March 17, 2023

Dr. Mike Hegedus
Department of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6

Re: ENSC 405W Design Specification for KickPro

Dear Professor Hegedus,

Within the attached document, you will find the design specifications for KickPro. KickPro is a soccer training mechanism that will analyze shot patterns and placements, and pass balls at the player - tracking them as they move around the designated area. KickPro is meant to be a retrofit tool that can be added to any net that meets the criteria and also provide training data to the player, such as the placement of the soccer balls as they cross the goal line.

KickPro will perform its functions through a physical ball-launcher system, a player tracking system, and a feedback system that records and analyzes the position of the ball as it crosses the goal line. These will be bundled up into a Graphical User Interface (GUI) designed for simplicity and ease-of-use.

IronFoot Technologies will outline our designs for the hardware and software systems in the document. The hardware specifications have been broken up into their mechanical and electric components, and the various software programs as well their external libraries are described.

Our team is comprised of Jagpreet Grewal, who is in the computer discipline; Amirali Farzaneh, Tao Li, and Minh Phat (Henry) Tran, who are in the systems discipline; and Alon Singh and Zehui (Jeffrey) Lin, who are in the electronics discipline.

Thank you for considering our proposal. We look forward to hearing from you soon. Should you have any questions, comments or concerns, please reach out to our Chief Communications Officer, Jeffrey at zla167@sfu.ca.

Sincerely,

Alon Singh
Chief Executive Officer
IronFoot Technologies

ENSC 405W
DESIGN SPECIFICATION: REVISION A
KICKPRO

A DESIGN SPECIFICATION DOCUMENT BY IronFoot Technologies

WRITTEN BY

ALON SINGH	CHIEF EXECUTIVE OFFICER	(CEO)
AMIRALI FARZANEH	CHIEF BRAND OFFICER	(CBO)
JAGPREET GREWAL	CHIEF FINANCIAL OFFICER	(CFO)
MINH PHAT TRAN	CHIEF TECHNICAL OFFICER	(CTO)
ZEHUI (JEFFREY) LIN	CHIEF COMMUNICATIONS OFFICER	(CCO)
TAO LI	CHIEF LEGAL OFFICER	(CLO)

Abstract

This design specification document of KickPro will cover all proposed design schemes that help training soccer players by launching the ball to the player's position automatically to reduce waiting time, promote the training effects as well as provide feedback to players. This design specification document covers all mechanical, electronic, firmware and software designs for KickPro. The ball launcher system is operated mechanically by servo, two motors with two flywheels. The tracking system is designed to track the player presented in the webcam's frame with the assistance from the real-time computer vision library OpenCV. Training feedback is provided through KickPro software graphical user interface based on the ball's position. Overall, a single-board computer Raspberry Pi 4 will be used to control the whole KickPro's system with servo's rotation angle and human tracking. Each of the design specifications within test plans will be discussed in depth in this document.

Table of Contents

List of Figures

List of Tables

Glossary

Term	Definition
AC	Alternating Current
CSA	Canadian Standards Association
DC	Direct Current
EMC	Electromagnetic Compatibility
GUI	Graphic User Interface
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
PCB	Printed Circuit Board
PC	Personal Computer
PWM	Pulse Width Modulation

TABLE I: Definition of Glossary Terms

1 Introduction

KickPro is a revolutionary tool that is set to elevate soccer training to unparalleled heights. Designed to retrofit any compatible net, this smart device harnesses the power of a launcher that can accommodate multiple balls at once (in the prototype phase). Not only does it swivel automatically per the player's position, but it also integrates seamlessly with a cutting-edge tracking system. This subsystem employs a sophisticated optical program to precisely calculate the ball's angle and velocity, ensuring that it is returned to the player with reasonable accuracy. And that's not all – KickPro also boasts a state-of-the-art training analyzer module that delivers real-time data on the player's positioning, shot placement, and progress, making every training session an immersive and comprehensive learning experience.

This design specification document discusses the technical aspects of the design requirements that must be met to make this project possible and meet the functionality outlined in the requirements document for the proof-of-concept.

1.1 Background

Automatic ball launchers are commonly used across various sports, including soccer, baseball, basketball, and lacrosse. However, these products fall short in addressing a critical issue in player training - the time-consuming process of manually adjusting the angle and speed of the launcher. Additionally, these launchers do not have the capability to accurately pass balls to moving targets, requiring additional player or staff involvement.

KickPro revolutionizes this process by automating this mundane task, allowing players and coaches to focus on the essential training and hard work that underpins the game. Moreover, KickPro provides comprehensive analysis and feedback throughout each session, enhancing the training experience even further.

The main challenges we expect to face are constructing the ball launcher to a reliable standard and achieving sufficient accuracy in tracking moving targets - as a Raspberry Pi is intended to serve as the main computing device for this system, processing power is a concern.. IronFoot Technologies takes pride in its products and as such, we wish to construct our device so that it works as intended every time. As might be expected, achieving this level of sophistication and integration can be difficult.

1.2 Scope

The development of KickPro will occur in three distinct phases: proof-of-concept, engineering prototype, and production version, each with varying levels of design specifications. The proof-of-concept stage will have the most lenient requirements, while the production version will have the most rigorous validation standards, encompassing general requirements, subsystem requirements, safety, sustainability, and legality.

Readers should be aware this document builds upon the requirements document from last month. This document also includes the proof-of-concept test plan and design alternatives in the appendices.

1.3 Design Classification

The design specifications in this document will use the following system:

D.X.Y.Z V

The definitions for the encoding are as such:

Encoding	Definition
X	Section
Y	Subsection
Z	Requirement Number
V	Product Stage

TABLE 1.4.1: Definition of Convention

The versions for the product stages are abbreviated as follows:

Version	Product Stage
A	Proof-of-concept
B	Engineering Prototype
C	Production Version

TABLE 1.4.2: Abbreviation of Product Stages

2 System Overview

KickPro consists of hardware and software systems that collaborate inside a pipeline to provide the desired outputs. The system starts with the push of a button that provides power to all of its components. When the system boots, the Graphical User Interface is displayed on KickPros touchscreen. A start screen is initialized, where the user can begin their training session through a slide and unlock mechanism. After swiping, the user will be provided with a human tracking calibration step, where the tracking system learns and detects the user through a specific physical attribute such as the color of their shirt. Once calibration is complete, the user will be directed to the main dashboard of their training session, where they can alter launch settings and visualize the position of their scored goals on the net in real time. Lastly, KickPro will track and record key performance metrics and will output them in the form of a graph or prompts to the user through the “Performance Analysis” page of the GUI.

During a training session, KickPro will continuously track the user as they move around the field. When the user is ready, KickPro will use a flashing light to indicate when the ball is going to be launched. The system will then calculate the distance between the user and itself, and will speed up its flywheel motors to launch the ball at the proper velocity. The player will then shoot the ball and if a goal is scored, the sensing mechanism installed on the net will measure the ball location in the net. The sensing mechanism is implemented through a camera that captures a photo of the ball as it enters the net, a motion sensor to sense the ball and trigger the camera, and software to measure the location of the ball.

The system contains one Raspberry Pi 4 Model B (8GB RAM) that controls and processes each crucial step in providing the correct outputs to the user. This microcontroller is used in processing the human tracking software during calibration, the flywheel motor speeds, sensors on the net, and all control circuits involved. A simple flowchart of the overall system is provided below in figure 1 to help the reader with a better understanding of the device’s processing pipeline.

2.1 Overall System FlowChart

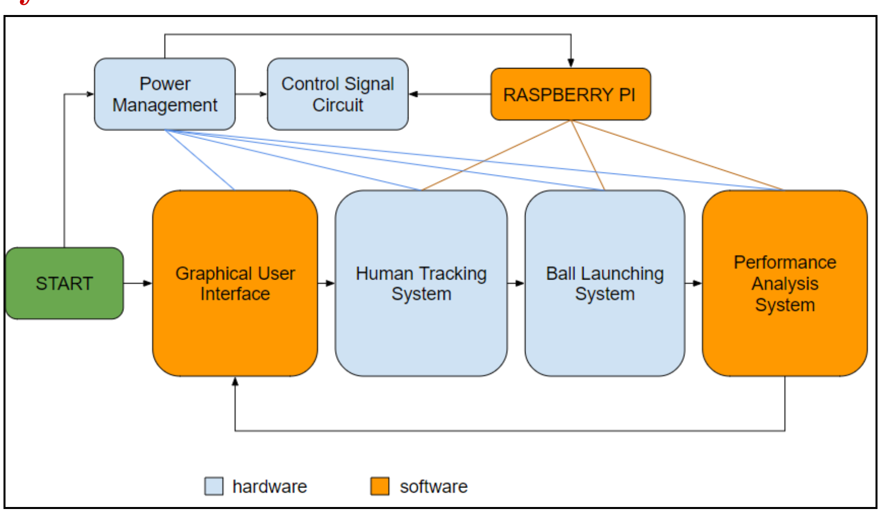

Figure 1: Overall flow chart of the system

3 Hardware Design Specifications

In this section, we will be providing detailed descriptions of KickPro's hardware design specifications. This section will be primarily split into two parts: the mechanical design and the electrical design. The mechanical design refers to the structural design of both the human tracking system with the servo motor alongside the 'Lazy Susan' bearing ring and the ball launching system with the brushed motors, flywheels and ball ramp as well as the overall aesthetic of the project. The electrical design refers to the power management system, which includes the power supply, regulators, fuses and its corresponding connections as well as the control signal system, which are relays and signal LEDs that interface with the Raspberry Pi. We will be referencing 03reqs.pdf to identify the corresponding requirements.

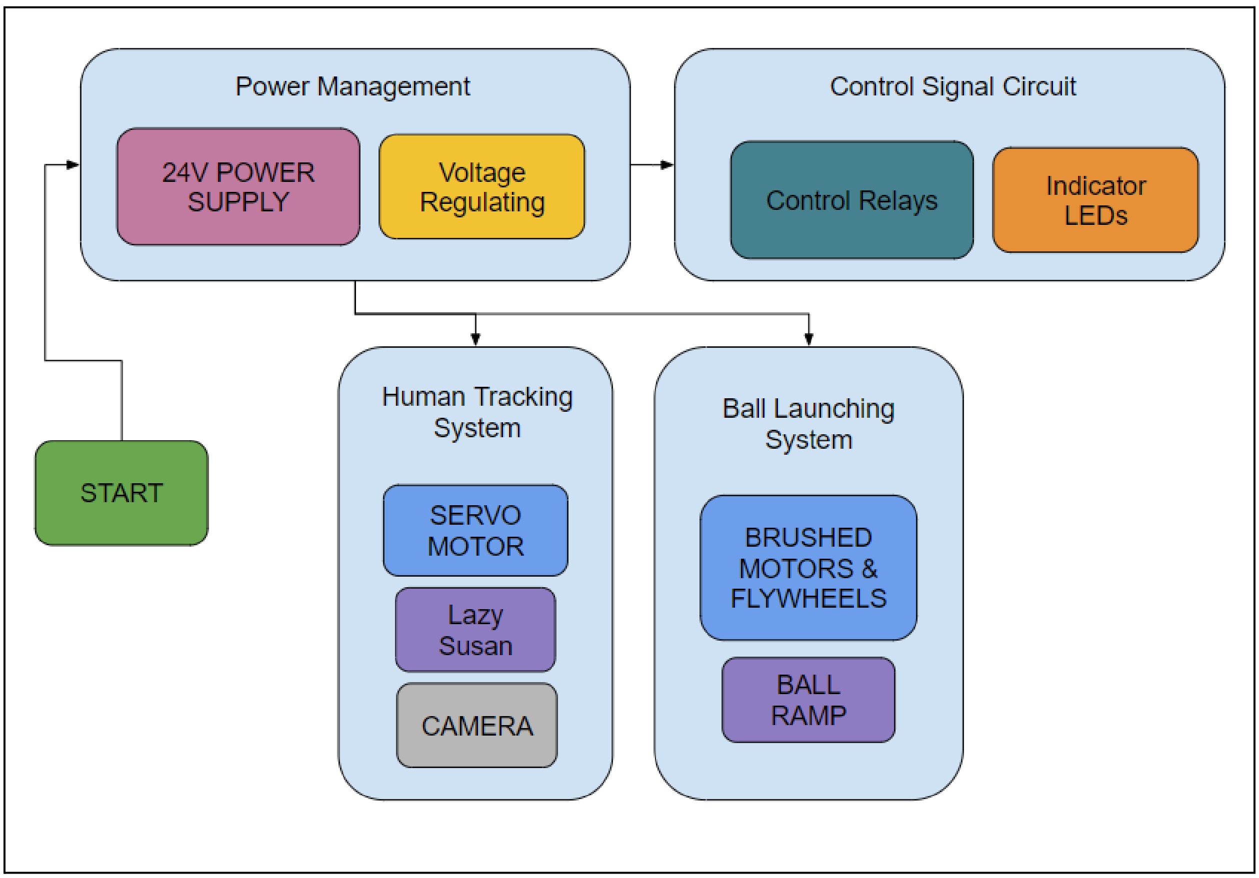

Figure 2: Hardware design flow chart

3.1 Mechanical Design Specifications

Design Specification ID	Description	Expected Changes for Future Designs	Related Requirement Specifications
D.3.1. A	Eight lockable caster wheels are installed at the bottom of the base to facilitate the portability.	Numbers and exact placement of the wheels might change over iterations of design due to possible change in center of gravity and weight.	R.4.1.1 A R.4.2.5 A
D.3.1. A	Fast moving and sharp mechanical parts are enclosed.	None.	R.6.0.1 A
D.3.1. A	Electronics circuits and connections are properly insulated using shrink wraps, enclosures, insulating connectors etc.	Types of connectors for power management might change due to space planning.	R.4.3.1 A R.4.3.2 A R.4.3.3 A R.4.3.4 A R.4.3.8 C R.6.0.2 A
D.3.1. A	A “Lazy Susan” is implemented to improve the fluidity of the rotation movement.	Choice of material for the turntable on top of the bearings is yet to be determined	R.4.2.3 A R.4.2.4 A

Table 3.1: Mechanical design specifications ID’s

3.1.1 Human Tracking System

The human tracking system of KickPro is mechanically controlled by an ANNIMOS 60KG Stainless Steel Digital Servo, which then is internally controlled by the PWM signal generated from the Raspberry Pi (logic explained in depth in software design specification section below). The parameters of the PWM signals are determined by the webcam positioned at the top of the project, where the Pi runs an object-detection algorithm to track the position of the user (details also explained in the software design section below).

