Title
Rack Attack

Authors
Caguimbal, Gregory
Chun, Thomas
Jones, Hunter
et al.

Publication Date
2017-05-25

Undergraduate

Concept Design Report
(Rack Attack)

Dr. Christopher Kang, Professor Robert Shaefer, Professor Tsu-Chin Tsao
TA: Matthew Gerber

Team #17
Gregory Caguimbal
Thomas Chun
Hunter Jones
Fadi Rafeedi
Samuel

MAE-162D: Mechanical Product Design-I
Winter 2016

Mechanical & Aerospace Engineering Department
University of California, Los Angeles

February 17th, 2016

Figure 1: Team 17; team members pictured from left to right: Greg Caguimbal, Thomas Chun, Fadi Rafeedi, Samuel, Hunter Jones

Abstract

The purpose of this report is to walk readers through the conceptual design development of an autonomous transporter that carries a golf ball and shoots it as far as possible. High and low-level design requirements are outlined to discuss intended functionality of the transporter. Preliminary design concepts are presented. An Objective Tree and a Pairwise Comparison Chart are then used to select the best concept based on desired qualities. CAD models of the chosen design are shown. Preliminary calculations for the drive and launch systems are formed and discussed to prove the robot's ability to succeed at accomplishing the proposed task.

Table of Contents

List of Figures

List of Symbols

Symbol	Definition	Units
$a_{inertia}$	Robot acceleration	m/s^2
F_f	Tire to plywood friction force	N
$F_{inertia}$	Force necessary to overcome inertia	N
F_m	Gravitational force on the device	N
F_{prop}	Required propulsion force	N
F_{rol}	Rolling friction force	N
F_{tf}	Traction force of both front wheels	N
F_{tr}	Traction force of both rear wheels	N
F_w	Force due to weight along ramp	N
g	Acceleration due to gravity	m/s^2
h	Disk height	m
h_c	Height of center of mass	m
I	Instant center of velocity	m/s
I_n	Motor no load current	A
I_s	Motor stall current	A
$J_{GL\rightarrow M}$	Reflected moment of inertia of load gear	$kg * m^2$
J_{GM}	Moment of inertia of motor gear	$kg * m^2$
J_l	Moment of inertia of load	$kg * m^2$
J_m	Moment of inertia of drive shaft	$kg * m^2$
J_r	Reflected load inertia	$kg * m^2$
J_{refl}	Load inertia reflected	$kg * m^2$
J_t	Total inertia at motor	$kg * m^2$
L	Distance between front wheel and rear wheel	m
L_c	Distance between rear wheel and center of mass	m
N_f	Normal force of both front wheels	N
N_r	Normal force of both rear wheels	N
P_{req}	Required power	W
R	Link vector	-
t_i	Section time	s
T_f	Friction torque	$N * m$
T_m	Motor torque	$N * m$
T_{pop}	Required propulsion torque	$N * m$
T_{req}	Required drive system torque	$N * m$
T_s	Motor stall torque	$N * m$

$V_{max,i}$	Section maximum velocity	m/s
α	Angular acceleration of wheel	rad/s^2
β	Weight distribution	-
θ_i	Incline angle	rad
μ_{4WD}	Four wheel drive coefficient of friction	-
μ_{FWD}	Front wheel drive coefficient of friction	-
μ_{rol}	Rolling friction coefficient	-
μ_{RWD}	Rear wheel drive coefficient of friction	-
μ_{wheel}	Tire to plywood friction coefficient	-
v	Disk velocity	m/s
ω_{in}	Crank angular velocity	rad/s
ω_n	Motor no load speed	rad/s
ω_{out}	Driver angular velocity	rad/s
y	Height	m
x	Horizontal distance	m
v_o	Initial velocity	m/s
k	Spring constant	N/m
m_{gb}	Mass of golf ball	kg
θ	Incline	rad
Δl_i	Initial spring extension length	m
Δl_f	Final spring extension length	m
h_f	Final height	m
h_i	Initial height	m
$\Delta P_{gravitational}$	Change in gravitational potential energy	$N * m$
ΔK	Change in kinetic energy	$N * m$
ΔP_{spring}	Change in spring energy	$N * m$
x_{land}	Horizontal distance between robot launch position and golf ball impact point	m
h_{land}	Vertical height of golf ball landing relative to initial height	m
$h_{launcher}$	Vertical height of golf ball relative to bottom of robot wheels at time of release	m

$h_{platform}$	Vertical height from bottom of ramp structure to top platform	m
t	Time	s
v_f	Final velocity	m/s
v_i	Initial velocity	m/s
N_{fcam}	Normal force between flapper short leg and cam	N
F_{fcam}	Frictional force between flapper short leg and cam	N
L_s	Flapper short leg length	m
L_l	Flapper long leg length	m
F_{rod}	Horizontal force on flapper pivot shaft	N
N_{frod}	Vertical force on flapper pivot shaft	N
F_{gb}	Normal force between long flapper leg and golf ball	N
M_{s1}	Moment due to torsion spring 1	$N * m$
M_{s2}	Moment due to torsion spring 2	$N * m$
mu_1	Friction coefficient between aluminum and aluminum	-
F_{spr}	Horizontal distance between robot launch position and golf ball impact point	N
N_{fgb}	Vertical height of golf ball landing relative to initial height	N
F_{fgb}	Normal force between tension spring beam and golf ball	N
mu_2	Friction coefficient between aluminum and golf ball	-
ΣF_x	Sum of forces in the x direction	N
ΣF_y	Sum of forces in the y direction	N
ΣM_{rod}	Sum of moments about the pivot shaft	$N * m$
Δl	Spring extension length	m
L_1	Distance between pivot shaft and pinion along arm	m
L_2	Distance between pivot shaft and rear upper guide along arm	m
L_3	Distance between pivot shaft and arm center of gravity	m
F_{py}	Vertical force between arm (rack) and pinion	N
F_{max}	Maximum force between pinion	N

	tooth and rack	
mu_3	Friction coefficient between aluminum and plastic roller wheel	-
R	Normal force between arm and rear upper guide	N
R_f	Frictional force between arm and rear upper guide	N

List of Tables

1. Introduction

Mechatronics is currently one of the most rapidly expanding fields in engineering, integrating more and more autonomous technology into our daily lives. From self-driving cars to artificially intelligent robots this field has proved very helpful. For example Professor Dennis Hong, now a researcher-cum-professor at UCLA, developed a car for the visually impaired. The car uses GPS to determine its location and laser range finder to sense obstacles [1].

Autonomous robots combine four areas of engineering consisting of mechanical, electrical, control, and computer systems. When implemented together, these disciplines can work as a team to perform the cycle of sensing, processing, and acting that allows an autonomous robot to complete a task.

1.1. The Challenge

Our team was tasked with developing a device that navigates its way up a ramp to a final platform where it must shoot a golf ball in a repeatable and accurate fashion. The ramp has 3 platforms, 2 inclines, and a 3-inch high safety wall on all sides, except at the high end, as shown in Figure 2. The overall length of the platform is 180 inches and the pathway is made of plywood. The robot must start in the lower square zone where it is manually loaded with a golf ball. When activated, it must navigate its way up the ramp autonomously. Once the platform has been reached the robot must launch the golf ball. Afterward it needs to return to its starting position and prepare to be loaded again for another trip. All this must happen in a timely manner and active steering must be employed to control the direction of the robot's movement. The fine details of the challenge are presented in Section 2.1.

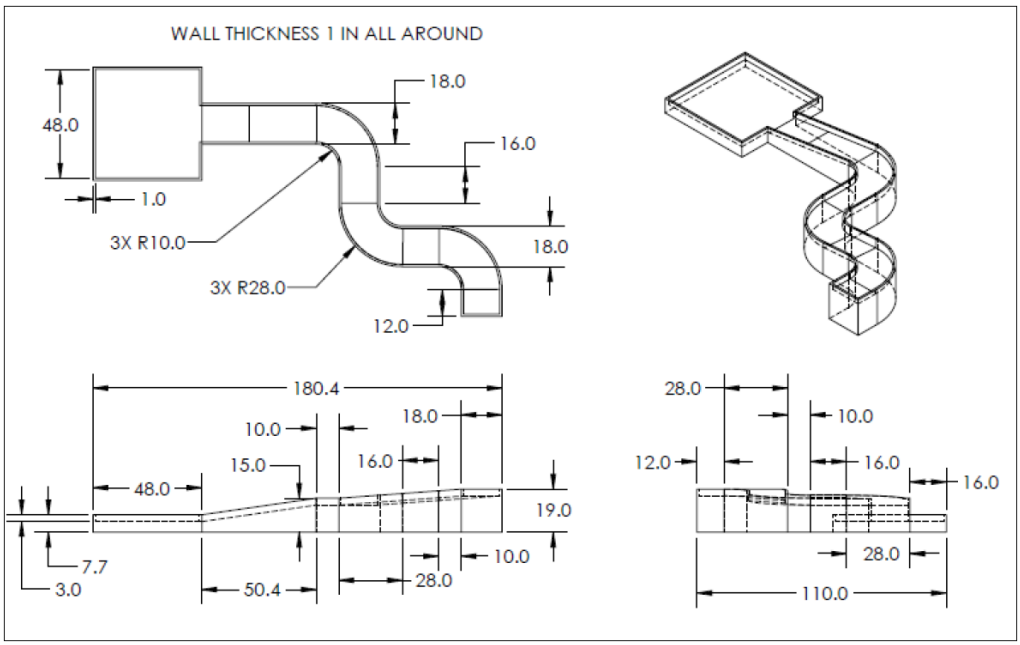


Figure 2: Ramp Dimensions and Views [2]

1.2. Literature Review

In order to first understand the state of the art in automated robotics and establish a starting point for the robot's design, literature discussing past successes of robots accomplishing similar objectives was investigated. In particular, methods for propulsion, steering, sensing, and shooting were reviewed and the findings are presented below.

Synthesizing a "perfect" Ackerman is very difficult to do with practical linkage designs. However, it can be closely approximated by a trapezoidal geometry as shown in Figure 3 [3]. Through the synthesis of the steering linkage for Rack Attack, the initial calculations began with this trapezoidal geometry and were then iterated on to optimize turning and the torque required to do so.

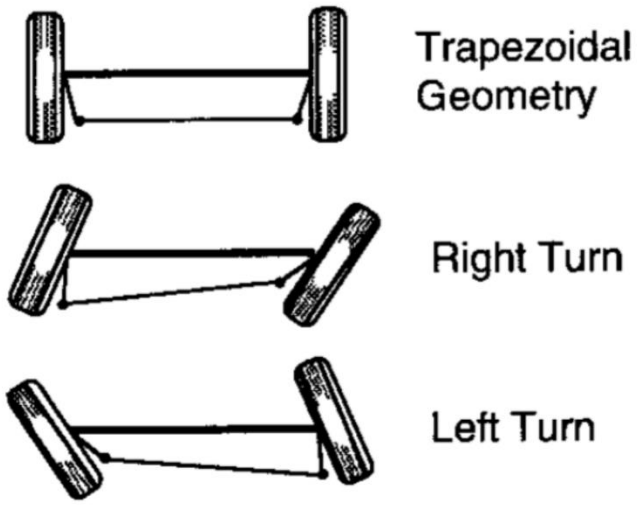


Figure 3: An illustration of the trapezoidal geometry that gives a close approximation to a perfect Ackerman Steering. [3]

The Wired popular literature article *This Incredible Hospital Robot Is Saving Lives. Also, I Hate It* by Matt Simon discusses a series of autonomous hospital robots called

named Tug that have become increasingly prevalent in recent years. The article touches on the robot's sensing system, which uses a laser and 27 infrared and ultrasonic sensors to avoid objects. [4]

The vision of our device is an essential aspect of the project. We consulted patents on obstacle detection and gained valuable insight on sensing technology and the execution of sensing from the perspective of an autonomous device. The first patent under the title "Proximity sensor" [5] discussed the use of a configurable ultrasonic sensors, in terms of power and the detection feature, in determining obstacles that is more efficient and reliable than the common ultrasonic sensor. The second patent under the title "Obstacle detection device" [6] uses image processing methods to detect obstacles in the path of the device. Both of these patents used multiple modules to process the information from their respective forms of sensors which translated to the detection of obstacles.

Associate Professor, Richard W. Wall and his students from University of Idaho developed a low-cost autonomous vehicle from a remote-controlled car. The purpose of their autonomous vehicle was to demonstrate that a safe autonomous vehicle with a low budget is not beyond reach. Similar to the setup of the car for the blind, this autonomous vehicle used GPS to determine its location, but also an ultrasonic range finder to avoid obstacles [7].

Figure 4. Low-cost autonomous vehicle developed by Assoc. Prof. Richard W. Wall and his students from Univ. of Idaho.

Autonomous vehicle has been in the news for quite some time now, with much research and resources devoted the field, making autonomous vehicle closer to reality. A news article by USAToday dated to 18 November 2015 shows Los Angeles Mayor, Eric Garcetti, testing a self-driving car. Although not autonomous, it is a big step towards the goal of autonomous vehicle driving alongside vehicles that is controlled by their driver. [8]

Specifications of the Tetrix motor were used to predict if it would be sufficient enough to provide power to our robot. Specification details can be seen in the Appendix. [9]

1.3. Our Design Solution

Our design concept, which incorporates single-axle non-differential drive, ackerman steering, and a spring-loaded rack and pinion cam launch system, provides a solution unlike the prior-art found in our review. The unique system strikes a novel balance of reliability, creativity, and robustness.

This report will outline the details of our design concept starting with the high level design requirements stated by the client and the low level requirements formed by the design team. Preliminary design concepts will be discussed briefly followed by an explanation of the scoring process used to downselect to one design. An overview of the selected design will be presented along with preliminary CAD models. Lastly, in-depth analyses of the robot's mechanical architecture and move profile will be presented to prove the validity of the design as a competitive solution to the presented challenge.

2. Design Concept

2.1. High Level Design Requirements

The High-Level Design Requirements are as follows:

1. The robot must navigate the ramp without unintentional contact with the ramp wall.
2. An active steering system must be utilized. No torque steering or skid steering is allowed.
3. The robot must launch a single golf ball from the top platform. The golf ball is initially placed on the device at the starting platform. The golf ball must be launched at least 24 inches horizontally away from the origin of launch. The distance of the ball launch must be predicted and must be delivered within 15% of the estimated target distance each time.
4. The golf ball may only be placed on the device when the device is on the starting platform.
5. The golf ball may not be dropped during its trip to the unloading platform.
6. The robot must return to starting platform after launching the golf ball.
7. The device must be initiated *once* by an "on/off" switch at the starting platform when starting each run. This will be the only time the switch is initiated throughout the run until completion.
8. Each run must be completed within 3 minutes.
9. The device must start within an 11 x 11 x 11 in^3 volume. However the device may change its dimensions after start.
10. The source of power for the device will strictly be ordinary batteries; disposable or rechargeable, no mixing of battery types. More specifically:
    a. Disposable option limit: Ten AAA, AA, C, or D (1.5 Volt) and three 9-Volt
    b. Rechargeable option limit: Twelve 1.2-Volt and three 9-Volt.
11. Only the robot may be on the runway during the runs.
12. The cost of the device must not exceed $375 [2].

A successful run is defined as follows: the device must carry the golf ball up the ramp to the unloading platform, without dropping the ball. It must launch the ball away from the unloading platform by a minimum horizontal distance of 24 inches. The device must return to the starting platform.

2.2. Low Level Design Requirement

1. Number of wheels: three to four
2. Drive system: RWD on forward trip and FWD on backward trip.

3. Number of drive motors: one
4. Type of drive motors: DC continuous [10]
5. Number of steering motors: one
6. Type of steering motors: servo motor
7. Battery system: rechargeable option (maximum twelve 1.2-Volt and three 9-Volt)
8. Number of runs: average of four-five runs in five mins
9. Types of gears: rack and pinion gears
10. Types of bearings: bushings, ball bearings, roller bearings
11. Materials used for chassis: Aluminum plate and sheet
12. Controller: NI myRIO microcontroller.
13. Type of sensors: five IR sensors (transmitters, receivers)
14. Shooting mechanism:
    - Launch distance expected: 6-15 feet
    - Launch angle: maximum 45°
    - Mechanism reload time: 20 - 50 seconds

2.3. Preliminary Design Concepts

Four preliminary design concepts were created to get an idea of what system could potentially work in order to achieve the task at hand. The four different design concepts are as follows: the pitching wheel robot, the looped track robot, the linear actuator robot, and the rack and pinion robot. Each of the design concepts are depicted and in the following figures and associated descriptions.

