Robot Integrated Pad Trainer (RIPT)

Brandon Noble

Jame Tran

Arshad-Labib Kazi

Hunter Ceranic

I. Introduction

The goal of the Robot Integrated Pad Trainer (RIPT) is to create a device that makes boxing training more accessible and expandable to any skill level. RIPT's innovative approach is shown through the nature of its design. The entire unit mounts easily on a wall, making it a perfect fit for both professional and home gym usage. By emulating a personal trainer, RIPT eliminates the high, reoccurring costs of hiring a coach, while also removing barriers associated with travelling to high end gyms. RIPT's most accessible features lay in its user-oriented design. Metrics collected during each sparring session are analyzed and applied to an adaptive training regime, expanding the device's coaching capabilities to all skill levels. This personal regime is synchronized with online servers, allowing a user to train from anywhere in the world. This project aims to encourage people to adopt healthier lifestyles through a fun, personalized training device that can be used by anyone.

The inspiration for this capstone project came from the movie *Real Steel* (2011) where robot boxing is a top sport akin to the UFC today [1]. The idea of combining boxing and robotics is not a novel idea from this movie though, as Rock 'em Sock 'em Robots have been a staple children's toy since 1964 [2]. Sports like tennis and baseball have products such as ball launching machines for training purposes, however when it comes to boxing, there are no widely known or available robotic training products. Creating a boxing robot seemed like an interesting challenge that would incorporate both robotics and A.I. principles into a product that would be fun to test and implement, which convinced the capstone group to move forward with the idea.

The design space for a product that combines boxing and robotics is quite large, as there are many different forms it could take. After exploring this design space thoroughly, two main directions were identified: solutions built for sparring and for training. Considering that sparring would require very advanced hardware to ensure the robot could both safely throw and absorb the impact force of a punch, as well as require software that would see and predict an opponent's movements, a sparring robot did not seem feasible given the time and budget constraints of the project. Therefore, a decision was made to create a training robot, allowing the group to begin research. First, four main physical traits of a good boxer were established so that the success of the project could be quantitively measured. These categories were: speed, which could be measured via reaction time, form which could be measured via computer vision, power which could be measured via force output, and endurance which could be measured via total training time [3]. Boxing as a sport can also simplified for instructional purposes: offense (punches) and defense (guarding).

Currently the only known competitor on the market is the STRYK RTX-1 robot. STRYK was built with sparring in mind using stationary leather pads as targets for attacking, and four foam robotic arms to simulate an opponent striking back from different angles [4]. The product is intended to be an all-encompassing training robot; the website advertises that you can train your defenses while the arms are striking at a chosen speed or use the moving arms like a pad coach where you strike the arms as they attack. However, since the arms are fixed at specific angles and strike in the same place every time, it appears this product is much better suited for defensive training rather than offensive, which pad coaches usually are targeted at improving. Taking that into consideration, clearly there is an underdeveloped niche in the market for a training robot that targets improving speed, accuracy, and endurance for offensive training, which is the main function this project hopes to address.

II. Technical Breakdown

Taking inspiration from 3D printers, the design consisted of two gantry-like systems, known as an H-bot pulley mechanism (see Appendix A). Atop the gantry, a two-axis robotic arm provided the means of securing and orienting the boxing pad targets. Mounted sensors collect data on the user, such as their reaction time, accuracy, and the force of a punch. Using a computer vision model, object detection on the user's face and gloves is used to judge their form and how well they guard. Utilizing these metrics, deficiencies in the user's skills are analyzed and the device tailors the training regime to them. The user interacts with the device through a web app, allowing them to customize routines, view their metrics, and start new training sessions.

The project was broken into four main modules: gantry system, electronics, pad control, and software. Each module was assigned a team member to oversee its development and completion, however, members provided their time and expertise across modules.

Module #1: Gantry System – 25% of Project – Lead: Brandon, Aid: Labib, Hunter, Jame

The purpose of the gantry system was to move the target pads into their required X and Y position. Inspired by the H-Bot pulley mechanism (see Appendix A), a central block is moved along these two axes by stepper motors (see Module #2 for details). Figure 1 is a photo of the final design for gantry system in isolation from the rest of the device.

<u>Key Items</u>

This module consisted of the following key items: brackets for the selected motors, brackets for idler pulleys, mounts to connect the rods and rails, a central sliding block, and zeroing components for the X and Y axis.

<u>Overview</u>

The gantry system is, in simple terms, a dual axis pulley system made of a continuous belt. Two motors simultaneously control the X and Y position of a central block upon which the robotic arm is mounted (more details in Module #3). A traditional cartesian gantry separates the X and Y axis from one another, such that each has its own belt and motor. H-Bot on the other hand utilizes a continuous belt for both axes, and allows the load of the central block to be shared between both motors [5]. This enables the torque of both stepper motors to be utilized at the same time, providing faster and more powerful movements.

Most of the construction for the gantry is 3D printed parts. These were custom modelled in Autodesk Inventor, and printed on an Anycubic Kobra FDM printer. Utilizing 3D printing allowed the gantry system to be prototyped quickly and with endless customization. The parts were printed with 6 walls, 0.3mm layer height, and a 15% cubic infill to provide strength, and minimize the weight they would add to the device. Each of the aforementioned items went through several iterations before arriving at the final design shown in Figure 1.

Four 70cm SBR12 steel rails were used in junction with eight SBR12UU linear bearings. These provided a rigid frame for the gantry and allowed the Y axis of the device to provide the bulk of the strength to the device. 3D printed mounts (rod/rail mount in Figure 1) were secured to the linear bearings upon which idler pulleys are attached to complete the H-bot pulley mechanism.

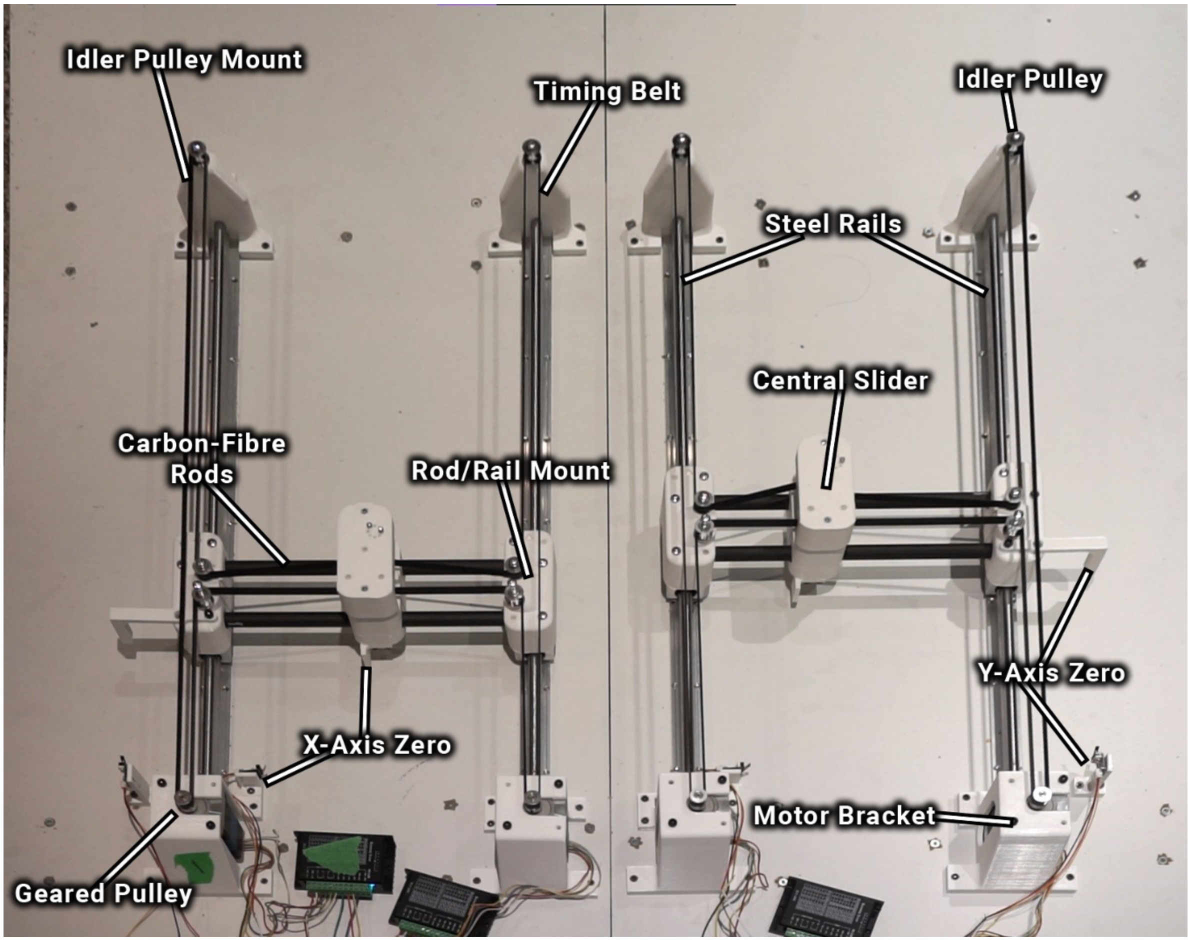

Figure 1: An overview of the gantry system.

The X-axis of the device is composed of four carbon-fibre rods upon which the central slider moves. These provided a rigid yet lightweight structure to reduce the load on the motors and ensure the frame could support the force of a punch.

Finally, to provide calibration of the gantry system, it was given zeroing capabilities inspired by 3D printers. One limit switch for each axis of the device is triggered when a 3D printed fin makes contact with it (shown in Figure 2). This method of calibration provides a means of finding the “origin” or the (0,0) coordinate of the gantry’s working area. Overall, the gantry had ~50cm of motion in the Y direction, and ~16cm of motion in the X direction.

One gantry was constructed for each side of the device as shown in Figure 1.

