Carnegie Mellon University
Electrical & Computer ENGINEERING

PongPal

Michael Bahner, Alex Kireeff, Seung Yun Lee
Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

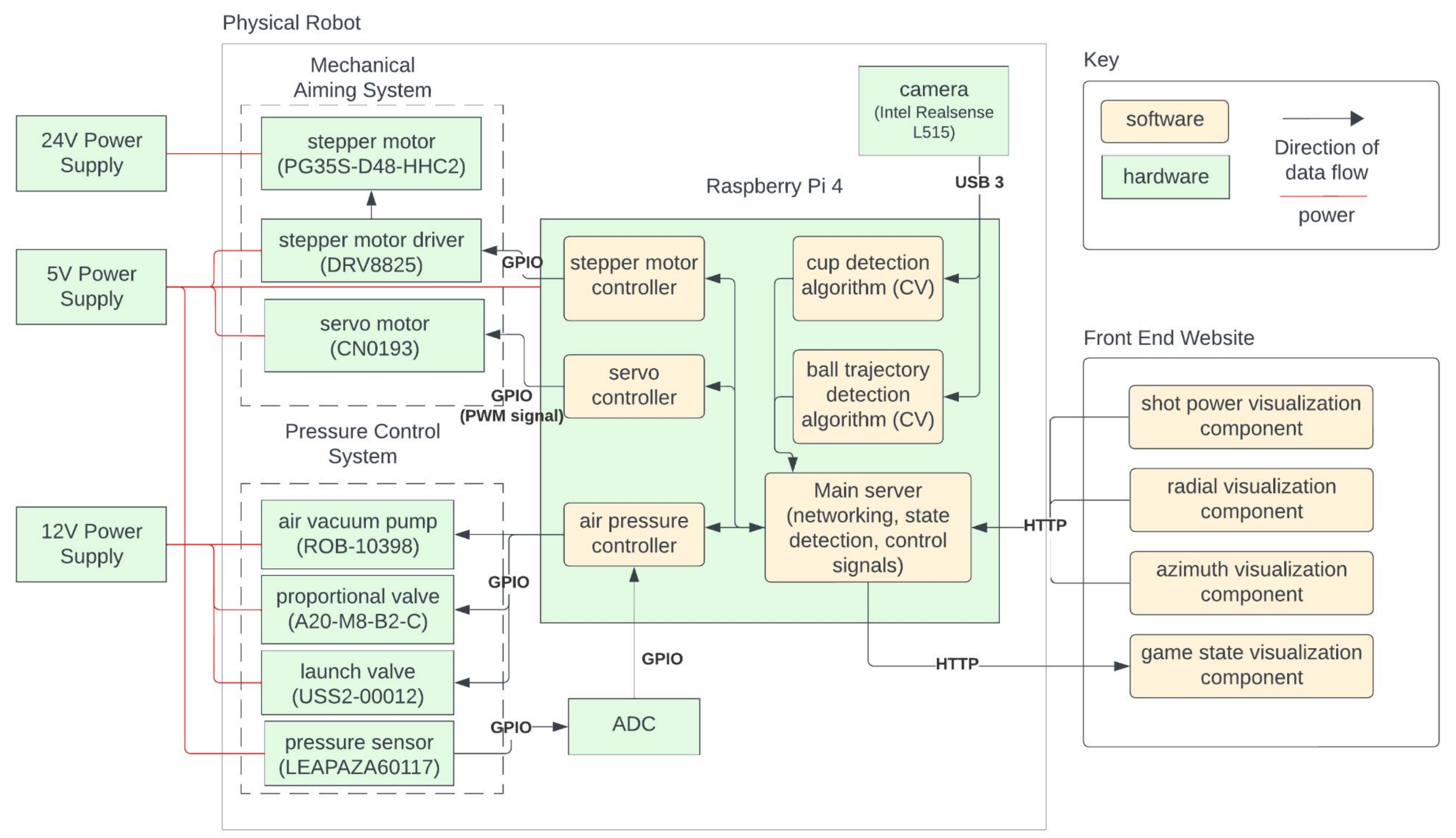
https://course.ece.cmu.edu/~ece500/projects/s24-teama8/

Product Pitch

While water pong is one of the most broadly enjoyed party games in United States, there hasn't been a platform in which a user can remotely play the game. **PongPal enables users to remotely enjoy the game of pong** by controlling the aiming and power settings, and get feedback on where the ball landed.

The project allows users to accurately control the robot and shoot a ping pong ball, so that the **variance in the ping pong balls' landing spot is less than 5cm**, which is the radius of a standard red solo cup. The feedback system also **accurately detects the balls' landing spot and the cups' true location with an error less than 5cm, within 5 seconds of firing**. These result in a pleasant game play experience that is responsive and accurate, as if the user is playing the game in real life.