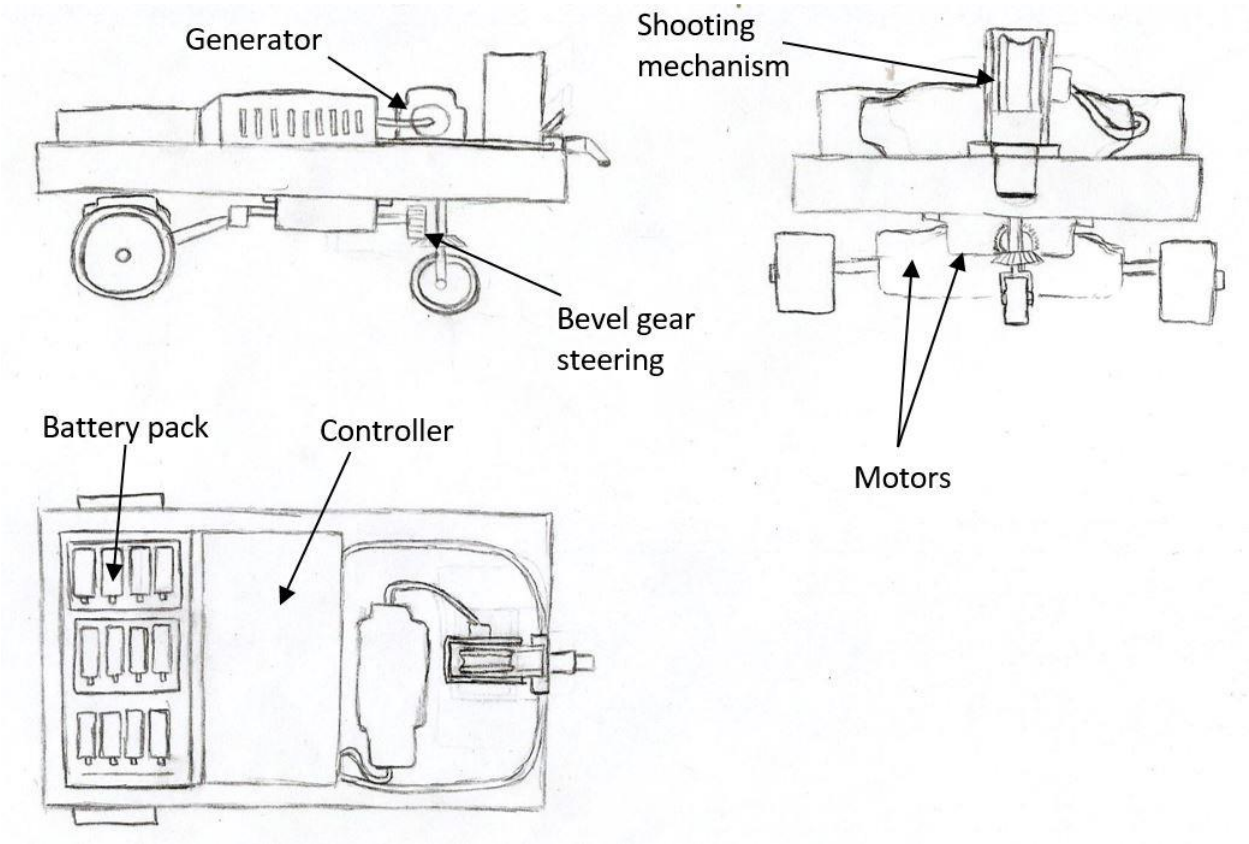


Figure 5: Pitching wheel robot- this design utilizes a motor and wheel shooting mechanism similar to a baseball shooter**.**

For the Pitching wheel robot, a bevel gear system for steering and a motor-wheel shooting mechanism placed at the center-front will be utilized, similar to a baseball pitching machine. Propulsion will be from the rear wheels and the battery pack and controller will be placed on the back of the robot.

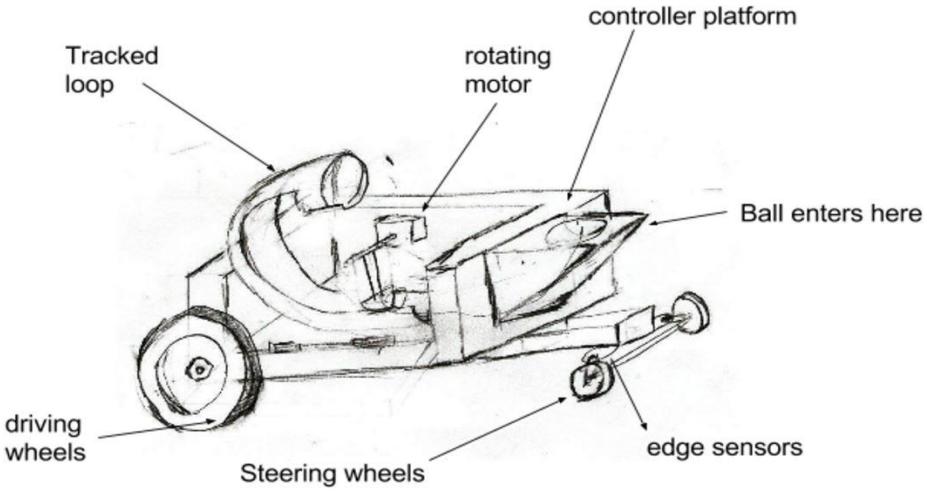


Figure 6: Looped track robot: utilizing a motor and a spoon arm to rotate and launch ball through

a track.

The looped track robot shown in Figure 5 uses a spoon-like shooter that rotates 360 degrees on a ramp made of industrial plastic tube. The ball can be placed in the hole on the right of the robot and slides down the track. When the sensor senses the edge of the open ramp, the spoon rotates clockwise to build up momentum and then hit and drive the ball around the ramp and then release it. The steering mechanism is placed in the back and uses a stepper motor to rotate a shaft connected to the handling wheels.

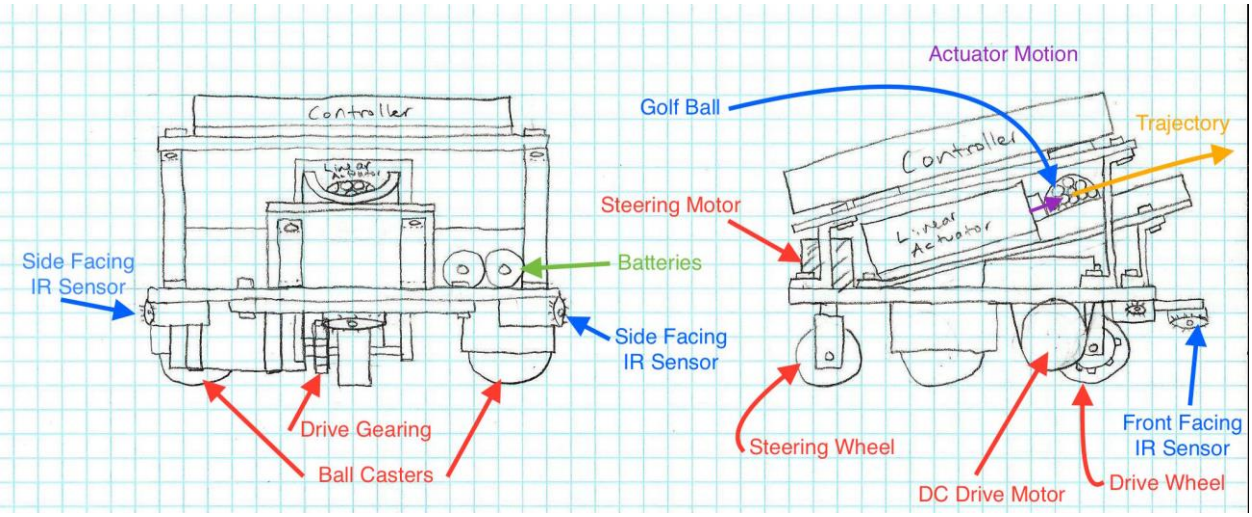


Figure 7: Design concept utilizing linear actuator, single geared FWD, and single rear active steering.

The design shown in Figure 6 uses a linear actuator for the projectile system. Steering is controlled with a single wheel powered by a stepper motor coaxial to the wheel's pivot axis. The steering wheel is centered at the rear end of the robot. The robot propels itself using a single wheel geared to a motor for simple, high-efficiency power transfer [9]. The structure is balanced with ball bearing wheels.

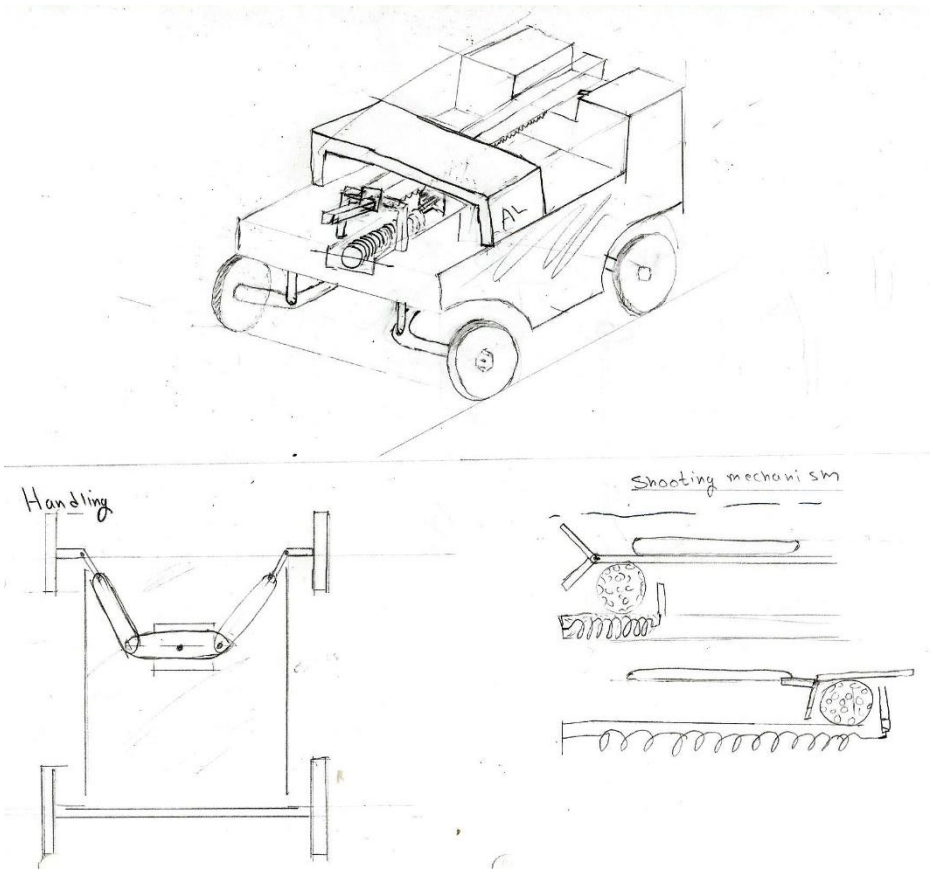


Figure 8: Design Concept using rack and pinion for the shooting mechanism, Ackerman steering, and RWD.

The rack and pinion design (see Figure 8) uses a rack and pinion system to pull back an extension spring, which accelerates the golf ball when released. A simple cam system controls timing of the launch such that the ball is released when the spring reaches maximum extension and thus holds the most potential energy to impart to the golf ball as kinetic energy. The steering system is a simple Ackerman configuration which is robust and allows the wheels to turn at different angles to compensate for different speeds of the inner and outer wheels. The drive system consists of a single solid rear axle powered by a single motor.

2.4. Selection Criteria and Process

The comparison chart is shown below, followed by the Objective Tree.[11] Design scores for the Objective Tree were scaled from 1-5 for each person in the group and added up for the total score for each category.

			# of Designers / Scale							
			5							
			Compared To							
	Is this Factor Used? (1 = YES, 0 = NO)	Factors	Cost	Weight	Reliability	Durability	Simplicity	Accuracy	Repeatability	Total
Asset	1	Cost	5	5	1	2	0	2	2	17
	1	Weight	0	5	0	0	0	0	1	6
	1	Reliability	4	5	5	5	4	4	5	32
	1	Durability	3	5	0	5	4	3	3	23
	1	Simplicity	5	5	1	1	5	1	5	23
	1	Accuracy	3	5	1	2	4	5	4	24
	1	Repeatability	3	4	0	2	0	1	5	15
									Total Sum:	140

Table 1: Pairwise Comparison Chart for three different design concepts and the associated deciding factors.

		Design Scores				Weighted Design Scores			
Factors	Weight	Pitching Wheel Robot	Looped Track Robot	Linear Actuator Robot	Rack and Pinion	Pitching Wheel Robot	Looped Track Robot	Linear Actuator Robot	Rack and Pinion
Cost	0.12	17	16	12	17	2.06	1.94	1.46	2.06
Weight	0.04	17	18	10	18	0.73	0.77	0.43	0.77
Reliability	0.23	16	12	19	19	3.66	2.74	4.34	4.34
Durability	0.16	15	20	19	20	2.46	3.29	3.12	3.29
Simplicity	0.16	15	17	16	17	2.46	2.79	2.63	2.79
Accuracy	0.17	18	13	18	18	3.09	2.23	3.09	3.09
Repeatability	0.11	17	14	11	17	1.82	1.50	1.18	1.82
	1			Total Score:		16.29	15.26	16.24	18.16

Table 2: Objective Tree Analysis with associated design scores

2.5. Selected Design

Using the Pairwise Comparison Chart, it was determined that the design configuration utilizing a rack and pinion was best suited for potentially meeting our weighted design criteria. Therefore, this is the design being adopted for the remainder of the project. This design won because it is anticipated to have above average reliability, accuracy, durability, cost, weight, and simplicity. This is especially significant because our PCC revealed reliability and accuracy as the two highest priority factors out of a list of seven.

The rack and pinion design received a high cost score (low expected cost) due to the simplicity and low part count of the projectile system. The motor required to power this system should not be particularly expensive or excessively power-depleting. Similarly, this design received a high weight score due to its simple, flat chassis, average weight projectile system, and lightweight steering mechanism. Reliability received an above average rating due to the potential for predictable trajectories from the rack and pinion shooting system. Expected durability was rated high relative to the other design options due to the steering system. While the rack and pinion system used for the shooting mechanism may require some finesse to perfect, it is still relatively simple and thus the robot received a slightly above average simplicity score. Accuracy was given a high score due to the nature of the rack and pinion design. Speed, while not exceptional, was ranked above the competing designs due to the relatively light weight of the robot's structure and steering setup and potential to launch quickly by beginning to load the golf ball for release as the robot travels up the ramp.

The looped track robot primarily lost out due to weaknesses in anticipated reliability and accuracy. This largely stems from the difficulty in designing and fabricating an efficient, high-quality loop track as well as the potential fragility of the steering system. The linear actuator transporter design, while reliable, fell short due to the heavy weight and high cost of an adequate linear actuator. After further research upon our original choice of a robot configuration, we found it best to not go with the pitching wheel configuration due to the variable accuracy that this machine could produce due to potential lag time of the motor to turn the wheel.

2.6. CAD Models

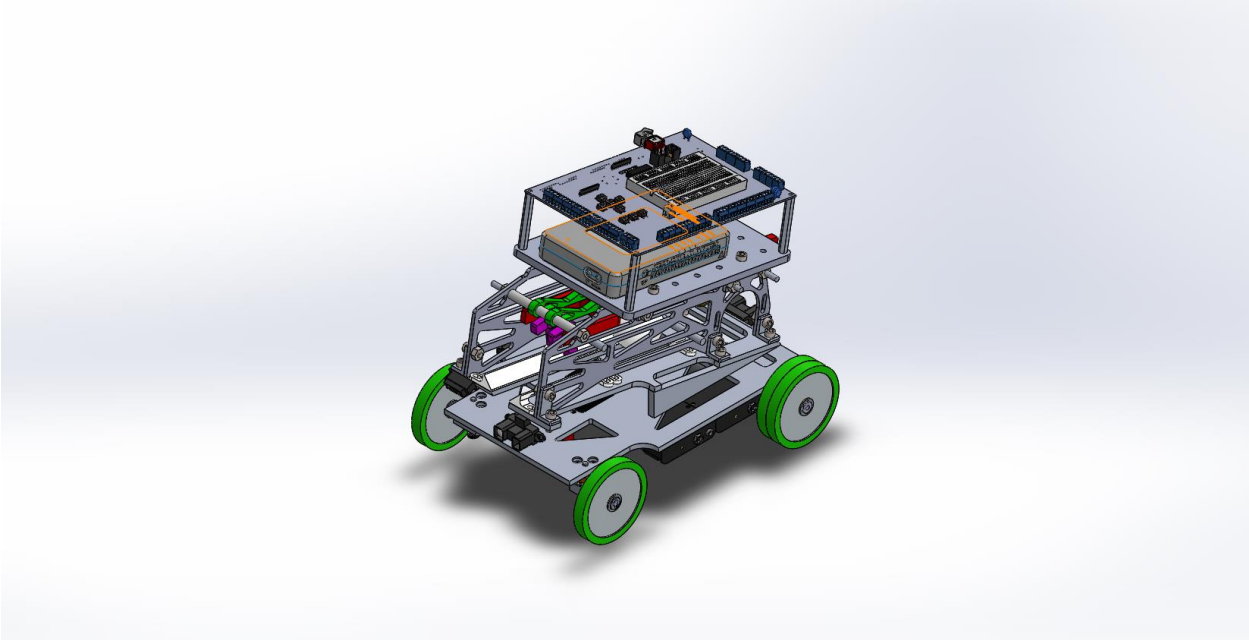


Figure 9: Isometric View of Rack Attack.

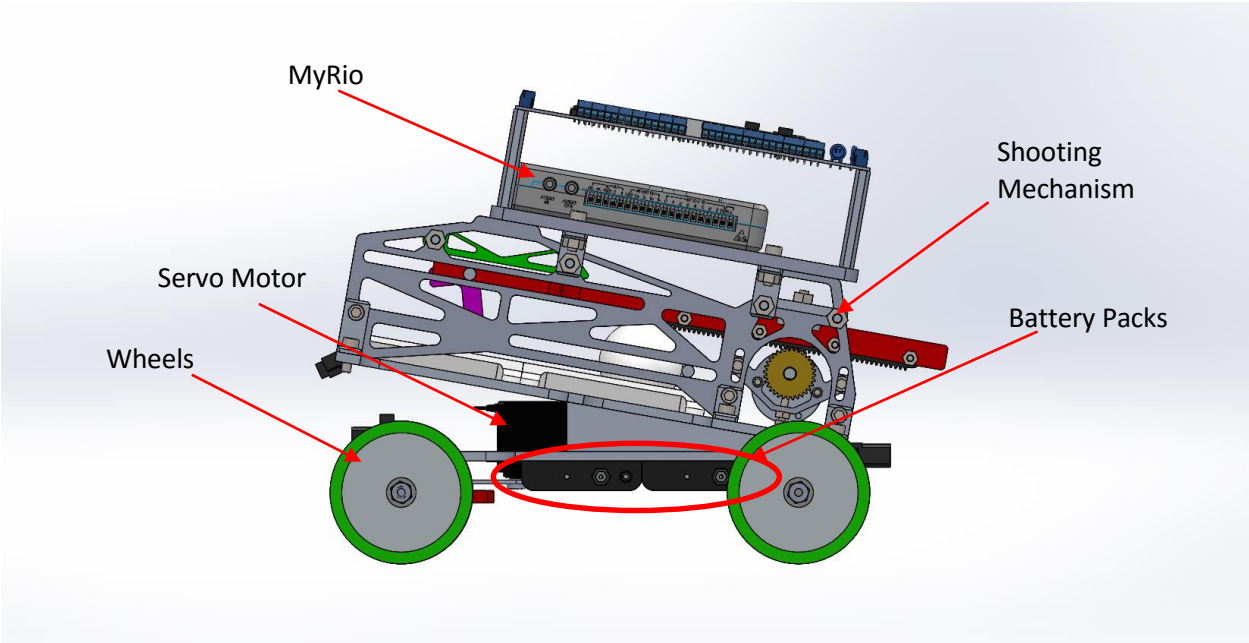


Figure 10: Right view of Rack Attack with labeled components.