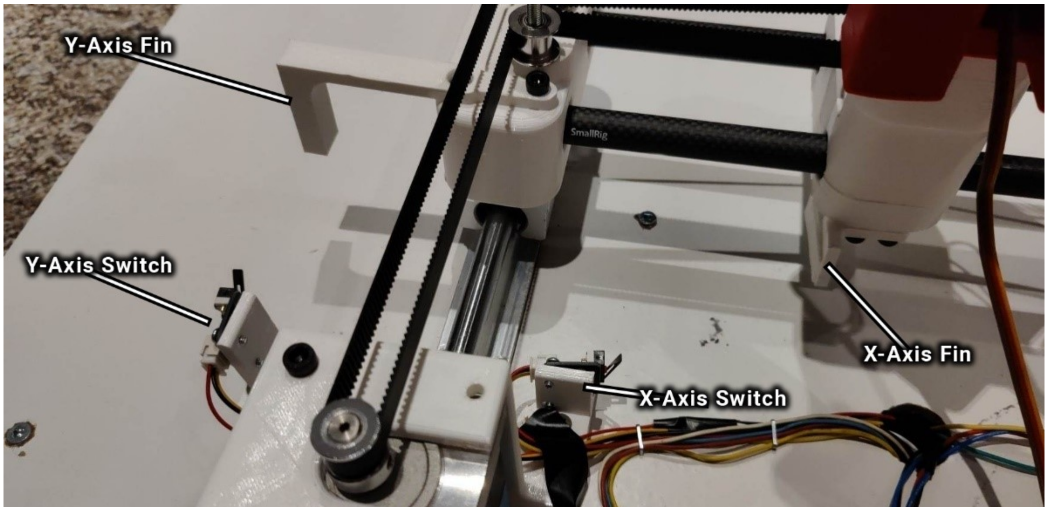

Figure 2: The limit switches and fins used to help calibrate the gantry

Item Weight and Responsibility

- Idler pulley mounts: 10% of module, assigned to Brandon
- Motor brackets: 10% of module, assigned to Brandon
- Rod/rail mount brackets: 10% of module, assigned to Brandon
- Central slider: 50% of module, assigned to Brandon
- Belt tensioners: 5% of module, assigned to Brandon
- Zeroing brackets/mounts: 5% of module, assigned to Brandon
- Installation/assembly: 20% of module, assigned to Brandon, Labib, Hunter, Jame

Time Spent on Module

- Brandon: 250 hours
- Labib: 40 hours
- Hunter: 30 hours
- Jame: 20 hours

Module #2: Electronics – 25% of Project – Lead: Labib, Aid: Brandon, Hunter

The electronics module covers the sourcing, configuration, and implementation of the electrical components used in the other modules. As this module integrates with all the other modules, there is a high degree of shared responsibility.

Key Items

As this module integrates heavily with the other modules, the key items are to provide the necessary functionality. For the gantry system, this means powering and controlling the stepper motors moving the pulley system. For the pad control module, this means using servos to provide roll and pitch to the pad and collecting and interpreting metrics data using load cells and sending it to software to process it.

Overview

The gantry system uses an H-Bot pulley system to control the position of the pad. This means that each gantry requires two stepper motors to control its movement, or four motors total. These motors will have their controls generated by a Raspberry Pi. As the Pi cannot provide sufficient power to the motors, a power supply is required to provide said power. This power supply was selected to be 24V 15A to power the four stepper motors that consume 3A each, and to fit within budget constraints. To allow the Pi to interface with the steppers, four TB6600 stepper motor drivers were acquired. These were chosen as they are the cheapest drivers that can handle the power requirements of the system. After all these decisions were made, wiring was completed and then the capabilities of the motors were tested for in Python on the Pi. This process was quite extensive as the motors were used previously and had lost some performance through the use. As RIPT required the pad to move quickly, we had to determine the maximum speed we operate the motors at with consistent performance.

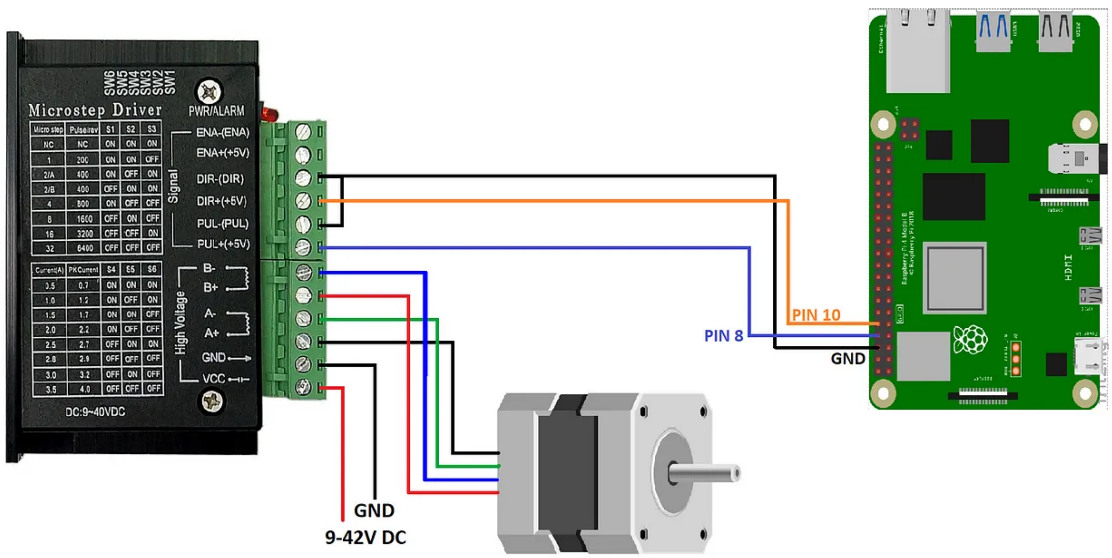

Figure 3: Circuit connecting gantry components together.

After completing the configuration for the steppers, we moved onto finding limit switches for the gantry system, which would allow us to track the position of the pad when moving it. As the steppers are moved by just sending pulses to it, without limit switches, we would be limited to being able track the position of the motor by tracking the movements made by the motor. Over time this would result in inaccuracies which could result in damage to the gantry system. As the switches just provide feedback to the Pi controlling the motors, they do not require any additional components and can be wired directly to the Pi. It was decided to use 3D printing limit switches due to their ease of implementation and access.

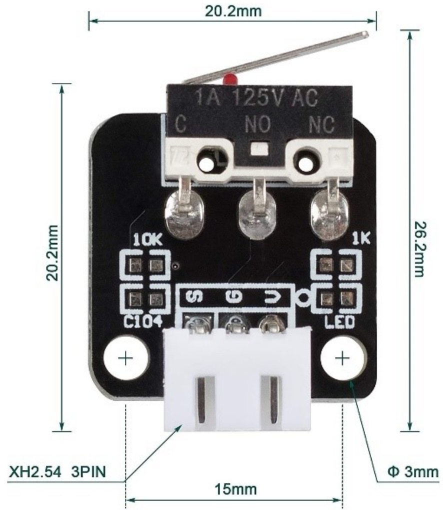

Figure 4: Single Pole, Double Throw (SPDT) Limit Switch

For the pad control module, the priority was to allow the pads pitch and roll movement to mimic straights, hooks, and uppercuts. This was done using 270° servo motos as they are easy control and have a good power to weight ratio. The motors sit with the pads, so they need to be light enough that the steppers driving the gantry to move them without a loss in performance. The servos require 5V 1A each, which also cannot be supplied by the Pi controlling them, so they also require a power supply. For this, two 5V 3A supplies were used as we already had access to them. As mentioned before, the controls were quite simple, with a single signal wire connecting to the Pi. Once wiring all the components was complete, we determined the controls required to get the pads in the required position and saved them for later use.

Figure 5: Servo Motor

After the servos, the pads also needed to be able to collect data about the user punching it. This was used to store metrics like force, reaction time, and accuracy. Load cells vary their resistance corresponding to the force applied to them, so they can be used to collect data for all three of these metrics. As the team already had some on hand, a few more were purchased to allow for two cells to be placed in each pad. The power requirements for the cells were small enough that they could be supplied using the same power supply as the servo motors. Control for these cells were a bit more complex as an analog signal is output by the load cells. To convert this signal to digital which can then be processed, an ADC was used. The ADC would take the output of the load cell as an input, sample it at some frequency, and convert the sampled signal to a corresponding digital signal. Once collected, we needed to convert this value to one that correlated accurately to the amount of force placed on the pad.

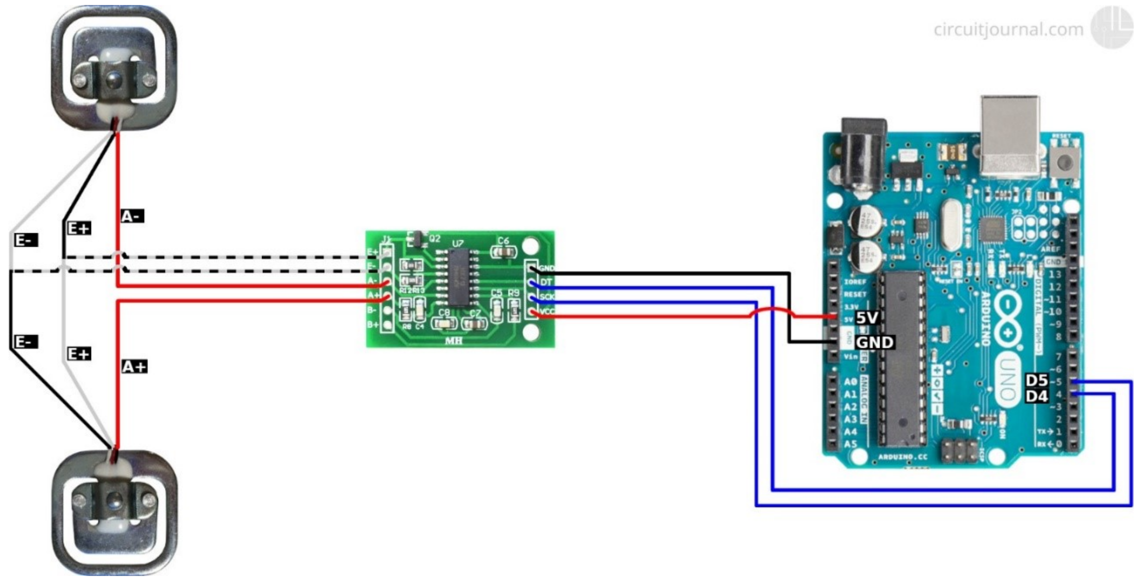

Figure 6: Load cell circuit.