System Architecture

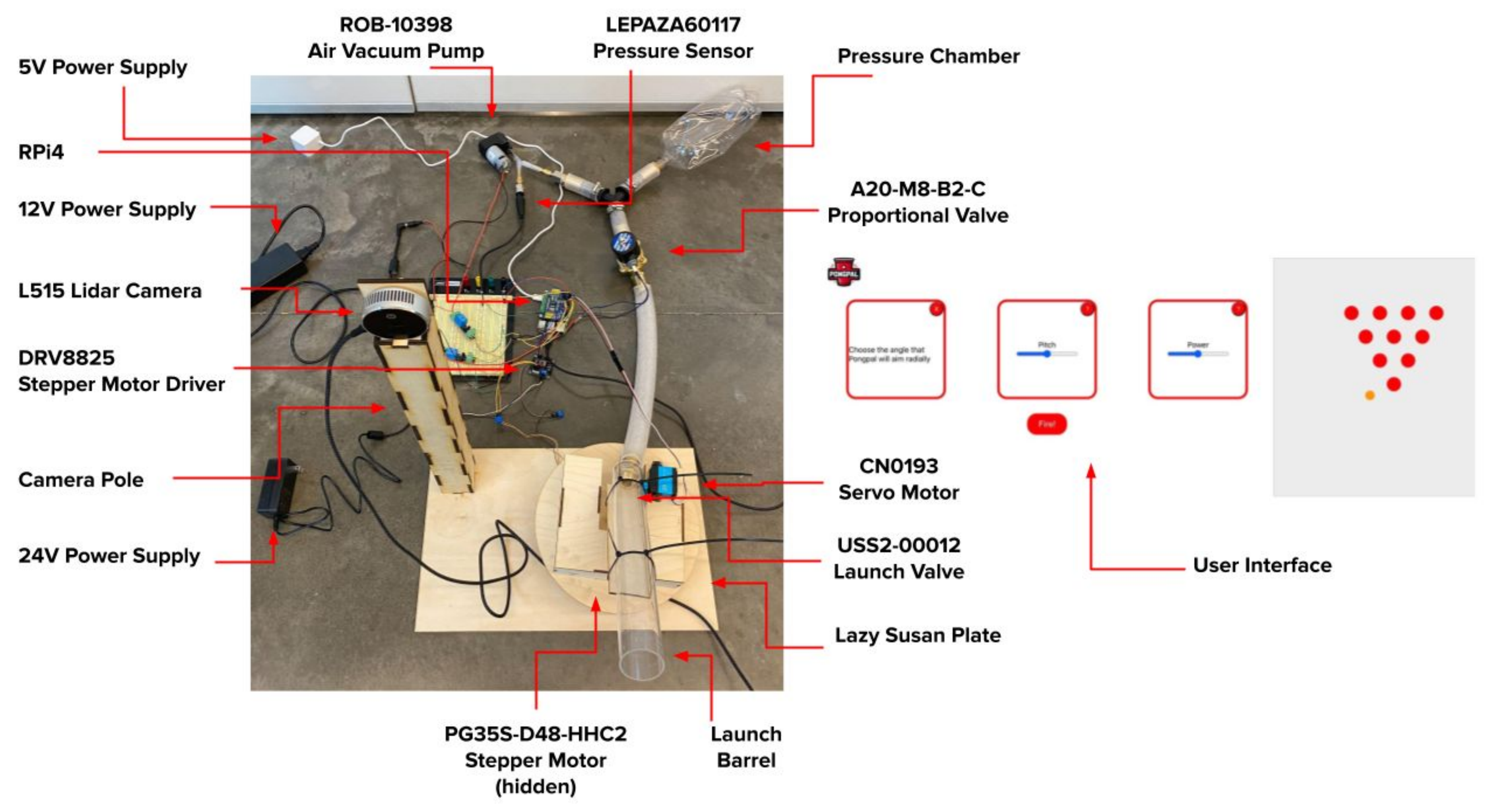

Physical Robot
Mechanical Aiming System
24V Power Supply
stepper motor (PG35S-D48-HHC2)
stepper motor driver (DRV8825)
5V Power Supply
servo motor (CN0193)
GPIO
GPIO (PWM signal)
Pressure Control System
12V Power Supply
air vacuum pump (ROB-10398)
proportional valve (A20-M8-B2-C)
launch valve (USS2-00012)
pressure sensor (LEAPAZA60117)
GPIO
ADC
camera (Intel Realsense L515)
USB 3
Raspberry Pi 4
stepper motor controller
servo controller
air pressure controller
cup detection algorithm (CV)
ball trajectory detection algorithm (CV)
Main server (networking, state detection, control signals)
HTTP
Key
software
hardware
Direction of data flow
power
Front End Website
shot power visualization component
radial visualization component
azimuth visualization component
game state visualization component

System Block Diagram

PongPal can be divided into **4 main subsystems**: aiming, vision, launching, and control.

