

LASER TRACKING GIMBAL

By

Aaron Zhang

Colin Smithsuvan

Jayden Kim

Final Report for ECE 445, Senior Design, FALL 2026

TA: Wesley Pang

Team 19

Abstract

This project uses a vision-guided two-axis gimbal to autonomously track a soccer ball for real-time on-device streaming and inference. A camera mounted on the gimbal will stream frames to the laptop, where an object detection model will identify the ball and estimate its position. The laptop will convert this information into movement commands, which are transmitted to a custom PCB via Ethernet (or CAN). The PCB acts as the main control unit to orchestrate the brushless motors and the closed feedback loop for moving the gimbal. The final product will have the gimbal successfully moving with the movement of the soccer ball, with a laser pointing at the ball at all times.

Contents

1. Introduction.....	1
1.1 Section head.....	1
2 Design.....	2
2.1 Hardware.....	2
2.1.1 Power.....	2
2.1.2 Motor and Motor Controls.....	2
2.1.3 Communications.....	2
2.1.4 Microcontroller.....	2
2.2 Software.....	2
2.2.1 Detection Inference.....	2
2.2.2 Preprocessing and Postprocessing.....	2
2.2.3 Communication.....	2
3. Design Verification.....	3
3.1 [Component or Block].....	3
3.1.1 [Subcomponent or subblock].....	3
4. Costs.....	4
4.1 Parts.....	4
4.2 Labor.....	4
5. Conclusion.....	5
5.1 Accomplishments.....	5
5.2 Uncertainties.....	5
5.3 Ethical considerations.....	5
5.4 Future work.....	5
References.....	6
Appendix A Requirement and Verification Table.....	7

1. Introduction

Modern day sports broadcasting typically relies on a camera man to track the ball movement throughout the game. This makes it difficult to sustain it for a long time in addition to capturing fast paced movements across milliseconds. Furthermore, this is especially difficult in settings where

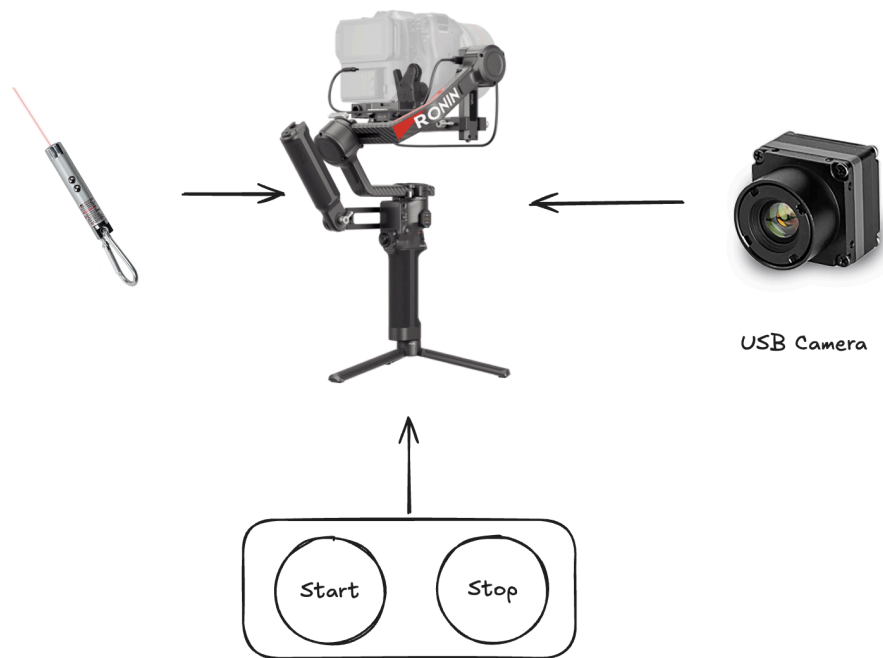
lower-production is required with fewer staff possibly missing out on capturing crucial moments within a game. This project can help smaller teams with fewer staff produce more consistent footage. However, our project still requires human operators and is intended to work in tandem with human operators rather than eliminate oversight. Finally, the gimbal must have safety constraints of a physical emergency stop, and it should never be aimed at people.

1.1 Solution

Our solution is a vision-guided camera platform that autonomously keeps a ping pong ball near the center of the camera frame. A USB camera mounted on the two-axis gimbal captures a live video stream. A laptop runs a real-time object-detection model to identify the soccer ball and calculate the difference between the ball's bounding-box center and the camera-frame center. This difference in space, along with knowing the offset between the laser and camera mounts, allows us to correct the laser to point toward the ball more accurately.