Once all the tasks above were completed independently, we worked to mount them all to the main unit and integrate them together. This was done by mounting all the components to a piece of plywood, named the wooden control board (WCB). The layout prioritized keeping wiring organized and by extension, control pieces as close to the components they were controlling as possible. A photo of this can be seen in Figure 7. After this, all that was left was to wire everything together and ensure that it all worked as a unit.

<u>Item Weight and Responsibility</u>

- Sourcing Components: 20% of module, assigned to Labib, Brandon
- Testing Stepper Motor Performance: 30% of module, assigned to Labib, Brandon
- Implementing Zeroing: 10% of module, assigned to Brandon
- Servo Configuration: 10% of module, assigned to Hunter
- Load Sensor Configuration: 5% of module, assigned to Hunter
- WCB: 5% of module, assigned to Labib
- Wiring/Cable Management: 20% of module, assigned to Labib

<u>Time Spent on Module</u>

- Labib: 350 hours
- Brandon: 150 hours
- Hunter: 100 hours

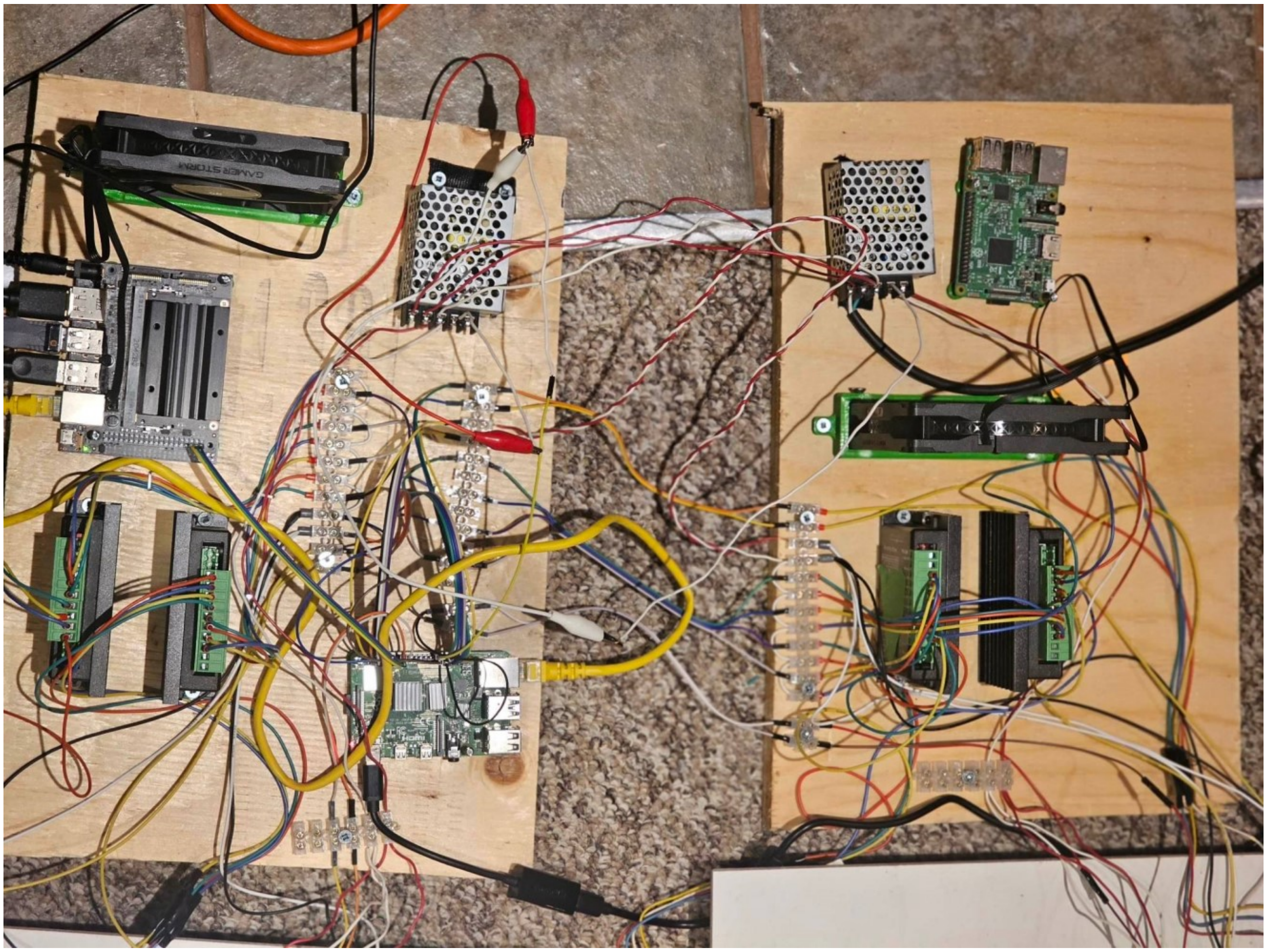

Figure 7: Testing connections on the wooden circuit board.

Module #3: Pad Control – 25% of Project – Lead: Hunter, Aid: Brandon, Labib

The pad control module consisted of two major components. The first of which was the system consisting of two servo motors, providing fine control on the pitch and roll of the pads to produce the desired orientations for straight, uppercut and hook punches, respectively. The second concerned the integration of the load cells in order to collect the metrics needed for both the software module and electronics module's motor control.

Key Items

This module consisted of the following key items: custom mounts for the servo motors that controlled the roll and pitch rotations of the pads, boxing training pads, an arm bracket to connect the pad to the motors, and load cells.

Overview

The basis of the pad control is two MG995 servo motors which established two additional degrees of freedom for the orientation of the boxing pads. The motors work in parallel to move the servos in the roll and pitch axes respectively. This parallel motion enabled the pads to be moved while the stepper motors from the gantry system were also in motion, improving the efficiency and speed of the system overall. Metal horns were attached to the servo motors such that the motors could be fastened to 3D-printed components to translate the motion to the pads.

These 3D-printed parts made up most of the motor mount and pad arm structures, and were custom modelled in Autodesk Inventor, and printed on an Anycubic Kobra FDM printer. Utilizing 3D-printing enabled us to use rapid prototyping methodologies to solve the design challenges the pad control module presented. The parts were printed with 6 walls, a 0.3mm layer height, and a 15% cubic infill to provide strength, and minimize the weight they would add to the device. Each part went through extensive changes considering both load distribution and motor configurations, in order to arrive at the final designs seen in Figure 8.

In addition to the 3D-printed parts a few other components were used to ensure the design would be able to withstand the force of a punch. These included two 3” radial bearings separating the roll and pitch servo mounts, two skateboard bearings to ensure frictionless movement of the motor directly controlling the pad arm, and two ½" diameter PVC pipes cut to a length of 20cm which provided the flexible, yet strong structural frame for the pad arm. Most of these components are hidden by 3D-printed housings, however they were integral to the function of the prototype.

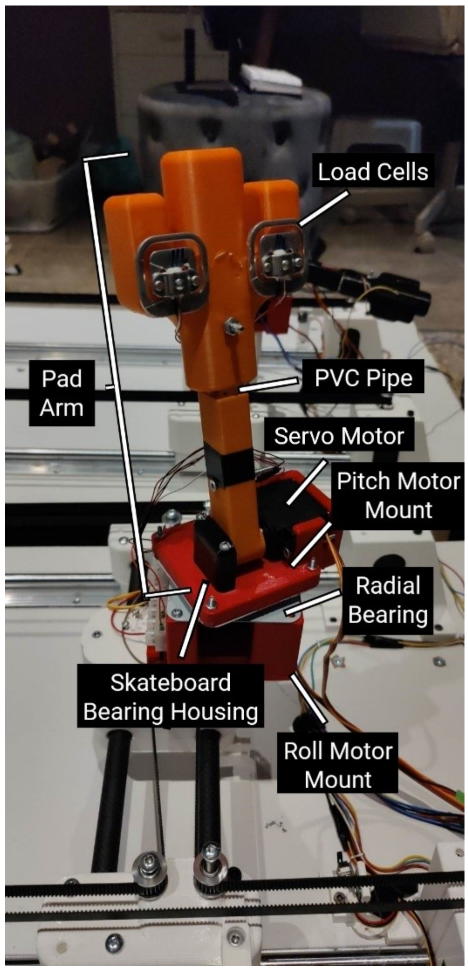

Figure 8 : An overview of the Pad Control Module

The final part of the pad control module concerned experimenting with the load cells to calibrate them and determine the best physical configuration, before they were implemented with the rest of the project. The load cells were used for collecting power, accuracy and reaction time metrics, as well as acting as a flag for the control loop to move the pad to a new position once hit. Ultimately the final design utilized two load cells per pad, in order to acquire a wider area for sensing and achieve more accurate results.

Two pad control units were assembled, one for each of the right and left gantry systems respectively.

