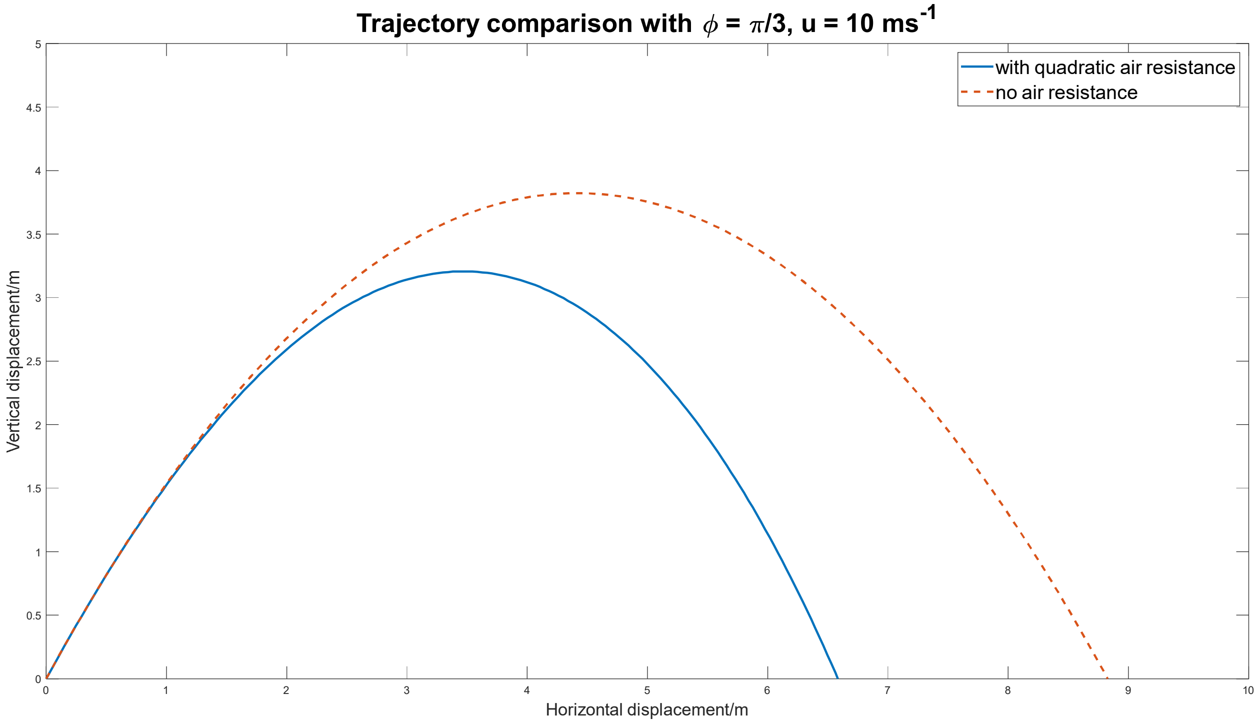


Figure 2-8: Graphical comparison between quadratic drag and drag-less trajectory, note the large decrease in maximum height and range (done with ode45 and cumtrapz in MATLAB)

There is a noticeable difference between both trajectories, primarily the large decrease in range which in this case is just below 2.24 m of range lost, about 25.5% reduction, as well the much larger increase in bullet-drop and maximum height, just below 1.19 m corresponding to about. Less visible in this figure as well is the non-symmetry of the trajectory, which can be made more apparent by changing the scale to show the negative displacement, and increasing the launch speed: -

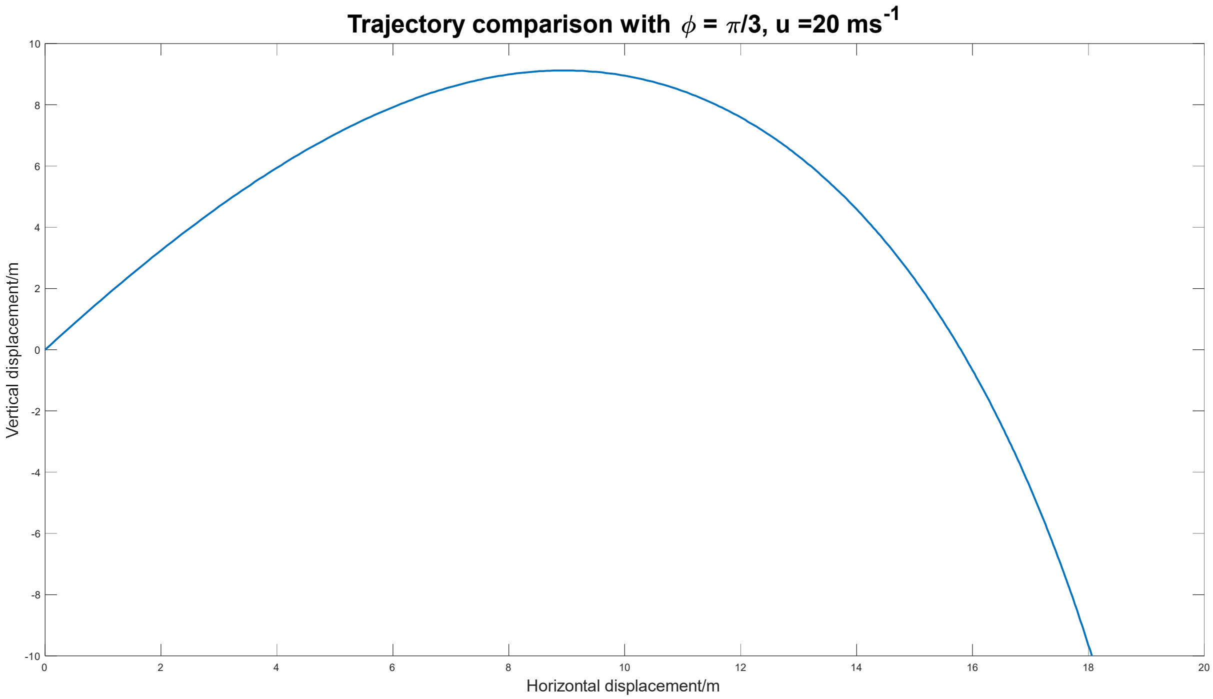


Figure 2-9: Trajectory plot for a quadratic trajectory, with higher launch speed to exaggerate asymmetry (done with ode45 in MATALB)

Thus, it is not suitable for a shooter- in this case, the BAS- to ignore air resistance, even if the darts were heavier than the assumed 1g of mass the test projectile has. Due to the large increase in bullet drop, there need be higher compensation for it by aiming higher, otherwise the range would be limited to shorter distances where it can be ignored (if criteria for bullet drop is 10% then that would be up to 6.2 m). This range however can still be increased by increasing the launch speed, while this will make the dart deviate at a quicker rate from the parabolic trajectory, it would travel a longer distance in this time, thus extending the effective range of the turret. The solution to this problem would be in the hands of the PDS.

Payload Delivery

- In this section, the modelling and analysis of the PDS will be discussed and will produce an approximation for its maximum range.

The principle of operation of the motor friction PDS may be confusing at first glance, it is often the case that friction is the adversary of motors; causing inefficiencies, wear, erosion, etc. and is combatted with all kinds of lubricants and innovations. It is rather the opposite here, where the friction from the motor roller would be the driving force in the propulsion system. It is the relative speed between the dart and the spinning motor that produces this friction, it is what could be compared to a devolution of a rack and pinion, where the entirety of motion transmission is frictive. The governing equation for such motion is not much different from the equation of a damped system: -

$$m\ddot{x} = -b(\dot{x} - r_{roller}\omega_{motor}) = -b(\dot{x} - v_m) \qquad \{2.33\}$$

Which is a 2nd order linear differential equation, with initial conditions: -

$$\dot{x}_0 = x_0 = 0 \qquad \{2.34\}$$

The solution for which: -

$$x(t) = tv_m + \frac{mv_m}{b}\left(1 - e^{-\frac{b}{m}t}\right) \qquad \{2.35\}$$

Where: -

- $v_m = r_{roller}\omega_{motor}$
- m is the dart mass
- b is the relevant friction coefficient, a function of several variables such as the normal force (N), and dynamic friction coefficient (μ)

It is not to be assumed that the domain of this function is $0 \le t < \infty$, this is due to the fact that there is no contact between the dart and the roller once the entire dart's length (l) has passed through, thus there is the problem of finding the domain given the range: -

$$x(t_{exit}) = l = t_{exit}v_m + \frac{mv_m}{b}\left(1 - e^{-\frac{b}{m}t_{exit}}\right) \qquad \{2.36\}$$

The unfortunate issue with this equation is that it cannot be solved explicitly for t, thus there are multiple options: -

a) Solution by numerical methods, which requires accurate estimates of b, m, v_m
b) Taylor expansion to a polynomial, which produces answers of questionable error

While method a) may well be feasible, it is unreliable as it is difficult to obtain estimates of b, errors in which may compound to be inaccurate

Starting with Taylor expansion of $x(t)$

$$l = x(t_{exit}) \approx \frac{bv_m}{2m} {t_{exit}}^2 \tag{2.37}$$

$$\therefore t_{exit} \approx \sqrt{\frac{2ml}{bv_m}} \tag{2.38}$$

While differentiating $x(t)$ and substituting $\sqrt{\frac{2ml}{bv_m}}$ may sound like the straightforward routine method to finding the launch speed, a better approximation can be obtained by substitution in only one term. This can be done by integrating {2.33} with respect to time, followed by the substitution the definition of l and the approximation of t_{exit}: -

$$\int_0^{t_{exit}} m\ddot{x}\,dt = mv_{exit} \tag{2.39}$$

$$\int_0^{t_{exit}} -b(\dot{x} - v_m)\,dt = -b(x(t_{exit}) - v_m t_{exit}) \approx -b\left(l - v_m\sqrt{\frac{2ml}{bv_m}}\right)$$
$$= -b\left(l - \sqrt{\frac{2mlv_m}{b}}\right) \tag{2.40}$$

$$\therefore v_{exit} \approx -\frac{b}{m}\sqrt{l}\left(\sqrt{l} - \sqrt{\frac{2mv_m}{b}}\right) \tag{2.41}$$