To further facilitate the fluidity of rotation, a ‘Lazy Susan’ is installed to support from the bottom of the rotated system. The ‘Lazy Susan’ is essentially a ring of bearing encased in metal that distributes the weight evenly throughout the surface area of the rotated system.

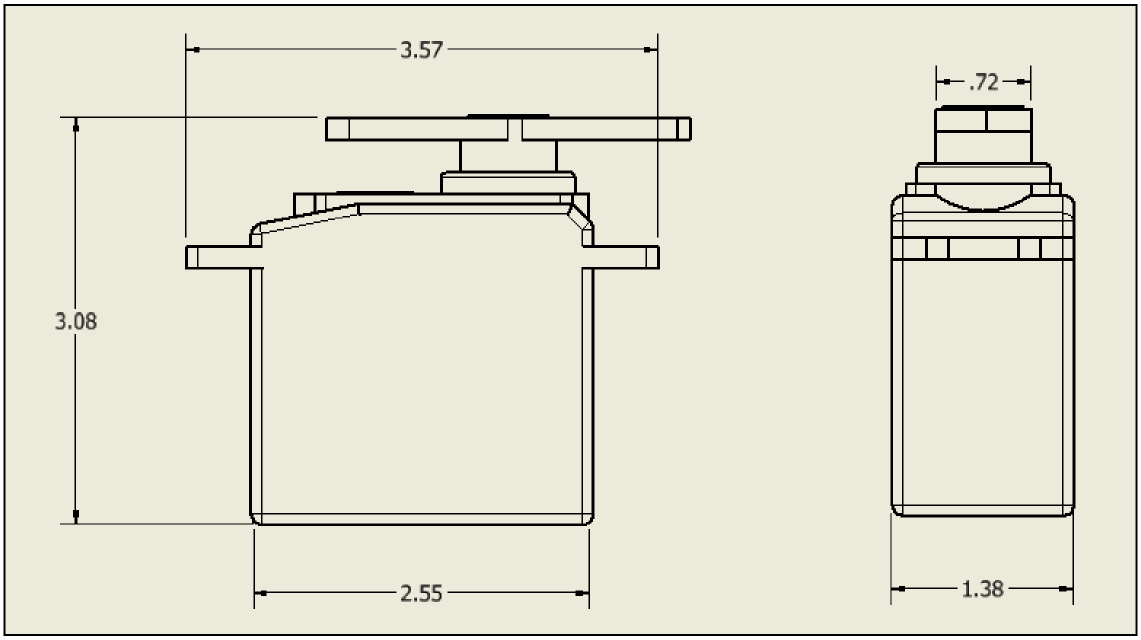

Figure 3: Rough dimensions of the ANNIMOS 60 KG servo motor

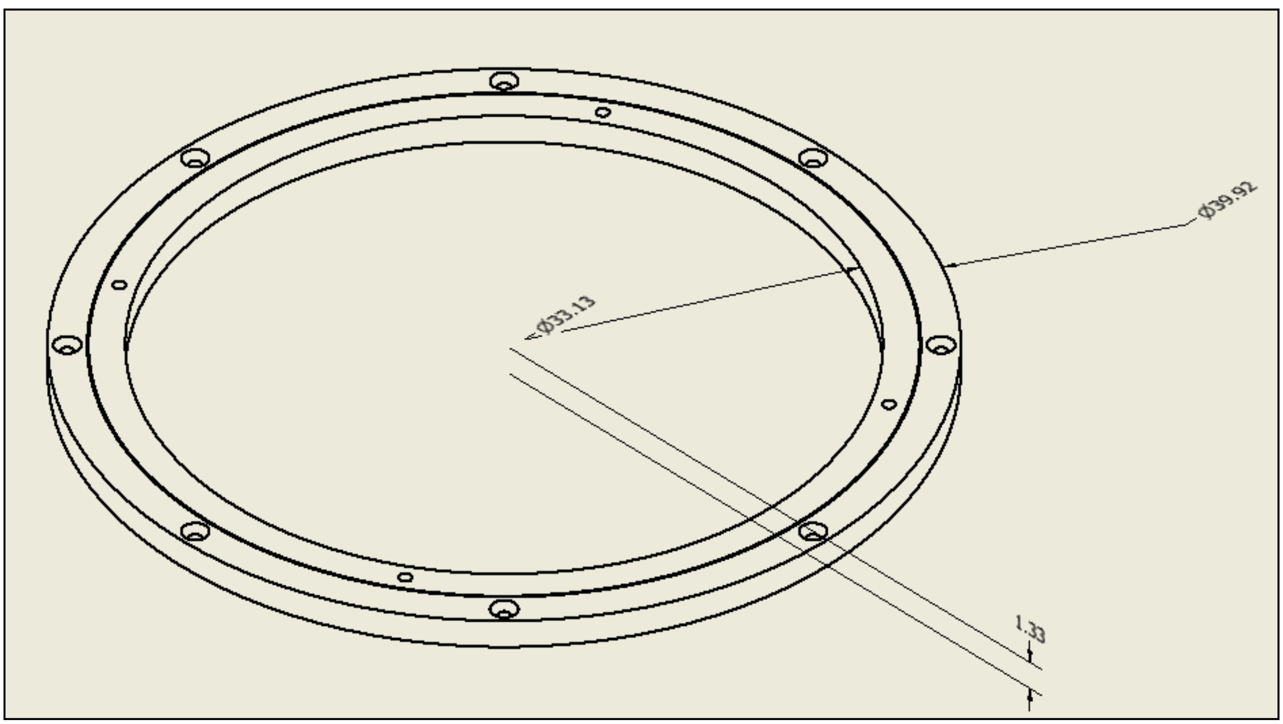

Figure 4: Rough dimensions of the 40-in "Lazy Susan" turntable bearings

3.1.2 Ball Launching System

The ball launching system of KickPro is mechanically controlled by two Yaetek 24V 250W Brushed Motors that are capable of producing an RPM of 2650 [1]. Two 10-inch dolly hand truck wheels are attached to the brushed motors via a customized axle. Friction created by the rubber material of the wheels while spinning produces the launching force for the soccer balls. In order to create the momentum that the balls need before contacting the launching mechanism, a plastic ramp is positioned slightly at the back of the motors for the balls to roll down. As shown in the figure below, the ramp is designed to be approximately at a height of 20 inches (0.508m) and at an angle of 45 degrees.

Below are some ideal calculations performed to validate the motor selection and the general mechanical design:

Soccer Ball Parameters:

Mass: 0.45kg

Diameter: 0.226m

Radius: 0.113m

Inertia of the Ball:

$$I_{ball} = (\frac{2}{3} \cdot m \cdot r^2)_{ball} = 0.00383\, kg\ * m^2$$

For these ideal calculations, we have decided to omit the friction of the ball rolling down the ramp.

We use the energy conservation formula to estimate the maximum velocity that the soccer ball is able to gain prior to coming into contact with the flywheels :

$PE_O = KE_T + KE_R$, where PEo is the initial potential energy, KET is the translational kinetic energy and KER is the rotational kinetic energy.

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2,$$

where $\omega = \frac{v}{R}$ is the angular velocity

$$(0.46kg) * (9.8m/s^2) * (0.508m) = \frac{1}{2}(0.46kg)v^2 + \frac{1}{2}(0.00383kg * m^2)\frac{v^2}{(0.113m)^2}$$

$$v \approx 2.45\, m/s$$

We know that a typical soccer pass is approximately at a speed of 20.5 m/s, therefore the flywheels must be able to achieve $20.5 - 2.45 = 18.05m/s$ of translational velocity under ideal operations.

Flywheel Parameters:

Mass: 3.06kg

Radius: 0.127m

Gear Ratio: 1:1 (directly interfaced with the motors)

Inertia of the flywheels:

$$I_{flywheel} = (\frac{1}{2} \cdot m \cdot r^2)_{flywheels} = \frac{1}{2}(3.06)(0.127)^2 = 0.02467737\, kg * m^2$$

Assume that it takes up to 0.5s for the motors to reach maximum angular velocity: t = 0.5s

$$Torque = I_{flywheel} \cdot \frac{\omega_{max}}{t}$$

Where we know that:

$$\omega_{max} = \frac{18.05}{0.127} = 142.125rad/s$$

So, our torque comes to:

$$Torque = (0.02467737kg * m^2) * \frac{142.125}{0.5} = 7.0146\, Nm$$

We know that:

$$1\, rad/s = 9.54929RPM$$

Therefore the minimum RPM needed is:

$$142.125 \cdot 9.54929 = 1357\, RPM$$

This means that our dual-motor system is entirely capable of producing a typical pass velocity. Note that depending on the player's distance from the launcher, the brushed motor speed can also be adjusted using PWM signal from the Pi.

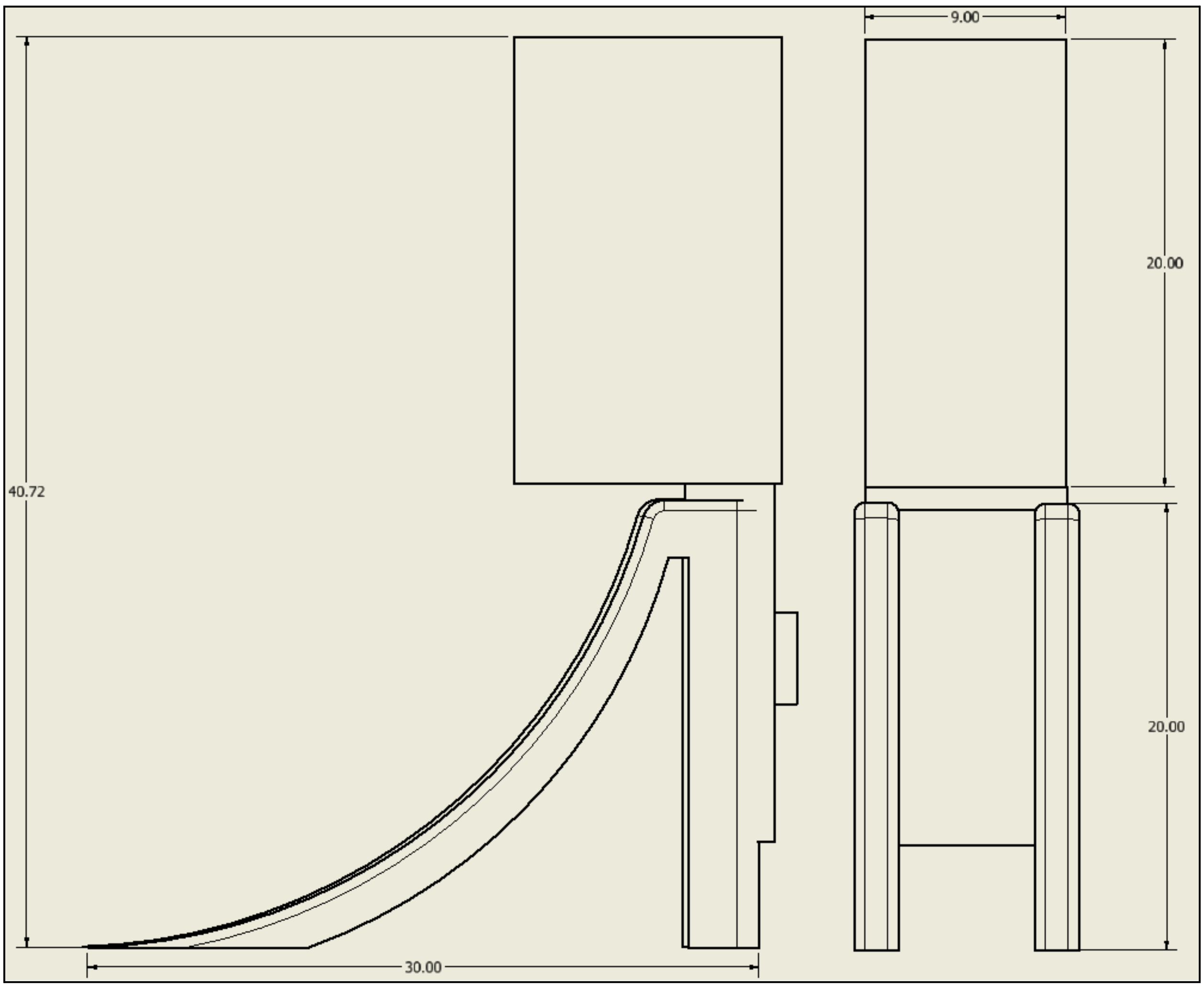

Figure 5: Rough dimensions of the ball ramp rails with the ball container mounted on top of it