Item Weight and Responsibility

- Roll servo motor mount: 20% of module, assigned to Hunter
- Pitch servo motor mount: 20% of module, assigned to Hunter
- Material and Pad selection: 10% of module, assigned to Hunter
- Pad arm bracket: 20% of module, assigned to Hunter
- Load cell testing: 10% of module, assigned to Hunter
- Installation/assembly: 20% of module, assigned to Hunter, Brandon, Labib

Time Spent on Module

- Hunter: 250 hours
- Brandon: 20 hours
- Labib: 20 hours

Module #4: Software – 25% of Project – Lead: Jame, Aid: Brandon, Labib

The software module can be thought of two systems: a computer vision stack and machine learning pipeline that allowed for the collection of face and boxing glove positions during the operation of the unit, and a user web app that allowed users to view their collected metrics.

Key Items

The key items for this module were the webcam used for object detection, and the two microcontrollers used for inferencing and serving the web app (the NVIDIA Jetson Nano and the Raspberry Pi 3 respectively). Of particular importance was the Jetson Nano, since it possessed an onboard GPU which was critical to allowing fast and accurate inference time for the computer vision model.

Overview

The system overall consists of a repository architecture, where various sensors and data sources collect data and are sent to a data fusion client on the motor control Pi. This data fusion client then changes information stored within a database. Users using the web application are also clients, but with limited permissions on writing data. A client server architecture is used to facilitate access to the data from users using the web app. This architecture was chosen because of its capacity to manage a large number of clients at the same time as client-server architecture has high fan out. The server also has specific API routes that separate what operations the data fusion client (the actual robot) and the user clients can perform, preserving separation of concerns.

While in our architecture the machine learning component is treated simply as a sensor, due to its complexity and importance to the functional requirements of the project, significant development and focus on that component was done. The machine learning component consisted of two systems: a development pipeline used on x86 devices for development, testing and model evaluation; and a deployed inference stack for use on the Nvidia Jetson Nano.

The development pipeline consisted of the dataset labelling tool Roboflow, which was then passed into a virtual GPU instance running Pytorch. This GPU instance would train our object detection model, which for this project used YOLOv7-tiny. After training, hyperparameter tuning, and final model evaluation, the model would be exported to a Pytorch weights file.

The deployment pipeline would take the exported Pytorch weight files and build them into a TensorRT inference engine. This is done because TensorRT is much faster at inferencing on

NVIDIA GPUs, which our inferencing device possesses. This is done by using TensorRTX, a custom library created for this purpose.

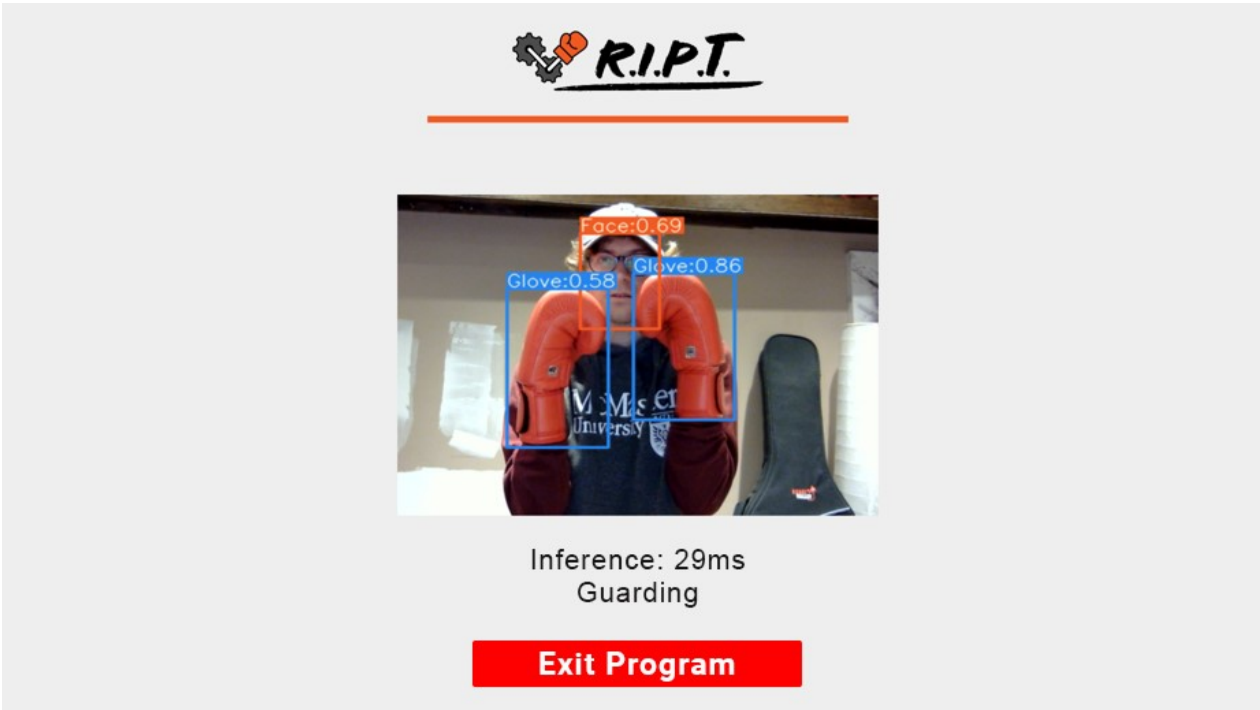

Figure 9: A deployed YOLOv7-tiny model on the Nano

For the server and web app a PERN (Postgres, Express, React, Node) stack was used. The server side logic was programmed using Express and Node.js, with PostgresDB as our database. Finally, the front end web app was built in React.

<u>Item Weight and Responsibility</u>

- Generating and labelling dataset: 10% of module, assigned to Jame, Labib, Brandon
- YOLO model deployment and training: 25% of module, assigned to Jame and Brandon
- YOLO model deployment and business logic implementation: 20% of module, assigned to Jame and Brandon
- Server development: 20% of module, assigned to Jame
- Frontend development: 15% of module, assigned to Brandon
- Networking and data fusion: 10%, assigned to Labib

<u>Time Spent on Module</u>

- Jame: 450 hours
- Brandon: 200 hours
- Labib: 100 hours

III. Progress and Results

Module #1: Gantry System – 100% Complete – Lead: Brandon, Aid: Labib, Hunter, Jame

Results

By the end of the project, all aspects of the gantry system were complete. The key items listed in *II. Technical Breakdown* were fully realized and functioned as intended. Figure 10 shows the first working iteration of the gantry system, while Figure 11 shows the final design for the gantry system.

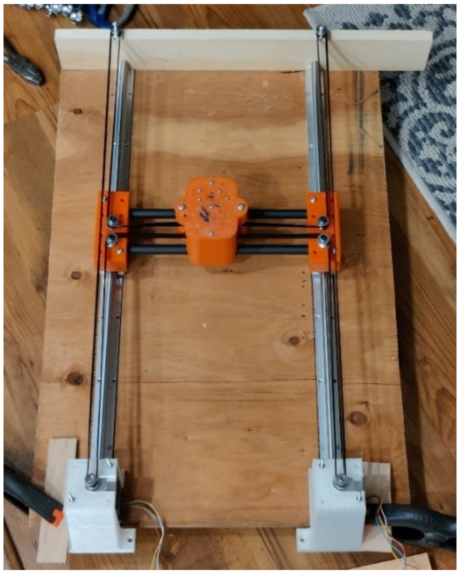

Figure 10: First working iteration of the gantry system

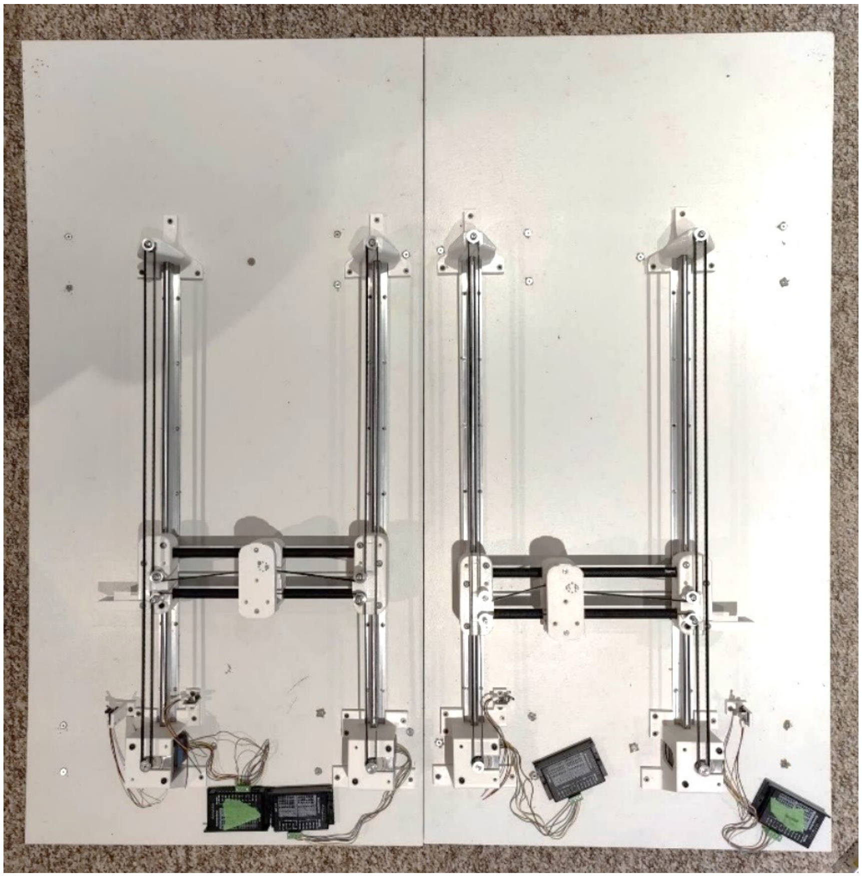

Figure 11: The final gantry system

Challenges

Over the duration of the project, several challenges arose. Component selection was a factor in this, as well as the lack of experience with CAD and mechanical design. By solving these challenges, I strengthened my CAD skills, learned to draw inspiration from my surroundings, and expanded my mechanical design abilities. I feel that this has broadened my skillset as an engineer, and allowed me to be more confident tackling large problems. The three main challenges faced were the central sliding unit, the belt tensioning system, and the frame.