From this, the recoil impulse and force can be approximated as such: -

$$I_{recoil} = -I_{launch} = -m\,dv = -mv_{exit} \approx b\left(l - \sqrt{\frac{2mlv_m}{b}}\right) \tag{2.42}$$

$$\therefore \vec{F}_{recoil} \approx -m\frac{v_{exit}}{t_{exit}} \approx \frac{b\left(l - \sqrt{\frac{2mlv_m}{b}}\right)}{\sqrt{\frac{2ml}{bv_m}}} = b(\sqrt{\frac{bv_m}{2lm}} - v_m) \tag{2.43}$$

Where Newton's 3rd and 2nd laws have been applied. However, several irregularities in {2.41}, plugging in 0 for motor speed somehow results in a non-zero exit speed. This partly due to the Taylor approximant being 2nd in order, so that it is quite inaccurate, and that inaccuracy increases quadratically. The second reason for this is a flaw in the mathematical reasoning, as the exit speed is defined as the speed of the dart once its entire length has exited the turret, which cannot possibly happen with the motor spinning. If the dart cannot move its entire length, then it would never reach the exit, it thus appears that the exit speed is poorly defined in such a scenario. It could also be defined as the dart length divided by t_{exit}: -

$$v_{exit} = \frac{l}{t_{exit}} = \frac{l}{\sqrt{\frac{2ml}{bv_m}}} = \sqrt{\frac{lbv_m}{2m}} \qquad \{2.44\}$$

Which satisfies the logical requirement of vanishing exit speed at 0 motor speed; however, it assumes constant speed throughout the motion, which implies that there is no net force acting on the dart, thus no means of acceleration, which completely negates the dart picking up any speed given the initial conditions.

Mechanical Design and Analysis

- In this section, the subject of discussion will be the design criteria, considerations, and process, in theory and simulation.

Design Criteria

The design criteria were established after it was decided to explore the world of 3D printing; with this first design criterion being put in place, the rest of the design criteria can take place. The design must be low in mass to decrease the cost while maintaining good safety and static and dynamic stability. The main criterion was to maintain low weight while being safe, so the design was adjusted multiple times to ensure it was as light as possible.

Considerations and Limitations

The main considerations and constrictions of the design were the sizes of the servos to be used. Initially the team decided to use 3 servos - 2 for the vertical and horizontal movement of the turret while the other servo will be in the launching mechanism of the darts. In order to be able to work with these limitations the servos to be used were modelled on Autodesk Inventor. The deflections and movement of the body due to the stress on it was a big consideration as it would affect the aiming and shooting. Since the body is already small and compact the body was vulnerable to bending and deflections.

Horizontal Movement

For the horizontal movement of the body, the main component in charge of this movement is servo model FS5106B(fig 2.10). In order to ensure all parts perfectly fit before printing them, it was decided to model every part of the project as well as the servos and motors to be used. This servo will also be the main part holding the rest of the body above it as well as the components. The servo will be held in place by the horizontal servo mount, which will also have the tripod holding the whole project.

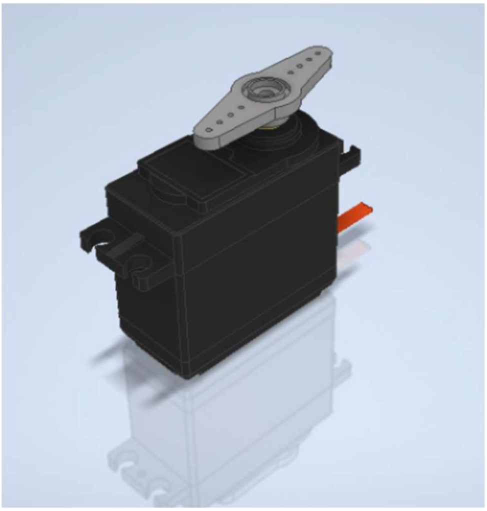


Figure 2-10: 3D model of FS5106B, initially used for horizontal motion

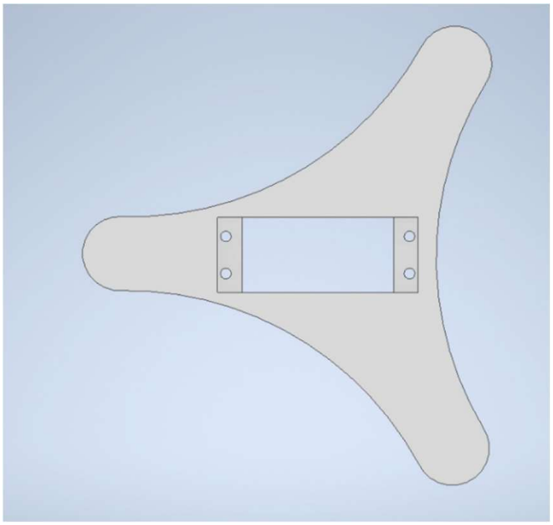


Figure 2-11: Top view of servo mount

Knowing the weights of all the components to be used and the final weight of the rest of the parts, we were able to do stress analysis on the body using multiple materials, including Nylon and PLA. The initial design of the servo mount weighed 41 grams but with the use of Inventors shape generator capability, it was possible to decrease the weight of the body to 30 grams while still maintaining safety which falls under the design criteria of keeping the design as light as possible.

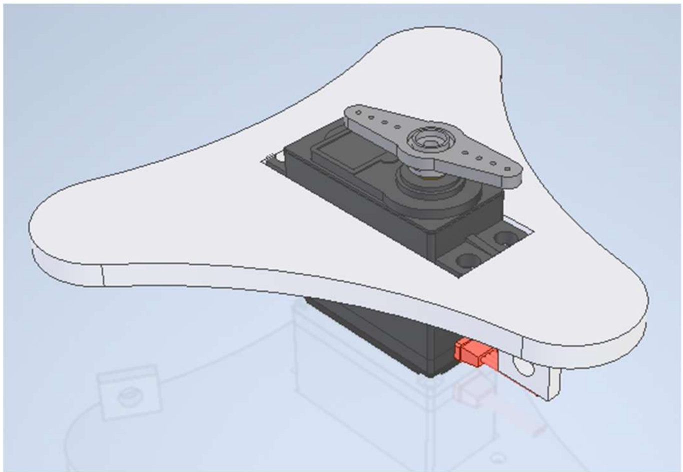


Figure 2-12: Orthographic projection of servo mount with FS5106B attached

Tripod

The tripod is used to lift the body parts above the ground, for that the design of modern camera tripods was taken into consideration, which can be changed in length and can be opened and closed. However, its mechanism is very complex and usually big. Considering the small size of the design a tripod as complex as a camera isn't the best option, therefore it was decided to create a more static design for the legs that don't allow for rotation or movement, which was then connected to the bottom of the Horizontal servo mount with a 5 mm screw and nut.

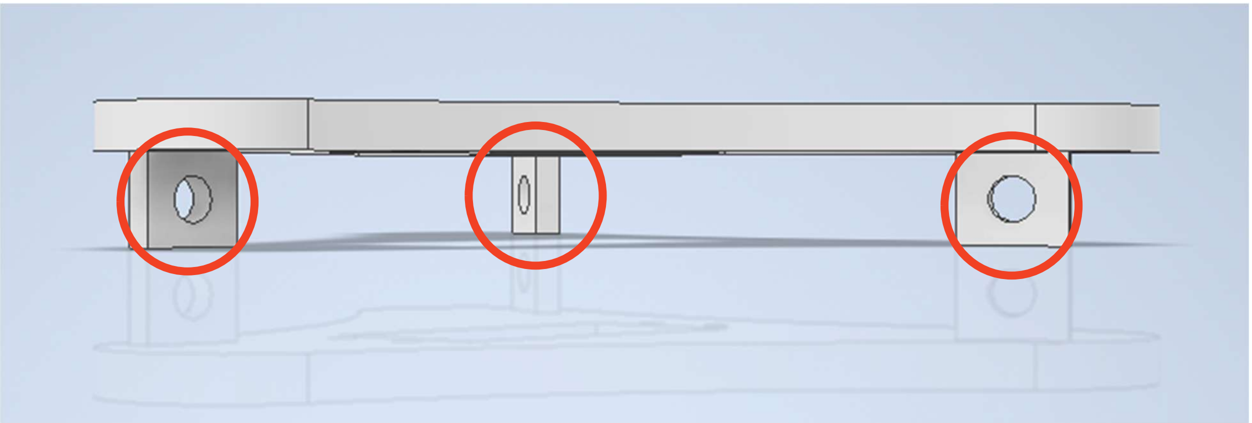


Figure 2-11: Side view of servo mount with the marked joints

The main features of the pods include a flat top to not allow rotation of the pillar and, have it fixed at the angle of the pillar's body itself as well as a flat base that is wider than the rest of the body.

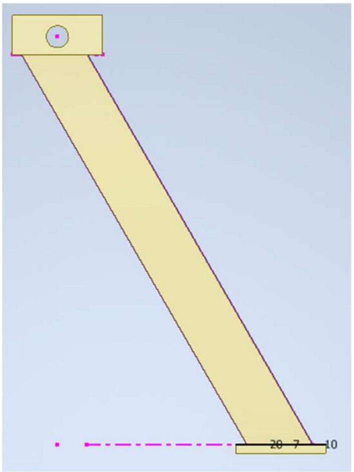


Figure 2-15: Side view of a pod in its preliminary design stage

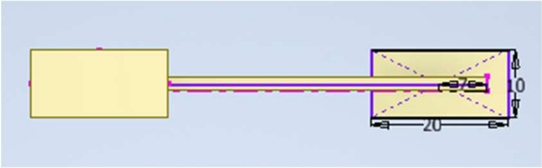


Figure 2-16: Top view

Initially the pods were mounted perpendicular to the body as shown in the figure below, unfortunately, this design proved to be unstable. Thus, a more casual approach was taken to the pods' angle by having it parallel to the body at a 45 degrees angle. This approach meant no change will be made to the pillar instead just the parts that connect the pods to the horizontal mount was changed by rotating them. This is shown in the figure below. This change in design proved to be more stable.