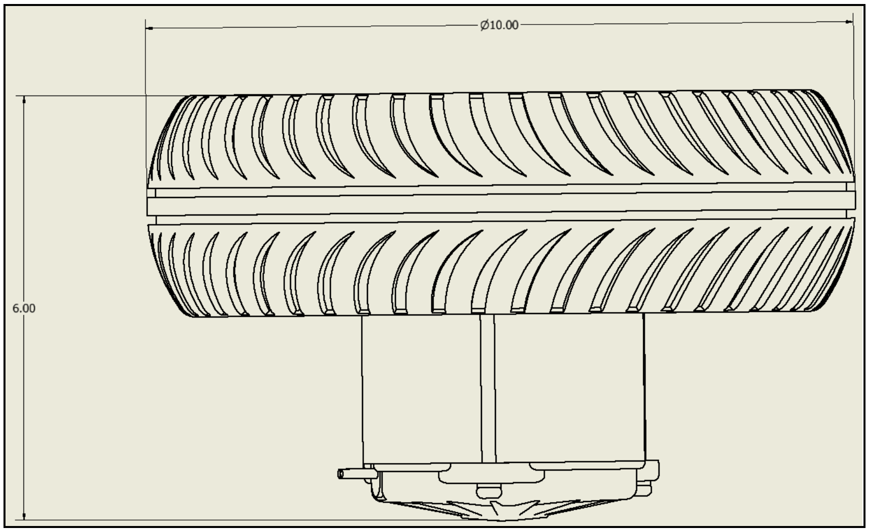

Figure 6: Rough dimensions of the brushed motor-flywheel interface

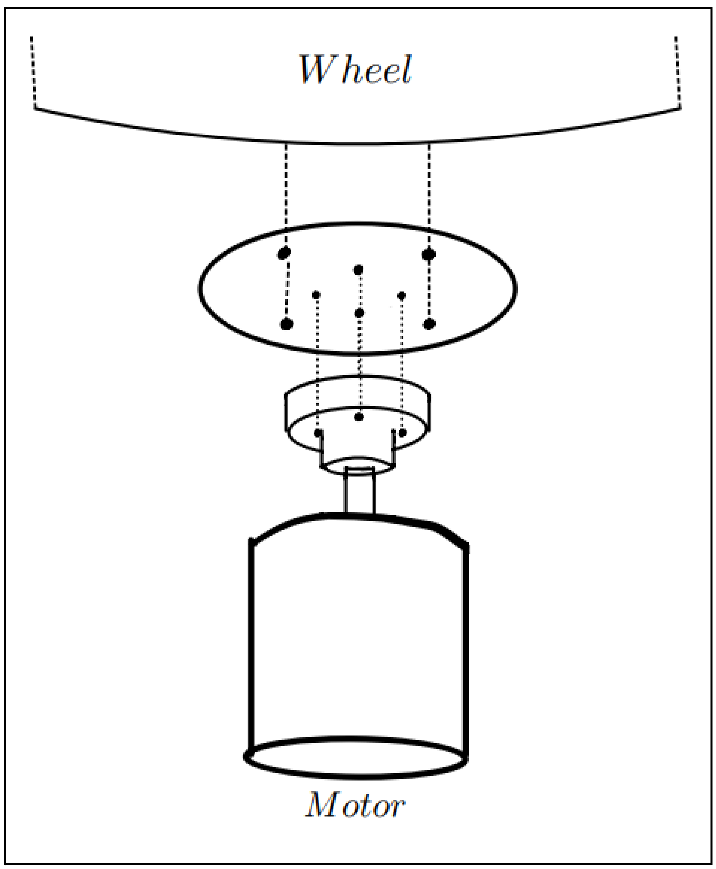

Figure 7: Mounting method for the brushed motor-flywheel interface.

3.1.3 Overall

The rectangular base of KickPro is constructed using wood for its structural integrity. Eight of the lockable caster wheels are installed at the bottom of the base to enable users to move the product around easily while still being able to be secured to the ground when in operation. Sharp edges of the wood structure are filtered out to optimize user's safety as well as maintaining an adequate aesthetic. Fast moving parts such as the brushed motors and wheels are properly enclosed to ensure there are no contacts with body parts.