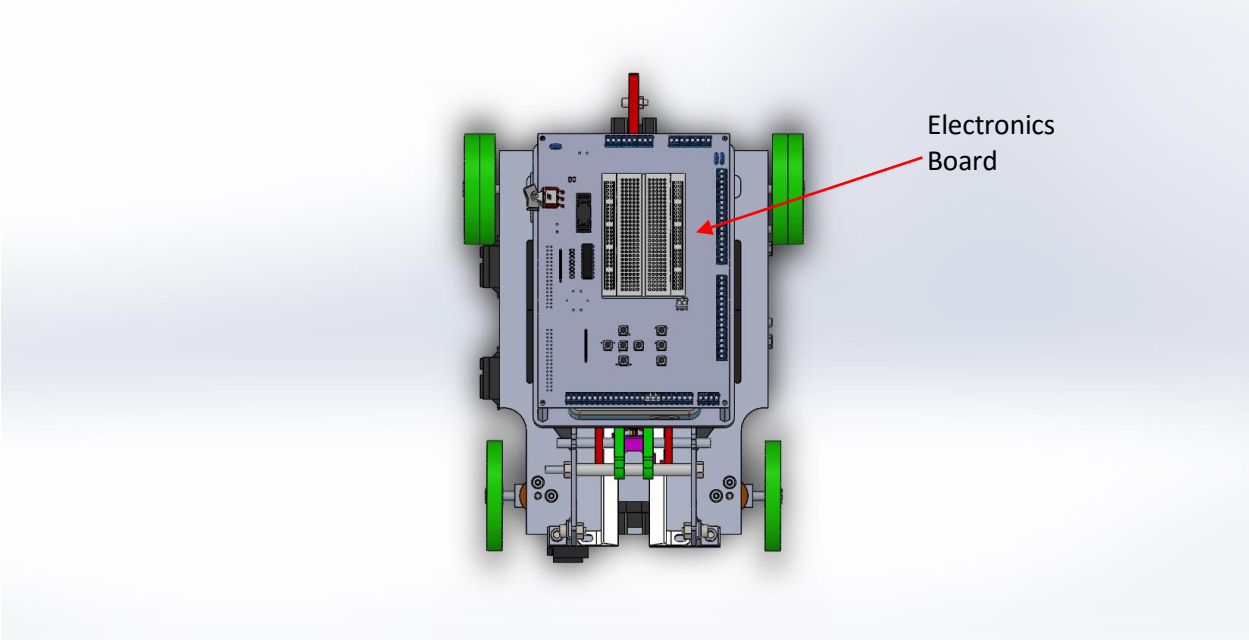


Figure 11: Top view of Rack Attack with labeled component.

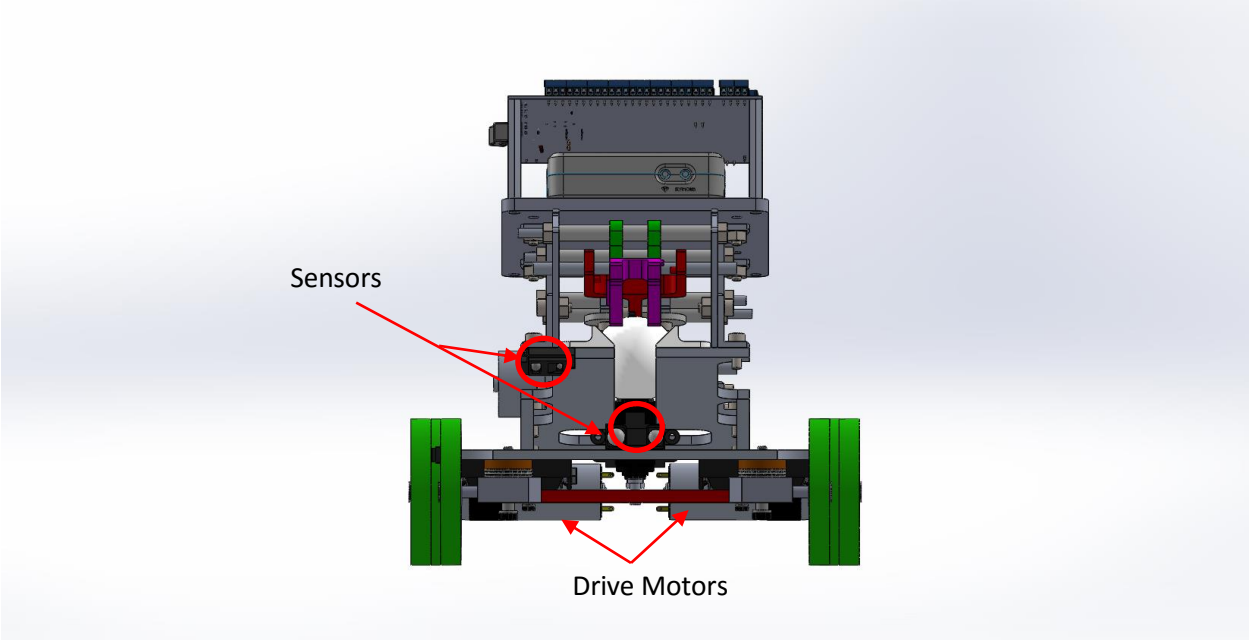


Figure 12: Front View of Rack Attack with labeled components.

The propulsion system is connected to the bottom face of the chassis. It will utilize one power motor that is connected to the rear axle and transmits power using two bevel gears. For the steering mechanism, the robot will use an Ackerman active steering mechanism. The front wheels of the robot will be for steering purpose of the robot. The robot will use two wheels that

rotate independently and can turn using a servo. The turning mechanism has been optimized to turn about the center of curvature of the ramp. Regarding the shooting mechanism of the golf ball, a spring will be used to drive the ball and shoot it. The ball will be pulled back using a rack and pinion and released by a mechanical trigger. The mechanical trigger that can be reversed to grab another ball from the loading area and reload to shoot again. To facilitate debugging the robot we decided to place the shooting mechanism right on top of the bottom plate and under the myRIO controller. Most of our batteries will ride under the plate of the launching mechanism.

3. Preliminary Calculations

The following subsections will be used to outline the calculations needed to be taken into consideration in order to find the necessary torque, power, move profile, and coefficients needed to drive our robot roundtrip. The main subsections are the drive system and the ball delivery system. **Note: different calculations in this report change unit systems due to keeping consistency with design parameters for the SolidWorks model and the standard parts for the robot.**

3.1. Drive System

In this section we show our calculations which led to determination of the type of tires we use on our robot.

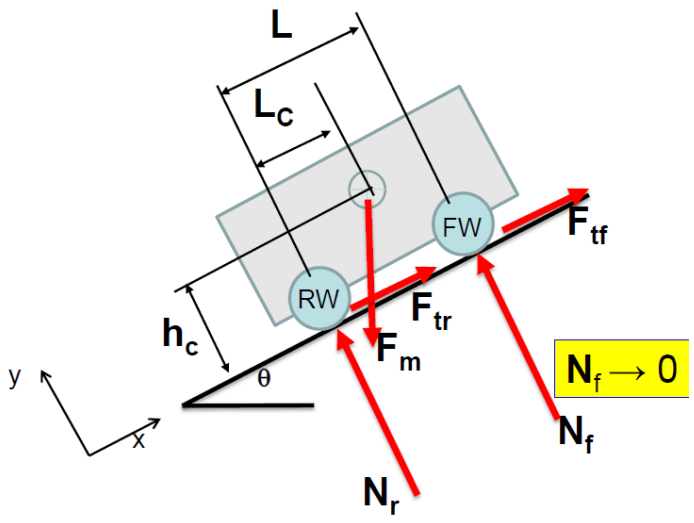


Figure 13: Free body diagram of robot going up an incline with associated tractive forces.

The normal forces for the front and rear wheels;

$$N_f = \frac{F_m \cdot L_C \cdot \cos\theta - F_m \cdot h \cdot \sin\theta}{L} \quad (1)$$

(2.1)

$$N_r = \frac{F_m \cdot (L - L_C) \cdot \cos\theta + F_m \cdot h \cdot \sin\theta}{L} \tag{2}$$

The tractive forces for the front and rear wheels;

$$F_{tf} = \mu N_f \tag{3}$$

$$F_{tr} = \mu N_r \tag{4}$$

A force conservation in the x-direction as defined in the figure above;

$$\sum F_x = 0 : F_{tf} + F_{tr} - F_m \cdot \sin\theta = \mu N_f + \mu N_r - F_m \cdot \sin\theta = 0 \tag{5}$$

Since we are not using an all-wheel drive system, but a rear-wheel drive up the ramp and front-wheel drive down the ramp, we must account for the absence of the respective tractive forces both up and down the ramp using the parameters γ_r and γ_f.

$$\sum F_x = 0 : \gamma_f \mu N_f + \gamma_r \mu N_r - F_m \cdot \sin\theta = 0 \tag{6}$$

$$\mu = \frac{L \cdot \sin\theta}{(\gamma_r - \gamma_f)(h \cdot \sin\theta - L_C \cdot \cos\theta) + \gamma_r L \cdot \cos\theta} \tag{7}$$

Going up the ramp with a rear-wheel drive;

$$\mu_{RWD} \geq \frac{1}{\frac{L - L_C}{L \cdot \tan\theta} + \frac{h}{L}} \tag{8}$$

Going down the ramp with a front-wheel drive;

$$\mu_{FWD} \geq \frac{1}{\frac{L_C}{L \cdot \tan\theta} - \frac{h}{L}} \tag{9}$$

The tip over angle for our device is determined by assuming the normal force on the front wheels are zero. Going up the ramp with a rear-wheel drive;

$$\theta_{max} = \tan^{-1}\frac{L_C}{h_{max}} \tag{10}$$

Going down the ramp with a front-wheel drive (note that L_C values are different going up and down the ramp);

$$\theta_{max} = \tan^{-1}\frac{L_C}{h_{max}} \tag{11}$$

3.1.1. Device Parameters

The following table summarizes the major components of Rack Attack and their associated materials and weights. The total weight of the robot is indicated in red.

Component	Quantity	Material	Weight (kg)	Total Weight (kg)
Chassis Plate	1	Aluminum 6061	0.53	0.53
Drive Motor	2	Aluminum 3003 Alloy	0.117	0.234
Servo Motor	1	Plastic	0.03	0.03
Wheels	4	Thermoplastic Rubber	0.022	0.091
Battery	15	Titanium Ti-8Al	0.032	0.481
Shooting Mechanism	1	Aluminum 6061	0.635	0.635
Battery Holder	4	PTFE	0.021	0.084
MyRio 1900	1	Plastic	0.193	0.193
AJB and Backing	1	Derlin 2700 NC010	0.322	0.322
Steering Linkage	1	Aluminum 6061	0.022	0.022
Drive Motor Mount	2	Aluminum 6061	0.012	0.024
Pinion Motor Mount	1	Aluminum 6061	0.004	0.004
Sensors	5	Plastic	0.007	0.035
				2.685

Table 3: Device components, quantities and associated weights.

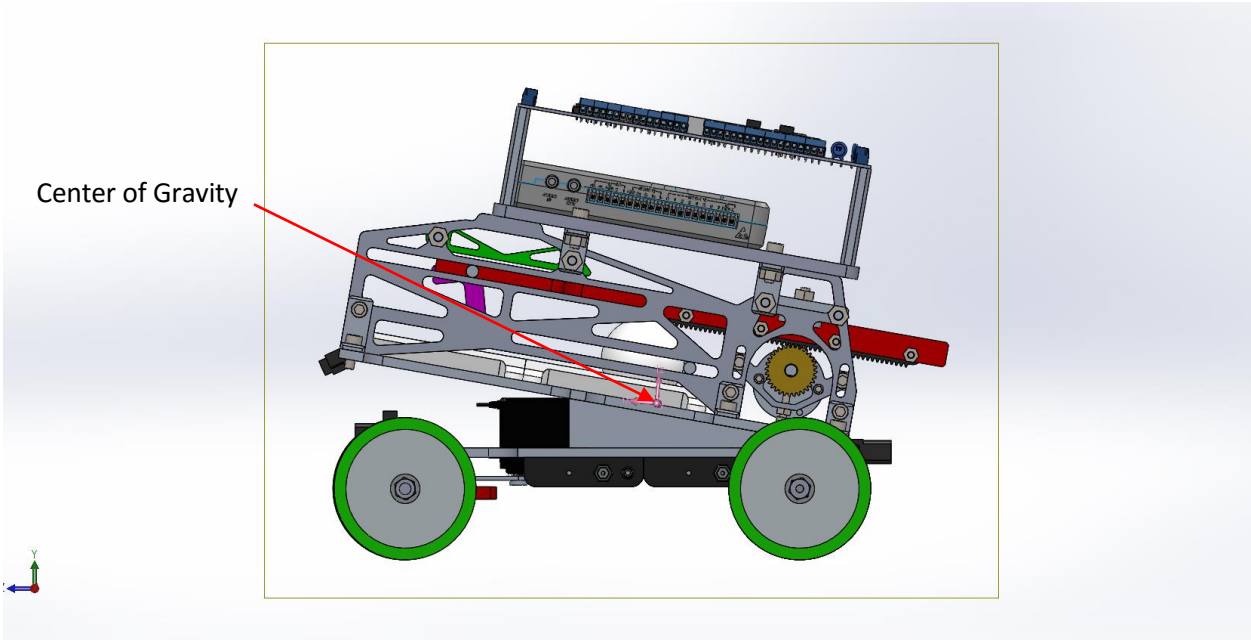


Figure 14: Center of gravity position as found by SolidWorks based on masses and positioning

of components for Rack Attack.

Dimension	Value	Units
Length of robot	0.276	m
Width of Robot	0.247	m
Height of Robot	0.235	m
Wheelbase	0.276	m
Wheel Diameter	0.073	m
Height of Center of mass	0.081	m
Distance of Rear Wheel to Center of Mass	0.096	m
Total Weight	2.685	kg

Table 4: Major dimensions of Rack Attack.

The above table lists the major dimensions of Rack Attack, along with its total weight. The ramp inclines and segments dimensions are provided in the move profile section of the report.

3.1.2. Friction Coefficients

From the free body diagram and equations given above, the required minimum friction coefficients can be calculated for front-wheel drive (FWD), rear-wheel drive (RWD), and four-wheel drive (4WD). The friction coefficient between the wheels and wood were determined experimentally. For our chosen schematic of the robot, the distance of center of gravity from the rear wheels, $L_{c,}$ height of the center of gravity, $h_{c,}$ and distance between the rear and front wheels, L, were predetermined for the coefficient of friction calculations.

Variable	Value	Units
$\mu_{wheel/wood}$	0.45	-
μ_{4WD}	0.27	-
μ_{FWD}	0.73	-
μ_{RWD}	0.42	-
L_c	0.096	m
L	0.203	m

h_c	0.081	m

Table 5: Coefficients of friction and required dimensions for drive system calculations.

For compatibility of both FWD and RWD, there is an optimal minimum coefficient of friction that must be met. This can be found by plotting the coefficient of frictions as a function of the distance of center of gravity from the rear wheels, which is shown in the following figure.

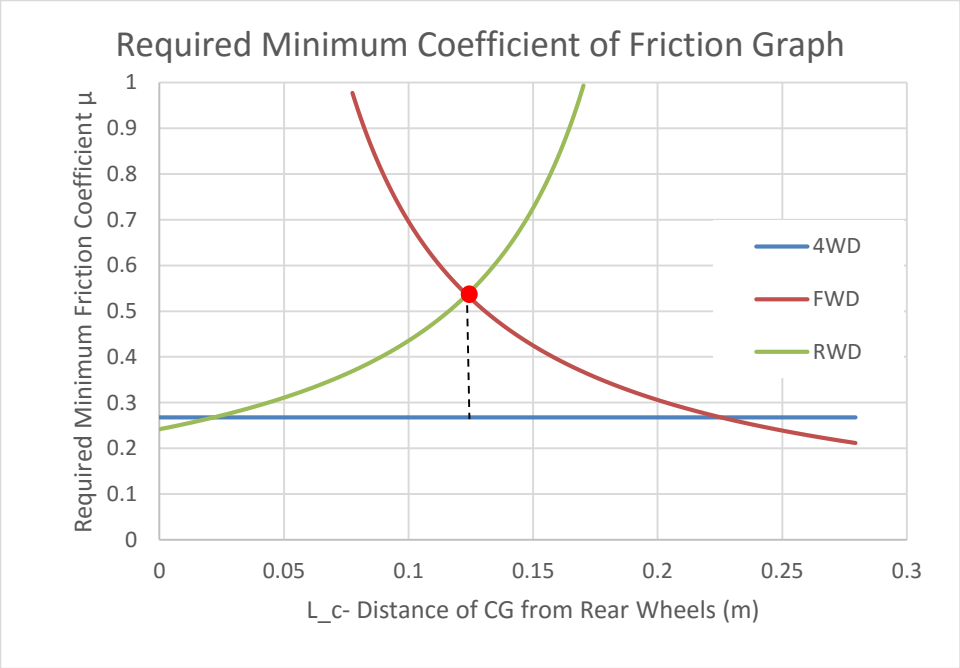


Figure 15: The above graph shows the minimum coefficient of friction needed for FWD and RWD interchangeable compatibility.

From above, it is indicated where L_c must be in order to have FWD and RWD compatibility. The minimum coefficient of friction associated to both RWD and FWD is 0.535 at an L_c of 0.123 m.