The central sliding unit went through the most variations in this module. Initially, it was proposed to use a video camera slider. The carbon-fibre rods implemented for the X axis are made for this function, and mounts to slide across them are readily available (see Figure 12). As the project progressed, it was found that a custom slider would enable better integration with other modules. Figure 12 shows a comparison the original concept, intermediate progress, and the final design. Modifications included making it thinner and the addition of mounting points for the robotic arm. By decreasing the footprint of the slider, the range of motion for the X-axis of the gantry increased.

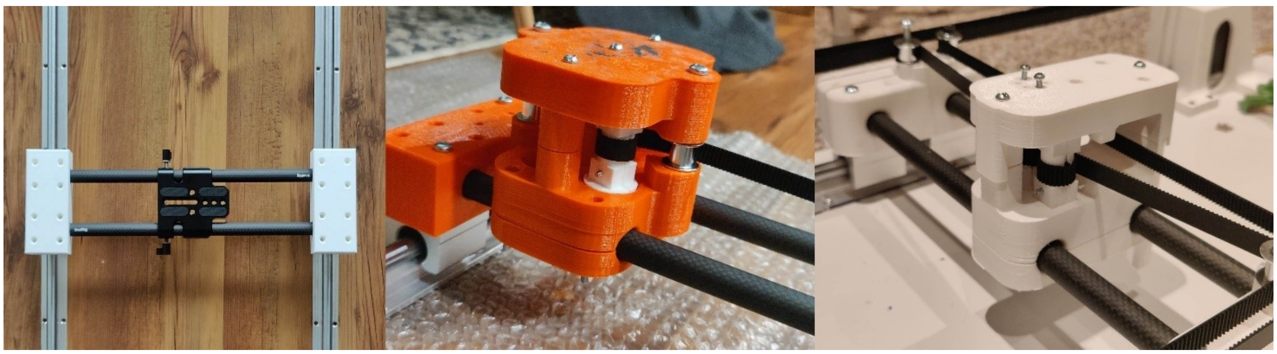

Figure 12: Left: initial design. Centre:intermediate design. Right: final design.

Another challenge for the gantry system was implementing belt tensioners. Any belt system requires sufficient tension to prevent it from slipping off of pulleys and gears. Since the H-Bot mechanism requires the belt to be fixed at the central slider, the tensioners were implemented here (see Figure 12). The design for this was inspired by spools for camera film. The belt sits between two 3D printed pieces. One is toothed with the same bore as the belt, while the other is flat as shown in Figure 13. This enabled the belt to be clamped between the two pieces, and then wound to provide proper tension. Later iterations added a square base to the tensioner so a wrench could be used to help tighten the system.

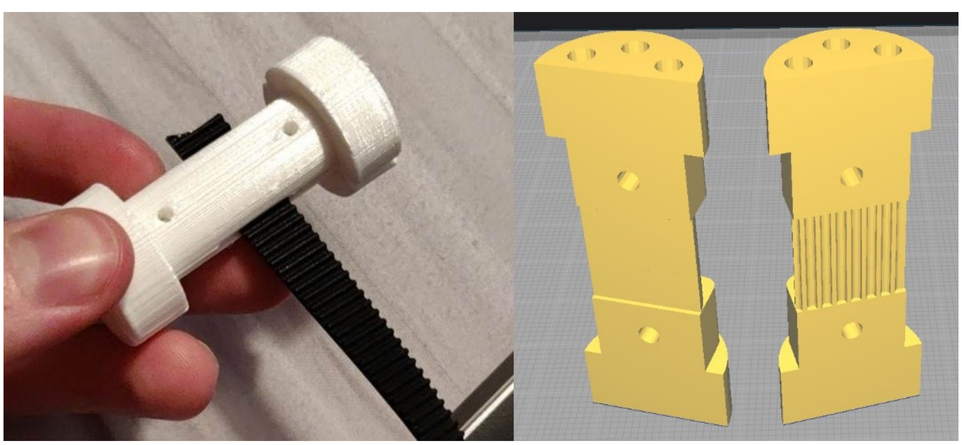

Figure 13: Spool-like design for belt tensioning.

RIPT is intended to be a wall mounted device for a home or a gym. It could not however, be mounted to a wall for demonstration at the McMaster Engineering Expo, so a frame was built to simulate a wall. Figure 14 shows the process of the frame from design to testing. Three support legs are joined by three cross beams, two of which have 45º mitres so the gantry system can hang from it. A sheet of plywood across the bottom of the frame was added so sandbags could be used to lower the centre of gravity and prevent it from toppling. The frame was designed to be taken apart into seven pieces, making transportation to and assembly at the McMaster Engineering Expo relatively easy.

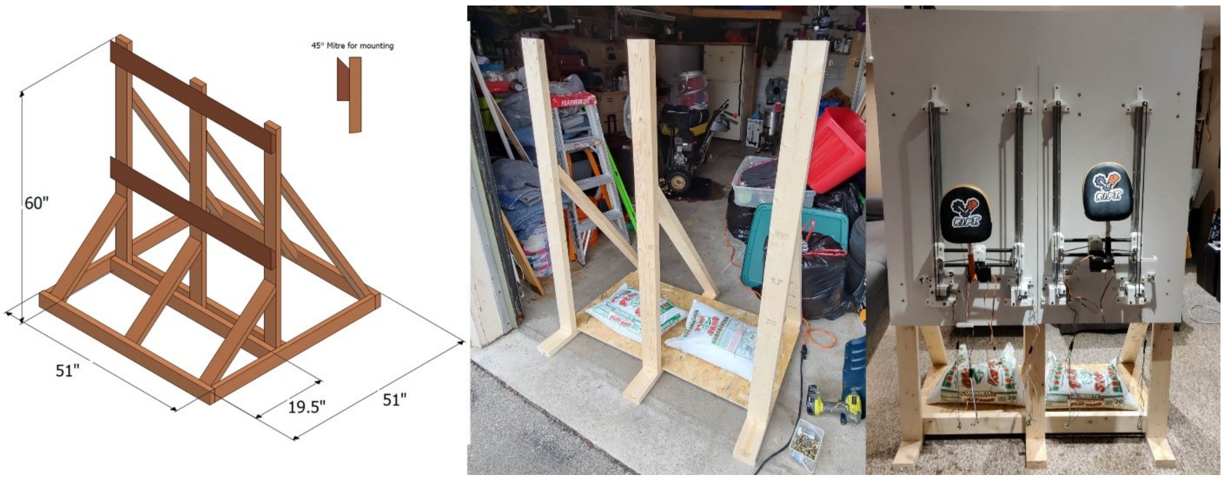

Figure 14: Left: frame plan. Centre: frame construction. Right: frame implementation

Items Completed

- Idler pulley mounts: 100% completed by Brandon in 20 hours
- Motor brackets: 100% completed by Brandon in 20 hours
- Rod/rail mount brackets: 10% completed by Brandon in 30 hours
- Central slider: 100% completed by Brandon in 100 hours
- Belt tensioners: 100% completed by Brandon in 35 hours
- Zeroing brackets/mounts: 100% completed by Brandon in 10 hours
- Frame: 100% completed by Brandon in 15 hours
- Installation/assembly: 100% completed by Brandon, Labib, Hunter, Jame in 20 hours

Items Not Completed

There were no items in this module that were not completed.

Module #2: Electronics – 100% Complete – Lead: Labib, Aid: Brandon, Hunter

Results

At the end of the project, all electrical aspects of the project were working. All items listed below, and above in *II. Technical Breakdown* have been completed.

Challenges

Most challenges with this section had to do with resource limitations, as can be seen in the items not completed section below. Though all the requirements were met, things like testing stepper motor performance could have been done in much less time given they performed to manufacturer specifications. As the motors were heavily used, the output from the motors were different and less than that of a new motor so they all had to be tested individually and as part of the system to determine the lowest common denominator in performance and match the output of all the motors to that. Another challenge with the stepper motors was to avoid them skipping steps. Through experimentation, it was determined that the motors could not be operated at their maximum speed without a slight acceleration. Large movements of the gantry were accelerated from 100kHz to 200kHz over two rotations, while smaller movements were operated at 100kHz.

Another challenge was with the load sensors, which managed to output data, but was not very responsive or reliable. The data needed to be scaled and shifted to represent the force of a punch accurately and once a user punch the load cell, the measured data would spike up before slowly going back to baseline. As a training session includes many punches in rapid succession, the load cell never had a chance to go back to baseline after a punch, resulting in inaccuracies in the results after some time.

Figure 15: Load Cell

Finally, the last challenge was the WCB. A goal with this project was to design and print a PCB containing all the control components to simplify construction and to increase scalability of the project. However, due to time constraints, the team never got a chance to develop a PCB.

<u>Items Completed</u>

- Sourcing Components: 100% completed by Labib and Brandon in 40 hours
- Testing Stepper Motor Performance 100% completed by Labib, Brandon in 200 hours
- Implementing Zeroing: 100% completed by Brandon, Labib in 20 hours
- Servo Configuration: 100% completed by Brandon, Hunter in 60 hours
- Load Sensor Configuration: 100% completed by Hunter in 40 hours
- WCB: 100% completed by Labib in 10 hours
- Wiring/Cable Management: 100% completed by Labib in 40 hours

<u>Items Not Completed</u>

All the tasks required to create a functional prototype for the expo were completed, however, given additional time and budget, more could have been done in this section. First, instead of the stepper motors, brushless DC (Direct Current) motors could have been used for faster and torquier performance moving the pads. Second, better load sensors with quicker, more accurate readings could have been used for more reliable measurements. Finally, a lot of the control components and controllers could have been integrated into one larger controller all connected through a PCB. This would have allowed for more streamlined flow between the different controllers used in this project, as well as have significantly reduced the need for wires between components, as the only things that could not be mounted directly on the board are the motors and load cells. Also ideally, a single power supply would be used, with converters being used to bring voltage down to necessary levels to power different components.