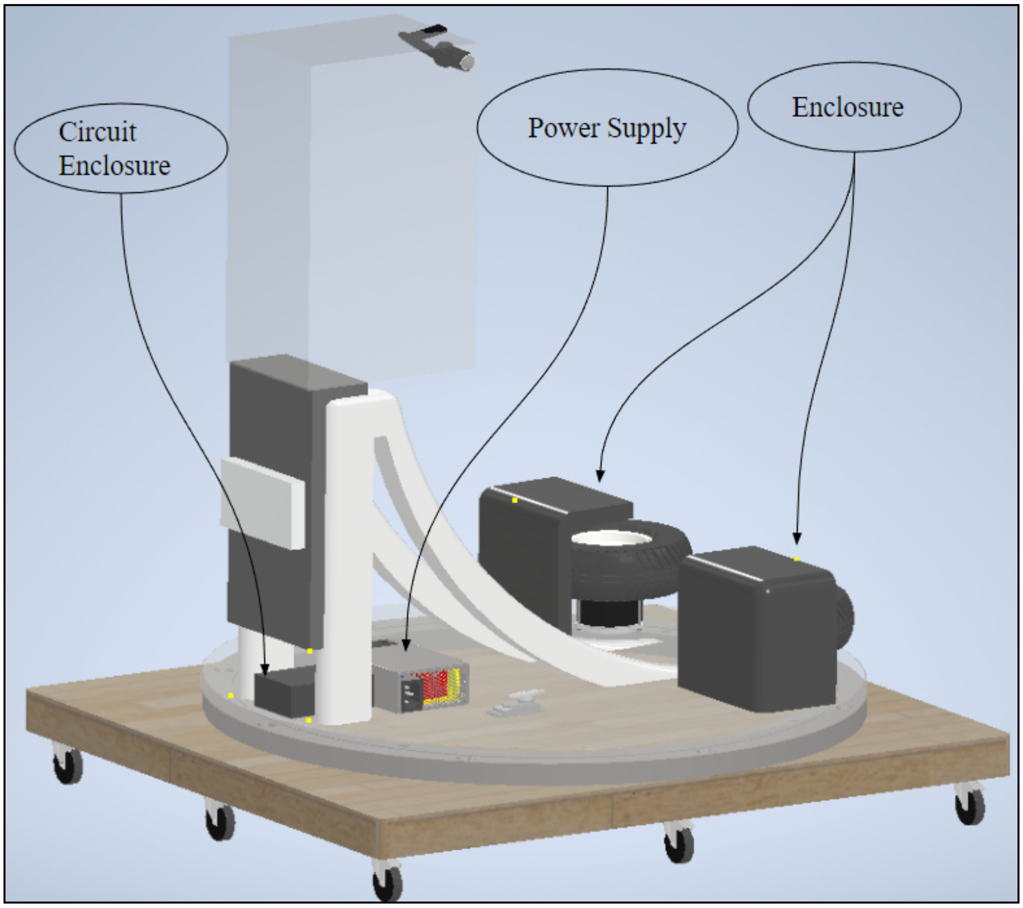

Figure 8: Overall aesthetics of the system

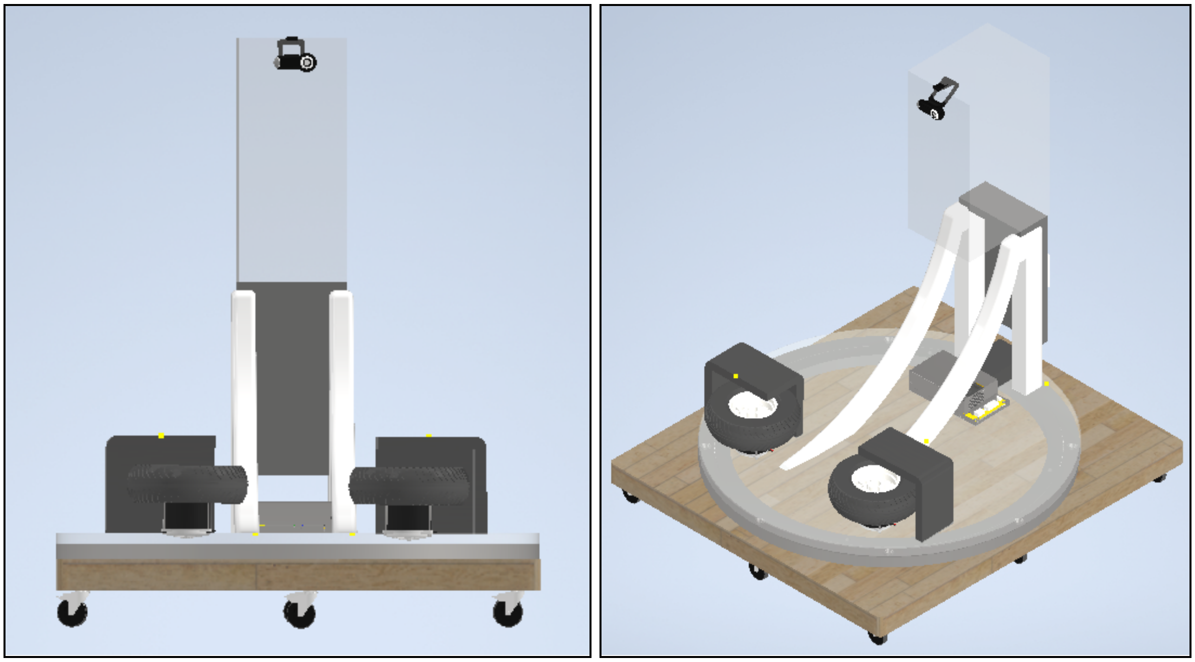

Figure 9: Overall aesthetics of the system from various angles

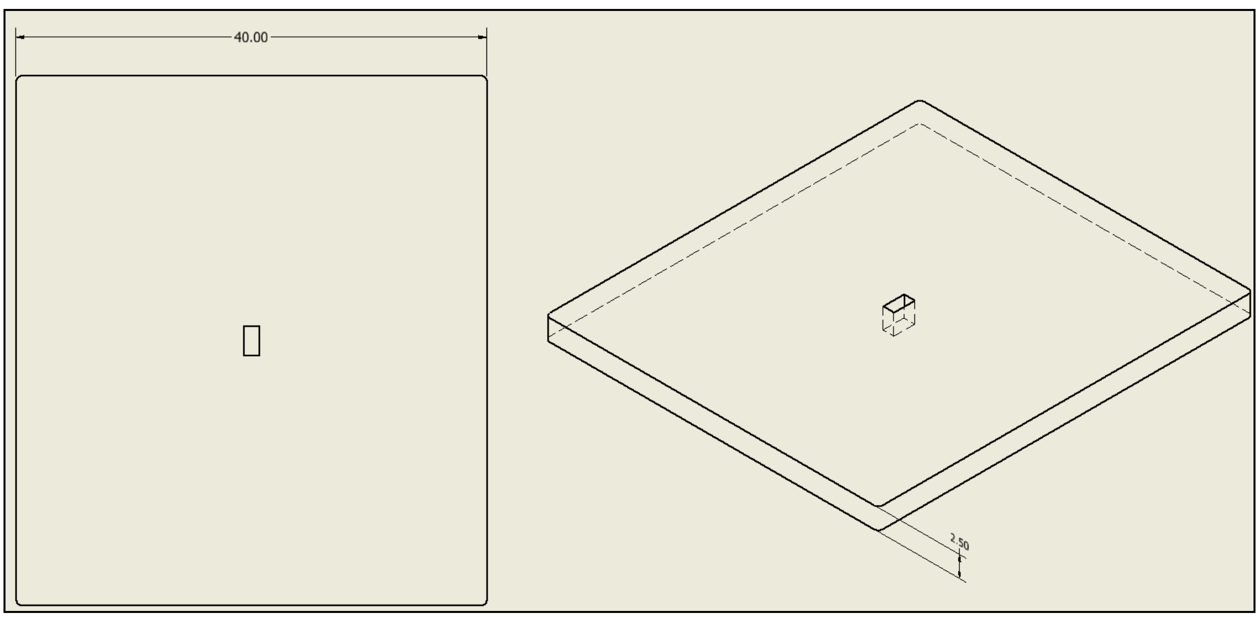

Figure 10: Rough dimensions of the wooden base plank

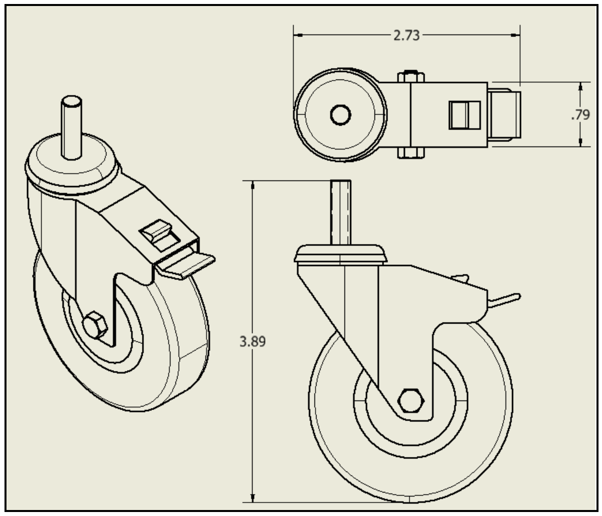

Figure 11: Rough dimensions of the lockable caster wheels

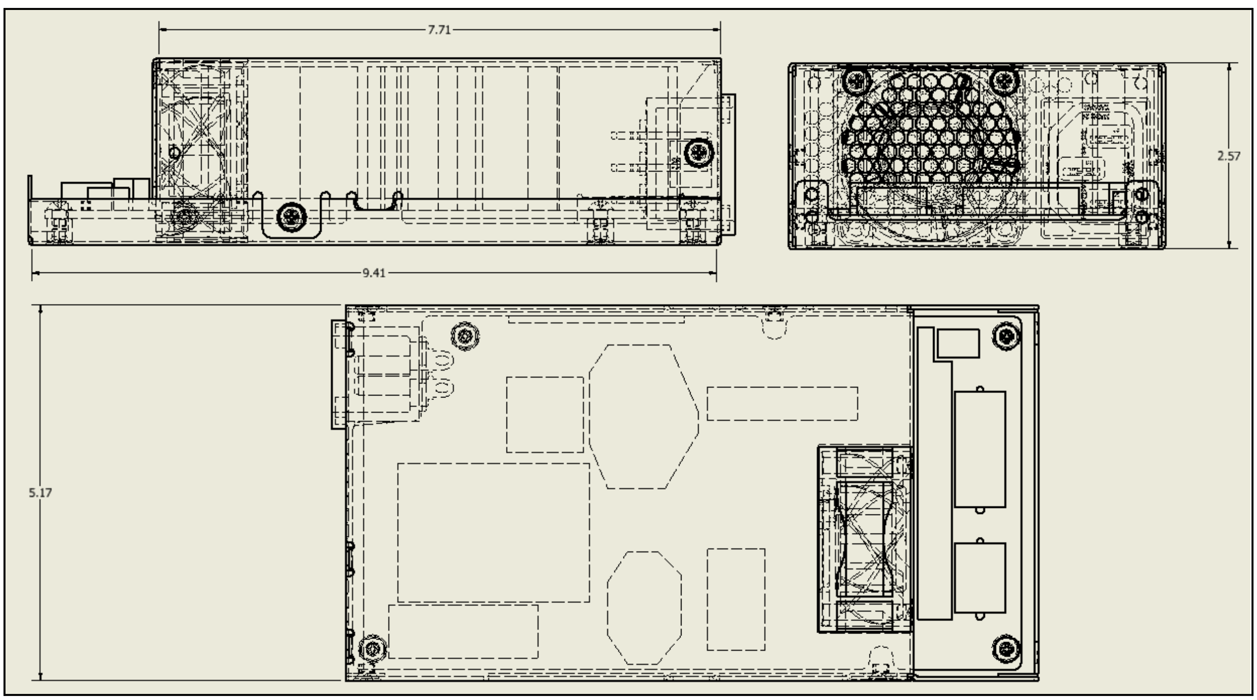

Figure 12: Dimensions of the 24V 40A power supply

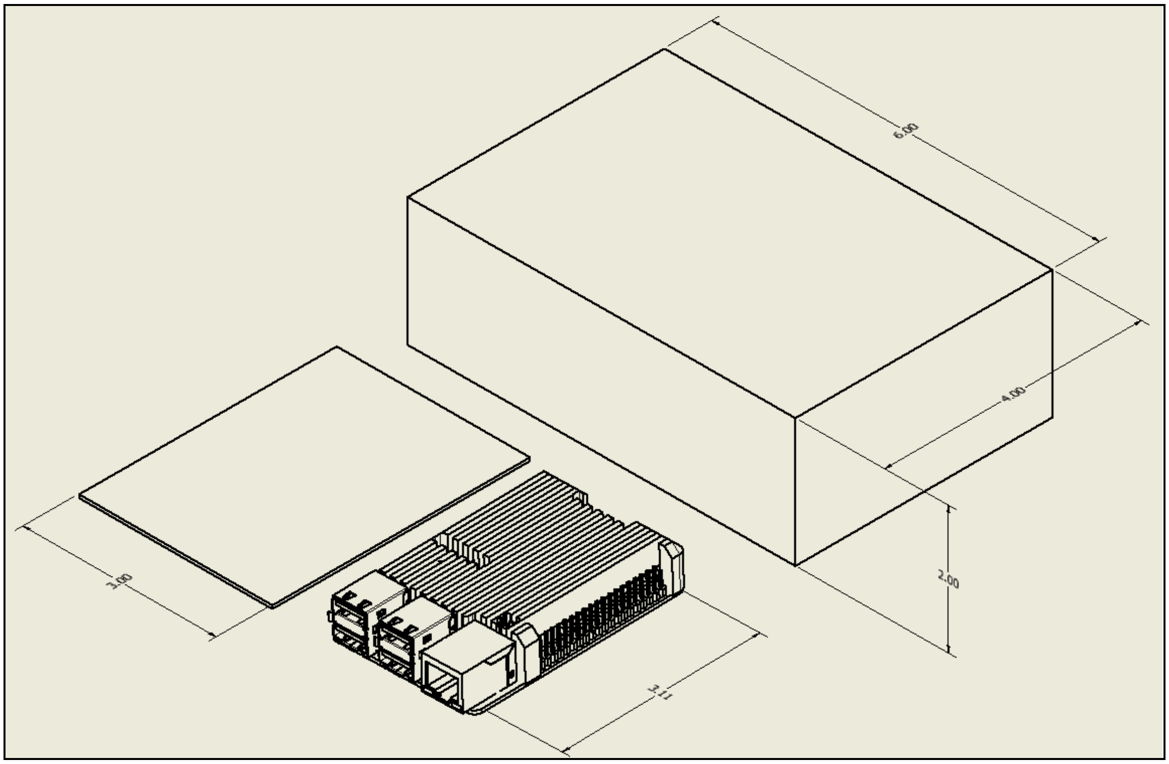

Figure 13: Rough dimensions of the electronic circuit enclosure for the Pi and an unstuffed power management/control signal PCB

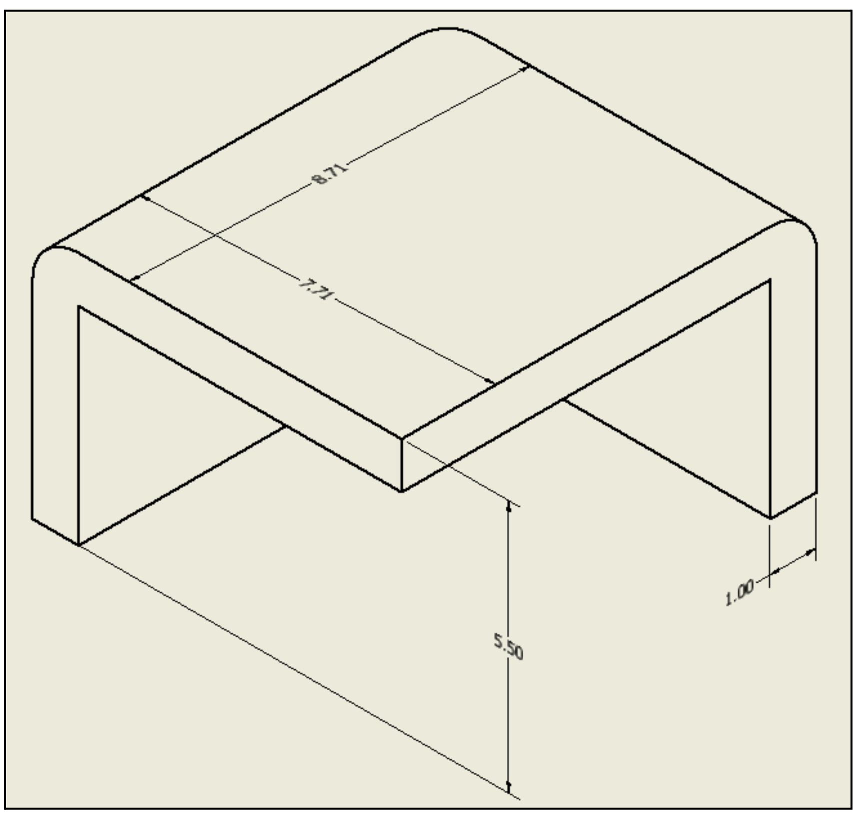

Figure 14: Rough dimensions of the flywheel protection enclosures

3.2 Electrical Design Specifications

Design Specification ID	Description	Expected Changes for Future Designs	Related Requirement Specifications
D.3.2.1 A	A large and well insulated power supply supplying sufficient current will provide power to the entire system.	None.	R.3.3.1 A R.4.1.1 A R.4.3.3 A R.4.3.4 A R.4.3.5 A R.4.3.6 B R.6.0.2 A
D.3.2.2 A	The components requiring a heavy electrical load will be protected using fuses.	None.	R.3.3.1 A R.4.3.1 A
D.3.2.3 A	The voltages available to the system will be 24, 8, and 5 volts.	If new parts/features are introduced that require a different voltage, a new circuit will have to be designed.	R.3.3.1 A R.4.3.2 A R.4.3.4 A R.4.3.5 A
D.3.2.4 A	Power connections will be made on covered screw terminal strips.	None.	R.3.3.1 A R.4.3.3 A R.6.0.2 A
D.3.2.5 A	Signal connections will be made on a protoboard.	None.	R.3.3.1 A R.4.3.3 A R.6.0.2 A
D.3.2.6 B	Signal connections will be made on a custom PCB.	None.	R.3.3.1 A R.4.3.3 A R.6.0.2 A

Table 3.2: Electrical design specification ID's

3.2.1 Power Circuit

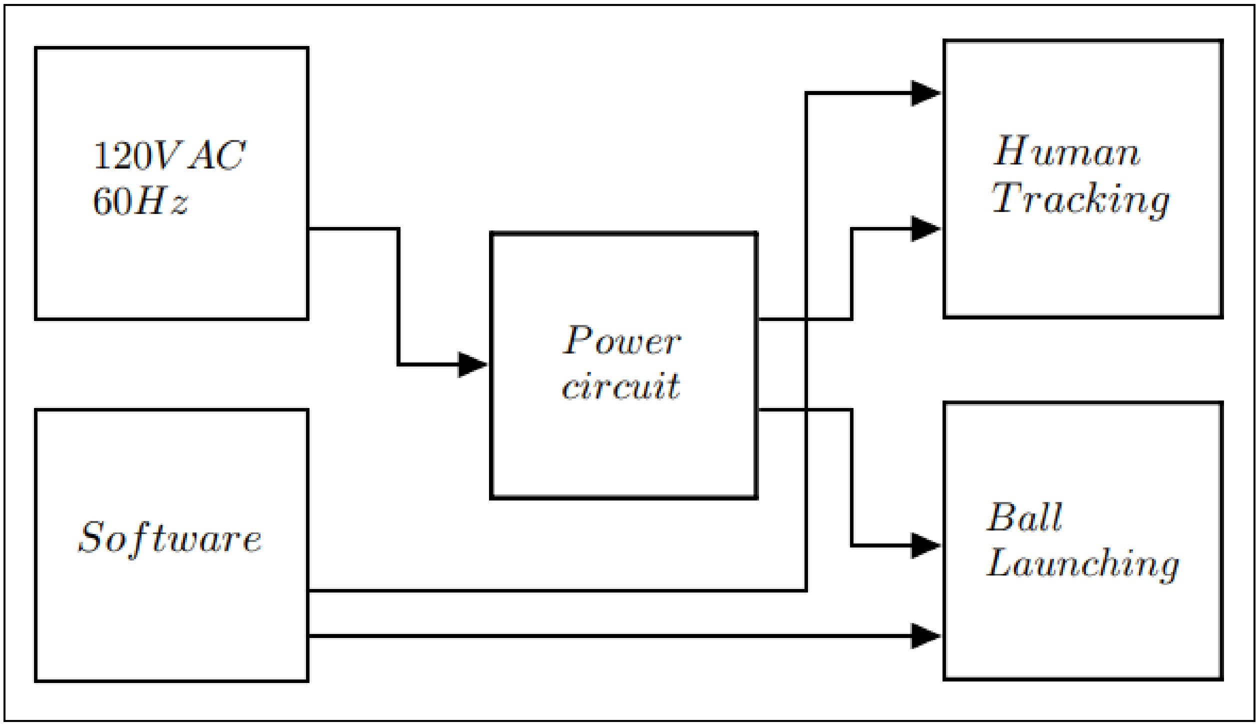

Figure 15: Overall electrical system block diagram

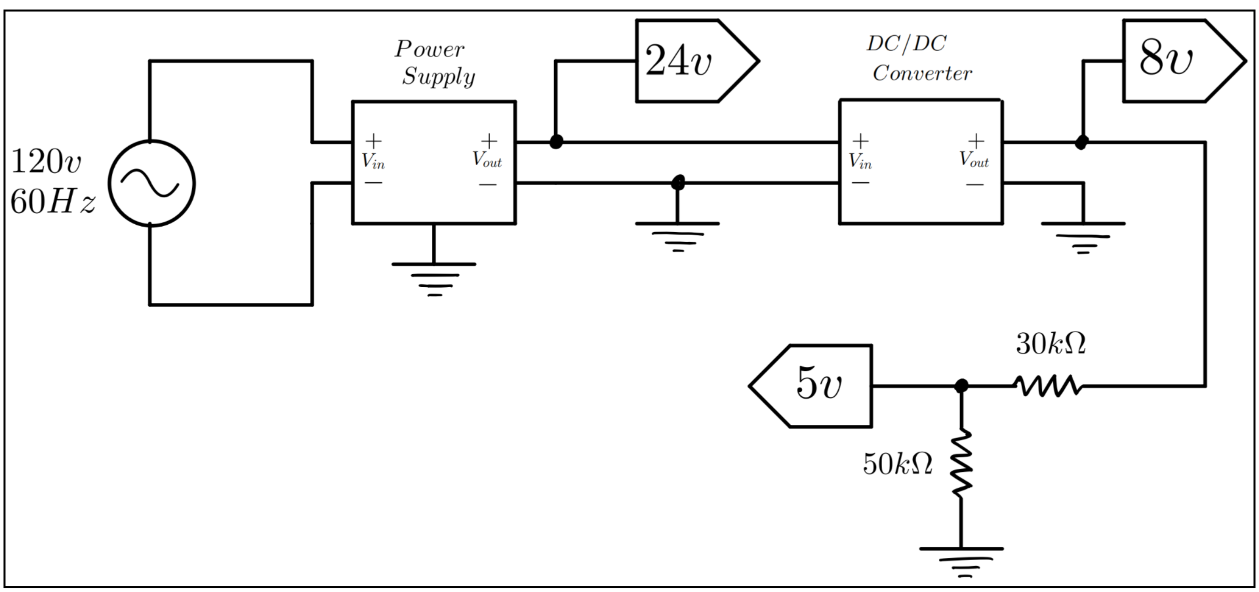

Figure 16: Power circuit diagram

The power circuit design of the KickPro will utilize the regularly available power of 120V AC at 60 Hz and using the power supply, convert it to 24V DC. The power supply is rated for 40 Amps which was chosen using the current specifications of each device.