3.1.3. Tractive Forces

Again, the normal, tractive, and gravitational forces on the robot along the ramp can be calculated in the equations discussed previously. The following table summarizes these forces as well as indicates the maximum angle the robot can travel up.

Variable	Value	Units
N_r	25.22	N
N_f	14.67	N
F_{tr}	10.69	N
F_{tf}	6.22	N
F_{tot}	17.91	N

F_g	41.3	N
Θ_{max}	49.9	deg

Table 6: Normal, tractive, and gravitational forces along with the maximum incline the robot can travel.

3.1.4. Move Profile

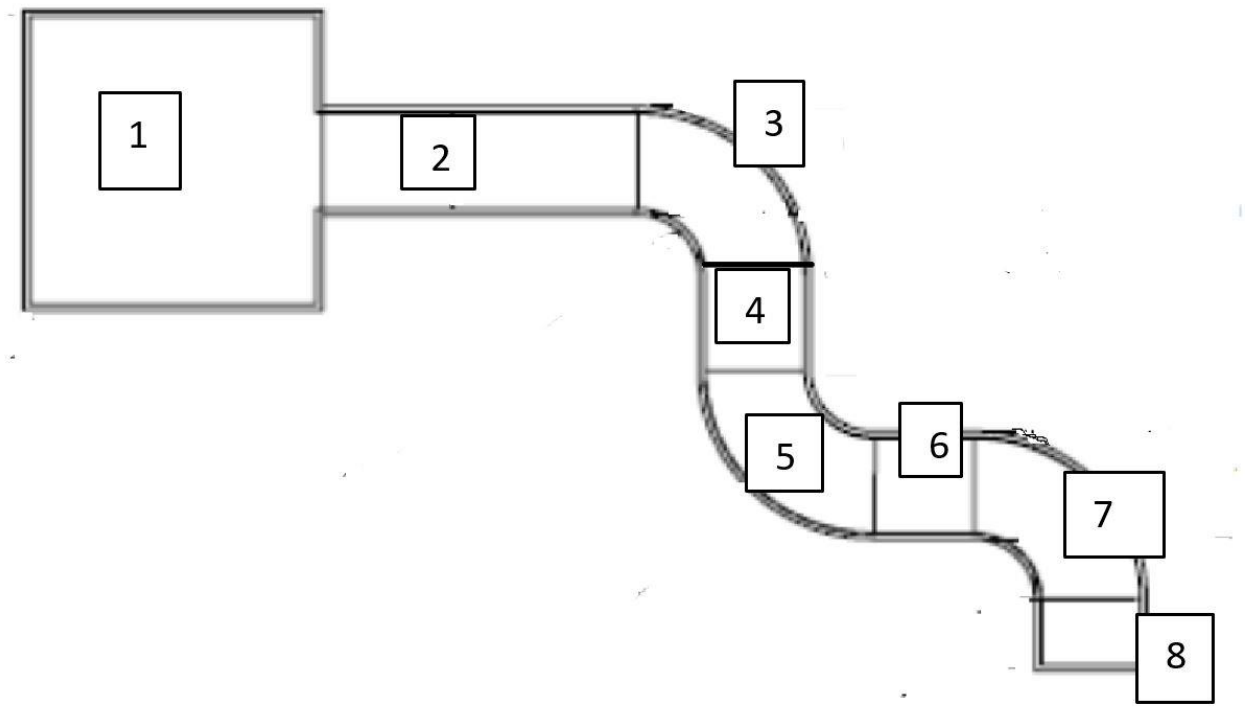


Figure 16: The map of the ramp used in profile picture.

The image above shows the regions that are studied in the move profile. The first region analyzed is region 1 and will end there after a round trip.

Region	Length (m)	Angle ˚
1	-	0.00
2	1.316	8.42
3	0.758	0.00
4	0.406	0.00
5	0.758	0.00
6	0.418	14.03
7	0.758	0.00

8	0.305	0.00

Table 7: Values of the length and angle for each ramp segment.

The table above shows the values of the length and angle for each segment in meters. Region 1 is left blank since it's a wide area and the robot will not go through the same path in it every time its navigating the ramp.

This part of the report is mostly an estimate of the velocities needed based on total time and distance traveled. Different sections of the ramp were studied differently depending on their conditions.

We will study two different move profile using different approaches, the first using a constant velocity throughout the ramp and with a varying torque. The second model uses a more realistic velocity profile that changes with each ramp segment. We will present calculations and reasoning throughout this section to help us decide which move profile we will implement.

As a primary approach to this problem we started by approximating the average time needed to travel the ramp and the average velocity throughout the trip.

Considering the time frame that we are limited to (3 minutes to deliver the ball and come back and the 5 minutes competition rule), we would like to have a total time for the trip to be approximately 1 minute 30 seconds.

Since the total distance of the ramp (ignoring the distance traveled at region 1) is 4.72 m. our average velocity on the ramp would be.

$$V_{ave} = \frac{Distance}{time} = 0.105 \frac{m}{s} \quad (12)$$

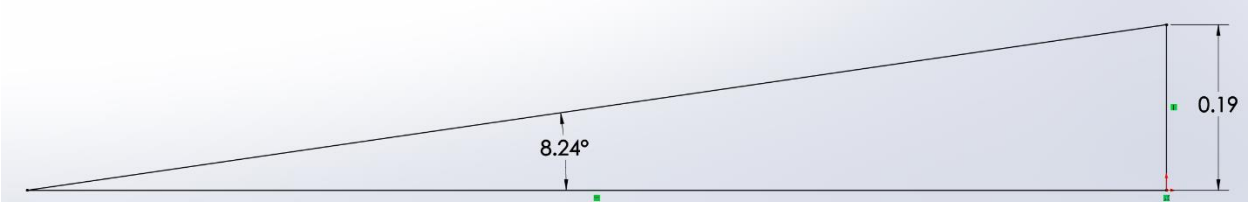
Then each segment is analyzed separately to try and preserve this average velocity or exceed it if possible. We will estimate the time for segment using the total time and the corresponding length.

Region 1:

An estimate to how much time the robot will spend in that area is about 10s before it figures the way to the ramp. A good velocity to accelerate to after the robot is turned on would be around 0.15 m/s. considering that when it goes on the ramp its speed will decrease because the gravitational force is added to the retarding forces.

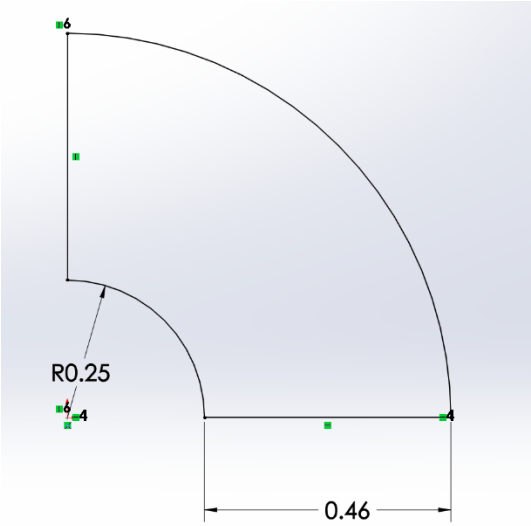
$$V_1 = 0.15\ m/s$$

Region 2:



Since region 2 is long (1.32m) we will assume a larger velocity throughout the section. Since our average velocity should be around 0.105 m/s, a good estimate would be 0.14m/s. We will also consider to slow down the vehicle at the end of the ramp to make sure we can reach a safe velocity for turning. This estimate is realistic compared to the larger region 1 velocity.

Region 3 and similarly regions 5 and 7:



In order to estimate how much velocity the robot needs to cross this circular section safely and efficiently we will first look at the maximum allowable velocity before radial slipping occurs.

Assuming static friction acts as a centripetal force to hold the robot.

$$F_{friction} = \frac{m\,V^2}{R} \tag{13}$$

$$V_{max} = \sqrt{\frac{N\,\mu_s R_{ramp}}{m_{robot}}} \tag{14}$$

$$V_{max} = \sqrt{9.81 * 0.4826 * 0.4} = 1.38\ m/s$$

Exceeding that speed is unrealistic hence we should not worry about slipping in the radial direction, however slipping is very possible in the steering wheels, hence using the same equation above for the front wheels alone.

$$V_{max} = \sqrt{\frac{N_{front}\,\mu_s R_{ramp}}{m_{robot}}} \tag{15}$$

$$V_{max} = \sqrt{\frac{5.833 * 0.4 * 0.4826}{4.08}} = 0.365\ m/s \tag{16}$$

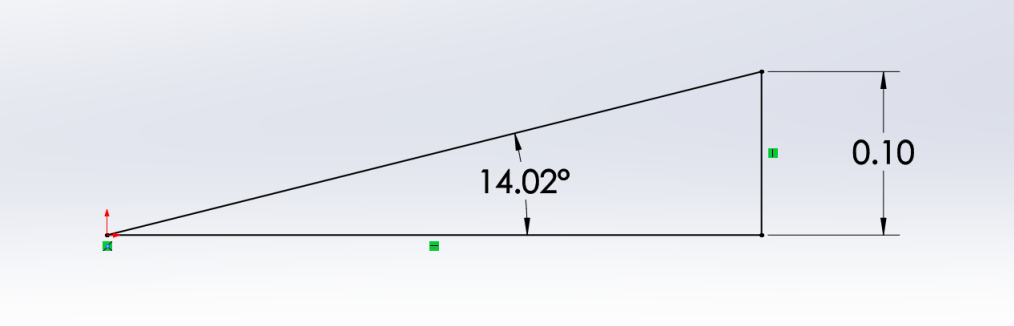
Complete slipping will occur in the front wheels at 0.365 m/s hence we need to be significantly lower and around the range of 0.105 m/s. A safe estimate would be 0.11m/s.

Regions 4 and 8:

Since these regions are short, straight and are proceeded by turns or stops it is best to estimate their velocities based on the velocities of the previous sections (curves). For both regions 4 and 8 the velocity can be approximated to be 0.11m/s for ease of calculations.

For section 8 the only exception is that it should be decelerating to zero to stop at the edge of the ramp to shoot the gold ball. Hence at the end of region 8 velocity will be 0.

Region 6:



Similar to region 2 the velocity can be chosen to be equal. Hence the velocity at the ramp section 6 should be around 0.14.

Important Note: *the velocity throughout each region is not constant and may vary to adjust to the upcoming section. For instance, at the end of region 2 the robot will start slowing down from (0.14m/s) to start the turn with a velocity of (0.11m/s). The time for the robot to accelerate or decelerate is estimated at 1-2 seconds at each interval.*

Region	cumulative time (s)	Velocity (m/s)	Acceleration (m/s^2)

1	0	0	0.05
1	3	0.15	0
1	10	0.15	-0.005
2	12	0.14	0
2	19	0.14	-0.03
2	20	0.11	0
3	26	0.11	0
4	28	0.11	0
5	34	0.11	0.03
5	35	0.14	0
6	37	0.14	-0.005
7	43	0.11	-0.022
8	46	0	0
8	48	0	-0.0367
7	51	-0.11	-0.005
6	57	-0.14	0
5	59	-0.14	0.03
5	60	-0.11	0
4	66	-0.11	0
3	68	-0.11	0
2	74	-0.11	-0.03
2	75	-0.14	0
2	82	-0.14	-0.005
1	84	-0.15	0
1	91	-0.15	0.05
1	94	0	0

Table 8: The estimated cumulative time, velocity and acceleration for one round trip.

The previous table displays an estimate of the cumulative time of the trip. It also shows at which region the robot will be at any time. The estimated velocities correspond to the estimates that were found in the previous section. Acceleration was calculated using the difference between two consecutive points.

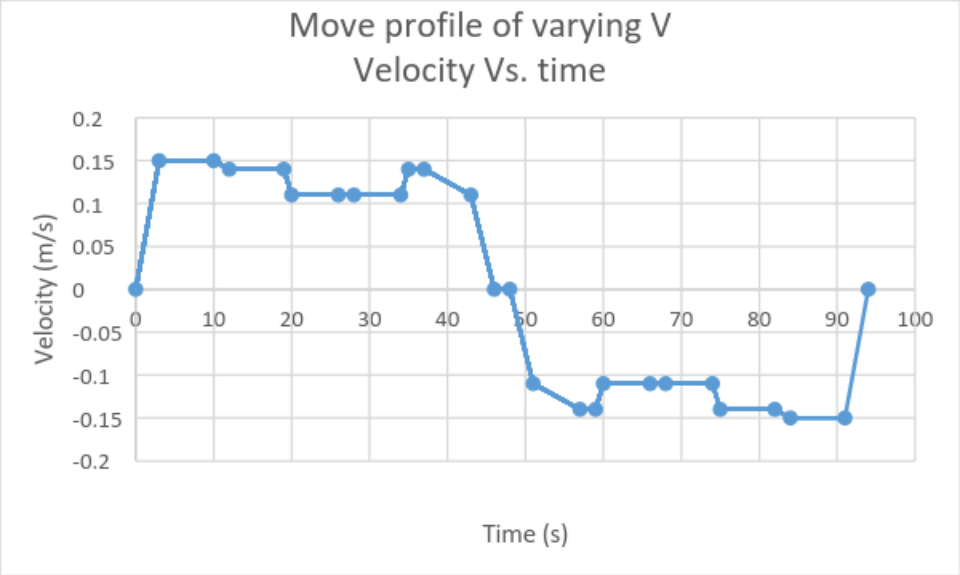


Figure 17: Move profile corresponding to the values of velocity from the previous table.

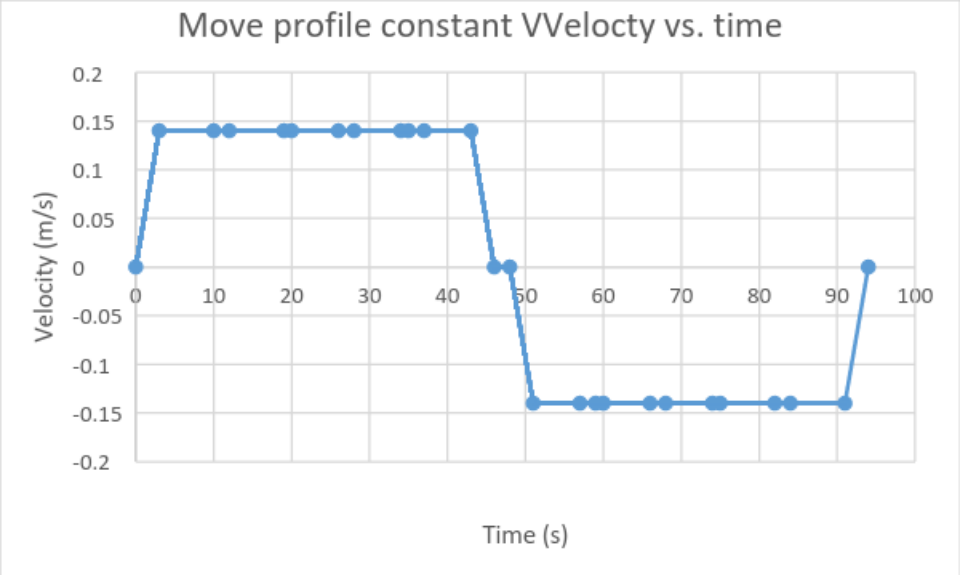


Figure 18: Move profile corresponding to an average velocity on the ramp. ***Note: This is for reference only.***

The weight distribution of the device is presented below. Notice that the front wheel weight distribution of the upward movement is just the rear wheel weight distribution of the downward movement. We have opted to "reverse" our device when we move down the ramp.

$$\beta_f = \frac{L_c * cos(\theta_i) - h * sin(\theta_i)}{L}$$

$$\beta_r = \frac{(L - L_c) * cos(\theta_i) + h * sin(\theta_i)}{L}$$

Upward		
Region	β_f	β_r
1	0.475	0.525
2	0.411	0.578
3	0.475	0.525
4	0.475	0.525
5	0.475	0.525
6	0.364	0.606
7	0.475	0.525
8	0.475	0.525

Table 9: Beta constants for upward travel.

Downward		
Region	β_f	β_f
8	0.525	0.475
7	0.525	0.475
6	0.606	0.364
5	0.525	0.475
4	0.525	0.475
3	0.525	0.475
2	0.578	0.411
1	0.525	0.475

Table 10: Beta constants for downward travel.

Furthermore, the coefficient of friction of the vehicle was assumed to be $\mu_{static} = 0.45$ and the rolling coefficient of friction was assumed to be $\mu_{roll} = 0.001$. The efficiency η=0.5.

3.1.5. Propulsion Force

Assumptions:

1. Rolling friction ($\mu_{roll} = 0.001$)

The value of the rolling friction between the robot's wheel and the ply wood track was predicted using the value of the rolling friction between a bicycles wheel and a wooden floor which approximately has the same roughness.

2. Total system efficiency was assumed to be 50% as the case with most electrical devices including motors, actuator and servos.
3. The overall drive train friction can be estimated using the theoretical values of the coefficient of friction $\mu_s = 0.45$.

In this section the values of the forces acting on the robot will be approximated using the following equations. The inertial force can be found from the mass of the robot and the desired acceleration from the table in the previous section:

$$F_{inertia} = ma \tag{17}$$

The gravitational force that acts in the robot can be found from the following equation using the angle of the ramp if present:

$$F_{grav} = mg\ sin(\theta) \tag{18}$$

The rolling reactive force can be calculated using the rolling coefficient of friction and the normal force.

$$F_{roll} = N\ \mu_{roll} \tag{19}$$

Note: the rolling force required when motion starts is approximately 2.5*the rolling force when the robot is moving.