Module #3: Pad Control – 97.5% Complete – Lead: Hunter, Aid: Brandon, Labib

Results

The overall design of the pad control module was completed by the end of the project, however due to unexpected issues with the load cells the module was not able to be 100% implemented by capstone expo. Figure 16 shows the initial design of the pad control module, and Figure 18 shows the final design at the capstone expo.

Challenges

The pad control module was likely the most mechanically challenging module due to the requirement of the pad arm to be able to withstand the force of an average human punch. The main areas these challenges arose included the spatial constraints of the mounted arm and the materials available to be used. The slider that the pad control brackets were intended to be mounted on was approximately 10.5 cm by 5 cm and due to the location of the belt tensioners, not all of that space was useable for the mounts. In addition, although the mounts were not required to be limited to the dimensions of the slider, it was important to be cognizant of the weight distribution of the mounts to ensure it did not affect the sliders movement along the gantry system. With all that needing to be considered, as well as the issue that the bracket needed to be able to distribute a substantial amount of force along its frame without breaking, the design of the motor mounts was a very difficult and delicate balancing act that required many iterations to get right.

Initially we were wanting to use a servo and stepper motor in tandem to control the pad rotation as the intended stepper motor was smaller in size than the servos (see Figure 16). However, after initial testing we realized this would result in a very complicated control loop and a lot of issues when it came to wiring, and weight of the design.

Figure 16: Motor Mount Design, with a servo controlling the roll axis and a stepper controlling the pitch

Quickly we swapped to using two servos, which is still seen in the final design. However this is where the materials we were using began to become a major constraint on the progress of the module. Through experimentation we found that when struck with any substantial amount of force, the bottom motor horn connecting the the roll axis motor to the top piece, would consistently fail. Initially we were using a plastic motor horn which fractured due to the shear stress it experienced, highlighted in Figure 17. We swapped these out for stronger metal motor arms and attached it with adhesives and fasteners to try to reinforce the fracture zone. However due to the size of the screws provided with the metal motor arms, one of the two pad control units also failed as the motor arm was not adequately fastened to the plastic mount. In addition, it was observed that the movement of the pads put a massive strain on the servo motors without force being applied, likely due to uneven weight distribution, and the system wobbled as it moved which needed to be addressed. Through research we came across 3" radial bearings, which are commonly used for turntables. It was thought they would be an excellent solution to redistribute the load throughout the plastic mounts instead of just along the motor horns. This required a redesign of the plastic mounts to be a bit larger to accommodate the size of these bearings, however in retrospect this probably was the most integral design change made in this module as it was the key to allowing the pad control unit to take a moderate amount of force without failing. This redesign to incorporate the radial bearing can be shown in Figure 18.

Figure 17 Double servo design, highlighting the discovered failure zone of the motor arm

Figure 18: The final pad control unit design, as seen from the front and the back

Overall, I am very happy with the progress made in regards to the mechanical challenges faced in this module. A lot was learned in regards to design and failure modes of different materials, and I think we really were able to show our ingenuity and resourcefulness to come up with solutions such as using the radial bearing to fix the problems we faced.

The final challenge that was encountered in this module was regarding the load cells. The challenge was twofold, and the first was finding an appropriate configuration for the load cells to obtain accuracy measurements. We realized that ideally, the load cells should have been sewed into the pads themselves, however due to lack of experience sewing and time constraints this configuration was never attempted. Instead layouts with one, two and four load cells per pad were tested to see which one was most efficient for our purposes. While four load cells would have been ideal for accuracy measurements, two load cells were much more feasible given the physical room we had to install them on the pad control arm. In turn, this meant we had to sacrifice the accuracy metric as the reading were not nearly precise enough to obtain valid data. Furthermore during this testing it was found that even post calibration the load cells in our possession experienced reading drift, such that whenever a load was removed from the cells they did not return to reading zero. It is hard to know exactly what the issue might have been. At first we thought it might have been potential damage from overloading the cells during testing or temperature damage as we used hot glue to fasten them to the pad control arm. However we had unused load cells from the same batch, and those displayed similar behaviour. Since we opted to buy the cheapest load cells available to us it may have just been the load cells or the wires themselves that were bad, or alternatively the ADC converter used to read the load cell output was damaged by ESD. Regardless, this issue was unable to be solved as we could not pin down the root of the problem in time, and as such the load cells were incorporated exclusively as proof of concept, and were not actually used in the control loop design.

Items Completed

- Roll servo motor mount: 100% completed by Hunter in 75 hours
- Pitch servo motor mount: 100% completed by Hunter in 75 hours
- Material and Pad selection: 100% completed by Hunter in 10 hours
- Pad arm bracket: 100% completed by Hunter in 50 hours
- Load cell testing: 75% completed by Hunter in 20 hours
- Installation/assembly: 100% completed by Hunter, Brandon, Labib in 20 hours

Items Not Completed

While all items were completed, in practice since the load cell drift problem could not be fixed in time for the capstone expo, the metrics they were collecting were not implemented into the system for the final product. That being said if we had more time and a larger budget, potetnailly buying different more expensive load cells may have yielded the results we were looking for. Additionally rebuilding the pad arm unit out of lightweight aluninum would have been ideal to further increase the strength of the frame. Replacing the servo motors themselves with higher torque DC motors would also improve both the ability of the design to take a punch as well as the speed of pad orientation.

Module #4: Software– 100% Complete – Lead: Jame, Aid: Brandon, Labib

Results

At the end of the project, we were able to get all major software components working. The computer vision model was able to achieve a high mean average precision (mAP). The mAP(.05:.95) was 70%, and the mAP(0.5) was over 95%. All business logic was implemented for the ML stack and all the sensors. Finally, the server, database and web app were fully implemented, tested and deployed.

Challenges

Major challenges in this section had to deal with resource shortages, the critical one being the lack of computing power we had. The model training required powerful GPUs, which we were unable to obtain until the final month of the project. In the final month of the project, we were able to switch to using a cloud GPU (RTX A4000), with 32GB of VRAM. This was critical to speeding up training and evaluation of the ML models. After that, the other major challenge was integrating all the components. The software components had many components that were able to be developed and unit tested individually, such as the database, server, web app and data fusion client. However, when it came to integration, many errors were encountered that had to be fixed.

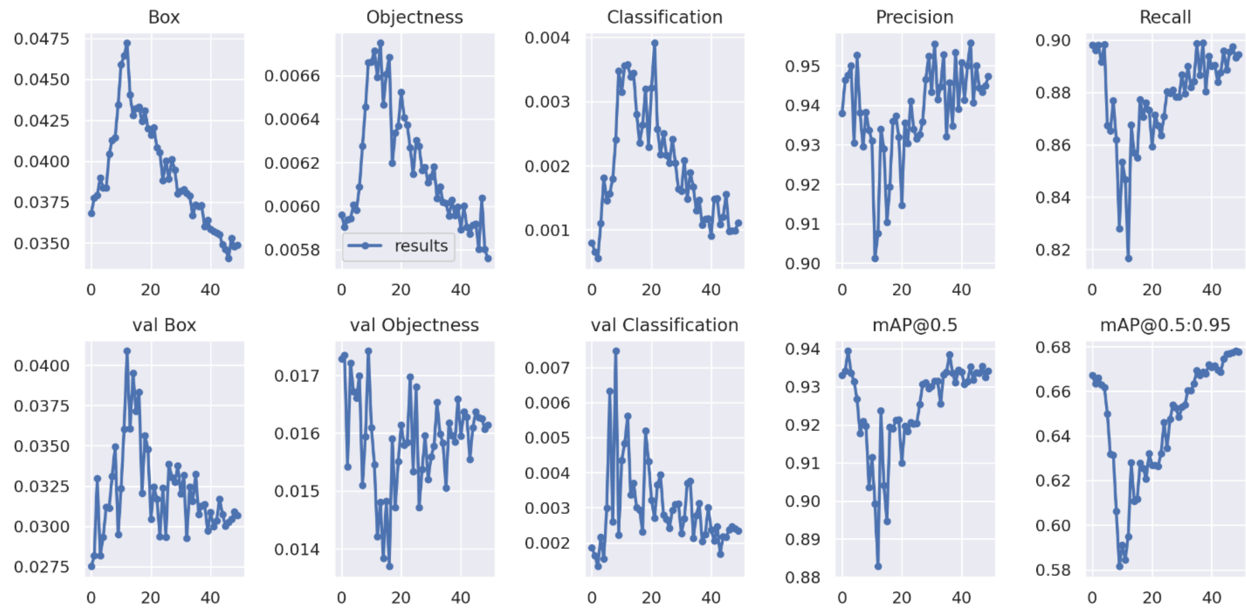

Figure 19: Metrics used for Hyperparameter Tuning

Items Completed

- Hyperparameter tuning: 100% completed by Jame in 75 hours
- Model deployment: 100% completed by Jame and Brandon in 175 hours
- Server deployment: 100% completed by Jame and Labib in 100 hours
- Database development: 100% completed by Jame in 50 hours
- Frontend development: 100% completed by Brandon in 50 hours

Items Not Completed

The overall module was completed 100% by the end of the project, with all components functioning as intended.

Progress of the Final Project:

After eight months of work, the prototype for RIPT was completed. The four modules were successfully combined with one another as shown in Figure 21.

Both the gantry and pad control systems operated without error and provided precise movements. They operated in conjunction with one another to provide a wide range of positions and orientations for the training pads. Unfortunately the load cell data could not be incorporated into the final version of the prototype, however this did not inhibit the gantry system and pad control unit from operating in tandem, it just reduced the accuracy of metrics that could be collected for the web app.

The electronics system effectively powered all the motors and controllers. Securing it on the back of the device prevented it from sustaining any damage during operation, and kept key components ventilated as well. The movement of the slider was accounted for when powering the servo motors and load cells by using a hanging cable with the appropriate amount of slack.