$$I_{Motor} = 14\ A$$

$$I_{Servo} = 6\ A$$

$$I_{Raspberry\ Pi} = 3\ A$$
$$I_{Camera} = 0.5\ A$$

Therefore, we have the max current draw as:

$$I_{Max} = 2(I_{Motor}) + I_{Servo} + I_{Raspberry\ Pi} + 2(I_{Camera}) = 28 + 6 + 3 + 1 = 38\ A$$

These calculations are assuming the motors and the servo are at their maximum load. Which, unless a ball is stuck in the system at the same time that something is causing the rotating assembly to halt, is highly unlikely to happen.

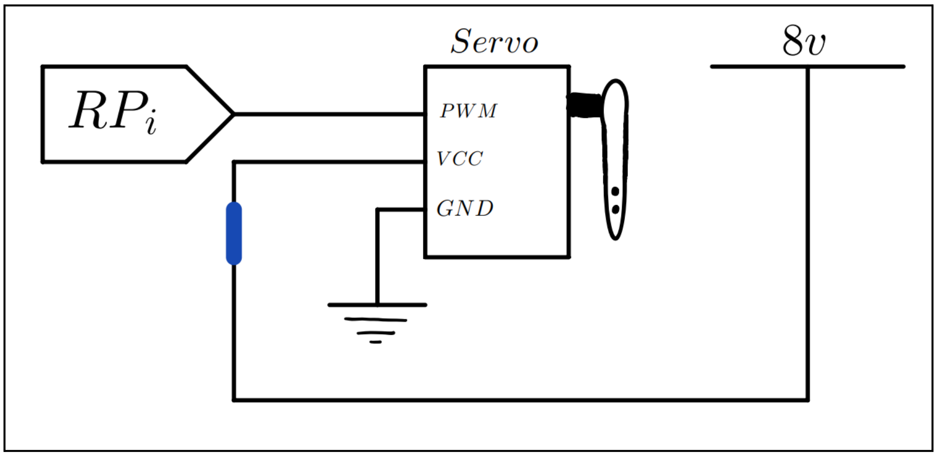

Figure 17: Servo wiring diagram

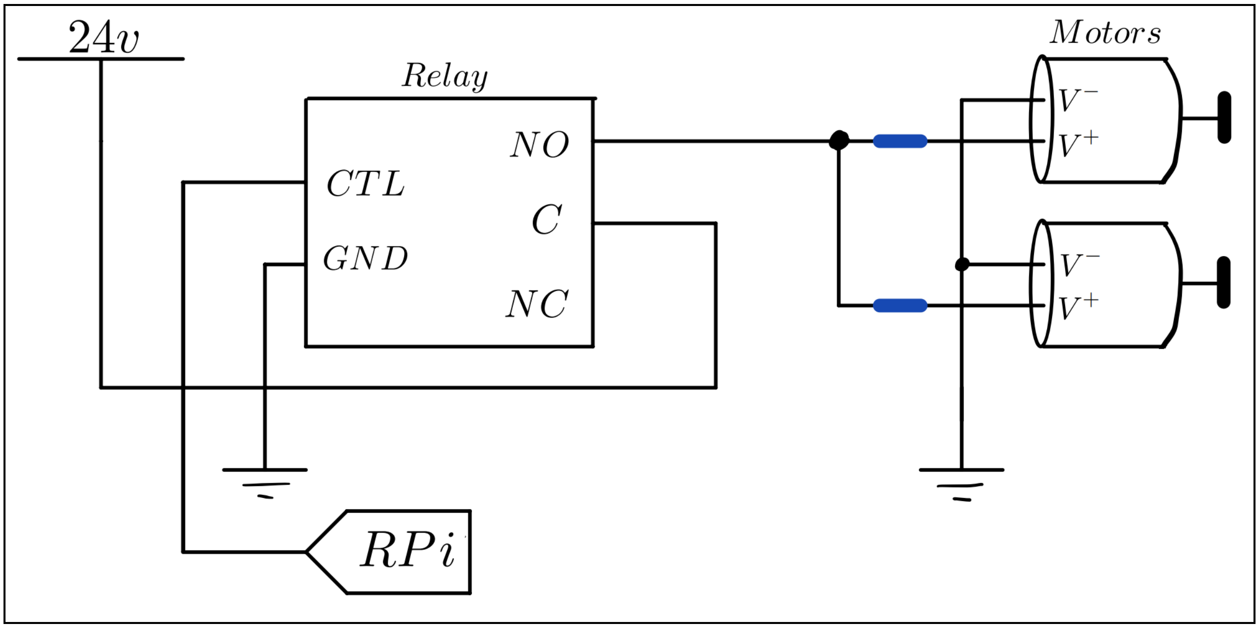

Figure 18: Motor wiring diagram

The power circuit will also include (at each of the thick blue marks in the circuit diagrams of figure 17 and figure 18) fuses inline with the high risk components. The motors run freely (without load) at approximately 7.5 amps. Only under extreme load (ie. the motors will not spin) will they draw the full 14 amps. Therefore, each motor is wired inline with a 10 amp fuse that can be easily replaced if blown during an operation failure. Similarly, the servo will contain a 6 amp fuse that will protect the component from complete failure in the event that the rotating assembly is lodged or stuck on any obstacles.

The power connections will be made on a covered dual-row terminal strip, much like the one shown in figure 19, to ensure that if any wiring connections fail from a soccer ball impact, it will not worsen the damage by short circuiting any high voltage connections rendering any components completely unusable.

Figure 19: Dual-rail barrier strip [2]

3.2.2 Signal Circuit

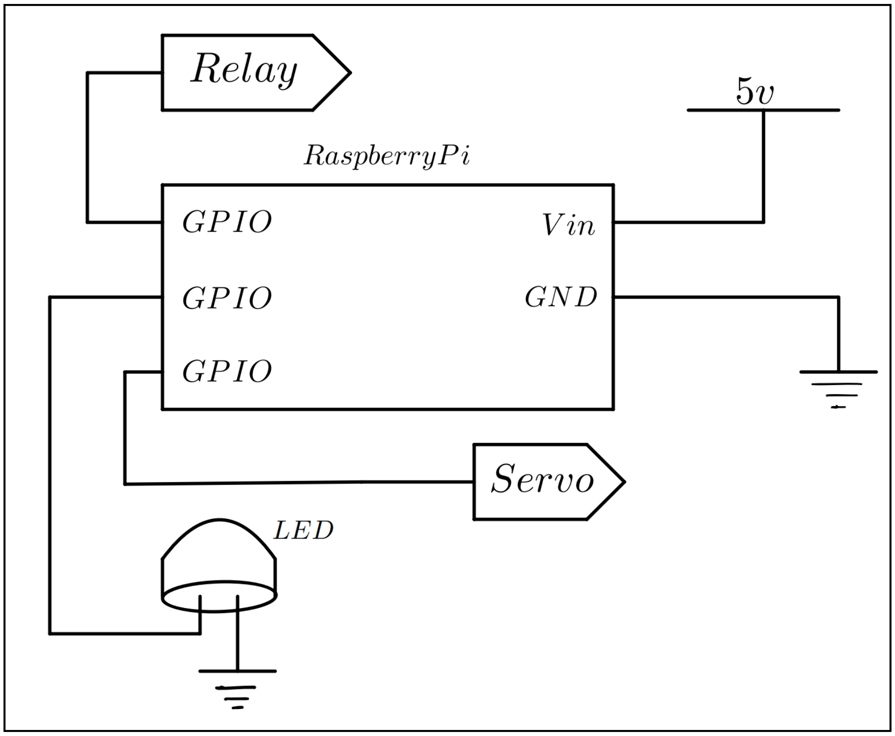

Figure 20: Signal circuit diagram

The relay used to provide power to the motors will have its control signal wired to the Raspberry Pi. Shown in figure 18, the motors will be designed to have an open circuit until triggered, known as Normally Open (NO). The control signal will cause the movable contact arm to make a connection between the common node of 24 volts(C) and the NO node. Also, as described in the section 4.1 of this document, the Raspberry Pi will control the servo via a PWM signal.

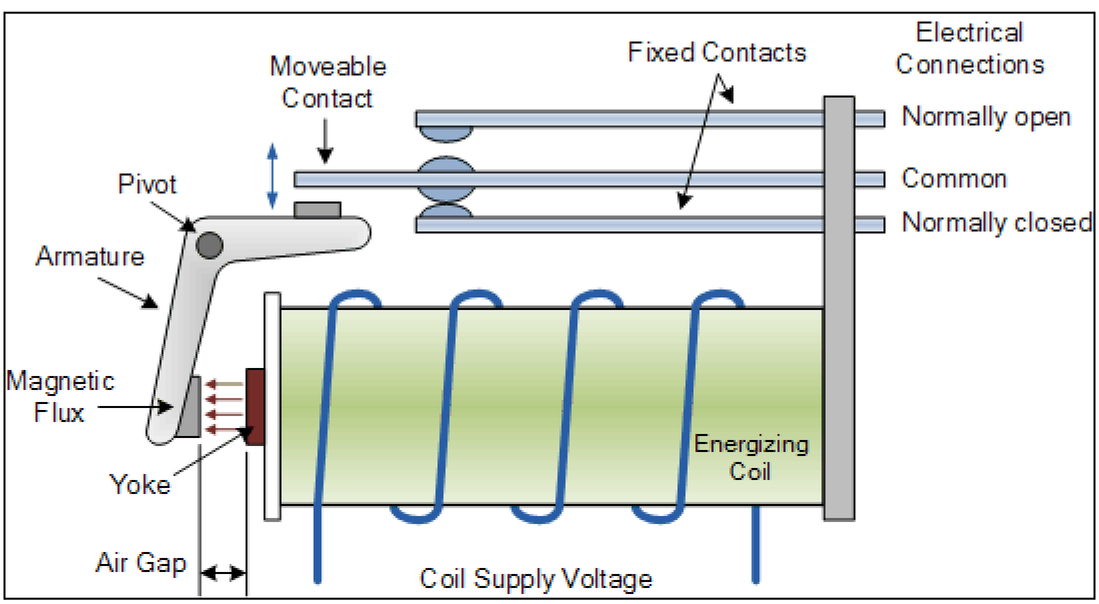

Figure 21: Diagram of how a relay works [3]

4 Software Design Specifications

The KickPro firmware and software systems are designed to satisfy the design requirements described below. The system has two main software components: the human tracking algorithm guiding the rotation of the servo, and the GUI. The former component, as per its name, tracks the user with a camera and assistance from various Python libraries such as OpenCV. It cross-examines the location of the player with the angle of the camera, then transmits the necessary adjustments to the part of the program controlling the servo. The latter component provides the user with a simple and intuitive GUI, allowing them to access the feedback provided by the system - such as the approximate position of the ball as it crosses the goal line.

4.1 ANNIMOS 60 KG Servo

Design Specification ID	Description	Expected Changes for Future Designs	Related Requirement Specifications
D.4.1.1 A	Raspberry Pi can send PWM signals to servo to rotate a whole system.	None	R.3.1.2 A R.3.2.1 A

TABLE 4.1: Serov design specification ID's

The KickPro project requires a servo that can carry weight up to 60kg to rotate a whole system by tracking the player's position on the field. The DS5160 SSG Big servo 60kg as shown in figure 22 is made of stainless steel gear with water-proof sealing silicon loop and heat dissipation. The kind of servo needs a power supply from 6V to 8.4V to operate in the pulse width range between 500 and 2500 µs. If the supply voltage is 6V, 7.4V and 8.4V approximately, the servo can rotate with the torque of 58 kg.cm, 65kg.cm and 70kg.cm respectively [4].

Figure 22: ANNIMOS 60 KG Servo [4]

The figure below demonstrates the Python pseudo code using pulse width modulation signal from Raspberry Pi to bring up and verify the servo that can be rotated at 0 degree, 90 degree and 180 degree. Our group has tested servo with input voltage at 7.4V with current limit of 2.9A and the servo has worked successfully as expected. The input voltage of 8.4V with current limit 6.2A will be applied for our product KickPro to optimize the rotation speed and flexibility.