(1) Aiming Subsystem
The aiming system can be divided into two smaller subsystems: vertical aiming using a servo motor, and radial aiming using a stepper motor. When a user sends a request to change the robot's settings, the request is received by the server hosted on Raspberry Pi 4. This request is forwarded to the individual modules that control the servo motor and the stepper motor through connected GPIO pins.

(2) Vision Subsystem
A LIDAR camera is used to make the vision system robust in all lighting conditions. Once the user decides to shoot the ball, the camera starts recording to acquire depth map. For the cup detection, we collect multiple frames of the depth make, aggregate it, and project it onto the floor to get a bird's-eye view. Then, the image is convolved to detect the circle, which is detected as a cup.



Once the ball is fired, we collect the frames in which the ball is flying, then find the contours and circles. From this, we can extract the 3d position of the ball. We then run regression on this data and drop the least explainable points until the residual error is less than 1e-3. From this, we can calculate the ball's trajectory and where it landed.

(3) Launching Subsystem
Once the user sets the desired power level, the air vacuum pump is used to increase the pressure. The coke bottle is used as the pressure chamber, as it can hold up to 120 psi which is more than enough for our use case. The current pressure reading is read by the pressure sensor (translated by ADC for the pi), and fired by releasing all the pressure at once through the launch valve.

(4) Control Subsystem
Pongpal is controlled by a locally run frontend on the client's device, and sends HTTP requests to the RPi to adjust robot's position, power level, and to read the game state. The user has control of three sliders for yaw, pitch, and power on the left side of the screen. Once the game state is detected by the CV system after the shot has been fired, that information is sent to the client and displayed on the right side of the screen.

System Evaluation

To verify the project's performance, we've performed multitude of unit testing and integration tests. The summarized metrics and the corresponding performance of the system is presented in the table on the right.

For the aiming system, we've set the angle of the yaw and pitch in their full range, varying by 10 degrees on each step. Then, the true angle was measured to calculate the average deviation. From this, we've verified that both **aiming axes' error is less than 0.5 degrees** as desired.

For the vision system, we've located the cups in the known position, shot the ball, and marked its landing location 10 times. We've then ran both cup detection and ball detection on this test setting environment to measure its deviation from ground truth, while also measuring the processing latency. For both detection algorithms, we've observed that the **detection error is less than 5cm** as desired, and the **processing latency is less than 5 seconds**.

Metrics / Test Performed	Performance
Aiming System's Angular Error	Radial aiming error **0.26 degrees**, vertical aiming error **0.14 degrees**
Cup Location Detection Error	Detection error (**x: 1.80cm, y: 1.62cm**)
Ball Trajectory Detection Error	Detection error (**x: 2.45cm, y: 4.29cm**)
Ball Landing Spot Variance	Depth variance **5.69cm**, horizontal variance **3.17cm**
CV Processing Latency	Processing latency **4.37 seconds**
Website UI/UX Rating	Overall enjoyment rating **8.6 / 10**

For the launching system, the ball was shot 10 times under the same power and aiming settings, and the variance between the shots was measured. The desired variance is under 5cm as that is the radius of a cup. The depth variance was slightly over this value, but the **horizontal variance was under 5cm** as desired.

For the website, we've performed user testing on 10 users, 5 of which are regular pong players and 5 of which are randomly selected. We've then asked them to rate their experience on a scale of 10, and got an **average enjoyment rating above 8** as desired.

Conclusions

We have successfully built a remote water pong playing platform that is enjoyable and performant. The project increases accessibility of the game by enabling remote game play. Through an intuitive website and accurate aiming/game state detection, anyone can enjoy the game only with an internet connection.

Throughout this project, we've learned about underestimating complexity, because we did it a lot. Making a robot that can precisely aim, launch, and track a ping pong ball all while maintaining an intuitive website proved to be more difficult than expected. Moreover, with so many moving parts, integration posed a huge challenge and design adjustments. Balancing the limited budget while quickly iterating different design was another project management challenge.

Although we've implemented a functional system, more work is necessary to make this project into a production-ready system. One of the biggest roadblocks is the cost - for an accurate CV system, we've used a LIDAR camera that is prohibitively expensive (~$500) for a casual entertainment system. The launching system is also rather heavy, which reduces the accessibility. We hope the future projects iterate on our design to bring down the form factor and the cost.

Acknowledgements

We would like to thank Professor Fedder and our TA Eshita Shrawan for the support and guidance throughout the semester. We would also like to thank the TechSpark staff for guidance on mechanical joints for the launching subsystem.