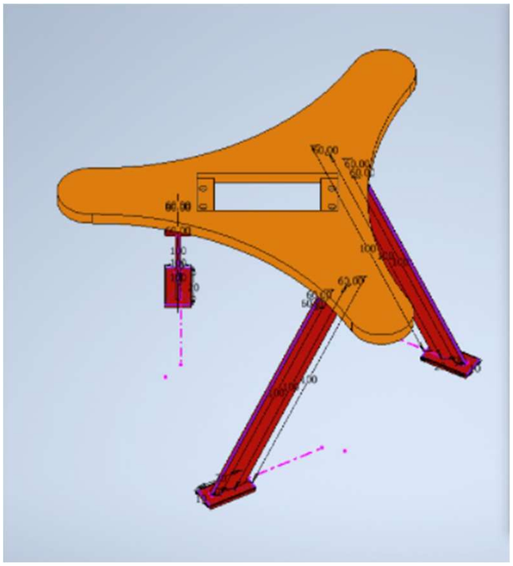


Figure 2-17: Orthographic view of initial servo mount design

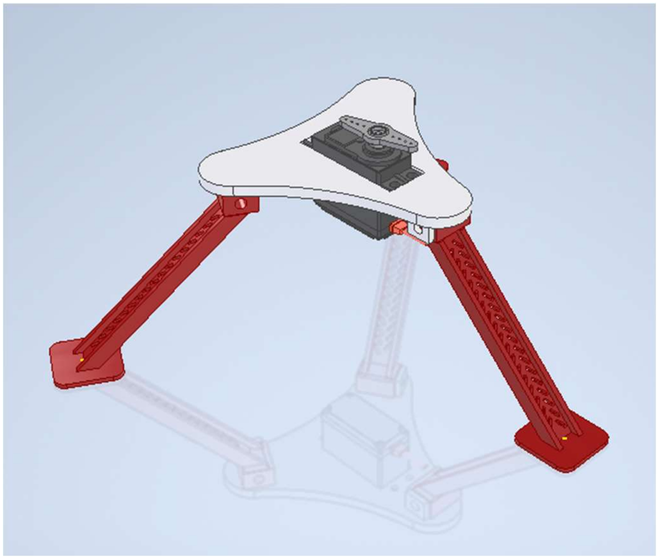


Figure 2-18: Orthographic view of final servo mount design

The pods (shown in figure 2.17) weighed about 14 grams and in order to fit the design criteria every extra gram mattered in the total cost of the project. However, having such a thin body (at only 2mm) meant that if any small adjustments are made to make it lighter it could affect its safety thus, stress analysis using Inventor had to be done after any readjustment to the body. There were 2 constraints to the body that couldn't be changed at all - top part of the body that connects using a screw to the horizontal mount and the wide base of the pillar both highlighted in red in the figure to the right. This meant adjustments can only be made to the part highlighted in blue. After trying multiple ideas, a redesign was made of the pillar that was safe to be used after applying stress analysis tests.

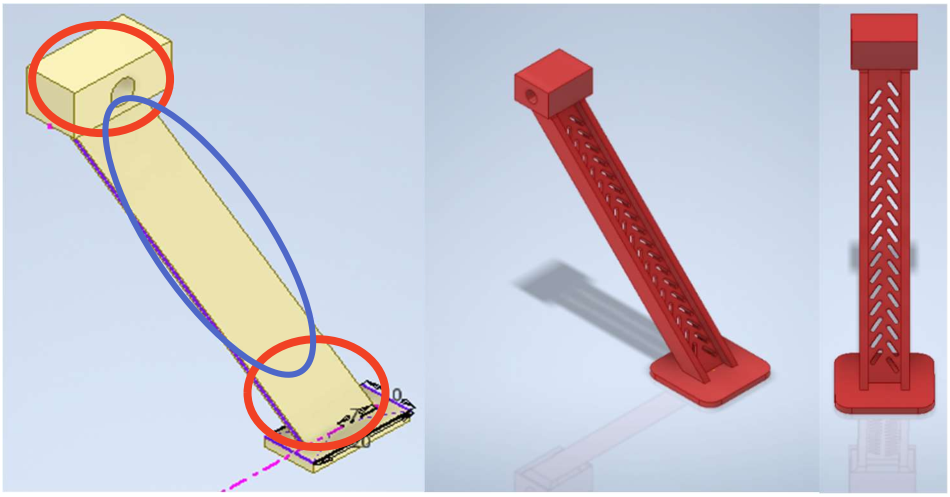


Figure 2-12: Orthographic view of pod at different design stages

Stress analysis showed safety when using all materials that were available for 3D printing including PETG and PLA in both criteria, SF and static deflection.

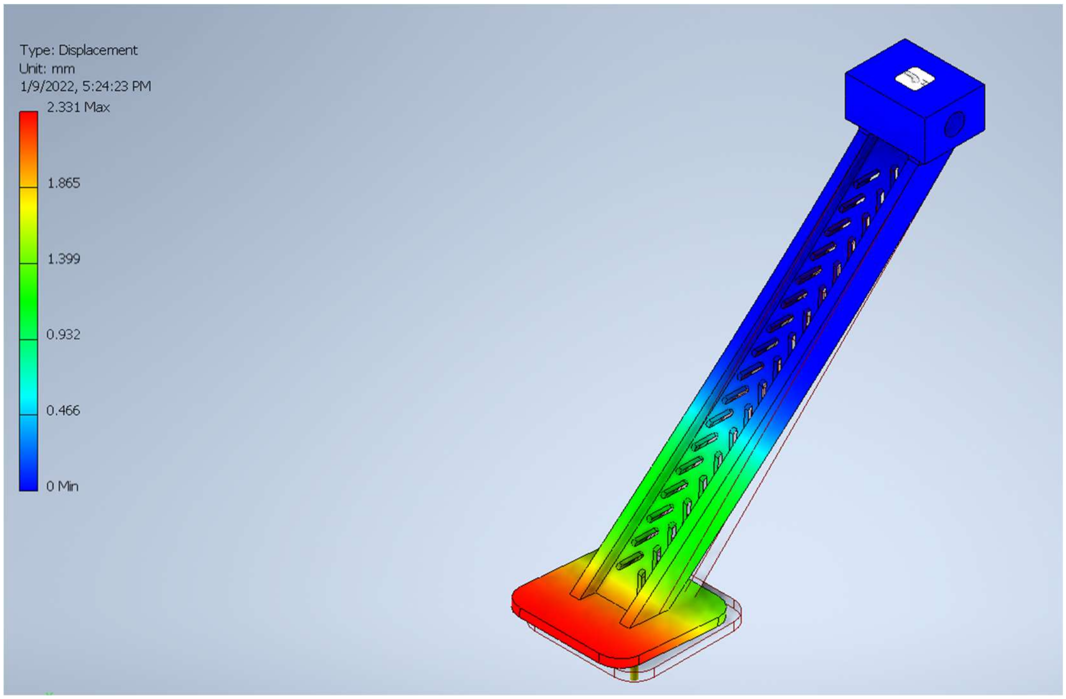


Figure 2-13: Orthographic view of simulation illustrating displacement distribution

As for the displacement of the pillar due to the stress on it, the body proved to be resistant and stable and only had a small displacement of 0.9 mm which won't affect or break the pillar making the pillar safe.

Vertical Movement

In order to transfer the horizontal movement to the rest of the body along with having a vertical mount an L shaped mount design was proposed, where the servos horns for both the servo in charge of the horizontal and vertical movement will. The Mount has the horn of the horizontal and vertical servos engraved on it so that the horn could be screwed in and held in place.

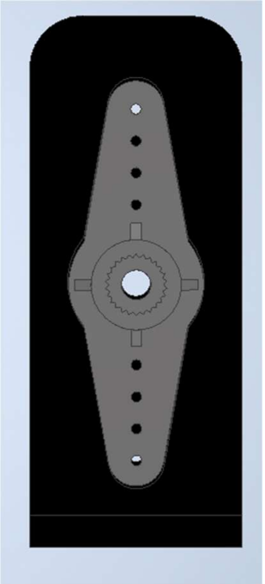


Figure 2-15: Top view of FS5106B horn on horizontal servo mount

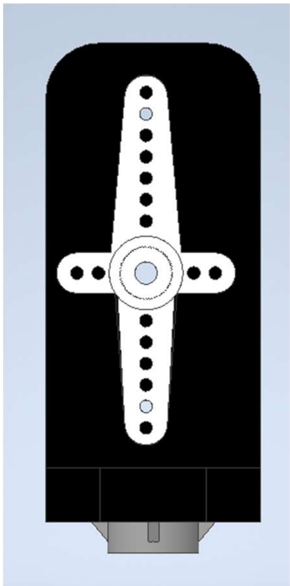


Figure 2-14: Top view of SG90 horn vertical servo mount

The vertical movement will be controlled by a servo. The team initially decided to use the SG90 servo due to its small size. As shown previously in the horizontal servo mount the servo is held in place by screws and fixated in the body so that the shaft rotates the attached horn. It was decided to look into a different approach for the vertical rotation. Rather than having the servo fixed in place and have the horn move, the opposite of this mechanism will be used, by having the horn grounded in the vertical servo mount, so that when the shaft of the servo moves the servo itself moves and having the body of the shooting chamber connected to the servo, so now when the servo moves the body moves along with it as show in the figures below.