The operation principle of controlling servo will use the pulse width modulation (PWM) which is a transformation from various digital signals to analog signals. To gain the range limit of angle rotation between 0 degree and 180 degree, the pulse width modulation signal needs to be calibrated correctly by the duty cycle. The duty cycle is the percentage of time when the digital signal is ON. The equation to calculate the duty cycle can be derived as below:

$$\text{Duty Cycle} = \frac{T(on)}{T(on) + T(off)} \text{ x } 100\ (\%) \quad [5]$$

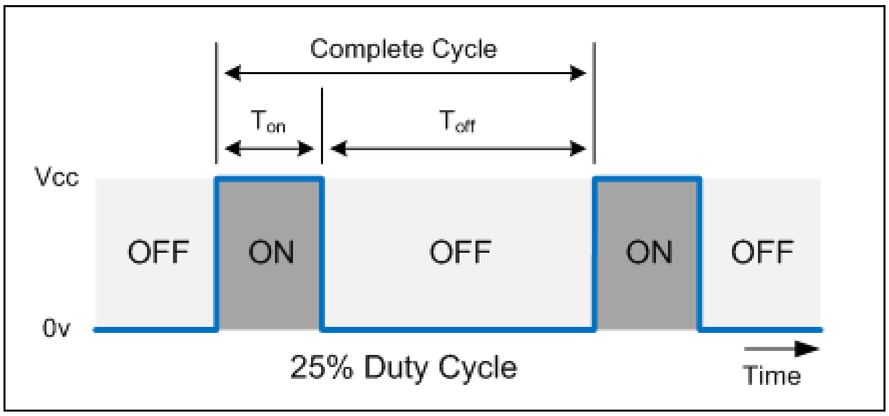

Figure 23: Duty cycle [5]

Furthermore, it is straightforward to find the rotation angle of servo thanks to the relationship between the duty cycle and rotation angle. After many tests and research has been done, we have come to the equation to express the relationship between the duty cycle and rotation angle which is used in ANNIMOS 60 kg servo [4]. The formula has been derived:

$$\text{Duty Cycle} = \frac{Angle\ (degree)}{27} + 2$$

Generally, the pulse width modulation for servo's rotation angle 0° usually has a period of 1ms when the digital signal is ON. The rotation angle of 90° and 180° have a period of 1.5ms and 2ms respectively when there is a high digital signal for almost all servos [6, 7]. However, for some special servos with high torques such as ANNIMOS 60kg servo, the pulse width modulation signal should be pre-calibrated to calculate the correct rotation angles before integrating into the project. For the project KickPro, the angle is converted to pulse width modulation which is passed to the Python function ChangeDutyCycle to control servo rotation to the desired angles as shown in the Python servo pseudo code below.

1. Import the RPi.GPIO and time libraries.

2. Disable GPIO warnings.
3. Assign pin number 17 to the servo_pin variable.
4. Set the mode to BCM.
5. Set up the servo_pin as an output pin.
6. Set the frequency of the pulse-width modulation (PWM) to 50 Hz.
7. Start the PWM signal with a duty cycle of 5%.
8. Enter a try-except loop for steps 9 to 17.
9. Within the loop, set the angle to 0 and calculate the duty cycle.
10. Change the duty cycle to the calculated value.
11. Wait for 3 seconds.
12. Set the angle to 180 and calculate the duty cycle.
13. Change the duty cycle to the calculated value.
14. Wait for 3 seconds.
15. Set the angle to 90 and calculate the duty cycle.
16. Change the duty cycle to the calculated value.
17. Wait for 3 seconds.
18. If a keyboard interrupt is detected, stop the PWM signal, clean up the GPIO, and print a message.

The servo diagram has shown the connection between servo and Raspberry Pi. The ANNIMOS 60 Kg servo consists of three wires: PWM signal, VCC and GND. To use an external power supply of 8.4V and 6.2A, the voltage between servo's VCC and GND needs 8.4V; and the signal PWM is connected from Raspberry Pi's GPIO 17 - PWM to PWM signal in servo. While the GND from servo is connected to both the GND pin in Raspberry Pi and the GND pin an external power supply.

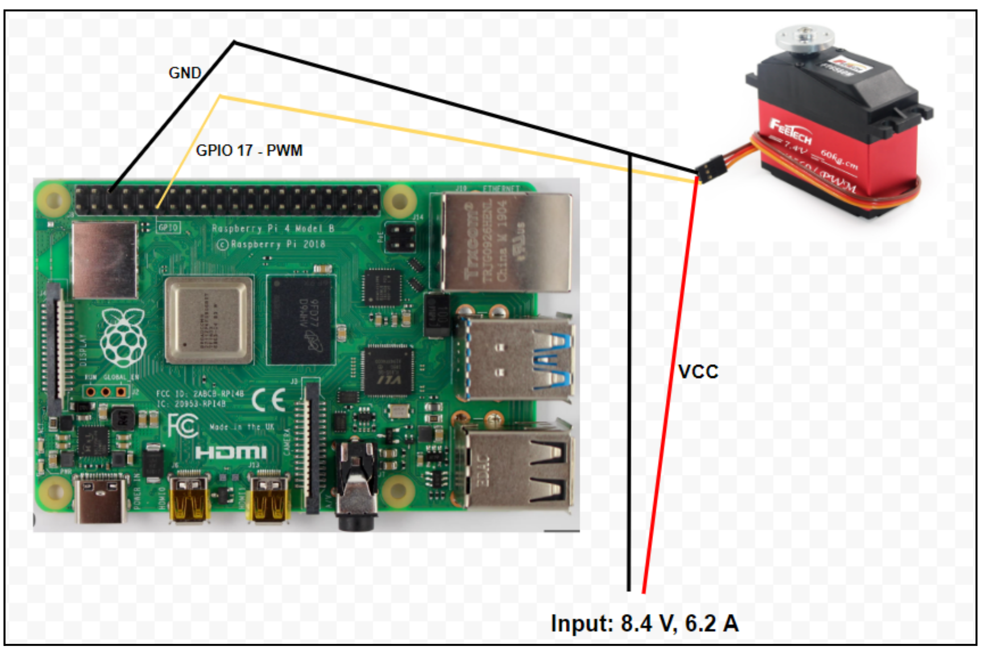

Figure 24: Servo diagram

4.2 Human Tracking

Design Specification ID	Description	Expected Changes for Future Designs	Related Requirement Specifications
D.4.2.1 A	Raspberry Pi can run a human detection algorithm by OpenCV to track a player via a webcam.	None	R.3.1.2 A R.5.1.1 A R.5.2.1 A R.5.2.2 B
D.4.2.2 B	Raspberry Pi can run human detection with distance estimation algorithm by OpenCV to track a player via a webcam.	None	R.5.2.3 B
D.4.2.2 C	Raspberry Pi can run human detection and predict the object movement algorithm by OpenCV to track a player via a webcam.	Raspberry Pi can run color detection to prevent the ball from being launched to the wrong people.	R.5.2.4 C

TABLE 4.2: Human tracking design specification ID's

To power its human tracking program, the KickPro backend makes heavy use of OpenCV (Open Source Computer Vision). OpenCV is an open-source library of computer vision and machine learning algorithms. It is designed to help developers create applications that can understand visual data from the world around them - like this very project. OpenCV provides a range of functions for image and video processing, including filtering, edge detection, segmentation, feature detection, and more [8].

One of the key features of OpenCV is its ability to detect and track objects in real-time. It offers various object detection algorithms such as Haar Cascades, HOG (Histogram of Oriented Gradients), and deep learning-based object detection models. These algorithms can detect objects of interest in an image or video feed and track their movement.

OpenCV also provides tools for image and video analysis, including background subtraction, optical flow, and motion detection. These tools enable developers at IronFoot Technologies to identify changes in a scene over time, such as object movement, camera motion, and more.

Below, the pseudocode and flowchart for the human tracking algorithm have been made available:

1. Import necessary libraries: numpy, cv2, RPi.GPIO, and time.
2. Turn off GPIO warnings.
3. Define the pin number of the servo and set the mode of the GPIO to BCM.
4. Set up the PWM with the pin number and frequency.
5. Print "Starting at zero..." to the console.
6. Initialize the camera object and set up the background subtractor and HOG descriptor.
7. Set the resolution of the camera.
8. Initialize variables for the position of the object and the angle of the servo.
9. Start a loop, from step 10 to 14, that reads frames from the camera.
10. Find the object in the frame using the HOG descriptor.
11. Draw a rectangle around the object and find the center of the object.
12. Draw lines on the frame to mark the center of the object and the center of the frame.
13. Determine the direction the object is in relation to the center of the frame and adjust the angle of the servo accordingly.
14. Print the angle of the servo, the position of the object, and the center of the frame to the console.
15. Exit the loop and release the camera when the user presses the "q" key.

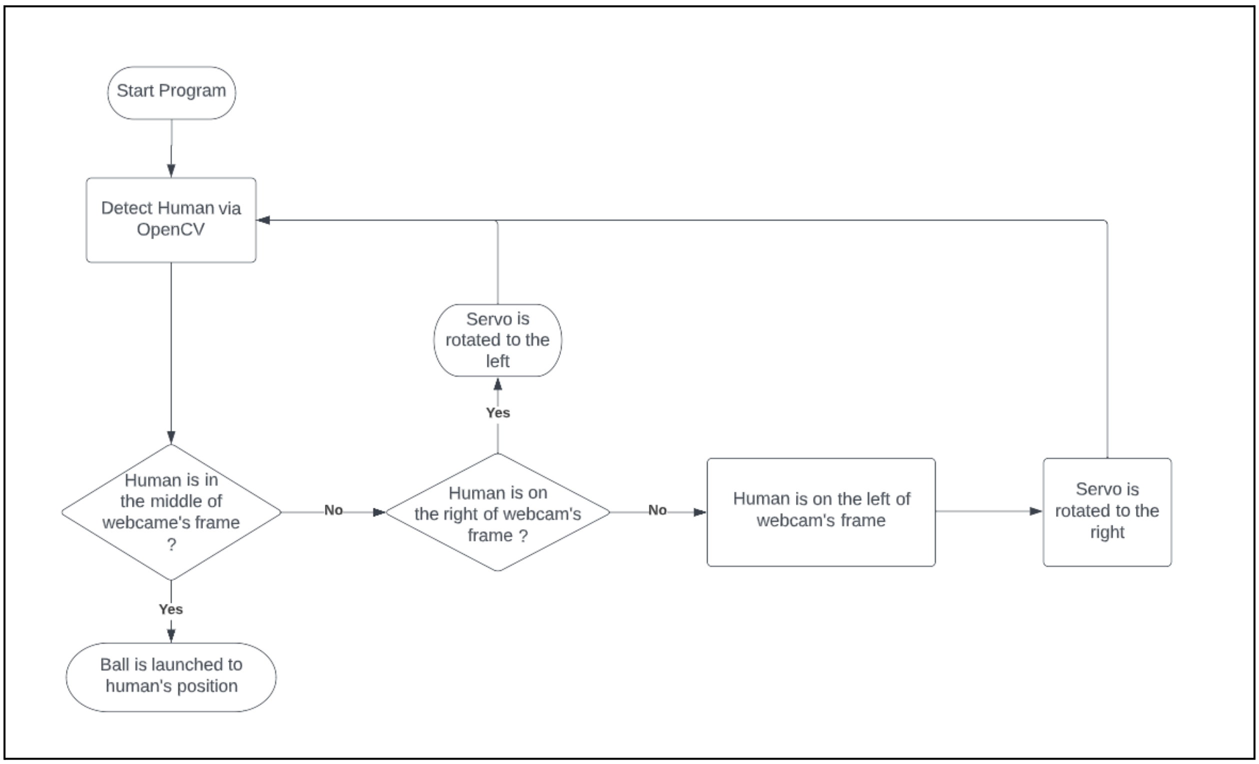

Figure 25: Human tracking flowchart

4.3 Software Application (GUI)

Design Requirement ID	Description	Expected Changes for Future Designs	Related Requirement Specifications
D.4.3.1 A	Graphic User Interface is able to provide feedback by Python Tkinter to the player.	None	R.5.1.1 A R.5.4.1 A R.5.4.2 A R.5.4.5 C
D.4.3.2 A	Graphic User Interface is able to detect the ball via a webcam by OpenCV when it crosses the goal lineline.	A new Raspberry Pi or a webcam with better resolution could be used.	R.5.1.1 A R.5.1.2 A

TABLE 4.3: GUI design specification ID's

When developing the frontend for KickPro, the developers at IronFoot Technologies considered tools like Python and various Javascript frameworks such as ReactJS. Ultimately, the former was chosen due to its ease of use and easy integration of the human tracking component. For which, the Tkinter, PIL (Python Imaging Library) Image, and ImageTk libraries proved quite useful.

The Tkinter library is a standard Python library that provides a GUI toolkit for developing desktop applications. It is a cross-platform library and is compatible with Windows, Mac, and Linux operating systems. Tkinter provides a set of tools for creating buttons, labels, text boxes, and other widgets that can be used to build interactive interfaces for desktop applications. It also provides functionality for handling user events such as mouse clicks and key presses [9].

PIL is a library for handling image files in Python. It provides support for opening, manipulating, and saving various image file formats. PIL Image is a module within the PIL library that provides functions for loading, saving, and manipulating image files. It supports a wide range of image file formats such as JPEG, PNG, BMP, and GIF.