The friction force acting on the robot while it moves can be approximated using the normal force and the coefficient of kinematic friction:

$$F_{roll} = N\ \mu_k \tag{20}$$

Now the total force required to push the robot throughout any section of the ramp would be:-

$$F_{prop} = F_{roll} + F_{roll} + F_{grav} + F_{inertia} \tag{21}$$

Time (s)	Grav force (N)	Rot force (N)	Normal force (N)	Friction force (N)	Prop force (N)
0	0	0.100119	40.04766	18.02145	18.32568
3	0	0.040048	40.04766	18.02145	18.06149
10	0	0.040048	40.04766	18.02145	18.04108
12	-5.8648	0.040048	39.61589	17.82715	23.732
19	-5.8648	0.040048	39.61589	17.82715	23.60953
20	-5.8648	0.040048	39.61589	17.82715	23.732
26	0	0.040048	40.04766	18.02145	18.06149
28	0	0.040048	40.04766	18.02145	18.06149
34	0	0.040048	40.04766	18.02145	18.18396
35	0	0.040048	40.04766	18.02145	18.06149
37	-9.70875	0.040048	38.85299	17.48385	27.21223
43	0	0.040048	40.04766	18.02145	17.97168
46	0	0.040048	40.04766	18.02145	18.06149
48	0	0.040048	40.04766	18.02145	17.91181
51	0	0.040048	40.04766	18.02145	18.04108
57	9.70875	0.040048	38.85299	17.48385	7.815144

59	0	0.040048	40.04766	18.02145	18.18396
60	0	0.040048	40.04766	18.02145	18.06149
66	0	0.040048	40.04766	18.02145	18.06149
68	0	0.040048	40.04766	18.02145	18.06149
74	5.864803	0.040048	39.61589	17.82715	11.87993
75	5.864803	0.040048	39.61589	17.82715	12.0024
82	5.864803	0.040048	39.61589	17.82715	11.98198
84	0	0.040048	40.04766	18.02145	18.06149
91	0	0.040048	40.04766	18.02145	18.26561
94	0	0.040048	40.04766	18.02145	18.06149

Table 11: Value of the forces and the total propulsion force.

Using the following equation, we can find the value of the power required instantaneously

$$P = F_{prop} V_{instn} \tag{22}$$

Cumulative time (s)	Total power (W)
0	0
3	5.418448
10	5.412325
12	6.644961
19	6.610669
20	5.221041
26	3.973529
28	3.973529
34	4.000472
35	5.057218
37	7.619425
43	3.95377
46	0
48	0
51	3.969038
57	2.18824
59	5.091509
60	3.973529
66	3.973529
68	3.973529
74	2.613584
75	3.360671
82	3.354956
84	5.418448

91	5.479683
94	0

Table 12: The value of the power required at each instance of the entire trip

.

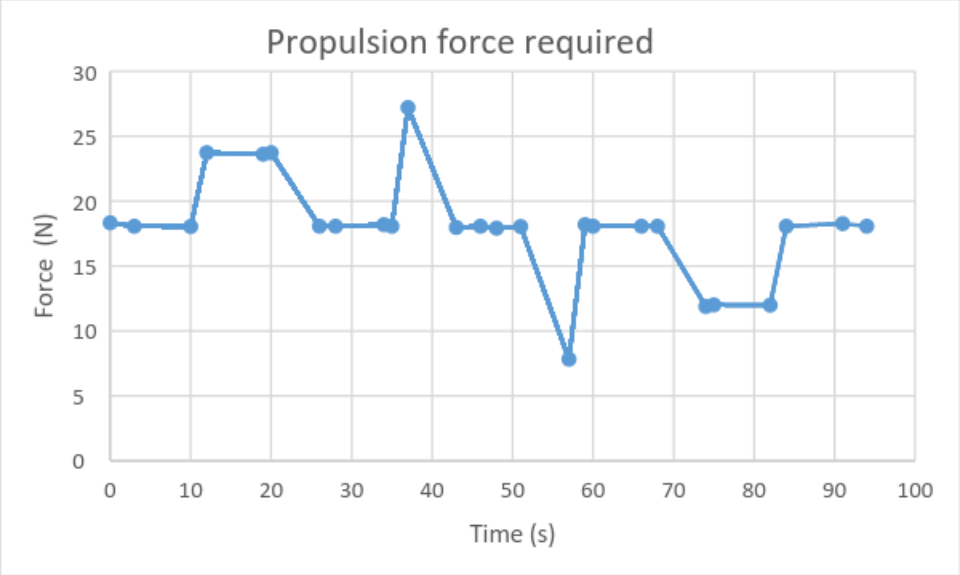


Figure 19: The value of power required against time

3.1.6. Propulsion Torque

Propulsion torque needed to drive our motor was estimated using the moments of inertia of all the connections to our drive motor and the friction force that the motor needs to overcome.

In our design we implemented a geared drive motor and transmits power to the wheels using and axle and bevel gears. The next figure illustrates the concept of our design.

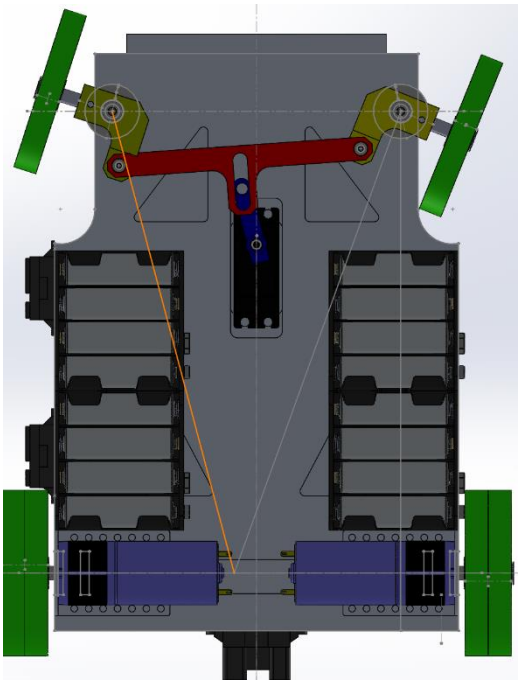


Figure 20 : Bottom plate of the robot showing the drive mechanism.

From the following equations we will try to approximate the torque needed from the motor in order to propel our robot: -

$$T_{req} = J_{tot}\alpha_{acc} + T_{fric} \tag{23}$$

The angular acceleration can be found from the linear acceleration of the wheel divided by the radius

$$J_{tot} = J_{load} + J_{mechanism} \tag{24}$$

For a geared motor as the one we are using

$$J_{load} = J_{motor\ shaft} + J_{refl} \tag{25}$$

$$J_{refl} = \frac{J_{load}}{N^2 e} \quad \text{hence from our design} \tag{26}$$

$$J_{tot} = J_{motor\ shaft} + J_{refl} + 2J_{bevelgear} + J_{axleshaft} + 2J_{wheel} \tag{27}$$

The previous equation takes into consideration the values of the moments of inertia of everything connected to the motor shaft.

The moment of inertia for all load will be approximated using a cylinder or hollow cylinder model and assuming all of these parts rotate with respect to the z-axis that goes through their centers. The following table illustrates the parameters for each moment of inertia mentioned:-

Part	inner radius (m)	outer radius (m)	Mass (Kg)	J (Kg.m^2)

Motor shaft	-	0.0055	0.03	0.00004125
Bevel gears	0.0047625	0.01397	0.01	2.1784E-06
Wheel	0.0127	0.02409	0.02409	0.00311
Axle	-	0.35	0.0888	0.1407119
Refl	-	-	-	2.6269E-08

Table 13: Parameters of each part and their moment of inertia.

The reflected moment of inertia was included in the table even though it is not due to rotating a part but rather the reflected load back on the motor. The equation for J_{refl} that was mentioned before was used and the gear ration N was approximated from a motor that is a possible candidate for our robot. *Note: the only gear ratio included in N is the internal gearing of the motor since the bevel gears have 1:1 gear ration.*

Other sources as to where the values of these table were found can be found in the Appendix.

Assuming that the frictional torque acting the wheel is always constant through the ramp (changes in normal force are minimal). We use the normal force acting on the back wheels.

$$F_{fric} = N_r \mu_s \tag{28}$$

$$T_{fric} = R_{wheel} N_r \mu_s = 0.2369N - m \tag{29}$$

Putting everything together we can deduce the torque needed at every segment of the ramp using the equations displayed above.

Cumulative time (s)	**Torque required (N-m)**
0	0.384621
3	0.2369
10	0.251672
12	0.2369
19	0.325533
20	0.2369
26	0.2369
28	0.2369
34	0.325533
35	0.2369
37	0.251672
43	0.301897
46	0.2369
48	0.345229
51	0.251672

57	0.2369
59	0.325533
60	0.2369
66	0.2369
68	0.2369
74	0.325533
75	0.2369
82	0.251672
84	0.2369
91	0.384621
94	0.2369

Table 14: Values of the propulsion torque require along each segment of the ramp.

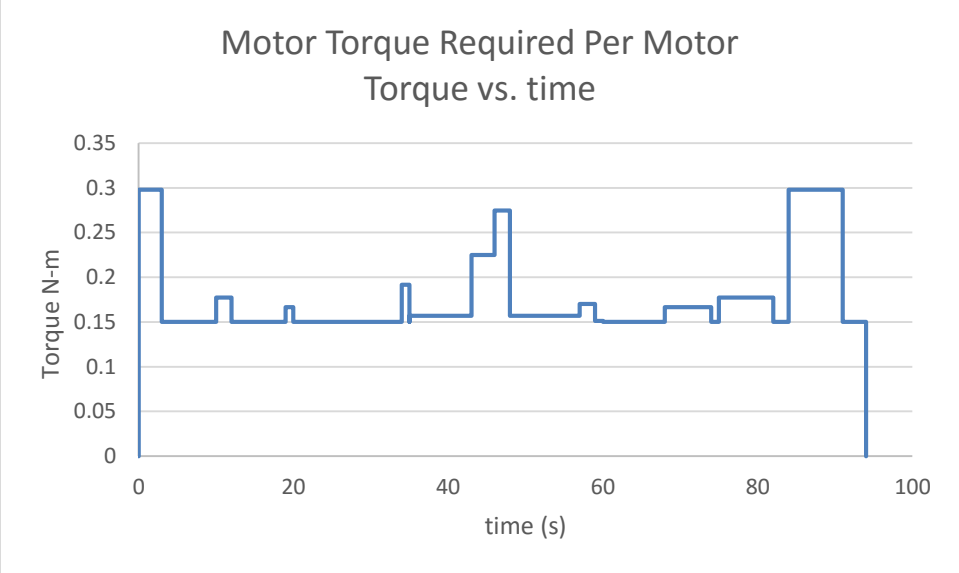


Figure 21: Plot of the torque required vs. the time on the ramp.

As briefed before the robot will be propelled using a rear wheel drive on the forward trip and when it reaches the end of the ramp it will go backwards without rotating. For this we devised a mechanism that can be used as forward and backward. Our drive mechanism consists of one big power motor that transmits power to an axle through two perpendicular bevel gears. The axle is then connected to the wheel.

3.1.7. DaNi Comparison Torque

Assuming the same motor as the DaNI robot has been used on our device, we have used the propulsion force to approximate the torque required to move our device. Furthermore, we have assumed the use of 1 DaNI motors. The motor under consideration is the Tetrix DC Motor W39530.

The propulsion torque was estimated from the product of the propulsion force, determined in the previous section, and the radius of the wheel. The largest torque value for each respective ramp region is the maximum torque. Similarly, the root mean square torque values were obtained for each ramp region.

According to the move profile, the maximum and the RMS torque are presented below.

Upward		
Ramp Segment	**Maximum Torque (Nm)**	**RMS Torque (Nm)**
1	1.338	1.323
2	1.733	1.733
3	1.319	1.319
4	1.319	1.319
5	1.323	1.320
6	1.988	1.988
7	1.311	1.311
8	1.319	1.319

Table 15: Maximum and RMS torque for upward travel.

Downward		
Ramp Segment	**Maximum Torque (Nm)**	**RMS Torque (Nm)**
8	1.306	1.306
7	1.318	1.318
6	0.573	0.573
5	1.319	1.319
4	1.319	1.319
3	1.319	1.319
2	0.876	0.876
1	1.334	1.323

Table 16: Maximum and RMS Torque for downward travel.

From the move profile, we demand a maximum angular velocity of $2.054\ rad/sec$ from the wheels of our device. The maximum angular velocity that the motor is operating at is $7.854\ rad/sec$ (or equivalently, $75\ RPM$). From this, we determine that the velocity ratio is 1:1; in other words, there is no need for a velocity ratio.

The data sheet of the motor can be found in Appendix. We have interpolated the data for the stall torque due to the absence of the information. Furthermore, we have made a bold assumption by using the approximation of

$$T_{max} = T_{stall}/2 \text{ and } S_{max} = S_{noload}/2 \text{ where } T_{max} \cdot S_{max} = P_{max}$$

Whereafter, T_{max} was used in computing the gear ratio necessary for our device.

The Torque-Speed profile of Tetrix DC Motor W39530 is presented below. The motor will run at its highest torque which corresponds to T_{max} but without exceeding S_{max} of the motor. Ultimately, we will be running in the "high power" capabilities of the motor.

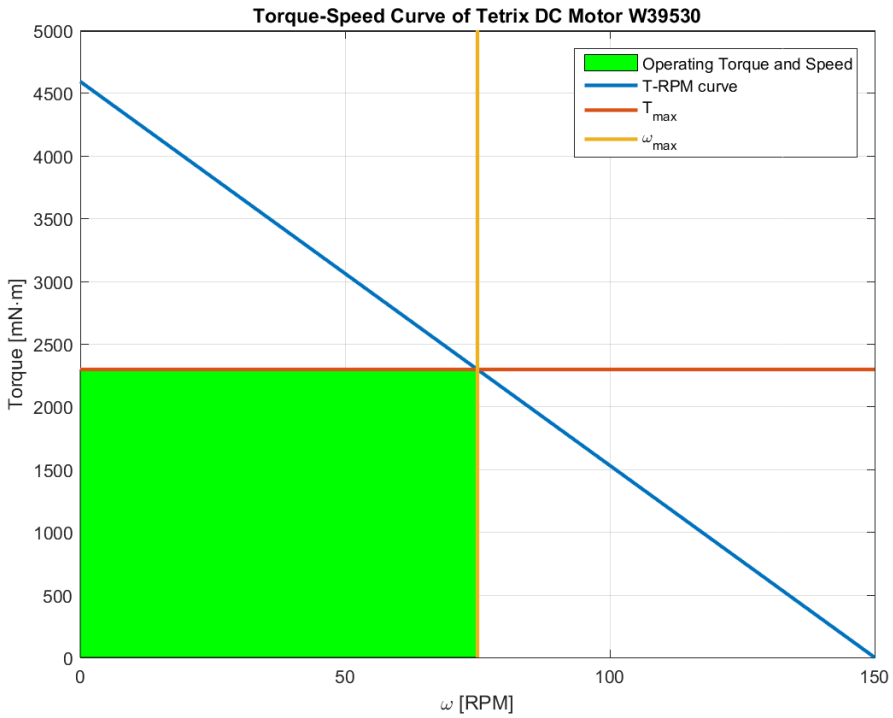


Figure 22: Torque-Speed Curve of Tetrix DC Motor W39530. Observe the operating area under the curve.

Using 1 motor, the torque from the motor are capable of moving our device across the runway both upward and downward. We have made assumptions leading to the values for T_{max} and S_{max}, and if our device demands a greater torque from the motor, we are able to harness the torque not in the range of the operating torque as shown in the figure above. Nevertheless, the compromise would be to move at a lower speed relative to the speed achievable in the operating speed range. Ultimately, the motors are sufficiently strong enough to move the device.

3.1.8. Steering Calculations

The following section will discuss the choice of the steering mechanism and how the parameters were chosen to achieve a complete turn successfully and without any slippage.

As briefed before, Rack Attack will use an Ackerman steering mechanism. This mechanism uses geometrical correlations between the angles of the wheels (with respect to vertical) and the radius of turning. The wheels of the Ackerman steering turn at different angles so that the wheel further from the center of rotation goes through a longer turn to preserve the total tangential velocity of the vehicle.

Radius of turning R:-

Since we are turning about a curved ramp that has an outer radius R_{out} = 28in and inner R_{inner} = 10, then our robot should be turning about the average of these radii.

$$R_{turn} = \frac{R_{out} + R_{inner}}{2} = 19\ in$$

For a perfect turn assuming no slipping in the system and perfect geometries of the wheels, our devise should turn prefect about radius R_{turn} = 19in.

Realistically there will be significant amount of slipping especially because we have a higher traction force on the rear wheels than the front wheels. As a result, we will reduce the radius of turning to about 17 in (estimate), which will make the turning angle much sharper. This will account for the slippage that will occur in the front wheels, which would likely cause our mechanism to turn further than expected. This estimation can be considered to be a factor of safety.

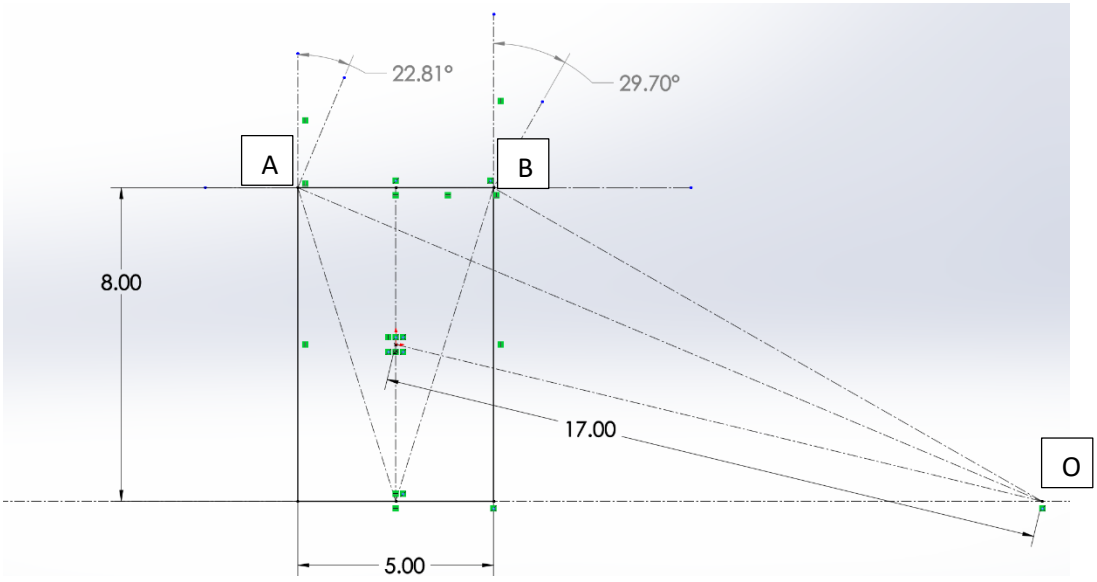


Figure 23: The geometry of the Ackerman steering

Note: 'O' marks the center of rotation and that can be found by connecting a line with

length R_{turn} from the midpoint of the line connecting the axles to the rear wheel axle extended construction line.