-

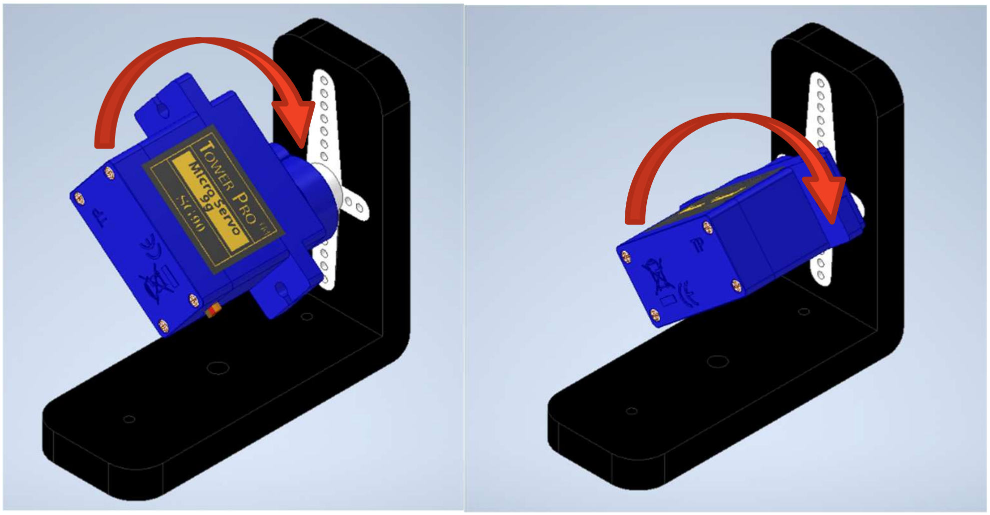


Figure 2-16: Orthographic view of vertical servo mount assembly illustrating its motion

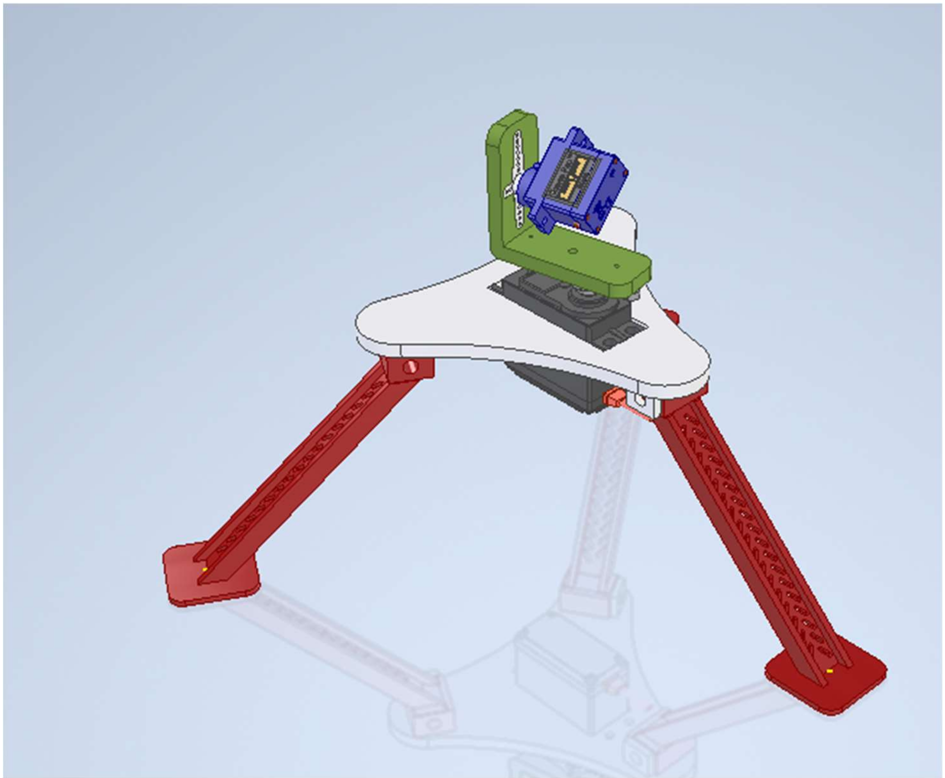


Figure 2-17: Orthographic view of vertical and horizontal assembly servo mount

Rollers

It was decided to use the motor friction as the projectile delivery system. This mechanism meant that we'll use 2 motors in the front of the magazine, where the dart can pass through and pick up speed from the high-speed rotating DC motors in the front. Multiple ideas were proposed on how to launch the darts, the team agreed to increase the diameter of the shaft of the motor by connecting it to a roller. This roller was intended to act as the new shaft of the motor, this will aid in being in full control of the distance between each shaft depending on the diameter of the dart being used, thus the thickness of the Roller can be changed or adjusted to provide enough distance between each. The dimensions of the darts being used were 13mm, thus, it was proposed to leave a spacing of less than 13mm to allow for friction of the dart. The rollers also needed to be wider than the DC motors used, for that and to accurately simulate this, existing models of the DC motor were used.

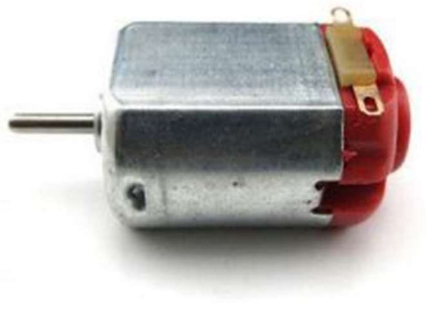


Figure 2-18: Photo of toy DC motors used for projectile delivery

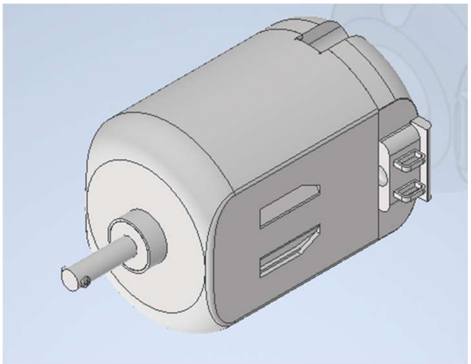


Figure 2-12-189: Orthographic view of toy DC motor

The maximum width of the motor was around 20mm, this meant that the minimum inside diameter of the roller had to be greater than 20mm, with 4mm thickness so that the distance between each roller can be maintained at 13mm. Furthermore, the bottom part of the roller had to be thick enough to insert the shaft of the motor in it. The roller should also be big enough to have the motor inside it.

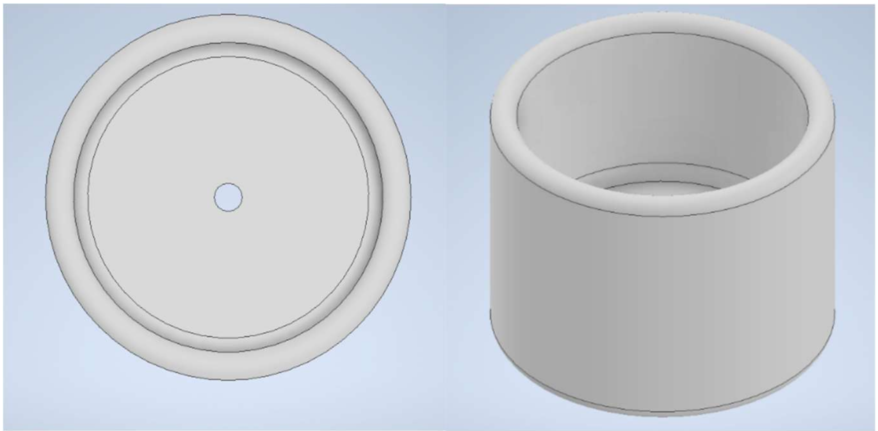


Figure 2-19: Top and Orthographic view of rollers used in projectile propulsion

Motor Mount

The 2 DC motors will be held in place by mount created to be placed in front of the magazine. Having a ready model of the motors aided in the design and, to design an accurate mount that can hold both motors. It was also proposed to make a space to place 2 screws that will aid in fastening the 3D part around the motors to hold them in place better. To hold the motor mount in the magazine, 4 holes with 4 screws were placed in it and the corresponding holes were placed in an extended part designed in the magazine to hold the motor mount.

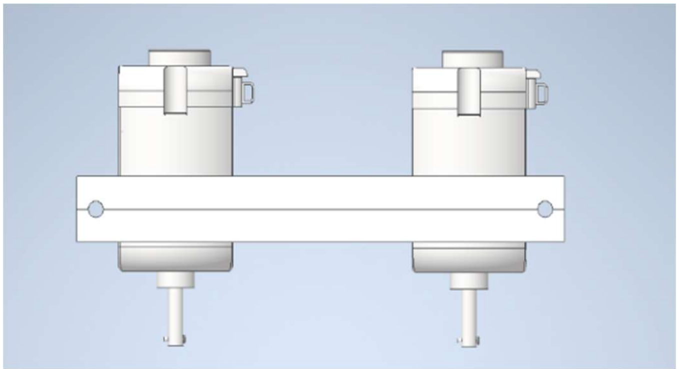


Figure 2-20: Motor mount assembly front view

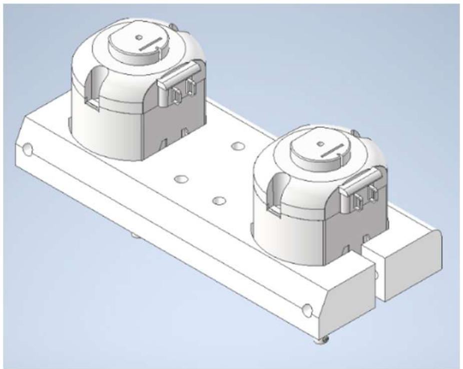


Figure 2-21: Motor mount assembly orthographic view

Magazine and Bolt Mechanism

The magazine is the part that will be holding the nerf darts, this part will also be connected to 2 different servos. A servo that is in charge of the bolt mechanism and another servo in charge of the vertical movement and connected to the vertical servo mount. The same approach as that of the rollers and the motor mount was used. It was used to model the motors in Auto Desk Inventor to be able to accurately simulate the movement as well as be able to precisely design a mount and holes for the servos to be placed in. Initially the 2 servos of choice were the servos model SG90. This allowed to accurately design 2 holes for each servo to fasten the servo in place as shown in the figure ()

