Objective

Automatically Fetch the Ball
Save Your Time during Training
**Power Subsystem**
- 11.1V battery connected to 2 buck converters
- Single 5v Pi battery

**Control Subsystem**
- Microcontroller connected to 2 H-Bridges and to 2 Servo Motors
- H-Bridges used for bidirectional control of motors

**Sensor Subsystem**
- 2 Ultrasonic sensors (Microcontroller)
- 2 Cameras (Raspberry Pi cameras)

**Fetching Subsystem**
- Raspberry Pi used for image processing and state machine control
**Microcontroller**

- ATMega328
- Basic loop for reading sensors and updating motors

**Runtimes (us)**
- WAIT: 432.88
- CHASE: 368.21
- ACQUIRE: 423.21
- FETCH: 402.27
- RETURN: 396.78
Microcontroller
- PWM outputs to motors
- Digital signal to/from Pi

Servo Motors
- Used for control of pincers

DC Motors
- Used for the wheels
- Controlled by DRV8848 H-Bridge
Buck Converters

- Misunderstanding of data sheet
- Lack of converter meant we needed battery with lower voltage

Power Source

- Originally 11.1V supply
- Success with 5V supplies
- Can run for 1-1.5 hours
Sensor Subsystem

Ultrasonic Sensors
• Placement of 2 sensors on front for ball and wall detection
• Placement of 1 sensor on the back for detection of waypoint and user

Cameras
• 1 camera in the front of the car and 1 camera in the back
• Front camera has narrower FOV than camera in the back
Success Rate
• Mostly reliable in clear environment and 3 meters
• At least 3 fps at 1080P
• 90% within 0.5 meters, 50% at 3 meters

Limitations
• Different lighting conditions
• Complex environment
Successes

**Sensor subsystem**
- Color detection to track the ball, user, and wait point

**Fetching subsystem**
- Displayed expected behavior for each state

**Control Subsystem**
- Moved the motors as expected for each state of the design software

**Power Subsystem**
- Pi Sugar battery supplies enough power to meet high-level requirements
Failures

Sensor subsystem
- Camera can't detect well in messy environment / long range

Control Subsystem
- Unexpected delay when power on

Power Subsystem
- Unable to utilize a second power supply

Fetching subsystem
- Unable to debug with field testing
- Unable to increase granularity of image processing code
Ultrasonic Sensors

- Tennis Ball Absorbs Sound
- Consistency varied with Distance and Positioning
- Attempt to use inconsistency (Partially Worked)
Future Work

- Consider Lidar as an alternative
- Parallelize Raspberry Pi code
- Avoid using python for non-image-processing tasks
- AI model for better edge case handling
- Replace the PiSugar battery with the PCB battery
- Space and weight for optical zoom lenses
- Hybrid Detection Method
- Camera Manual Exposure
Lessons Learned
- Plan and review for requirements & verification more carefully
- Take the limitations of key components into closer consideration
- Prepare for hardware failure by including redundancies in the PCB design

What should've been done differently
- Connect the Pi to the output of extra buck converters to simplify the power subsystem.
- More research on the PCB components
- Earlier work on System Integration