The figure above describes how the steering mechanism will be set up. The two horizontal construction lines represent the wheel axles of the front and the rear wheels (top side is the front of the robot). Since we want the robot to be rotating about the center of the ramp's curve and according to the Ackerman algorithm, the wheels should be tangent to the line connecting them with the point of the center of rotation (O). Solving the geometry for these conditions will give us an outer wheel maximum turning angle of 22.81˚ and an inner wheel maximum turning angle of 29.70˚. These angles mark the maximum expected turning angles. Moreover, the link that connects the wheels to the chassis should be approximately the angle of the lines that connect the midpoint of the rear axle to each of the points A and B *referred to in the diagram above.*

The steering for this device is implemented through the following mechanism:

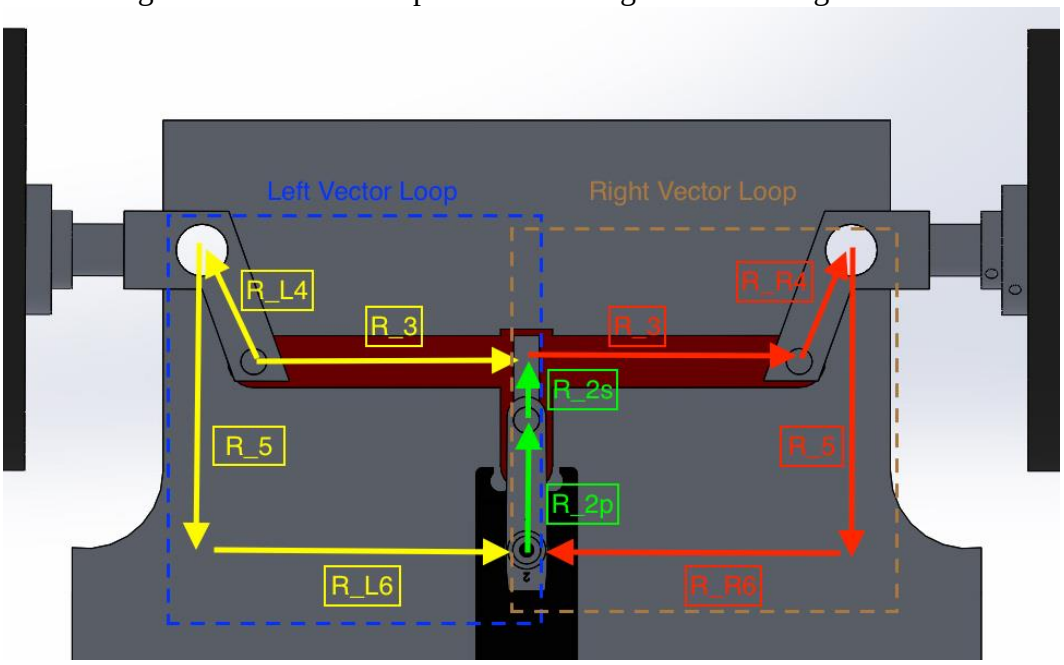


Figure 24: This figure illustrates the mechanism used to control Ackerman steering in the vehicle. The link labels will be referenced in the following discussions.

This implementation of Ackerman steering features a four bar linkage between the left and right turning wheels. This ensures that all four wheels on the device rotate about the same point in order to minimize slipping. This linkage is driven by a servomotor as shown.

In order to size the servomotor, we will take the following approach:

1. Determine how much torque is required to turn a wheel.
2. Determine the mechanical advantage our mechanism provides.
3. Calculate the torque required by the servomotor.

In the following discussion, we will use the variable names depicted in Figure 21.

1. **Determine how much torque is required to turn a wheel.**

   To turn a wheel, the worst-case, most torque-demanding scenario will occur when the wheels are at rest, meaning zero angular velocity. This is shown in Figure 22.

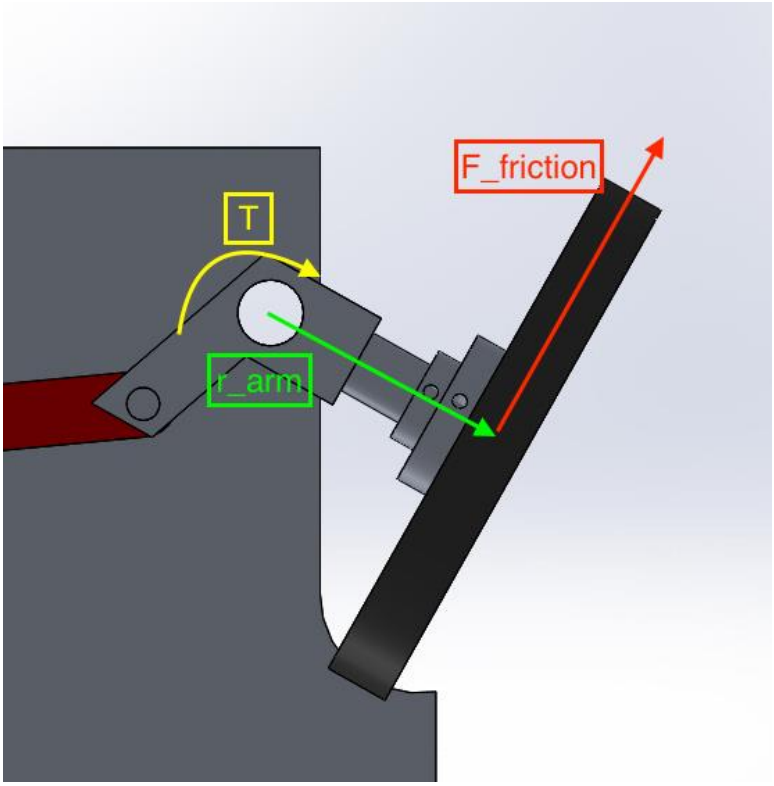


Figure 25 : Illustration of the front wheel assembly and the torque required to turn a single wheel.

In this case, we can estimate the torque required to turn a single wheel with the following equation:

$$T = F_{fric} r_{arm} \quad (30)$$

$$T = \mu_S N r_{arm} \quad (31)$$

For our case, r_{arm} is known to be 0.0395m. The load distributed to each front wheel is 0.297kg. In addition, the friction coefficient between the surface of the track and the wheel is 0.45. As a result, the torque required to turn a wheel is 0.0518 N-m.

2. Determine the mechanical advantage our mechanism provides.

To determine the mechanical advantage our mechanism provides, we use the vectors assigned to the linkages as shown in Figure 24. We will start by creating two vector loop equations.

$$\overline{R_{2p}} + \overline{R_{2s}} + \overline{R_3} + \overline{R_{R4}} + \overline{R_5} + \overline{R_{R6}} = 0 \tag{32}$$

$$\overline{R_{2p}} + \overline{R_{2s}} - \overline{R_3} + \overline{R_{L4}} + \overline{R_5} + \overline{R_{L6}} = 0 \tag{33}$$

Eq 32 contains information pertaining to the vector loop including the right wheel and Eq 33 contains information pertaining to the vector loop including the left wheel. We are able to split each of these vector loops into their constituent X and Y components, leaving us with the following system of equations:

$$\begin{aligned} r_{2p}c_{2p} + \boldsymbol{r_{2s}c_{2s}} + r_3\boldsymbol{c_3} + r_{R4}\boldsymbol{c_{R4}} + r_5c_5 + r_{R6}c_{R6} = 0 \\ r_{2p}s_{2p} + \boldsymbol{r_{2s}s_{2s}} + r_3\boldsymbol{s_3} + r_{R4}\boldsymbol{s_{R4}} + r_5s_5 + r_{R6}s_{R6} = 0 \\ r_{2p}c_{2p} + \boldsymbol{r_{2s}c_{2s}} - r_3\boldsymbol{c_3} + r_{L4}\boldsymbol{c_{L4}} + r_5c_5 + r_{L6}c_{L6} = 0 \\ r_{2p}s_{2p} + \boldsymbol{r_{2s}s_{2s}} - r_3\boldsymbol{s_3} + r_{L4}\boldsymbol{s_{L4}} + r_5s_5 + r_{L6}s_{L6} = 0 \end{aligned} \tag{34}$$

NOTE: ***In this system of equations, for the sake of clarity, we utilize the following notation convention:***

$$|\overline{R_x}| = r_x, \qquad sin\,\theta_x = s_x, \quad cos\,\theta_x = c_x$$

In this system of equations, the bolded variables are unknowns:

$r_{2s}, \theta_{2s}, \theta_3, \theta_{R4}, \theta_{L4}$

We can simplify this one step further by recognizing there is a constraining equation between θ_{2s} and θ_3:

$$\theta_{2s} = \theta_3 + 90° \tag{35}$$

With this, we are able to reduce our system of nonlinear equations to four equations, four unknowns, which makes is solvable. Due to the difficulty associated with solving nonlinear systems with coupled variables, we resort to using a nonlinear system solver in MATLAB. Our code can be referenced in Appendix.

With this, we have a completely solved position solution, giving us the angles and magnitudes of all vectors in our setup. We will now go one step further and derive all equations in Eq 36 with respect with θ_{2p}:

$$-r_{2p}s_{2p} - \boldsymbol{r'_{2s}}s_3 - r_{2s}c_3\boldsymbol{\theta'_3} - r_3s_3\theta'_3 - r_{R4}s_{R4}\boldsymbol{\theta'_{R4}} = 0$$

$$
\begin{aligned}
r_{2p}c_{2p} + \boldsymbol{r'_{2s}}c_3 - r_{2s}s_3\boldsymbol{\theta'_3} + r_3c_3\theta'_3 + r_{R4}c_{R4}\boldsymbol{\theta'_{R4}} &= 0 \\
-r_{2p}s_{2p} - \boldsymbol{r'_{2s}}s_3 - r_{2s}c_3\boldsymbol{\theta'_3} + r_3s_3\theta'_3 - r_{L4}s_{L4}\boldsymbol{\theta'_{L4}} &= 0 \\
r_{2p}c_{2p} + \boldsymbol{r'_{2s}}c_3 - r_{2s}s_3\boldsymbol{\theta'_3} - r_3c_3\theta'_3 + r_{L4}c_{L4}\boldsymbol{\theta'_{L4}} &= 0
\end{aligned}
\tag{36}
$$

Once again, the unknown quantities are bolded. For this calculation, we are faced with a system of linear equations that can be easily solved. This is also done in MATLAB (**See Appendix**).

With this we are able to obtain values for θ'_{R4} and θ'_{L4}. With these two values, we are able to obtain a value for mechanical advantage.

From the conservation of energy:

$$P_{in} = P_{out} \tag{37}$$

Input power is from the servomotor and output power is from the turning wheels

$$P_{2p} = P_{R4} + P_{L4} \tag{38}$$

$$
\begin{aligned}
T_{2p}\omega_{2p} &= T_{R4}\omega_{R4} + T_{L4}\omega_{L4} \\
&= T_{R4}\omega_{2p}\theta'_{R4} + T_{L4}\omega_{2p}\theta'_{L4}
\end{aligned}
\tag{39}
$$

$$T_{2p} = T_{R4}\theta'_{R4} + T_{L4}\theta'_{L4} \tag{40}$$

We realize that the torque in links L4 and R4 are the values calculated as the torque required to turn a wheel from Eq 41.

$$T_{2p} = T_{wheel}(\theta'_{R4} + \theta'_{L4}) \tag{41}$$

We realize that the mechanical advantage our mechanism provides is approximately:

$$MA = \frac{1}{(\theta'_{R4} + \theta'_{L4})} \tag{42}$$

3. Calculate the torque required by the servomotor.

Finally, to calculate the torque required by the servomotor, we know that the power transmission provided by the linkage is not perfect; there will be energy lost to friction. As a result, we assume an "efficiency", e of our linkage of 0.75

Finally we can calculate the torque required to turn our linkage as follows:

$$T_{2p} = \frac{T_{wheel}(\theta'_{R4} + \theta'_{L4})}{e} \tag{43}$$

Through these three steps of calculations, we are able to calculate the torque required to turn the wheels. However, it is important to distinguish that this is the torque required for a given position. In order to calculate the maximum torque required, we must consider these

calculations over the entire range of turning angles expected. To do this, we run the calculation on MATLAB, repeating the calculations over the entire range of turning angles the mechanism we expect to engage. We expect to have a maximum inside wheel turning angle of 30°. We iterate through different positions for the mechanism until the inside turning wheel hits this angle (SEE FIGURE 23).

Doing this gives us a maximum expected torque of 41.48 in-oz. Using this value, we are then able to specify the servomotor we will choose to use.

3.2. Ball Delivery System

The golf ball delivery system selected for our robot uses tension springs stretched by a rack and pinion system. The pinion is directly driven by a high-torque brushed DC motor. The rack is attached to an arm which moves horizontally above a channel that houses the golf ball. Several guide slots keep the arm positioned correctly. At one end of the arm is a pivot shaft which allows an L-shaped piece, called the flapper, to rotate. The longer leg of the flapper serves the purpose of retaining the golf ball during spring pullback. The shorter leg controls the timing of retaining and releasing the ball via a cam system. When a golf ball is not present, contact between the flapper and cam is maintained via torsion springs attached to the pivot shaft and flapper. Below the cam and rack parts, a tension spring is located on either side of the golf ball channel. The two are connected across the channel by a beam. When the arm pulls the golf ball back, this pushes the beam and extends the springs. The beam's motion is kept linear by guide slots on either side of the mechanism. The basic components and motion of the system can be seen in the following prototype CAD model images:

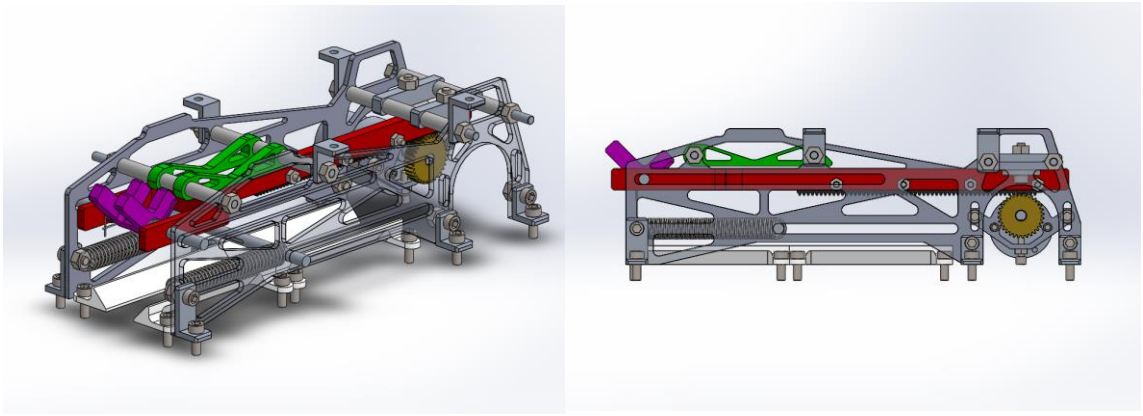


Figure 26: Mechanism prepared to accept golf ball.

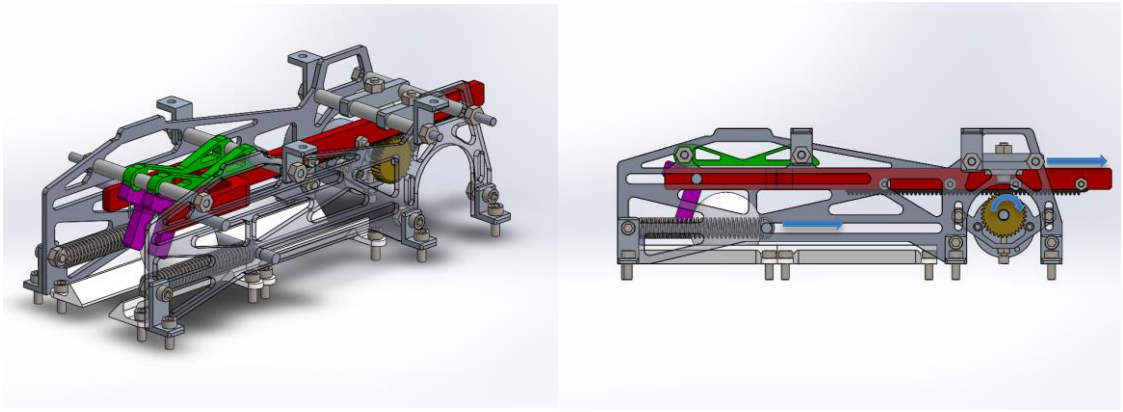


Figure 27: Golf ball captured; motor pulling golf ball and springs to the right.

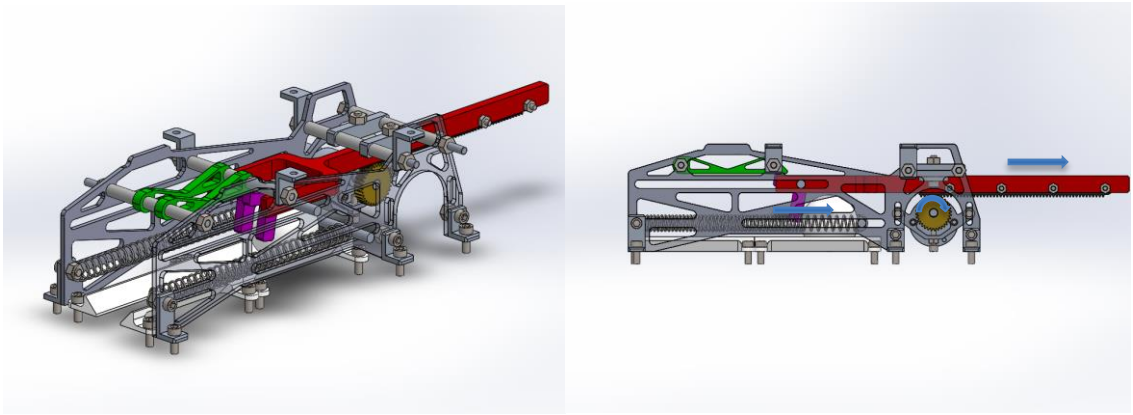


Figure 28: Mechanism about to release golf ball; motor pulling golf ball and springs to the right.