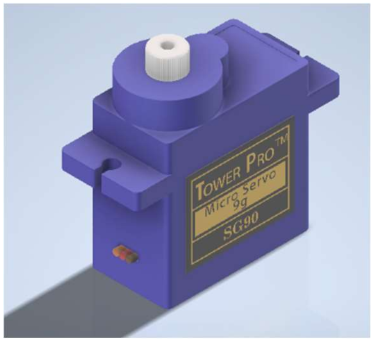


Figure 2-233: Orthographic view of SG90 servo motor used for bolt mechanism

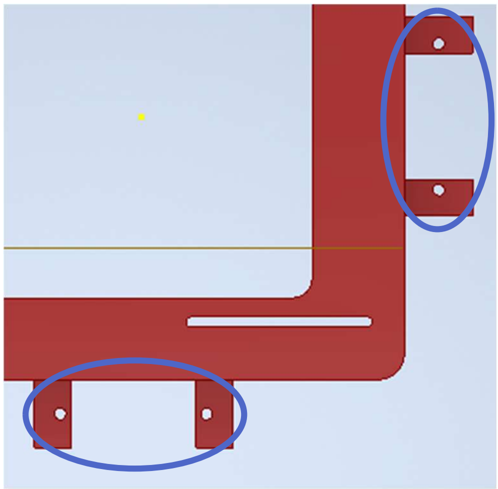


Figure 2-224: Side view of magazine and mechanism design illustrating locations of attachment

Bolt Mechanism

The bolt mechanism is the mechanism that will push the darts to be received by the 2 DC motors in the front. Using knowledge from theory of machines a design of the bolt mechanism was developed. The servo horn was placed in the first part and the other part was connected with a rod that had to be placed through the magazine so that when the servo moves the part highlighted in red it will move the white part and since the white part is constrained to move a transient movement in the rod, it can be moved in both directions.

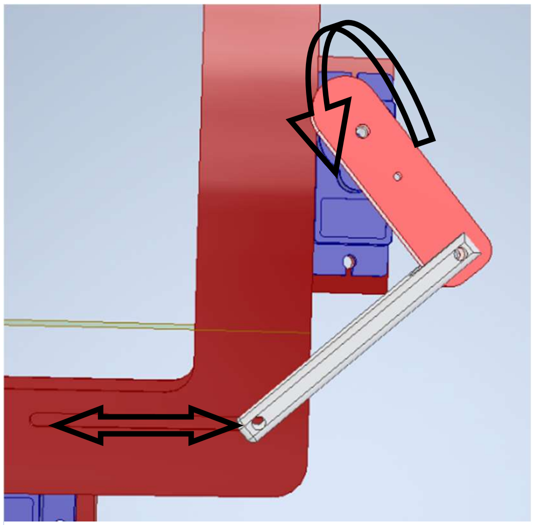


Figure 2-24: Side view of bolt mechanism assembly, with arrows illustrating slider crank mechanism

Magazine Design

The design criteria of the magazine were to make it capable of holding 4 darts at the same time and be able to hold the 2 servos and the 2 DC motors, this meant that the design had to be safe and resistance to displacement as much as possible. Any crucial change especially in the displacement would not only be the cause of failures such as fractures or breaking but the minute error in elevation can compound over the trajectory, accumulating to a complete miss. The initial design was based on a dart of size 8cm in length and width of 2 cm. As stated above in the motor mount, it was to be placed on the magazine, so a design of an extended part was developed - a part that can hold the motor mount in place. With the above constraints, the magazine had a big design and had a weight of 70 grams, compared to the rest of the parts it was heavy. It was possible to reduce this weight to 60 grams while maintaining a high safety factor.

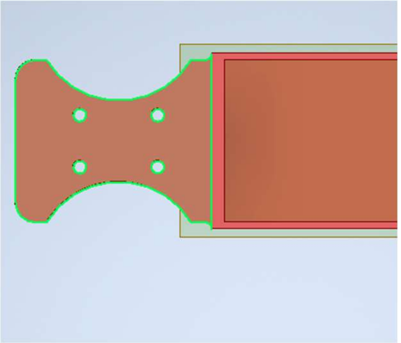


Figure 2-266: Top view illustrating close up of motor mount location

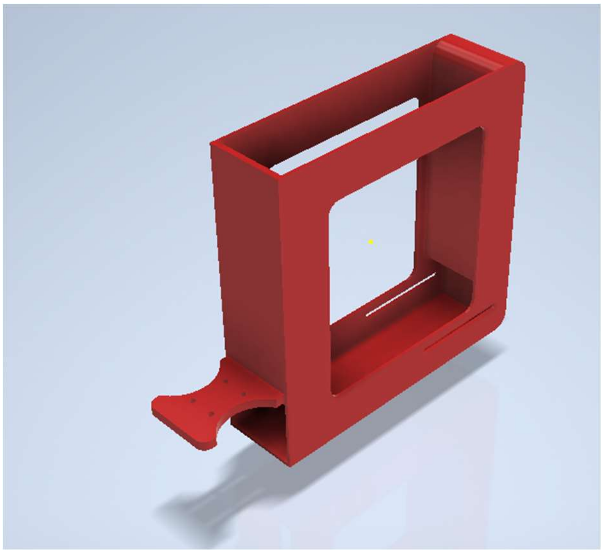


Figure 2-257: Orthographic view of magazine design

Initial Design

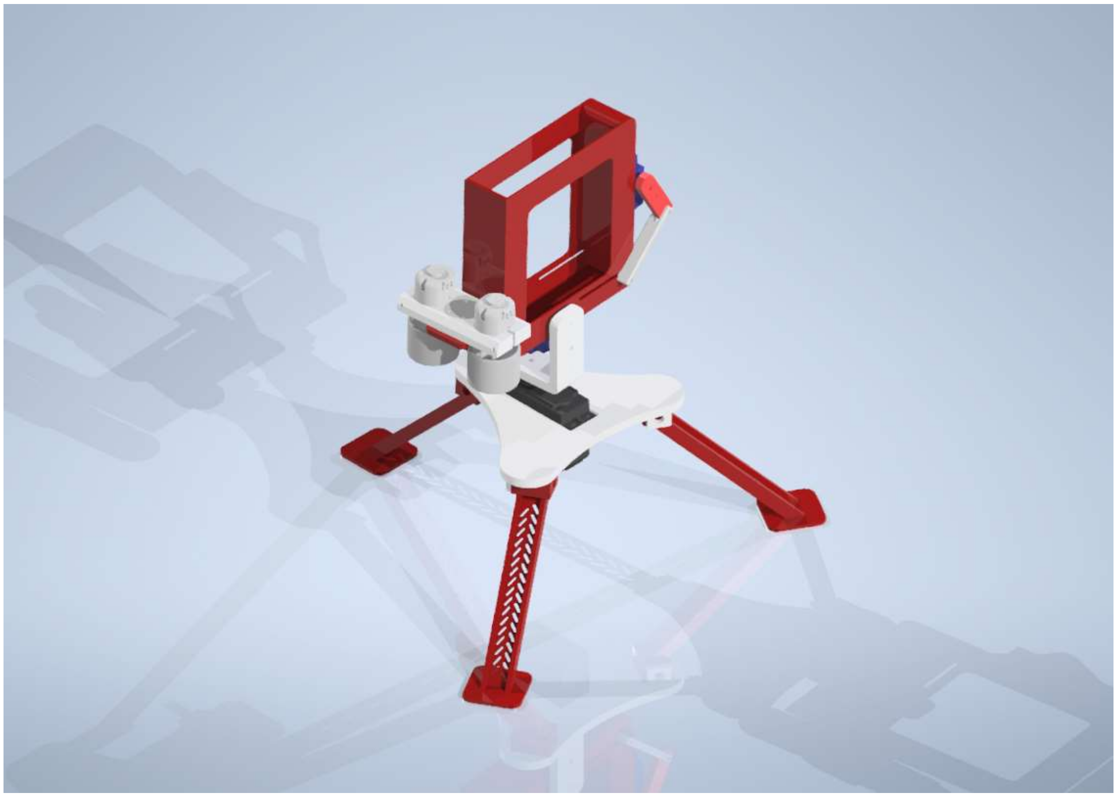


Figure 2-27: Orthographic view of complete design assembly

Chapter 3 Manufacture and Test

Manufacture

3D Printing and Material Selection

The parts were printed in PLA material, as chosen in the previous section corresponding to this one. The choice of material was influenced by its advantages over other options, such as PETG, in cost-efficiency, whilst maintaining a similar deflection as well as safety factor.

Assembly

The Tripod joints, which were designed to be fastened in place using screws and nuts to the Horizontal servo mount.

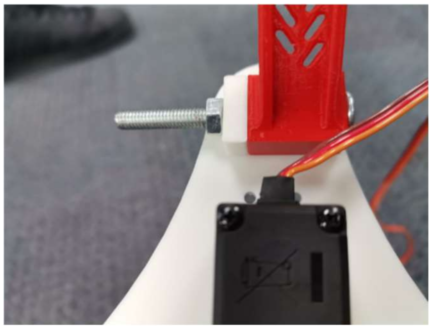


Figure 3-1: Bottom view photo of pod-servo mount joint assembly

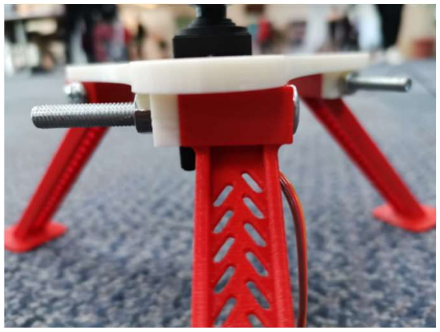


Figure 3-2: Side view photo of pod-servo mount assembly

The servo in charge of the horizontal movement was then attached to the horizontal mount by 4 screws and nuts in the designated holes that align with the corresponding holes in the servo. As designed above the vertical servo mount shown in dark green the figure below has slots where the servo' horns are to be placed in and fixated in place using a screw that fixates the horn to the vertical mount and connects them with the servo.