Results from the computer vision model and metric collection were accurate, and successfully posted from the NVIDIA Jetson Nano to the RIPT website for analysis.

Overall 99% of the project modules were completed, and given more time and resources, substantial improvements could still be made to the prototype. The silver milestone outlined in our initial proposal was completely met, and some portions of the gold milestone such as the web app were accomplished as well. Ultimately as a multidisciplinary team we feel that our project was able to fully encapsulate the engineering principles we learned in our respective programs, as well as improving our ability to work in interdisciplinary settings in the future.

To help show the final functionality of the RIPT prototype, the following videos are available.

- Explanation video: https://youtu.be/TRa0bQ9WmGY
- Demonstration videos: https://youtu.be/QMfrOVIuLZo and https://youtu.be/PSHmU63E7Yo
- Web application: https://brlnoble.github.io/RIPT_Capstone/

Figure 20: Results from one of the sparring sessions during the 2023 McMaster Engineering Expo

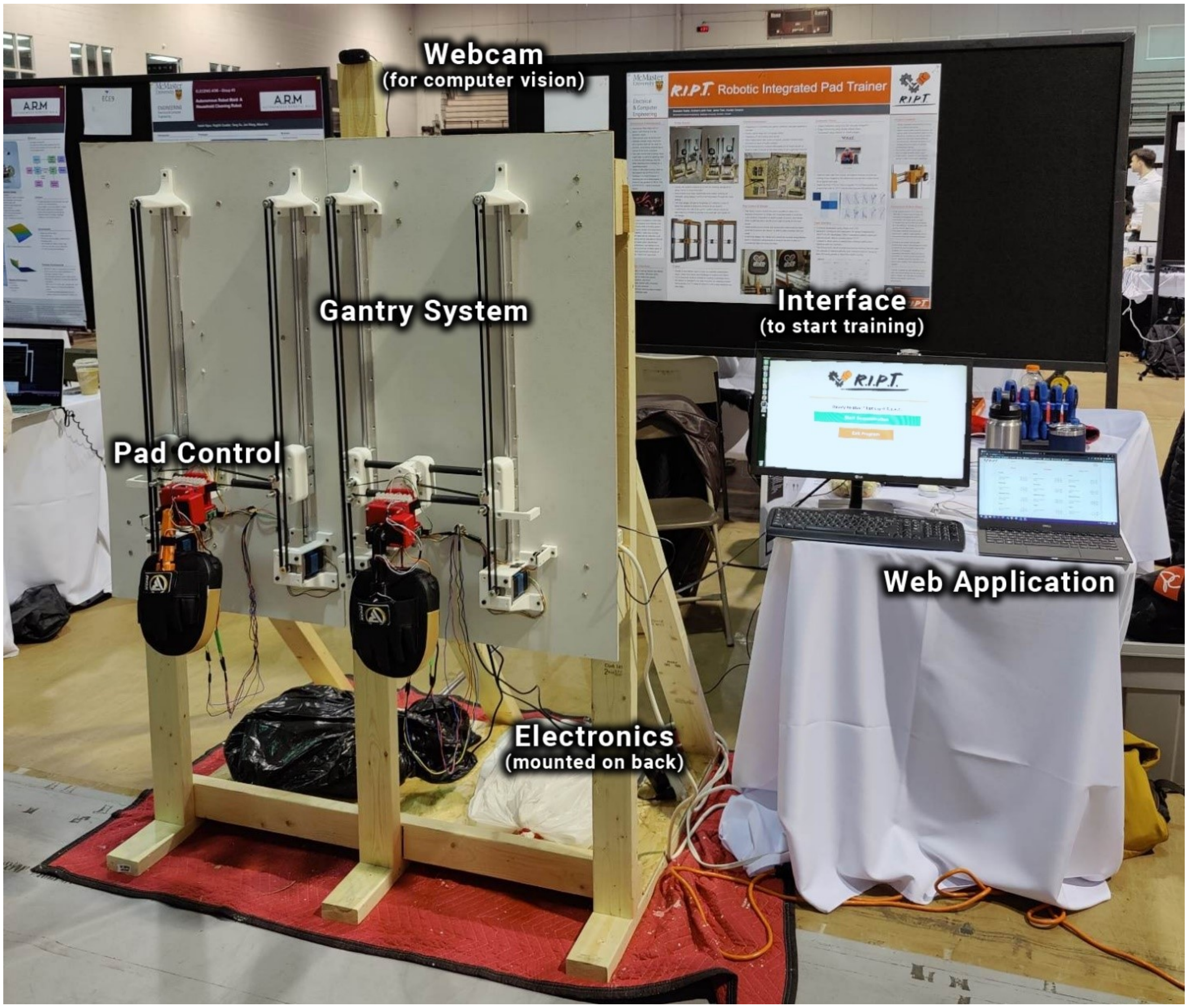

Figure 21: Overview of the project on display at the 2023 McMaster Engineering Expo

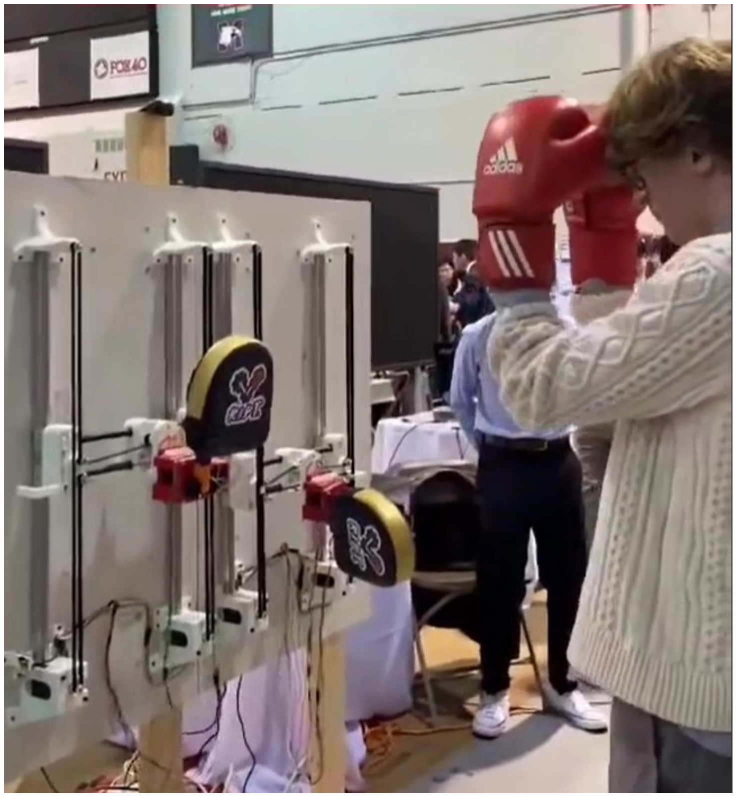

Figure 22: A photo of RIPT being demonstrated at the 2023 McMaster Engineering Expo

IV. Conclusion/Summary

As a prototype, RIPT is incredibly successful. All physical aspects of the device work as intended, and the computer vision model is fully functional. Metrics are collected on a per session basis and uploaded to an SQL server for deployment on the RIPT website.

The final result is a promising product whose full potential has yet to be released. Although the prototype is fully functional, there are improvements that could be made. The robotic arms for controlling the pad orientation are the least robust elements of the design. In their current state, they lack the strength to withstand the full force of a punch. With a larger budget, these arms could be fabricated out of steel or aluminum, and accompanied with stronger motors to improve structural integrity. In addition, using higher quality load sensors would result in substantial improvements concerning metric collection. Regarding the gantry system, the current prototype is incredibly successful. Movements are precise and the current rail system is strong. Upgrading from NEMA 24 stepper motors to brushless DC motors would increase the speed of movements. Furthermore, the computer vision model effectively identifies both faces and boxing. In its current state, the model is trained on 10,000 images and has a relatively high accuracy. Improvements to this could come in the form of additional training images, and more powerful hardware to reduce the inference time. The brains of RIPT are a Raspberry Pi 4 for motor control, and an NVIDIA Jetson Nano to run the computer vision model. Given a larger budget, the project would make use of more powerful hardware, and integrate these components on a custom-made PCB.

Overall, the group is very pleased with the outcome of the RIPT prototype and looks forward to expanding its capabilities in the future. The project was a great success at the 2023 McMaster

Engineering Expo, receiving lots of praise from faculty members, industry judges, and other students. It was a rewarding experience to have our hard work recognized and to have it inspire future engineering graduates of McMaster.

Figure 23: The project saw lots of interest at the 2023 McMaster Engineering Expo

Appendix A: H-Bot Pulley

H-bot is a belt path used by the 3D printing community [5]. It utilizes one long belt, with two fixed points on the central block (D) as shown below. Two motors (M_1 and M_2) control the movement of the belt, while a series of idler pulleys create the H design it is named for. When only one motor rotates, the central block (D) moves diagonally. If both motors rotate in opposite directions, it moves along the Y axis; if in the same direction, then it moves along the X axis. This straightforward design allows for very precise control over the movement of the block.