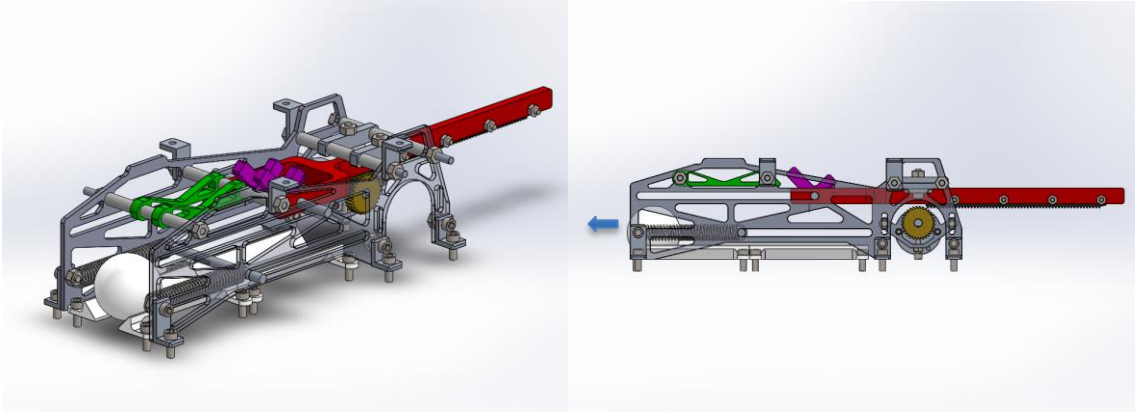


Figure29: Golf exiting the robot; the flapper has returned to the neutral position.

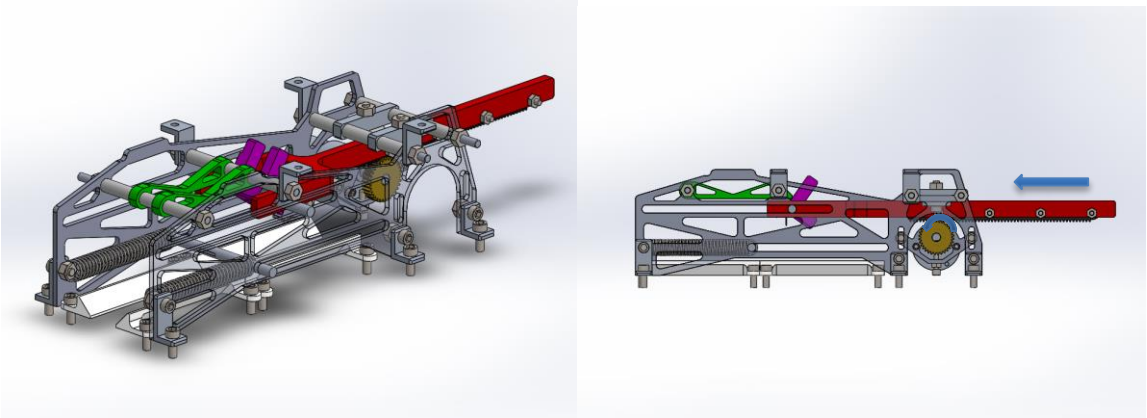


Figure 30: Post launch; the rack begins returning to loading position, moving to the left.

3.2.1. Relevant Ball Delivery Sub-System Calculations

In analyzing the mechanics of this ball delivery system to ensure its success and size components, three main areas of focus were most relevant in terms of calculations: initial velocity, spring energy, and motor torque. In completing these calculations, the following noteworthy assumptions were made:

- Negligible air resistance acting on the golf ball during its trajectory.
- Negligible dynamic friction between the golf ball and launcher channel during launch.
- No spinning or rolling of the golf ball along the channel during launch.
- Tension and torsion springs behave under their respective versions of Hooke's Law.
- Motor efficiency is constant throughout the mechanism's range of motion.
- Elastic deformation in components other than the springs during pullback is negligible.
- The bulk of the robot remains stationary during ball launch (no energy transferred into chassis or wheel movement).
- Negligible friction between the rack arm and the rear lateral guides.
- For the purposes of statics calculations, forces due to gravity on the flapper and golf ball are negligible.

Basic vertical and horizontal kinematics equations were used as the foundation for predicting the golf ball's trajectory, selecting a reasonable target launch distance window, and calculating the initial velocity needed for the golf ball to exhibit the desired trajectory:

$$y(t) = -½gt^2 + v_0 sin(\theta)t \quad (44)$$

$$x(t) = v_0 cos(\theta)t \quad (45)$$

In these equations, the origin is taken to be the golf ball's position when it leaves the robot at $v=v_0$ and $t=0$. Thus $y(t_{land}) = h_{land}$, where h_{land} is negative due to the launch location being located at the top of the ramp. h_{land} can be expressed in the form:

$$h_{land} = -h_{launcher} - h_{platform} \quad (46)$$

where h_{launch} is the difference in height between the bottom of the robot and the location of the golf ball's departure. By solving for t_{land} in the horizontal kinematic equation evaluated at $y(t_{land}) = -h_{land}$, evaluating $x(t_{land}) = x_{land}$, and solving for v_0, the following equation is acquired:

$$v_o = \frac{x_{land}\sqrt{g * \sec(\theta)}}{\sqrt{2(x_{land}\sin(\theta) - h_{land}\cos(\theta))}} \quad (47)$$

This formula outputs the initial velocity needed to achieve a horizontal trajectory distance of x_{land} for a given launch angle and height. For example, for a target distance of 1.88 meters and launch angle of 8.6 degrees, the necessary launch velocity is 4.81 meters per second (h_{land} is -0.48 meters).

With the needed initial velocity known, a range of spring options can be formed from energy relationships. In particular, conservation of energy between the golf ball's resting state with maximum spring tension and the time at which the golf ball loses contact with the spring beam can be employed:

$$\Delta P_{gravitational} + \Delta K + \Delta P_{spring} = 0 \quad (48)$$

$$m_{gh}g(h_f - h_i) + \frac{1}{2}m_{gb}(v_f^2 - v_i^2) + \frac{1}{2}n_{spr}k(\Delta l_f - \Delta l_i) = 0 \quad (49)$$

In the case of this launch mechanism, h_i, v_i, and Δl_f can be set to zero. Solving for k yields the following relationship:

$$k = \frac{m_{gb}(v_0^2 + 2gsin(\theta) * \Delta l_i)}{n_{spr} * \Delta l_i^2} \quad (50)$$

To assist in selecting a spring, this equation was plotted as shown below:

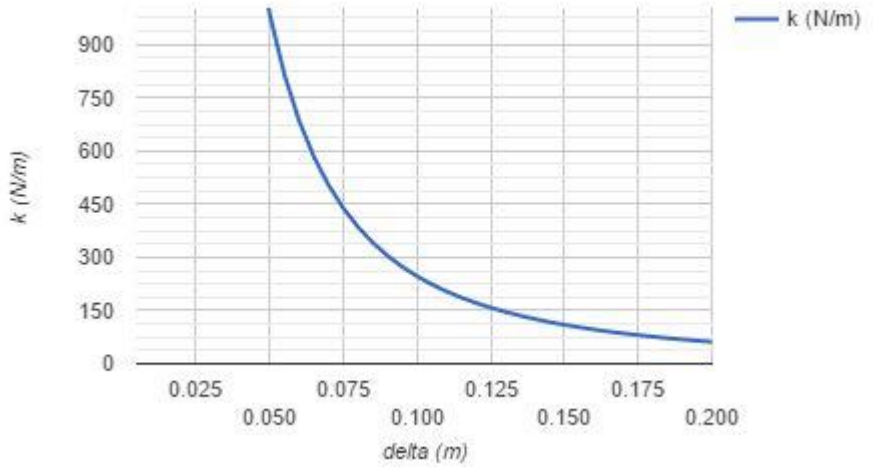


Figure 31: Needed spring extension versus needed spring stiffness to achieve target trajectory. Parameters: m_{gb} = 45.9g, theta = 0.17rad, n_{spr} = 2, x_{land} = 5.5m, h_{land} = 0.483m.

While our design permits some flexibility in spring constants in that the springs are easily interchangeable, currently the intended spring combination consists of an extension length of 0.093 meters and a stiffness of 275 Newtons per meter.

With the spring stiffness and maximum extension known, the final critical specification to determine is the torque capability of the motor. To obtain an acceptable estimate, a system of statics equations can be formed and solved. While other methods are possible, in this instance the system was split into three free body diagrams: one for the arm and rack, one for the flapper, and one for the golf ball. For the arm and rack, a system of four equations was formed including sum of forces in the horizontal and vertical directions, as well as sum of moments about the pivot shaft and the force due to friction against the upper rear guide. For the flapper, a system of four equations was formed including sum of forces in the horizontal and vertical directions, as well as sum of moments about the pivot shaft and the force due to friction against the cam. For the golf ball, a system of three equations was formed including sum of forces in the horizontal and vertical directions, as well as the force due to friction against the bottom plate. To simplify the calculations, all three subsystems of equations were combined into a matrix, which was solved numerically using MATLAB (see Appendix for code).

In these statics equations, the torques due to the torsion springs are taken into account. Two torsion springs lie on the pivot shaft, each with one leg fixed to the flapper and the other to the arm. The two torsion springs share the same torsional spring coefficient and can each accept 270 degrees of rotation from fully uncompressed to fully compressed. The springs are mated to the system in such an arrangement that under their own forces they cause the flapper to reach an equilibrium where the angle measured clockwise from the horizontal to the short leg of the flapper is 135 degrees, with each flapper in the middle of its range of rotation. This way the flapper can rotate 135 degrees in either direction as is needed for its path of motion, while always tending toward the 135 degree offset shown in Figures 30 and 32 above.

In addition to the tension spring constant, number of tension springs, and extension of the tension springs, the system of static equations accounts for the following as input parameters:

- Torsion spring coefficients
- Torsion spring neutral angular offsets
- Intended flapper neutral angular position
- Flapper angular position
- Coefficient of friction between aluminum and aluminum
- Coefficient of friction between aluminum and plastic rollers
- Coefficient of friction between aluminum and golf ball
- Length of short flapper leg
- Length of long flapper leg
- Horizontal distance between pivot shaft and pinion
- Horizontal distance between pivot shaft and arm center of mass
- Horizontal distance between pivot shaft and upper rear guide

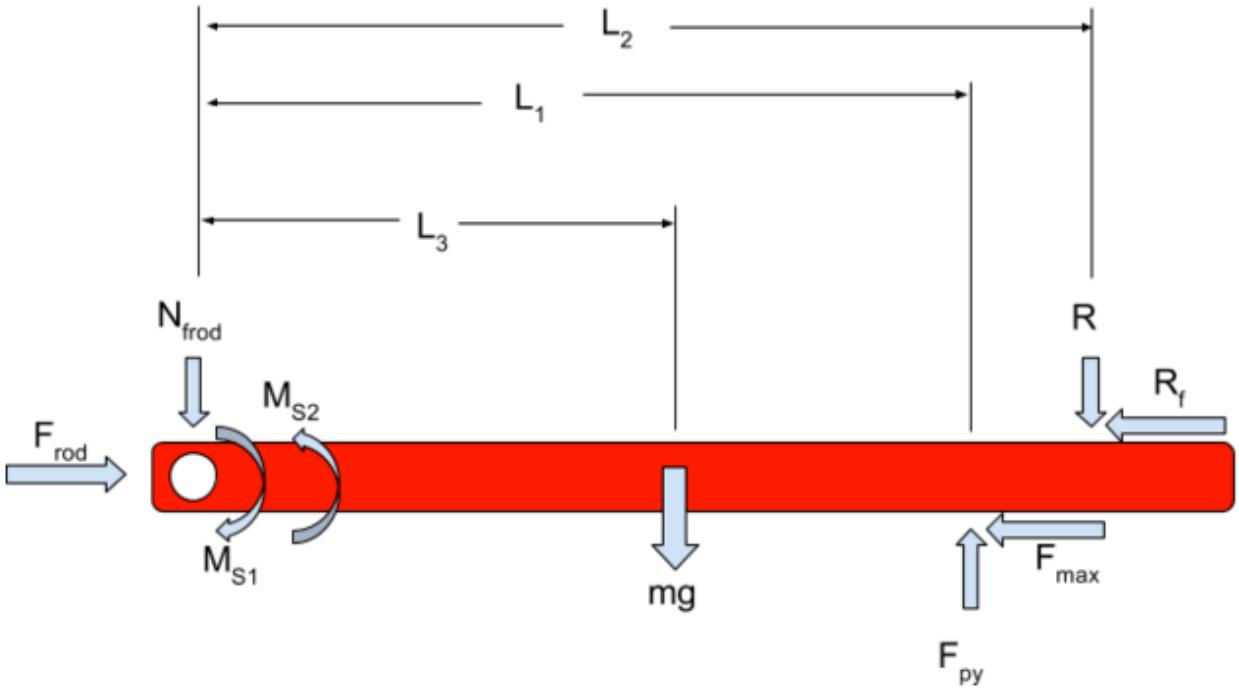
Values used for the input parameters are listed in Table 17.

Parameter	**Value**	**Units**
Tension spring constant	275	N/m
Number of tension springs	2	-
Max tension spring extension	0.093	m
Torsion spring constant	0.0163	Nm/rad
Torsion spring 1 neutral offset	-2.356	rad
Torsion spring 2 neutral offset	2.356	rad
Flapper angle	0.17	rad
Flapper neutral angle	2.356	rad
Aluminum-Aluminum	1.0	-
Aluminum-plastic roller	0.1	-
Aluminum-golf ball	0.7	-
Short flapper leg length	0.020	m
Long flapper leg length	0.028	m

Pivot shaft to pinion length	0.095	m
Pivot shaft to arm CG length	0.095	m
Pivot shaft to rear guide length	0.099	m

Table 17: Selected Input Parameters for the Statics System of Equations.

The three free body diagrams used to construct the system of equations are shown below:



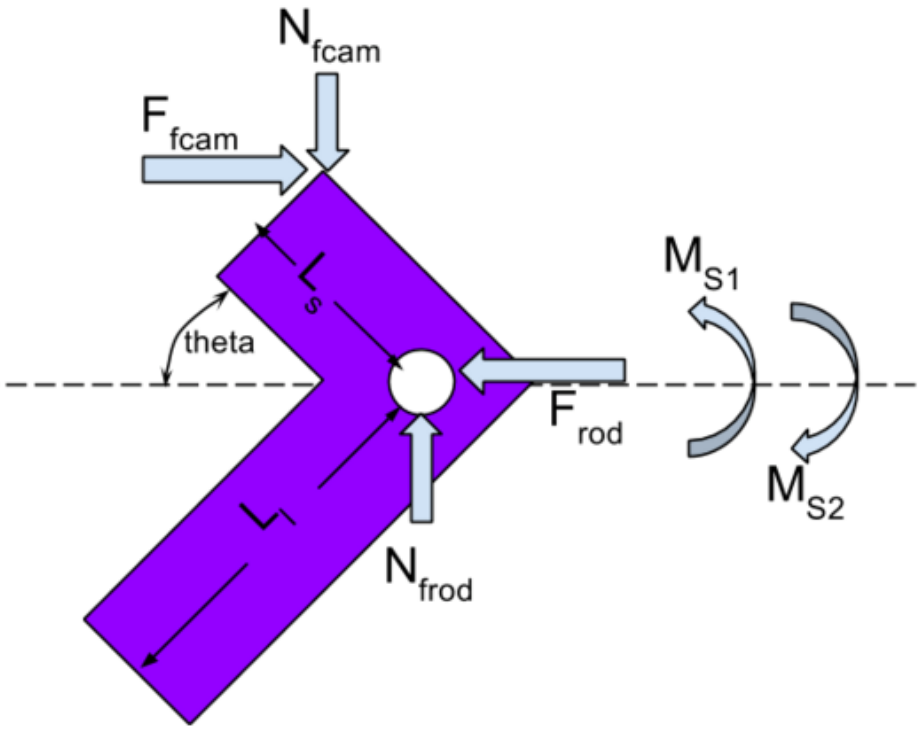
$$\overset{+}{\rightarrow} \sum F_x = 0 = F_{rod} - F_{max} - R_f$$

$$+\downarrow \sum F_y = 0 = F_{py} - R - N_{frod} - mg$$

$$R_f = -R\mu_3$$

$$\circlearrowleft \sum M_{rod} = 0 = M_{s2} - M_{s1} + F_{py}L_1 - RL_2 - mgL_3$$

Figure 32: Arm and rack free body diagram.