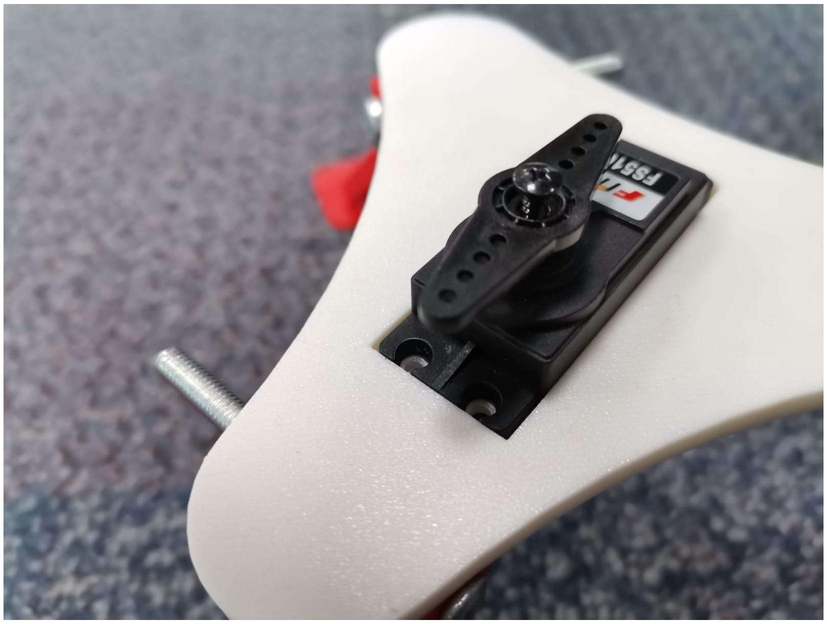


Figure 3-3: Photo of servo-servo mount assembly illustrating location of screws

The same process was reciprocated for the horn of the vertical servo using the slot designed in the vertical servo mount. Since the vertical servo mount will be basically floating and carrying the full weight of the magazine and the components attached to it. This would be creating a lot of stress on the horn and even though the lot is only big enough to only allow the horn to it in without any slight movement a safety measure and precaution had to be taken. Therefore, extra holes were designed in the vertical mount to allow 2 extra screws to be screwed in which will add extra safety and ensure stability of the horn to the vertical mount.

The magazine the is the part with the most components attached to, the servo in charge of the vertical movement is attached to the magazine using the designed holes in the magazine and using 2 screws and nuts. The same mechanism of assembly is repeated for the servo that runes the bolt mechanism. The 2 motors have then been placed in their designed places in the motor mount and tightened in place by a screw on each side. Motor mount was then screwed in place in the designed area in the magazine using 4 screws and nuts. With the motor mount fully screwed in this ensures that the shaft and roller of the motors align with the exit of the magazine.

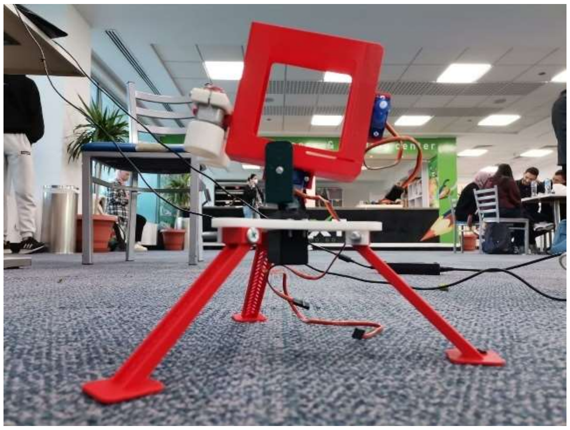


Figure 3-4: Photo of project assembly

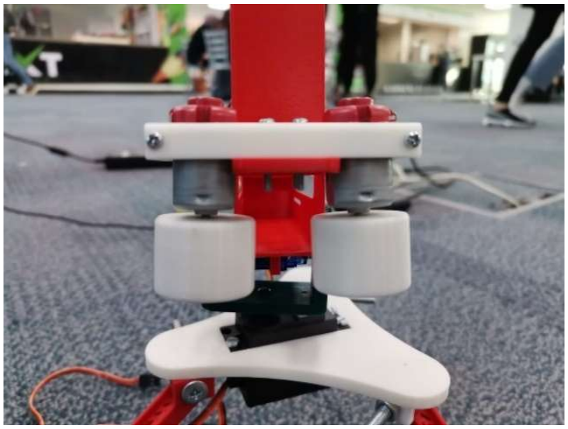


Figure 3-5: Photo illustrating dart path through roller then bolt mechanism

The bolt mechanism had 2 parts, the first part is designed to have a slot for the servo’s horn, then fixated in place with a screw that goes through the part, the horn and attaches them to the servo, this allows the new part to act as the new horn for the servo. And to translate that movement of the force horizontally another part was created, which attaches to the first pat with a screw, the other end of the second part is connected with a rod that goes through the magazine in its designed place, the rod goes through the full width of the magazine.

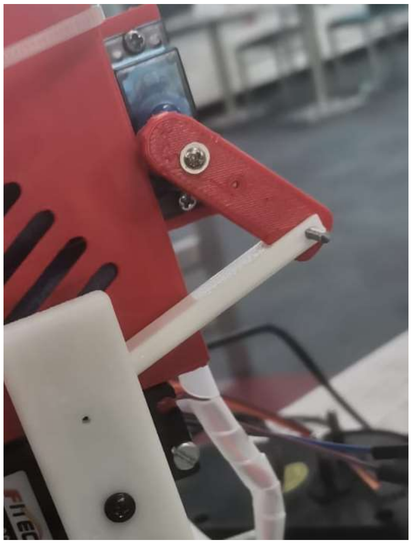


Figure 3-6: Photo showing slider crank mechanism in bolt

Bill of Materials

Component	Weight	Cost
Vertical Servo mount	12.6	46.57
Frontal DC motor mount	10	36.96
Rollors	13.2	48.79
Pods	34.2	126.41
Magazine	30	110.88
Horizontal Servo mount	35	129.36
Link 1	1.497	5.53
Link 2	1.485	5.49
FS5106B	80	330
SG90	9	50
Toy DC motor	20	120
Darts	2.4	50
Camera	79	250
Arduino Nano	0	125
Buck Converter		45
Power Supply	100	100
Total	328.382	1580

Testing and Improvement

Mechanical Failures and Fixes

<u>Joint</u>

One of the failures faced was the sudden fracture that led to the breaking of one of the joints, this meant that the part had to get through small adjustment to the joints and to get printed again.

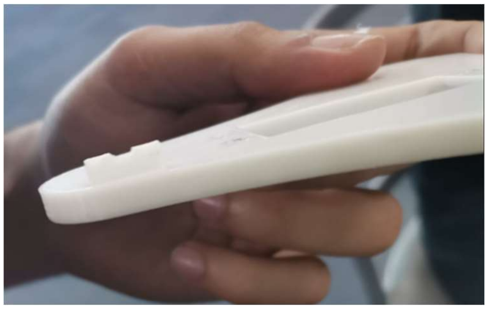


Figure 3-7: Close up photo illustrating mechanical failure of joint/

Adjustment was made to make the part wider and increase its thickness. This reduced the stress on the parts and increased the factor of safety.

Dart size

Initially the dart size was assumed to be 8x2cm and this meant a change in the design of the magazine had to take place to fit the darts that were purchased which were smaller in size compared to the initial assumption.

Vertical Servo

After the assembly was done, it was apparent that the SG90 servo wasn’t able to rotate the magazine, that is due to the DC motors in front of the magazine that exerted a high moment. This caused the servo motor to have the tendency to always tilt towards the direction of the motors due to the high opposing moment. This generated high instability. This required an immediate change to get a higher torque servo, the most feasible decision was to use the same servo as the one used in the horizontal movement which is the FS5106B servo model. This change of Servo required a rework of the design of the parts that were attached to the old servo.

Vertical Servo Mount

With the need to use FS5106B servo, which is bigger in size, a change to the design of the vertical mount had to take place, to change the design to allow for a spacing for the bigger servo horn, as well as increase the length of the vertical servo mount

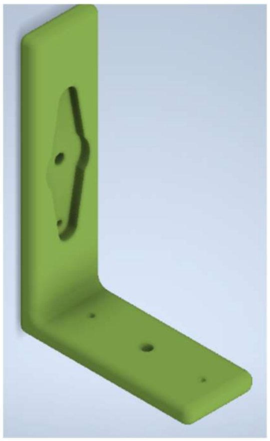


Figure 3-8: Orthographic view of vertical servo mount

Magazine

With crucial changes coming from the change in the servo and the size of the projectile being used, this would ultimately change the design of the magazine, by making smaller and to adjust the spacing between the servo holes to fit the bigger servo being used. The same approach was used as the Design criteria when making the magazine and that is to keep it as light as possible.