ImageTk is a module in PIL that provides support for displaying images in Tkinter GUIs [10]. It allows images to be displayed in Tkinter widgets such as labels, buttons, and canvas objects. ImageTk provides functions for converting PIL Image objects to tkinter-compatible PhotoImage objects. PhotoImage objects can then be used to display images in Tkinter widgets. ImageTk also provides support for resizing and scaling images to fit within Tkinter widgets.

Unfortunately, the developers also ran into processing issues when the Raspberry Pi was tasked to track designated objects on two separate webcams - the program hung on the terminal. The same was observed if only one of the webcams was used to track and the other used as no more than a live video stream. Only when both of the webcams streamed unaltered video could the Pi process them.

This is a problem as both the soccer ball's location needs to be recorded when it crosses the goal line and the player needs to be tracked for the ball-launching system. As such, adjustments will need to be made for the proof-of-concept design. One expensive but obvious solution is buying another Pi to process the second webcam. Another is using the touch-screen tablet in the final design to handle some of the computation and image processing. At this moment, IronFoot Technologies is leaning towards the latter solution as it is a simple repurposing of resources that will already need to be acquired. As well, it provides a greater justification for such a costly component of the KickPro system.

The following is the pseudocode and flowchart for the GUI:

1. Import Tkinter library and all of its functions, as well as cv2, PIL Image and ImageTk, pathlib, datetime, and strftime functions from time.
2. Create three page frames for KickPro Application, named frame1, frame2, and frame3
3. Set background image for frame1
4. Add two buttons to frame1 to navigate to frame2 and frame3 respectively
5. Set background color for frame2 to pink
6. Display an image of a ball in a net with its position marked by an oval shape
7. Add a textbox for feedback to players
8. Add buttons for exit, reset, home, and next to a toolbar in frame2
9. Set background color for frame3 to pink
10. Add buttons for exit, reset, webcam, home, and back to a toolbar in frame3
11. Define a function to enable the webcam when the webcam button is pressed

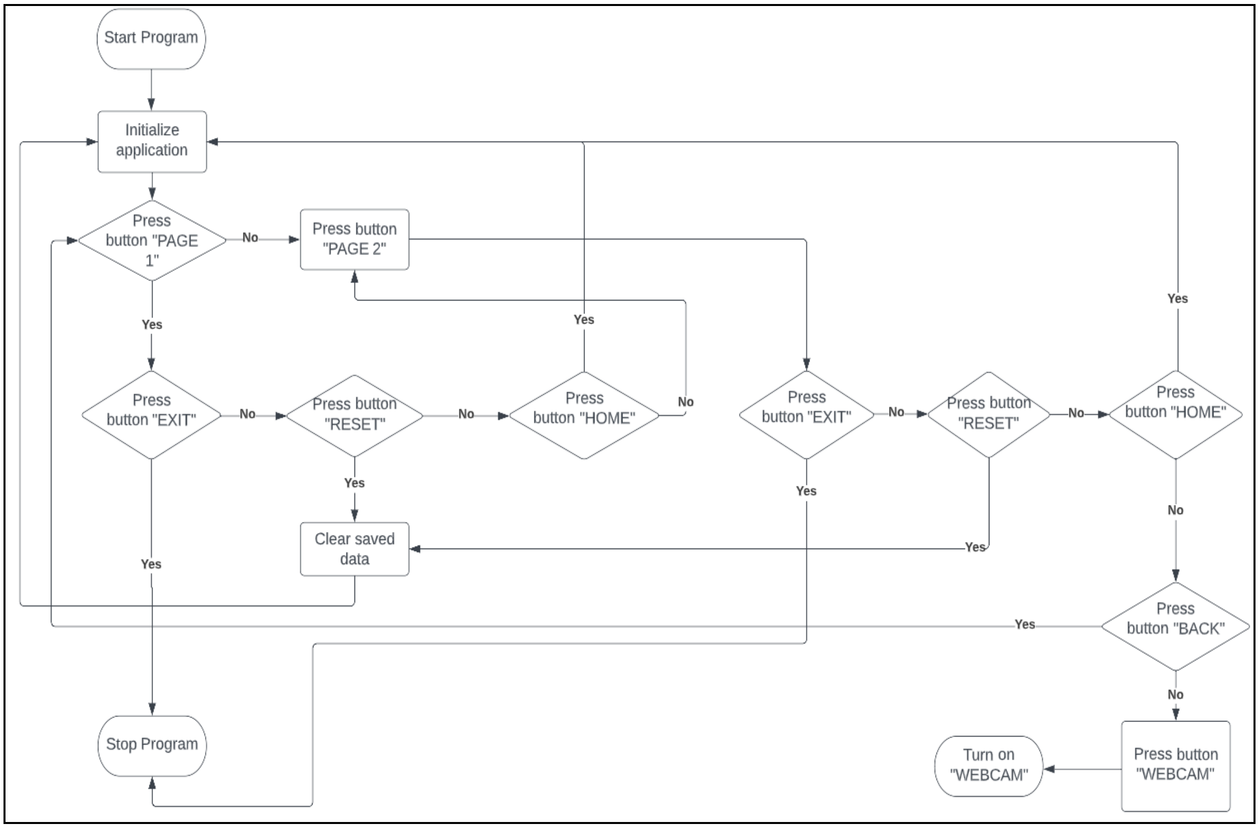

Figure 26: KickPro software application flowchart

Those figures below are the graphic user interface that our developers have been designing so far for KickPro. The GUI includes three frame pages. The first page is covered by a background image with the displayed date and time. There are two buttons which are pointing to the second page and the third page that users can easily reference with just a click on it. The second page will provide performance rate feedback to users and help them improve their soccer skills. Within the second frame, users can navigate to the first page or homepage by clicking on the "home", exit the program immediately by "exit" button, reset the program by "reset" button and move to the next page with the "next arrow" button. The third page of GUI also has buttons with similar functions with the buttons in the second page. Besides, there is a "webcam" button on the third page to enable and disable the webcam that can do human detection to control the servo.

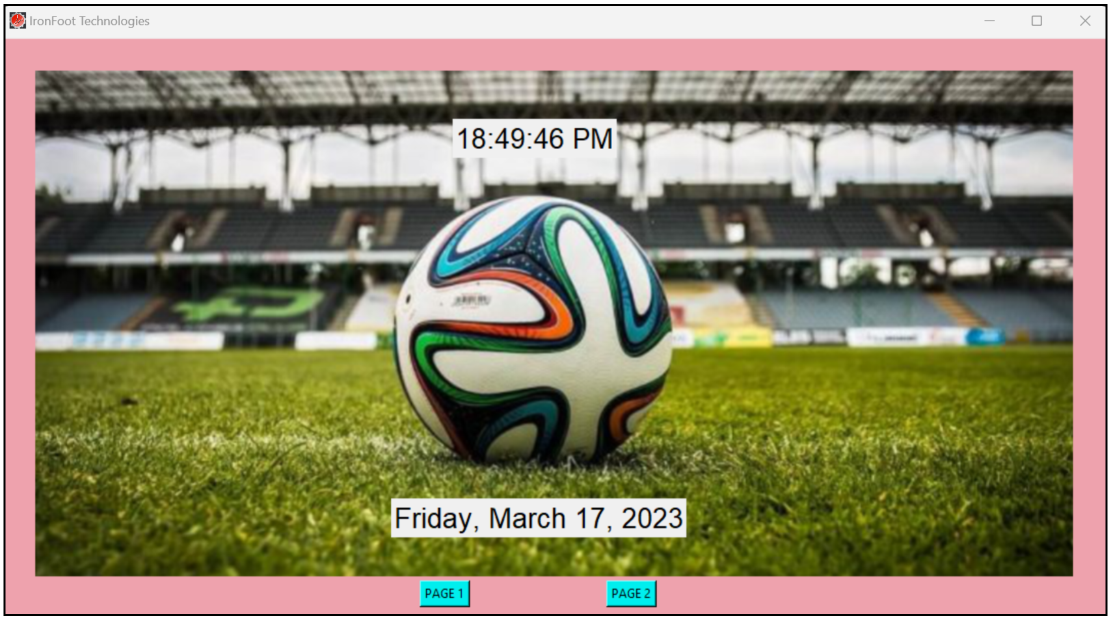

Figure 27: Home page

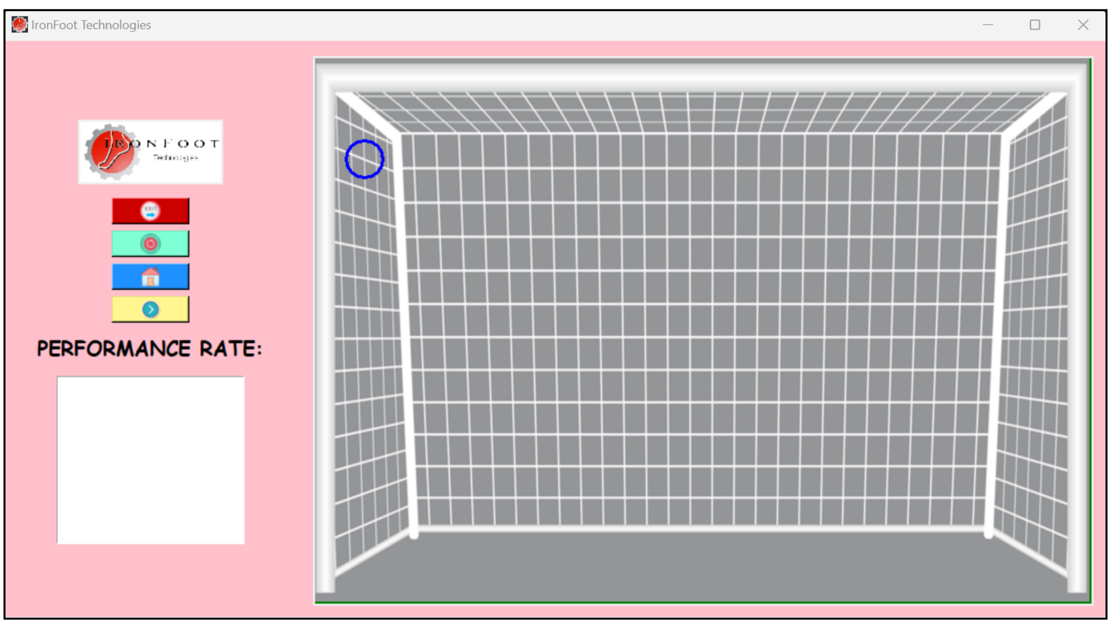

Figure 28: Performance analytics page

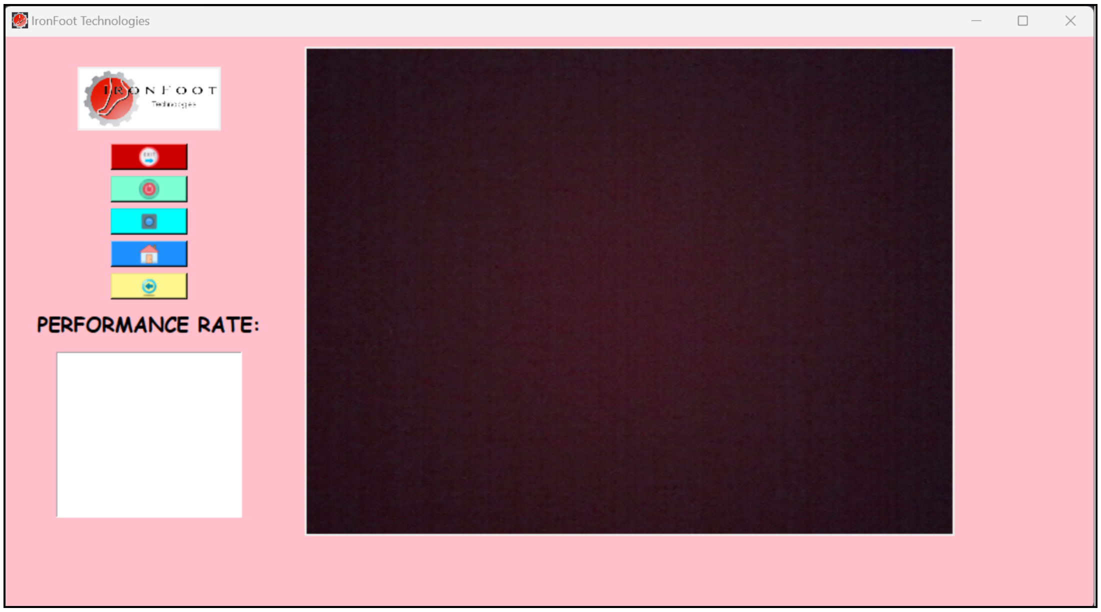

Figure 29: Box to display webcam

5 Conclusion

In conclusion, this document offers a detailed analysis of the design specifications of KickPro, outlining its features and functionalities, and lays the groundwork for future development. IronFoot Technologies has subdivided the KickPro design into 2 main components - hardware and software - which are also split in turn, forming the subsystems of this project: mechanical, electric, servo software, human tracking software, and GUI. Working together, KickPro becomes a soccer training mechanism that will analyze shot patterns and placements, and pass balls at the player. In the future, the prototype model design will be heavily influenced by the assessments completed during the testing phase and the specific requests of our clients.

The two components that work together to make the KickPro are:

1. Human Tracking
   a. Software views the players on the field via a camera.
   b. Electronic signals are sent from the software to the servo telling it where to rotate to.
   c. Mechanical servo, with its signal and power connection, rotates the entire assembly.
2. Ball Launching
   a. Software tells the motors when to turn on at the beginning of the session.
   b. Electrical powers and limits the spin of the motors.
   c. Physical wheels attached to the motors grip the ball as it comes between them and passes it to the player.