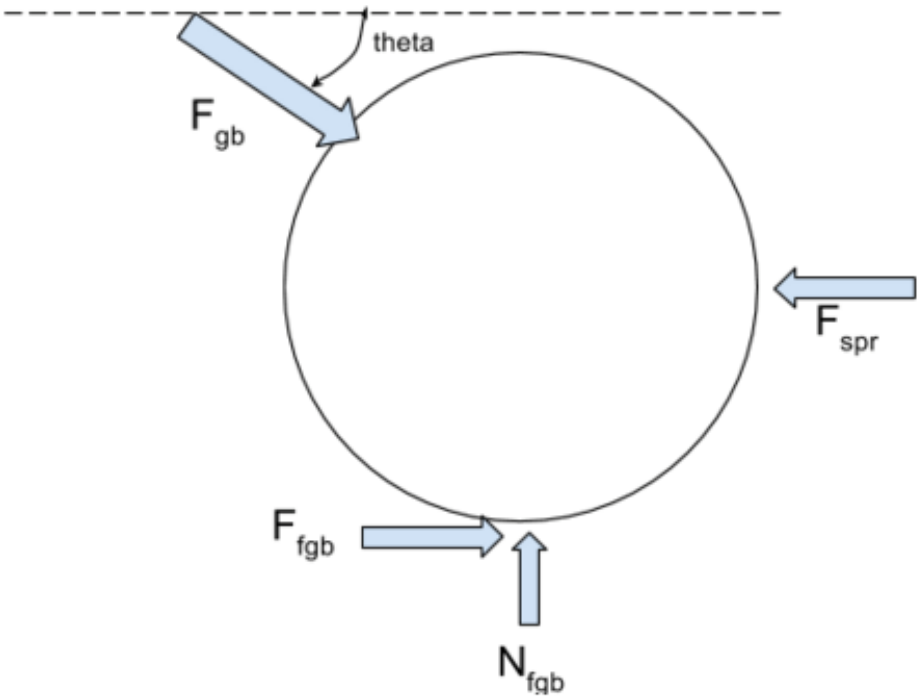
$$\overset{+}{\rightarrow} \sum F_x = 0 = -F_{rod} + F_{cam} - F_{gb}cos\theta$$

$$+\downarrow \sum F_y = 0 = -N_{fcam} + N_{frod} - F_{gb}sin\theta$$

$$\circlearrowleft \sum M_{rod} = 0 = M_{s1} - M_{s2} + F_{gb}L_l - F_{fcam}L_s sin\theta + N_{fcam}L_s cos\theta$$

$$F_{fcam} = N_{fcam}\mu_1$$

Figure 33: Flapper free body diagram.



$$\overset{+}{\rightarrow} \sum F_x = 0 = -F_{spr} + F_{fgb} + F_{gb}cos\theta$$

$$+\downarrow \sum F_y = 0 = N_{fgb} - F_{gb}sin\theta$$

$$F_{fgb} = -N_{fgb}\mu_2$$

$$F_{spr} = n_{spr}k\Delta l$$

Figure 34: Golf ball free body diagram.

Using the input values from Table 17, a maximum required motor force at the teeth of 315 N was calculated. However, for the mechanism to completely pull back the golf ball while the robot drives up the ramp, the motor must also spin at a certain speed. This creates a set of interdependent parameters: the target launch distance and initial launch angle; the spring's maximum travel and rate; the motor's speed and torque; and the selected pinion gear. The combination chosen consists of the following:

- A target distance of 5.5 meters using an initial launch angle of 10 degrees
- Two stainless steel extension springs with a nominal end-to-end length of 63.5 mm, a max extended length of 157 mm, and a spring rate of 275 N/m
- A pinion gear with a 25.4mm pitch diameter, thus requiring 3.9 Nm of torque from the motor and an average speed of 1.63 rpm to complete the pull back in a ramp traversal time of 43 seconds
- One 12 V DC brushed gear motor with a stall torque of 7.8 Nm at 0.5 A, and a no load speed of 3 rpm (thus approximately 3.9 Nm of torque at 1.63 rpm should be achievable)

Altogether, the results of these calculations prove the feasibility of the design and provide a strong theory-based starting point for sizing and creating detailed specifications for the golf ball launch mechanism.

4. Summary and Conclusions

Our team was tasked with developing an autonomous device to transport a golf ball up a ramp and shoot it a minimum of two feet horizontally while exhibiting and maintaining accuracy. We began by outlining the high and low-level design requirements. We then brainstormed three unique design concepts, which presented different combinations of drive systems and shooting mechanisms. Through the Pairwise Comparison Chart the designs were compared and evaluated on a weighted list of criteria comprised of cost, weight, reliability, durability, simplicity, accuracy, and speed. It was determined that the spring rack and pinion robot was best suited for the challenge due to its anticipated reliability, durability, and accuracy. With one design in mind, a rough CAD model of the system was created to confirm that reasonable packaging and assembly were achievable while maintaining an acceptable weight distribution. Following this, detailed calculations were performed to prove the validity of the drive, steering, and launch systems. The results show that as expected, the design is practical and ready to move to the fabrication phase.

5. References

[1] D. Hong. (2011). *Making a car for blind drivers* [Online]. Available: http://www.ted.com/talks/dennis_hong_making_a_car_for_blind_drivers

[2] Lecture 1, MAE-162D- Mechanical Product Design-1 Course Introduction, Winter 2016, UCLA

[3] T. D. Gillespie, "The Steering Linkages," in *Fundamentals of Vehicle Dynamics*: SAE, 1992, ch. *8*, pp. *275–277*.

[4] M. Simon. (2015, Feb. 10). *This Incredible Hospital Robot Is Saving Lives. Also, I Hate It* [Online]. Available: http://www.wired.com/2015/02/incredible-hospital-robot-saving-lives-also-hate/

[5] S. Pozzi et al., "Proximity sensor," U.S. Patent 9 251 706, February 2, 2016.

[6] M. Liaw, "Obstacle detection device,' U.S. Patent 9 245 342, January 26, 2016

[7] R. Wall *et al*., "Creating a Low-Cost Autonomous Vehicle," Univ. of Idaho, Moscow, ID, 2002.

[8] Robotics Universe. *How to choose a motor for your robot* [Online]. Available: http://www.robotoid.com/howto/choosing-a-motor-type.html

[9] Pitsco, *Technical Specs- TETRIX MAX DC gear Motor 39530*, Pitssco Ed., 2013.

[10] YouTube, *Robot Drive Systems by Greg*: https://www.youtube.com/watch?v=NjXK5Fzq6Fs; accessed Jan. 15, 2016

[11] Aoyama, A. T., Pairwise Comparison Chart Template, Fall 2010

[12] J. Graham, USA Today (2016, Nov. 18). *L.A. mayor tests semi self-driving car to ease commuting* [Online]. Available: http://www.usatoday.com/story/tech/2015/11/17/l-mayor- -traffic- improve-5- years/75934114/

[13] Boys' Life (2013). *Build a pingpong ball launcher* [Online]. Available: http://boyslife.org/hobbies-projects/projects/55008/build-a-pingpong-ball-launcher/

[14] P. Gittings, CNN (2012, Jan. 11). *RoboPutt: Rise of the golfing machines?* [Online]. Available: http://edition.cnn.com/2012/01/10/sport/golf/golf-robots/index.html

6. Appendix

Projectile Calculations

```
clc
clear
```

Solve the statics system of equations regarding projectile system

```
% INPUTS:
% theta_home: intended neutral location of flapper, in radians measured
% from the horizontal CW (around the pivot) to the short leg of the flapper
% this will be achieved if both springs have the same constant and opposite
% theta offsets.
% theta_X_offset: angular offset of neutral spring location from theta_home
% thetasXmax: torque at max spring twist
% kX: spring constant (Nm/rad or N/m depending on spring type)
% n_tenspr = number of tension springs
% delta_x: tension spring extension (.096 meters is roughly max) to use in
% statics calculations
% theta: angle to use in statics calculations, in radians measured
% from the horizontal CW (around the pivot) to the short leg of the flapper
% muX: coefficient of friction for material combination X
% r_p: pinion gear effective radius (pitch dia/2)
% L1: horizontal distance between flapper pivot and rack/pinion contact (lever arm)
% L2: horizontal distance between flapper pivot and upper arm guide (lever arm)
% Ls: short flapper leg distance to pivot (lever arm)
% Ll: long flapper leg distance to pivot (lever arm)


% OUTPUTS:
```

```
% Primary output is the torque the motor must create to produce a static
% system given the parameters chosen. Also outputs ~11 other potentially
% useful-to-know forces/torques in the system. See sketched FBDs for
% details.

%-------------------------------------------------------------------------%

g = 9.81; % m/s^2

theta_home = 2.356; % rads

theta_1_offset = -2.356; % rads; angular offset of neutral spring location from theta_home
ts1max = .0768; % Nm; torque at max spring twist
thetas1max = 4.712; % rads; max spring twist angle
k1 = .0163 %ts1max/thetas1max; % Nm/rad

theta_2_offset = 2.356; % rads; angular offset of neutral spring location from theta_home
ts2max = .0768; % Nm; torque at max spring twist
thetas2max = 4.712; % rads; max spring twist angle
k2 = .0163 %ts2max/thetas2max; % Nm/rad

k3 = 275; % N/m; tension spring constant
n_tenspr = 2; % number of tension springs
delta_x = .093; % m; tension spring extension (.096 meters is roughly max)

r_p = .0127; % m; pinion gear effective radius (pitch radius)

theta = 0.17 % radians (3/4*pi results in balanced springs)

delta_theta_1 = theta - theta_home - theta_1_offset; % rads
delta_theta_2 = theta - theta_home - theta_2_offset; % rads
```

```
Ms1 = -k1*delta_theta_1 % Nm
Ms1imp = Ms1*8.8507
Ms2 = -k2*delta_theta_2 % Nm
Ms2imp = Ms2*8.8507

mu1 = 1.0; % aluminum-aluminum estimated friction coefficient
mu2 = 0.7; % aluminum-golf ball estimated friction coefficient
mu3 = 0.1; % aluminum-plastic (rolling guide wheel) estimated friction coefficient

L1 = .095; % m; horizontal distance between flapper pivot and rack/pinion contact (lever arm)
L2 = .099; % m; horizontal distance between flapper pivot and upper arm guide (lever arm)
Ls = .020; % m; short flapper leg distance to pivot (lever arm)
Ll = .028; % m; long flapper leg distance to pivot (lever arm)
L3 = .095; % m; distance from flapper pivot to arm CG

m_arm = 0.0949; % kg; mass of arm

A = ...
[0 0 0 -1 1 -cos(theta) 0 0 0 0 0 0 0 0;
 0 0 0 0 0 sin(theta) -1 1 0 0 0 0 0 0;
 0 0 0 0 -Ls*sin(theta) Ll Ls*cos(theta) 0 1 -1 0 0 0 0;
 0 0 0 0 1 0 -mu1 0 0 0 0 0 0 0;
 0 0 0 0 0 cos(theta) 0 0 0 0 -1 1 0 0;
 0 0 0 0 0 -sin(theta) 0 0 0 0 0 0 1 0;
 0 0 0 0 0 0 0 0 0 0 0 0 1 -mu2 0;
 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 -1;
 0 1 -1 0 0 0 0 -1 0 0 0 0 0 0 0;
 0 0 mu3 0 0 0 0 0 0 0 0 0 0 1;
 0 L1 -L2 0 0 0 0 0 -1 1 0 0 0 0;
 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
```

```
 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

b = [0 0 0 0 0 0 0 0 m_arm*g 0 m_arm*g*L3 n_tenspr*k3*delta_x Ms1 Ms2]';

x = A\b % torques in Nm, forces in N; represents [Tmax Fpinion,vertical
Reactive_force_at_rear_guide,vertical Reactive_force_at_flapper_pivot,horizontal
Frictional_force_at_cam,horizontal Force_between_flapper_and_golfball
Normal_force_on_cam,vertical Normal_force_on_rod,vertical Ms1 Ms2
Force_on_golfball_due_to_tension_springs Frictional_force_on_golfball_at_bottom,horizontal
Normal_force_on_golfball_at_bottom,vertical Frictional_force_at_rear_arm_guide,horizontal]
```

```
k1 =

    0.0163


k2 =

    0.0163


theta =

    0.1700


Ms1 =

   -0.0028
```

Ms1imp =

-0.0245

Ms2 =

0.0740

Ms2imp =

0.6553

A =

Columns 1 through 7

```
         0         0         0   -1.0000    1.0000   -0.9856         0
         0         0         0         0         0    0.1692   -1.0000
         0         0         0         0   -0.0034    0.0280    0.0197
         0         0         0         0    1.0000         0   -1.0000
         0         0         0         0         0    0.9856         0
         0         0         0         0         0   -0.1692         0
         0         0         0         0         0         0         0
   -1.0000         0         0    1.0000         0         0         0
         0    1.0000   -1.0000         0         0         0         0
         0         0    0.1000         0         0         0         0
```

```
    0    0.0950   -0.0990         0         0         0         0
    0         0         0         0         0         0         0
    0         0         0         0         0         0         0
    0         0         0         0         0         0         0
```

Columns 8 through 14

```
    0         0         0         0         0         0         0
 1.0000         0         0         0         0         0         0
    0    1.0000   -1.0000         0         0         0         0
    0         0         0         0         0         0         0
    0         0         0   -1.0000    1.0000         0         0
    0         0         0         0         0    1.0000         0
    0         0         0         0    1.0000   -0.7000         0
    0         0         0         0         0         0   -1.0000
-1.0000         0         0         0         0         0         0
    0         0         0         0         0         0    1.0000
    0   -1.0000    1.0000         0         0         0         0
    0         0         0    1.0000         0         0         0
    0    1.0000         0         0         0         0         0
    0         0    1.0000         0         0         0         0
```

x =

```
  1.0e+03 *

  -0.3146
  -2.0238
  -1.9422
  -0.1204
```

-0.0747
0.0463
-0.0747
-0.0826
-0.0000
0.0001
0.0512
0.0055
0.0078
0.1942

Convert Nm to oz-in and in-lb

```
max_torque_Nm = x(1)*r_p

max_torque_oz_in = max_torque_Nm*141.6119

max_torque_in_lb = max_torque_Nm*8.8507
```

max_torque_Nm =

-3.9958

max_torque_oz_in =

-565.8497

max_torque_in_lb =

-35.3654

Ackerman Steering Servomotor Torque Calculations

```
% SteeringSystem_PINSLOT
clear all;
close all;
clc;

% INPUT_____________________________________________

% Servo Crank Dimensions
L0 = 1;

% Steering Arm Dimensions
theta = 115.07;
L1 = .0254*(1.36+.39/2); %TO WHEEL
L2 = .95; %TO LINKAGE

% PIN TO PIN HORIZONTAL DISTANCE
L00 = 5
%Connecting Link Dimensions
L3 = L00-2*L2*abs(cosd(theta));

%Distance between fixed joint and servo center
L4 = 2.31;

%Coefficient of Static Friction Between Wheel & Surface
mu = 0.45;

%Load on each front tire (kg)
m = 0.297;

%Linkage Friction Factor
e = 0.75;
```

```
%Maximum Expected Wheel Turning Angle (INSIDE WHEEL)
%SET TO 360 TO CHECK MAX TURNING ANGLES
thetaLIM = 30;

%Resolution of graph (servomotor degrees)
res = 0.5;

%_____________________________________________
%Torque Required to Turn a Wheel
T = mu*m*9.81*L1; %(N-m)
global r2;
r2 = L0;
global r3;
r3 = L3/2;
global r4;
r4 = L2;
global r5; global theta5;
r5 = L4;
theta5 = 270;
global r6; global thetar6; global thetal6;
r6 = r3 + r4 * sind(theta-90);
thetar6 = 180; thetal6 = 0;

global theta2;
theta2=90;
count = 0;
maxTorque = 0;
table=zeros(2,1);

LeftTurnAngle = 0; RightTurnAngle = 0;
while (abs(LeftTurnAngle) < thetaLIM && abs(RightTurnAngle) < thetaLIM)
   count = count + 1;
fun = @positionSolution_PINSLOT;

x0 = [r5-r4*cosd(theta-90)-r2,0,180-theta,theta];
[x,fval,exitflag] = fsolve(fun,x0);
   if exitflag ~= 1
      SERVO_NON_CONVERGENCE_ANGLE = 90-theta2
      LeftTurnAngle
      RightTurnAngle
      break;
   end
r2s = x(1); theta3 = x(2); thetar4 = x(3); thetal4 = x(4);
LeftTurnAngle = thetal4-theta;
RightTurnAngle =thetar4-(180-theta);
```

```
A = [-sind(theta3) -r2s*cosd(theta3)-r3*sind(theta3) -r4*sind(thetar4) 0;...
     cosd(theta3) -r2s*sind(theta3)+r3*cosd(theta3) +r4*cosd(thetar4) 0;...
    -sind(theta3) -r2s*cosd(theta3)+r3*sind(theta3) 0 -r4*sind(thetal4);...
     cosd(theta3) -r2s*sind(theta3)-r3*cosd(theta3) 0 +r4*cosd(thetal4)];

B = [r2*sind(theta2);
   -r2*cosd(theta2);
    r2*sind(theta2);
   -r2*cosd(theta2)];
fokc = A\B;
thetar4p = fokc(3); thetal4p = fokc(4);
reqTorque = (abs(thetar4p) + abs(thetal4p))* T
if reqTorque > maxTorque
   maxTorque = reqTorque;
end
table(1,count) = 90-theta2;
table(2,count) = reqTorque;

theta2 = theta2-res;
end

plot(table(1,1:end),table(2,1:end))
xlabel('Servo Angle (Deg)');
ylabel('Torque (N-m)');
title('Torque vs Servo Angle'); grid on;

maxTorque_Nm_IDEAL = maxTorque
maxTorque_Nm = maxTorque/e
maxTorque_Kg_cm = maxTorque/e * 10.1971621298
maxTorque_in_Oz = maxTorque/e * 141.611932278
```

FUNCTION FILES:

```
%positionSolution_PINSLOT
function F = positionSolution_PINSLOT (x)

global theta2; global r2; global r3; global r4; global r5; global theta5;
global r6    ; global thetar6; global thetal6;
F(1) = r2*cosd(theta2) + x(1)* cosd(x(2)+90) + r3*cosd(x(2)) + r4*cosd(x(3)) +
r5*cosd(theta5) + r6*cosd(thetar6);

F(2) = r2*sind(theta2) + x(1)* sind(x(2)+90) + r3*sind(x(2)) + r4*sind(x(3)) +
r5*sind(theta5) + r6*sind(thetar6);

F(3) = r2*cosd(theta2) + x(1)* cosd(x(2)+90) - r3*cosd(x(2)) + r4*cosd(x(4)) +
r5*cosd(theta5) + r6*cosd(thetal6);
```

```
    F(4) = r2*sind(theta2) + x(1)* sind(x(2)+90) - r3*sind(x(2)) + r4*sind(x(4)) +
r5*sind(theta5) + r6*sind(thetal6);

    %x(1) = r2s, x(2) = theta3, x(3) = thetar4, x(4) = thetal
```