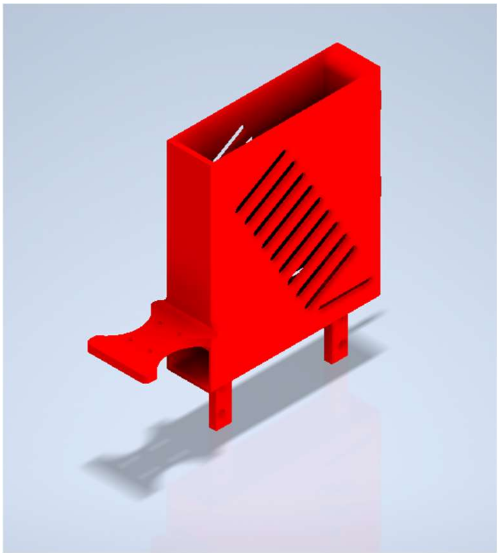


Figure 3-9: Orthographic view of magazine post-redesign

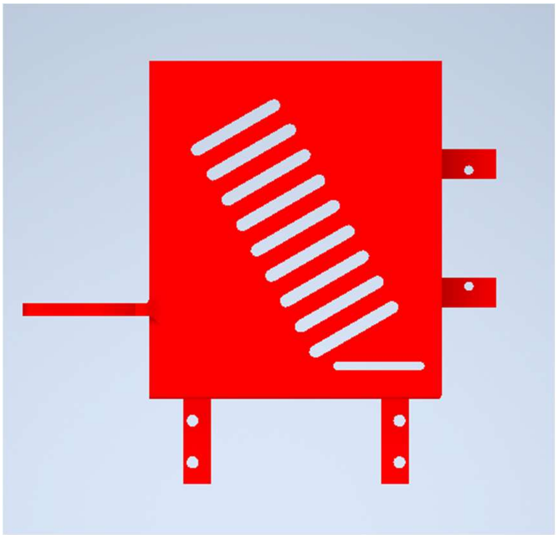


Figure 3-10: Side view of magazine post-redesign

Final Design

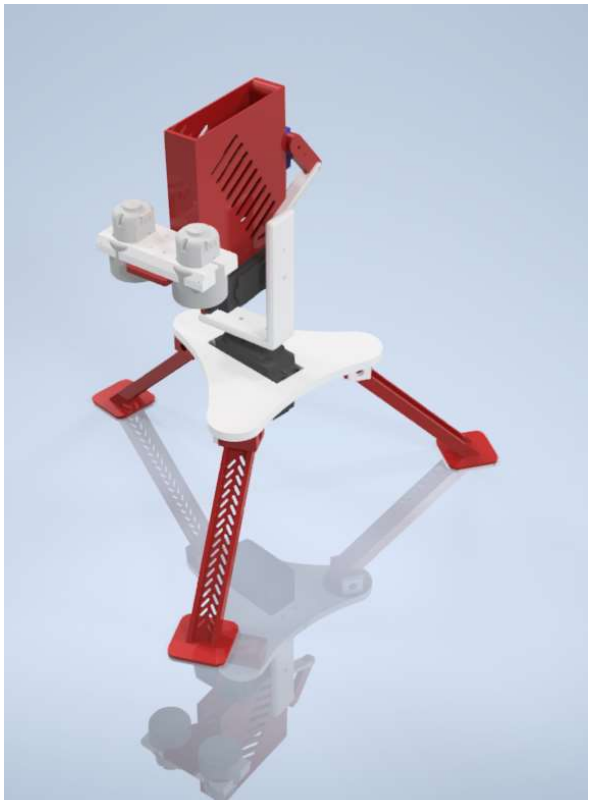


Figure 3-11: Orthographic view of final design after all modifications

Electrical Failures and Fixes

Power

While the testing of individual components was successful, as well as that of individual subsystems, such as the face detection, tracking, and shooting, when these were unified on the power circuit and tested, it didn't respond as planned. Servo motion was uncontrollable and seemed chaotic, while motor motion was slow and didn't seem to operate on any certain conditions. It was initially thought that the problem was a logic error within the code or that the processing was too much for the Arduino to handle. After thorough investigation of the code, each section was retried separately, several times until the code was cleared off all errors. The tracking algorithm was revised and cleared of errors as well, this left the power supply circuit as the only culprit in this matter.

As power distribution across circuit was inefficient, an investigation of the power source had to be commenced, here is an overview of the circuit diagram that was initially used: -

A SG-90 servo motor for the slider crank bolt mechanism rated at 4.8-6V
Two FS5106B servo motors for the elevation and panning motion rated 4.8-6V
Two toy DC motors with 1.6Ω of resistance (as measured by Avometer) running at 9V, no datasheets for which have been found.
An adjustable voltage power supply with a maximum of 12V and rated at 1A
A buck converter to variably adjust and stepdown the DC Voltage from the power supply

These were each examined for short circuits using Avometer and most were found to be functional, with the exception on one FS5106B that was found dysfunctional, probably due to the overcurrent burning the motor driver. After consulting with several doctors about this issue, we were directed that the issue was that the power supply current was too low and was insufficient.

Trials

Several suggested solution configurations were tried and tested: -

1. Connecting the motors to a battery rather than a power supply, thinking that the power supply current was insufficient or that the power supply had internal circuit issues that were undetectable for the team, as each component worked individually, but when the circuit was assembled, some components didn't work, when the voltage on the devices was raised, the system experienced overcurrent but was protected from complete failure thanks to the fuse in the power supply circuit.
2. In order to troubleshoot the issue, each component was separately tested on by one, then the load on the power supply was varied to check and compare the response 0f each component, after each component was tested, it was note that one the toy DC motors was spinning slower than the other one. The team was unable to identify the cause of this, as both were tested alone and on the same voltage level and the same power supply.
3. An external power supply capable of 9-12V and 2A of current was used as toy DC motor supply, albeit stepped down to 9V, while the other two FS5106B servo motors were powered by the adjustable plug-in power supply, while the SG90 servo motor, being the smallest motor in terms of power demand, only controlling the bolt mechanism was powered by the Arduino Uno 5V. While the latter is usually meant for sensors, as it

incorrect to demand a high current from a microcontroller, it was appropriate as the current demand was not high. This method is the only one that succeeded in solving the power issue, it was thus chosen as the configuration to be used in the final circuit. This also had the obvious positive side effect of increasing the motor RPM, which magnified the launch speed, causing the trajectory to be much straighter in its initial phase before giving into air resistance and deviating from the airless trajectory. This made it much more accurate, especially aiming with the camera system, as it vastly decreased bullet-drop over the effective face-detecting range. The Arduino Uno was initially used for testing due to the ease with which connections are made, once the circuit was finalized, the connections were transported to the Arduino Nano and soldered on the PCB.

Chapter 4 Conclusion

Conclusion

The aim of this project was to prototype and develop a working model of an autonomous defensive system, upon which improvements can be made and grow into a full-scale system for military or policing purposes. The need for more advanced weapons is steadily growing, at an ever-increasing rate, with AI-integrated weapons systems forecasted to become the new norm in the near future and becoming so even sooner than that. Thus, the need to catch up with the rapidly changing weapons industry, of which this project is a manifestation of. The mechatronics design process, marked by parallelism and overlap in each of the mechanical, electrical, and control systems was integral in the making of this project, as such a weapons system needs to have all its components work in tandem together, integrated to work together.

References

[1] [Online]. Available: https://www.armyupress.army.mil/Journals/Military-Review/English-Edition-Archives/May-June-2017/Pros-and-Cons-of-Autonomous-Weapons-Systems/.

[2] "Little French Kiev," [Online]. Available: https://www.littlefrenchkev.com/bluetooth-nerf-turret.

[3] Sciencish, "Youtube," [Online]. Available: https://www.youtube.com/watch?v=cy3QToyba4s.

[4] U. Navy, "Navy," [Online]. Available: https://www.navy.mil/Resources/Fact-Files/Display-FactFiles/Article/2167831/mk-15-phalanx-close-in-weapon-system-ciws/.

[5] "Raytheon Missiles and Defence," General Dynamics, [Online]. Available: https://www.raytheonmissilesanddefense.com/capabilities/products/phalanx-close-in-weapon-system.

[6] "Navy Look Out," [Online]. Available: https://www.navylookout.com/last-ditch-defence-the-phalanx-close-in-weapon-system-in-focus/.

[7] J. P. Hartke, "Characterization And Magnetic Augmentation of Low Voltage Railgun," December 1997. [Online]. Available: https://core.ac.uk/download/pdf/36701634.pdf.

[8] "Open CV Homepage," [Online]. Available: https://opencv.org/.

[9] M. J. Paul Viola, "ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 200," [Online]. Available: https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf.

[10] "My Great Learning," [Online]. Available: https://www.mygreatlearning.com/blog/viola-jones-algorithm/#sh21.

[11] "AP Monitor," [Online]. Available: https://apmonitor.com/pds/index.php/Main/CascadeClassifier.

[12] "AP Monitor," [Online]. Available: https://apmonitor.com/pds/index.php/Main/AdaBoost.

[13] N. Henelsmith, "Projectile Motion: Finding the Optimal Launch Angle," Whitman College, May 12, 2016.

[14] D. E. C. a. S. S. Jacobson, Ballistics: Theory and Design of Guns and Ammunition, CRC Press.

[15] U. Navy, "NavWeaps," [Online]. Available: http://www.navweaps.com/Weapons/WNUS_Phalanx.php.

[16] N. Shachtman, “WIRED,” [Online]. Available: https://www.wired.com/2008/12/israeli-auto-ki/.

[17] “Analytics Vidhya,” [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/learn-how-to-implement-face-recognition-using-opencv-with-python/.

Appendix

Concepts

Lorentz Force: $\vec{\boldsymbol{F}}_{Lorentz} = q(\vec{\boldsymbol{E}} + \vec{\boldsymbol{B}} \times \vec{\boldsymbol{v}})$

Ampere's Law: $\oint \vec{\boldsymbol{B}} \cdot d\vec{\boldsymbol{l}} = \mu_0 I$

Frictive Force: $F_{motor} = -b(\dot{x} - v_{motor})$

Hooke's Law: $\vec{\boldsymbol{F}}_{spring} = -k\vec{\boldsymbol{s}}$

Taylor Series: $f(z) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(z-a)^n$

Newton's 2nd Law: $\vec{\boldsymbol{F}} = \frac{d\vec{\boldsymbol{p}}}{dt}$

Newton's 3rd Law: $\vec{\boldsymbol{F}}_{\mathbf{12}} = -\vec{\boldsymbol{F}}_{\mathbf{21}}$