6 References

[1] "YaeTek 24V electric motor brushed 250W 2650rpm chain for E Scooter Drive Speed Control," *Amazon.ca: Sports & Outdoors*. [Online]. Available: https://www.amazon.ca/YaeTek-Electric-Brushed-2750RPM-Scooter/dp/B07KVVL1NM/ref=asc_df_B07KVVL1NM/?tag=googleshopc0c-20&linkCode=df0&hvadid=459365705466&hvpos=&hvnetw=g&hvrand=8279872115552552388&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9001538&hvtargid=pla-635620784133&psc=1. [Accessed: 17-Mar-2023].

[2] "Dual-row barrier strip, 20A, 4-circuit," *All Electronics Corp.* [Online]. Available: https://www.allelectronics.com/item/ts-204/dual-row-barrier-strip-20a-4-circuit/1.html. [Accessed: 17-Mar-2023].

[3] R. Dickinson, "How does a relay work? learn relay basics: MRO Electric," *MRO Electric Blog*, 28-Apr-2022. [Online]. Available: https://www.mroelectric.com/blog/how-does-a-relay-work/. [Accessed: 17-Mar-2023].

[4] "Buy ANNIMOS 60KG digital servo 8.4V high voltage stainless steel gear large torque high speed waterproof 15 Baja servos 270 degrees online at lowest price in Ubuy Canada. B07KTSCN4J," *Ubuy Canada*. [Online]. Available: https://www.you-buy.ca/en/product/CKFJ89Y-annimos-60kg-digital-servo-8-4v-high-voltage-stainless-steel-gear-large-torque-high-speed-waterproof. [Accessed: 14-Mar-2023].

[5] "Onion Omega2 Maker kit," *Controlling Servos*, 31-Jan-2023. [Online]. Available: https://docs.onion.io/omega2-maker-kit/maker-kit-servo-controlling-servo.html. [Accessed: 14-Mar-2023].

[6] "Servo motor control using Raspberry Pi," *rhydoLABZwiki*, 03-Jul-2015. [Online]. Available: https://www.rhydolabz.com/wiki/?p=8271. [Accessed: 15-Mar-2023].

[7] E. G. Projects, "Interfacing Tower Pro SG90 9G servo motor with 8051(89c51, 89C52) microcontroller.," *Engineers Garage*. [Online]. Available: https://www.engineersgarage.com/servo-motor-sg90-9g-with-89c51-microcontroller/. [Accessed: 15-Mar-2023].

[8] *OpenCV*. [Online]. Available: https://opencv.org/. [Accessed: 17-Mar-2023].

[9] "Tkinter - Python interface to TCL/TK," *Python documentation*. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed: 17-Mar-2023].

[10] "Python - GUI programming (TKINTER)," *Tutorials Point*. [Online]. Available: https://www.tutorialspoint.com/python/python_gui_programming.htm. [Accessed: 17-Mar-2023].

Appendix A: Design Alternatives

With our current design, possible risks may affect system performance in terms of errors, system failures, or software bugs. Furthermore, the product can face non-technical issues such as budget constraints and user-defined problems. It is important to consider multiple design options for different components of the device to prepare for unforeseen events during the development phase.

IronFoot Technologies considers many alternate designs for components such as the ball launching mechanism, performance feedback sensors, and the swivel mechanism motors to compensate for future design problems and obstacles.

A.1 Ball Launcher

The ball launching mechanism's function is to launch the ball at the desired target location within the soccer field. According to the requirement numbers **R.5.2.1 A** and **R.5.2.2 B** of KickPro, we are required to detect the player within the penalty box during phase A, and outside of the penalty box during phase B. This leads to the assumption that the launcher must be capable to launch the ball at an initial velocity $\mathcal{V}_i$, towards the target, in order to land the ball at a distance **x** from KickPro.

To fulfill these requirements, we devised multiple solutions involving different mechanisms. First solution is to use a hydraulic controlled mechanical launcher to send the ball at long distances. Second option involves two flywheels on DC motors placed side by side, rotating in opposite directions, which can be used to launch the ball at different speeds.

We decided to choose method 2 for our design. This is due to the fact that it will contribute to satisfying requirement **R.3.3.1 A** and **R.3.3.2 B** by being far more cost effective than the hydraulic alternative. Additionally, requirement **R.4.1.1 A** will be met as hydraulics can be too heavy to move around and DC motors can be a lightweight solution. Lastly, we decided to move forward with the DC motors as they will be easier to use in controlling the speed of the ball.

Assuming that the maximum distance to launch the ball is within half of the soccer field (110x75 meters) from the corner kick line, we can find that the launcher is required to pass the ball at a maximum distance of 95 meters. With certain trajectory calculations, and an optimum launch angle of 45°, we find that our motors are required to spin at 1858 rpm. The motors used in our current design are more than adequate for such purposes.

A.2 Performance Feedback

The performance feedback system will provide the player with different insights on their play styles. The software will use sensors to find the position of the ball when scored through the net, track players position on the field, and display the results onto the GUI for the coaches/players to visualize.

The ball position sensing mechanism is required to be easy to install with minimal electronics knowledge as referred to by requirement **R.3.1.7 B**. They must also be able to detect the ball, and send the information to the GUI wirelessly. To accommodate for this requirement, IronFoot Technologies designed multiple solutions. Our first solution involves using semi-long ranged ultrasonic sensors, triangulated at the side goal posts to detect the ball when going through the net and determine its distance from each end of the net. Through discussion with Dr. Michael Hegedus, we realized that this solution may include blindspots as the sensors only have a range of 15° in the induction angle. Additionally, through some testing, we realized that the ball may not be detected at high velocities and accuracy decreased as we reached its near maximum range of 4.5 m, which will be smaller than the net length.

The second solution includes using cameras, software, and motion sensors. The motion sensor detects when the ball is about to enter the net, the camera will capture a photo of the ball while going in, and object detection software is used to determine the position of the ball. This solution is best, as it is cost effective, easy to set up, and will not require much processing power, i.e. it can be connected directly to the launcher, meeting requirement **R.3.1.5 A** and partially **R.3.1.7 B** as it requires minimal electronics knowledge.

As we test the second option, we have determined a third backup solution to ensure the functionality of the product in cases where option two will not suffice. This solution involves the usage of LIDAR technology and real-time object detection software. A 2-D LIDAR can scan its surroundings at high frequencies accurately. However, this solution may be costly in both budget and processing power. An additional processor may be required to use the LIDAR for object detection and communication with the launcher, making it difficult to meet requirement **R.3.3.1 A** and potentially **R.3.3.2 B**. Furthermore, it may be a great alternative for the production version of KickPro.

A.3 Swivel Mechanism

When KickPro tracks the player across the field, it will require a swivel mechanism to rotate itself and point towards the player. In the proof-of-concept phase of our product, we are required to swivel KickPro at precise angles with acceptable tolerance, and do so in a timely manner (**R.4.2.3** and **R.4.2.4.**)

To swivel the launcher with respect to such requirements, we observe that two main electronic components are best for the application. First option is a stepper motor, which provides high precision 360° rotations at high speeds. Second option is a servo motor, providing high torque rotations up to 180° accurately. The difference in choice will be based on torque requirements, which is favored with the servo motors. Servo motors can be a bit pricier than stepper motors. However, a good servo motor will include an internal gearbox, allowing us to rotate heavy objects at high speeds/precision. The stepper motor will require a custom made gearbox which may increase the project cost and its complexity. To conclude, our main design choice is to use the servo motor for accurate, high torque rotations.

Appendix B: Test Plan

The appendix contains the deliverables that will be presented during the proof-of-concept demonstration, as well as different methods of testing that will be employed to guarantee the safety and effectiveness of the KickPro apparatus.

B.1 Proof of Concept Deliverables

IronFoot Technologies will exhibit the functionalities of the KickPro during the proof-of-concept presentation scheduled for April 12, 2023, which include:

- Using KickPro device to track a player on the field.
- Using KickPro device to launch a soccer ball to the player.
- Using KickPro device to record positions where a soccer ball enters the net.

B.2 Test Plan

The following test procedures will be performed to ensure the functionality and safety of all subsystems of the KickPro device.

<table>
<tr><th>Test: Human Tracking System Verification</th><th>Time:</th><th>Date:</th></tr>
<tr><td colspan="3">Test Procedure:
The tester will set up the KickPro.
The tester will power on the KickPro device and press the start button.
The tester will use KickPro touchscreen and perform tracking calibration.
The tester will move in an unpredictable manner in front of the camera and observe the monitor.
The tester will repeat the test with different players and movements.</td></tr>
<tr><td colspan="3">Expected outcome: The tracking system can accurately track the player without significant delay.</td></tr>
<tr><td colspan="3">Observed outcome:</td></tr>
<tr><td colspan="3">Comment:</td></tr>
<tr><td>Score: ☐ Pass ☐ Fail</td><td colspan="2">Tester Signature:____________________</td></tr>
</table>

<table>
<tr><th>Test: Launching System - Servo Verification</th><th>Time:</th><th>Date:</th></tr>
<tr><td colspan="3">Test Procedure:
● The tester will power on the Raspberry Pi and servo motor.
● The tester will use the Raspberry Pi to drive the servo motor and adjust the orientation of the launcher to a specific angle.
● The tester will repeat the test with different angles and speeds.</td></tr>
<tr><td colspan="3">Expected outcome: The servo can accurately adjust the orientation of the soccer launcher to a desired angle.</td></tr>
<tr><td colspan="3">Observed outcome:</td></tr>
<tr><td colspan="3">Comment:</td></tr>
<tr><td>Score: ☐ Pass ☐ Fail</td><td colspan="2">Tester Signature:____________________</td></tr>
</table>

<table>
<tr><th>Test: Launching system - Launcher Verification</th><th>Time:</th><th>Date:</th></tr>
<tr><td colspan="3">Test Procedure:
● The tester will set the angular velocity of the brushed motors.
● The tester will power on the brushed motors.
● The tester will manually load the soccer to the launcher.
● The tester will repeat the test with different soccer balls.
● The tester will repeat the test with different angular velocities of the brushed motors</td></tr>
<tr><td colspan="3">Expected outcome: The motors and flywheels can launch a soccer ball with a desired speed.</td></tr>
<tr><td colspan="3">Observed outcome:</td></tr>
<tr><td colspan="3">Comment:</td></tr>
<tr><td>Score: ☐ Pass ☐ Fail</td><td colspan="2">Tester Signature:____________________</td></tr>
</table>

<table>
<tr><th>Test: Ball Returning Verification</th><th>Time:</th><th>Date:</th></tr>
<tr><td colspan="3">Test Procedure:
● The tester will perform this test after successfully passing the above test procedures.
● The tester will power on the kickPro and load a soccer ball to the launcher.
● The tester will move around the field.
● The tester will repeat the test.</td></tr>
<tr><td colspan="3">Expected outcome: The camera will continuously track the position of the tester, the ball launcher will rotate towards the direction of the tester and return soccer to the tester.</td></tr>
<tr><td colspan="3">Observed outcome:</td></tr>
<tr><td colspan="3">Comment:</td></tr>
<tr><td>Score: ☐ Pass ☐ Fail</td><td colspan="2">Tester Signature:____________________</td></tr>
</table>

<table>
<tr><th>Test: Loading System Verification</th><th>Time:</th><th>Date:</th></tr>
<tr><td colspan="3">Test Procedure:
● The tester will assemble the ramp and the ball launcher.
● The tester will place the soccer on the ramp.
● The tester will retrieve the soccer and repeat the test.</td></tr>
<tr><td colspan="3">Expected outcome: The ramp can successfully deliver a soccer to the launcher.</td></tr>
<tr><td colspan="3">Observed outcome:</td></tr>
<tr><td colspan="3">Comment:</td></tr>
<tr><td>Score: ☐ Pass ☐ Fail</td><td colspan="2">Tester Signature:____________________</td></tr>
</table>

Test: Feedback System - Sensor Verification	**Time:**	**Date:**
Test Procedure: • The tester will set up the sensors on the goal post. • The tester will power on the sensors. • The tester will kick a soccer ball into the net. • The tester will repeat the test by kicking the soccer into different positions of the net.		
Expected outcome: The sensors can accurately record positions where a soccer enters the net.		
Observed outcome:		
Comment:		
Score: ☐ Pass ☐ Fail	Tester Signature:____________________	

Test: Touchscreen Verification	**Time:**	**Date:**
Test Procedure: • The tester will power on KickPro and press the start button. • The tester will use the touchscreen and test each feature of the software.		
Expected outcome: The tester can have access to performance data on the touchscreen.		
Observed outcome:		
Comment:		
Score: ☐ Pass ☐ Fail	Tester Signature:____________________	

Test: Collision Verification	**Time:**	**Date:**
Test Procedure: ● The tester will set up the KickPro device on the field. ● The tester will kick/throw a soccer ball towards the KickPro. ● The tester will repeat the above test.		
Expected outcome: The KickPro should not move significantly from its original position, nor should it tip over.		
Observed outcome:		
Comment:		
Score: ☐ Pass ☐ Fail	Tester Signature:____________________	

Test: Weatherproof Verification	**Time:**	**Date:**
Test Procedure: ● The tester will set up the KickPro device on the field. ● The tester will spray water onto the device. ● The tester will power on the device. ● The tester will load a soccer ball to the device and launch it. ● The tester will repeat the above test.		
Expected outcome: The KickPro should be weatherproof and have no problem launching a soccer ball.		
Observed outcome:		
Comment:		
Score: ☐ Pass ☐ Fail	Tester Signature:____________________	