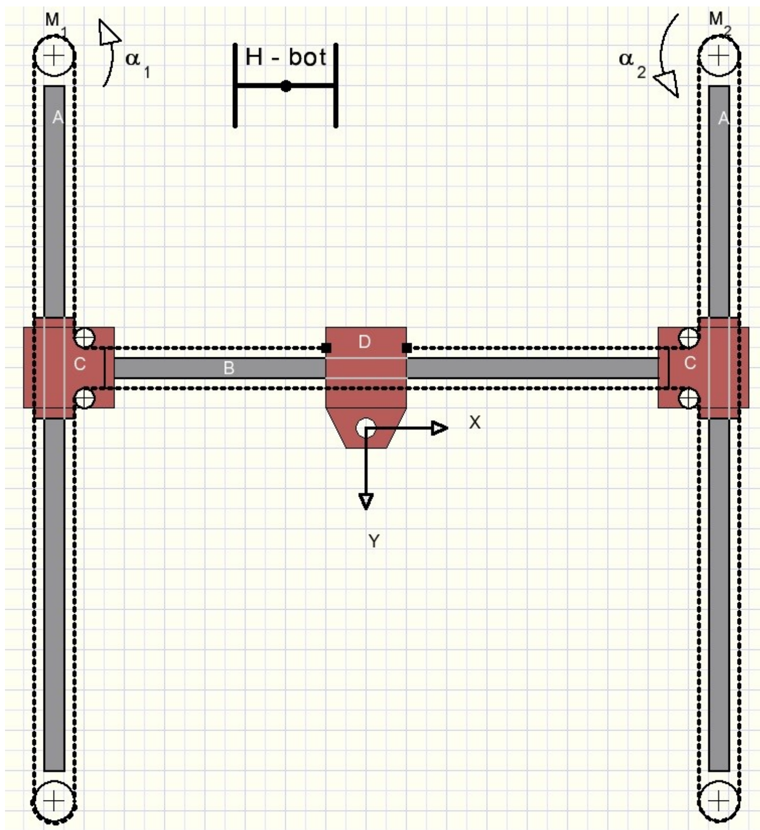

Figure 24: H-bot pulley system, courtesy of [3]

Movement of the central block (D) is described in the following table [3].

Motor 1	Motor 2	Net Movement
Stationary	Counter-clockwise	↙
Stationary	Clockwise	↗
Counter-clockwise	Stationary	↖
Clockwise	Stationary	↘
Counter-clockwise	Counter-clockwise	←
Counter-clockwise	Clockwise	↑
Clockwise	Counter-clockwise	↓
Clockwise	Clockwise	→

Appendix B: Bill of Materials

Item	Quantity	Cost	Source/Notes
NEMA 24 Stepper Motors	4	$0.00	Salvaged from co-op
TB6600 Motor Drivers	4	$54.22	Purchased from Amazon
SBR12 Steel Rails, 70cm	4	$106.28	Purchased from Amazon
SBR12UU Linear Bearings	8	$23.87	Purchased from Ali Express
Geared Pulley Wheels	4	$29.24	Purchased from Amazon
Idler Pulley Wheels	12	$39.44	Purchased from Amazon
Carbon Fibre Rods, 30cm	4	$63.47	Purchased from Amazon
4 Load Cells w/ HX711 ADC	2	$28.44	Purchased from Amazon
24V, 15A Power Supply	1	$35.99	Purchased from Amazon
MG995 Servos	4	$0.00	Already owned
3” Radial Bearing	2	$19.96	Purchased from Amazon
Boxing Pads	2	$33.89	Purchased from Amazon
Raspberry Pi 3	1	$0.00	Salved from old 3D printer
Raspberry Pi 4	1	$0.00	Already owned
NVIDIA Jetson Nano	1	$250.00	Purchased from Amazon
Screws, Bolts & Nuts	-	~ $50.00	Purchased from Home Depot
Plywood, 2x6 & 2x4 Boards	-	$0.00	Leftover from home renovations
Skateboard Bearings	2	$0.00	Already owned
10ft long ½” PVC Pipe	1	$0.00	Leftover from home renovations

The total cost of the project is approximately $734.80.

Appendix C: Code Repository

The project contains approximately 38,000 lines of code. Of this, 72.2% is Python, 18.6% JavaScript, 6.1% CSS, 2.5% C++, and 0.6% HTML.

To prevent bloating the report, all the code and CAD files have been made publicly available on GitHub at https://github.com/brlnoble/RIPT_Capstone.

Below are descriptions of the key code files contained within the repository:

MotorControl.py: Acts as the main program for controlling the stepper motors, servo motors, calling other scripts, and dictating the operation of the device.

StepperMovement.py: A custom class that provides functions for controlling the stepper motors.

ServoMovement.py: A custom class that provides functions for controlling the servo motors.

Analysis.py: Converts JSON formatted results from each session into useable metrics. Calculates averages and numerics for each punch, quadrant, and provides the basis of the adaptable training regime.

punch.py: A custom class that provides functions for creating punch objects. Assigns quadrants, types, and positions based on the probabilities assigned. Realizes the adaptability of the device.

interface_yolo.py: Implements the custom trained YOLOv7 tiny model. This is a modified file that was provided by Wang Xinyu [7]. It implements our own custom routines to perform object detection, track the users guarding with intersection over union, interact through a user interface, and send the results for processing to the Raspberry Pi.

server.js: Hosts the PostgreSQL server for the web application.

index.js: The main container for the React.js based web application.

Although this list covers key scripts and files the group created for this project, it is non-exhaustive and we emphasize that exploration of the GitHub repository will provide a better basis upon which to gauge the scope of the programming for RIPT.

APPENDIX D: GANTT CHART

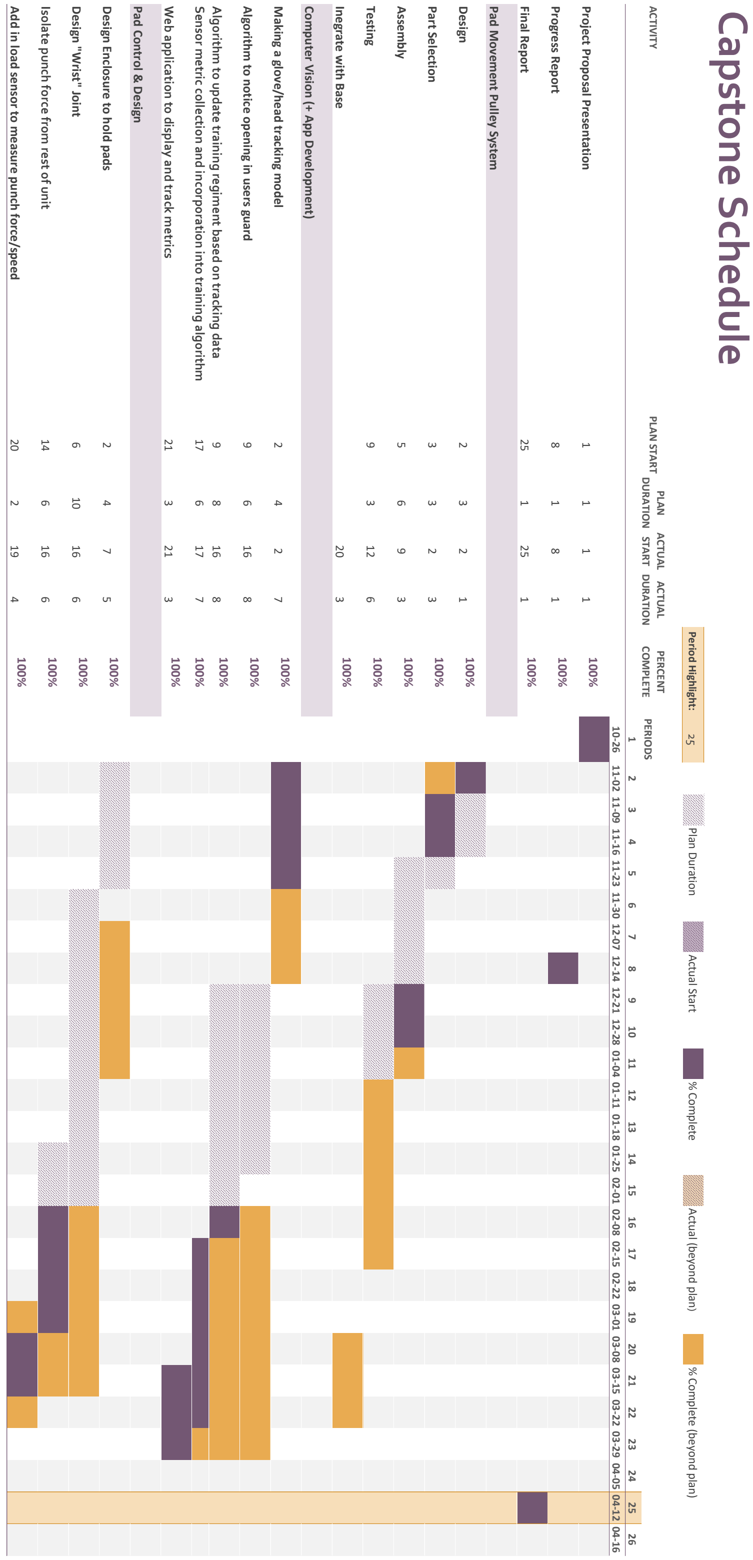

Figure 25: Gantt Chart showing the progress of the project throughout the semester

References

[1] “Real Steel,” IMDb, 07-Oct-2011. [Online]. Available: https://www.imdb.com/title/tt0433035/

[2] “Rock 'em Sock 'em Robots,” Wikipedia, 11-Sep-2022. [Online]. Available: https://en.wikipedia.org/wiki/Rock_%27Em_Sock_%27Em_Robots

[3] “What makes a great boxer?,” Boxfit UK Ltd, 22-Sep-2017. [Online]. Available: https://www.boxfituk.com/blog/what-makes-a-great-boxer/

[4] “The world's first striking robot: Stryk USA,” STRYK. [Online]. Available: https://www.strykusa.com/

[5] M. Lütkemeyer, “CoreXY vs. H-bot: Which system is better for a 3D printer?,” The 3D Printer Bee, 30-Nov- 2021. [Online]. Available: https://the3dprinterbee.com/corexy-vs-hbot/.

[6] D. E. Dirkse, “The H-Bot,” The HBOT Robot. [Online]. Available: https://davdata.nl/math/hbot-eng.html.

[7] W. Xinyu, “TensorRTX - YOLOv7,” GitHub, 29-Nov-2022. [Online]. Available: https://github.com/wang-xinyu/tensorrtx/tree/master/yolov7.