Linear velocity on an arc: $v_\theta = r\dot{\theta}$

Reynolds Number: $Re = \frac{\rho v D_h}{\mu}$

Characteristic length scale: $D_h = \frac{4A}{P}$

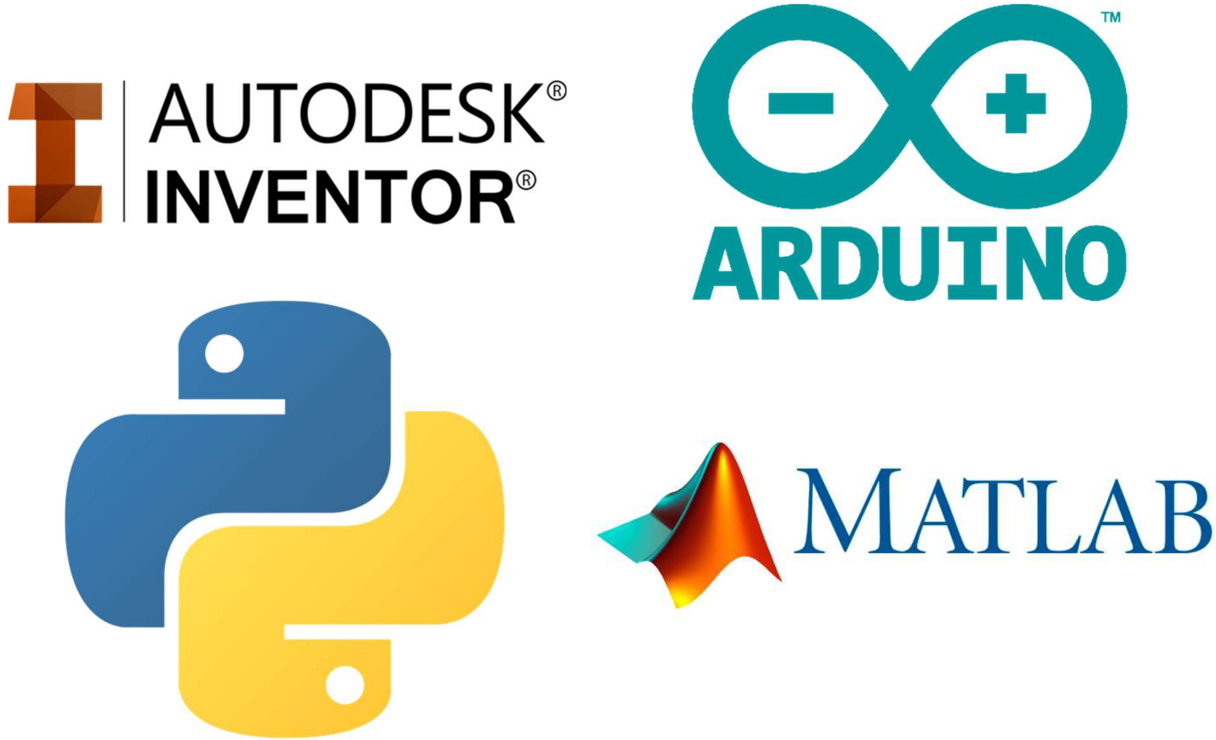
Runge-Kutta Method: $\frac{dy}{dt} = f(t, y),\ y(t_0) = y_0$
$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), t_{n+1} = t_n + h$

$$k_1 = f(t_n, y_n)$$
$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

Software



Code

Plot (2-5): -

```
y0 = 1;x0 = 0;
g = 9.81;
d = 0.013;
c = 0.25*d^2; m = 3/5*1e-3;
b = d*1.6e-4;
kc = c/m;
kb = b/m;
phi0 = pi/3;
u = 20;
uy = u*sin(phi0); ux = u*cos(phi0);

% no air resistance graph plot
step = pi/15;
for n = 0.5:0.25:10
    phi0 = n*step;
uy = u*sin(phi0);ux = u*cos(phi0);
r = (u^2/g)*sin(2*phi0);
x = (0:0.001:r);
y = x*uy/ux-g*0.5*(x/ux).^2;
plot(x,y)
plot(x,u^2/(2*g)-(g/2*u^2)*x.^2, 'linewidth',2)
xlabel('Horizontal displacement / m','Fontsize',20)
ylabel('Vertical displacement / m','Fontsize',20)
title("Various trajectories  with u = 10 ms^{-1} at interspaced
\phi",'Fontsize',24)
```

```
hold on
end
axis([0 12 0 8])
```

Plot (2-6): -

```
y0 = 1;x0 = 0;
g = 9.81;
d = 0.013;
c = 0.25*d^2; m = 3/5*1e-3;
b = d*1.6e-4;
kc = c/m;
kb = b/m;
phi0 = pi/3;
u = 20;
uy = u*sin(phi0); ux = u*cos(phi0);

% linear air resistance plot
steptime = 0.01;
t = 0:steptime:2.5;
xmax = ux/kb;
x = xmax*(1-exp(-kb*t));
y = (uy-g*t)/kb+(g-exp(-kb*t)*(g+kb*uy))/(kb^2);
plot(x,y,'linewidth',1)
hold on
x1 = ux*t;
y1 = -4.9*t.^2 + uy*t;
plot(x1,y1, '--','linewidth', 1)
xlabel('Horizontal displacement/ m','Fontsize',16)
ylabel('Vertical displacement / m','Fontsize',16)
title("Comparison between zero drag and linear drag, \phi =
\pi/3",'Fontsize',24)
legend('with linear air resistance','without air resistance','Fontsize',18)
axis([0 10 0 4])
```

Plot (2-7): -

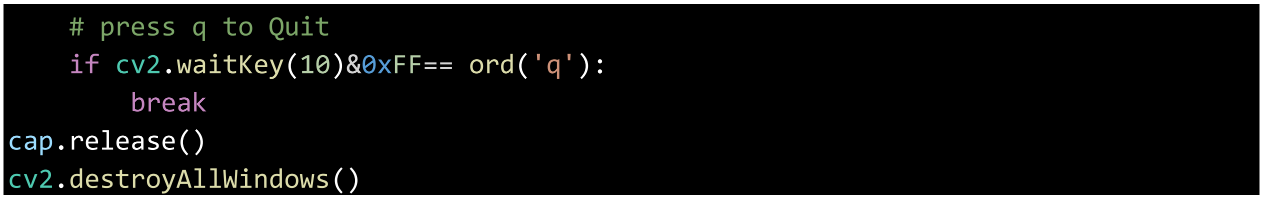
```
function dydt = odefun(t,y)
dydt = zeros(2,1);
dydt(1) = -0.042250000000000*y(1)*(y(1)^2+y(2)^2)^0.5;
dydt(2) = -9.81-0.042250000000000*y(2)*(y(1)^2+y(2)^2)^0.5 ;
end
tspan = 0:0.01:5;
yinit = [ux;uy];
[t,y] = ode45('func', tspan, yinit);
n1 = 0:0.1:15;
y1 = (n1*uy/ux-9.81*0.5*(n1/ux).^2);
v1 = y(:,1);
v2 = y(:,2);

plot(t,y,'linewidth',2.5)
legend('x-component','y-component', 'fontsize', 16)
xlabel('time/ s', 'Fontsize', 16); ylabel('Velocity/ m s^{-1}', 'Fontsize',
16);
title('Velocity-time graph of trajectory at \phi = \pi/3, u = 10 ms^{-1}',
'Fontsize', 24);
```

Plot (2-8): -

```
% plotting distance
d1=cumtrapz(t,v1);
d2=cumtrapz(t,v2);
plot(d1,d2,'linewidth', 2)
axis([0 40 -10 30])
hold on
plot(n1,y1,'--','linewidth', 2)

legend('with quadratic air resistance','no air resistance','fontsize',18)
xlabel('Horizontal displacement/m','Fontsize',16); ylabel('Vertical
displacement/m','Fontsize',16);
title('Trajectory comparison with \phi = \pi/3, u =20 ms^{-
1}','Fontsize',24);
```

OpenCV Python

```
import cv2
import serial,time
face_cascade = cv2.CascadeClassifier('haar_face.xml')
cap=cv2.VideoCapture(1, cv2.CAP_DSHOW)

ArduinoSerial=serial.Serial('com12',9600,timeout=0.1)

time.sleep(1)

while cap.isOpened():
    ret, frame= cap.read()
    frame=cv2.flip(frame,1)  #mirror the image
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces= face_cascade.detectMultiScale(gray,1.1,6)  #detect the face
    for x,y,w,h in faces:
        #creating string for coordiantes
        string='X{0:d}Y{1:d}'.format((x+w//2),(y+h//2))
        print(string)

        ArduinoSerial.write(string.encode('utf-8')) #send string to arduino
        time.sleep(0.05)

        #plot the center of the face
        cv2.circle(frame,(x+w//2,y+h//2),2,(0,255,0),2)
        #plot the roi
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),3)

    cv2.imshow('img',frame)

    #for testing purpose
    read= str(ArduinoSerial.readline(ArduinoSerial.inWaiting()))
    time.sleep(0.05)
    print('data from arduino:'+read)
```



```
    # press q to Quit
    if cv2.waitKey(10)&0xFF== ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Tracking Arduino

```
#include<Servo.h>

Servo x, y;
int width = 640, height = 480;  // total resolution of the video
int xpos = 90, ypos = 90;  // initial positions of both Servos
int shoot = 15; //initial position of shooting servo
void setup() {

  Serial.begin(9600);
  x.attach(9);
  y.attach(10);
  s.attach(11);
  x.write(xpos);
  y.write(ypos);
  s.write(shoot);
}
const int angle = 1;   // degree of increment or decrement

void loop() {
  if (Serial.available() > 0)
  {
    int x_mid, y_mid;
    if (Serial.read() == 'X')
    {
      x_mid = Serial.parseInt();  // read center x-coordinate
      if (Serial.read() == 'Y')
        y_mid = Serial.parseInt(); // read center y-coordinate
    }

    int errorx = x_mid - width/2;
    int errory = y_mid - height/2;
    int error = sqrt(errorx*errorx + errory*errory);

    /* adjust the servo within the squared region if the coordinates
        is outside it
    */
    if (x_mid > width/2 + 20)
      xpos += angle;
    if (x_mid < width/2 - 20)
      xpos -= angle;
    if (y_mid < height/2 + 20)
      ypos += angle;
    if (y_mid > height/2 - 20)
      ypos -= angle;
    if (error <=20)
      s.write(70);
      delay(100);
      shoot=15

    // if the servo degree is outside its range
    if (xpos >= 180)
      xpos = 180;
    else if (xpos <= 0)
      xpos = 0;
    if (ypos >= 180)
      ypos = 180;
    else if (ypos <= 0)
      ypos = 0;

    x.write(xpos);
    y.write(ypos);
    s.write(shoot);
  }
}
```