The laptop communicates movement commands to an STM32 microcontroller on a custom PCB. The microcontroller reads motor encoder feedback, drives the motors through the inverter stage, and returns telemetry to the laptop. The firmware will implement several states, such as a searching state, initialization, calibration, idle, tracking, and fault states. Camera calibration will calculate the frame center through the camera's intrinsic parameters.

1.2 Visual Aid

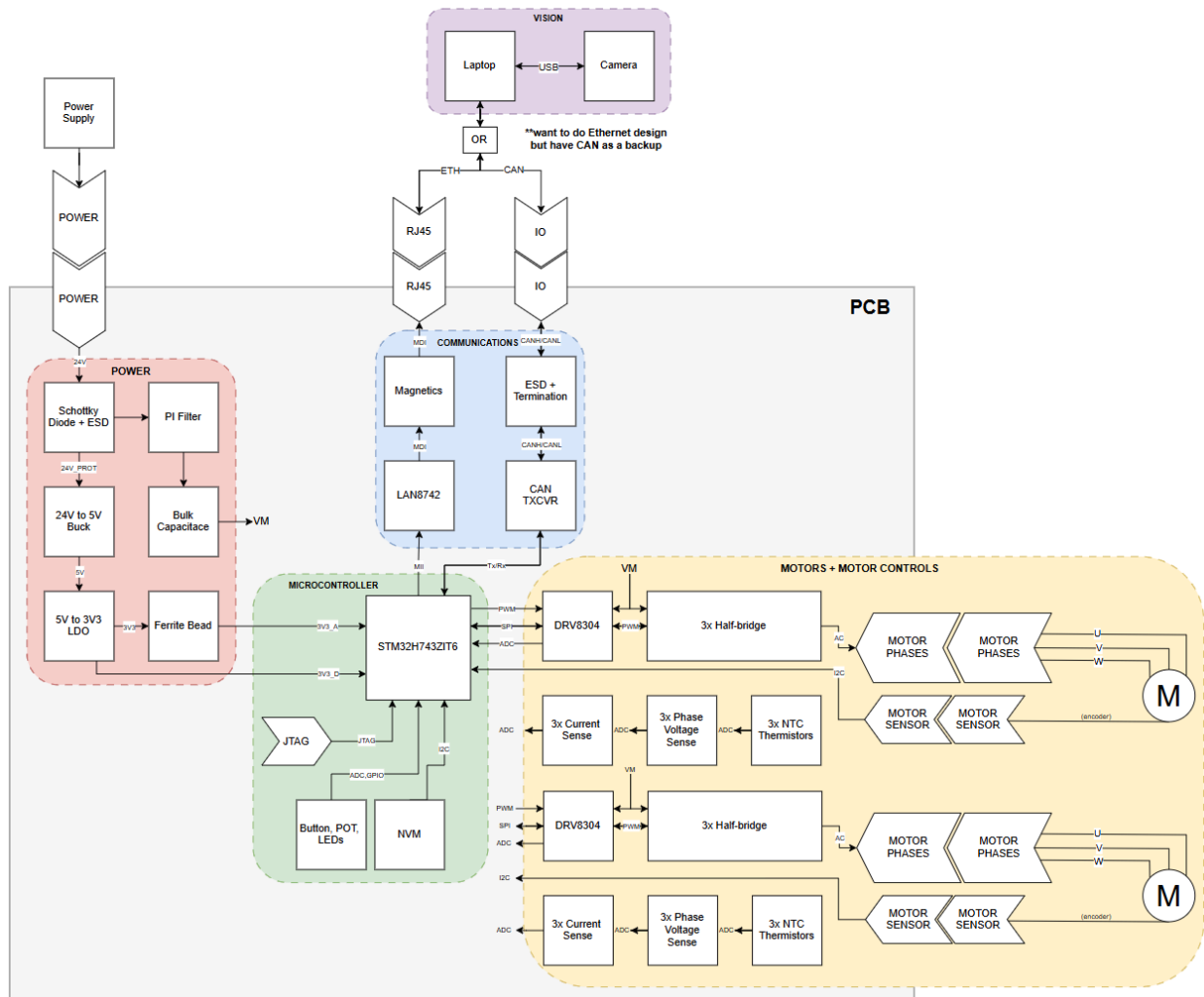


1.3 High Level Requirements

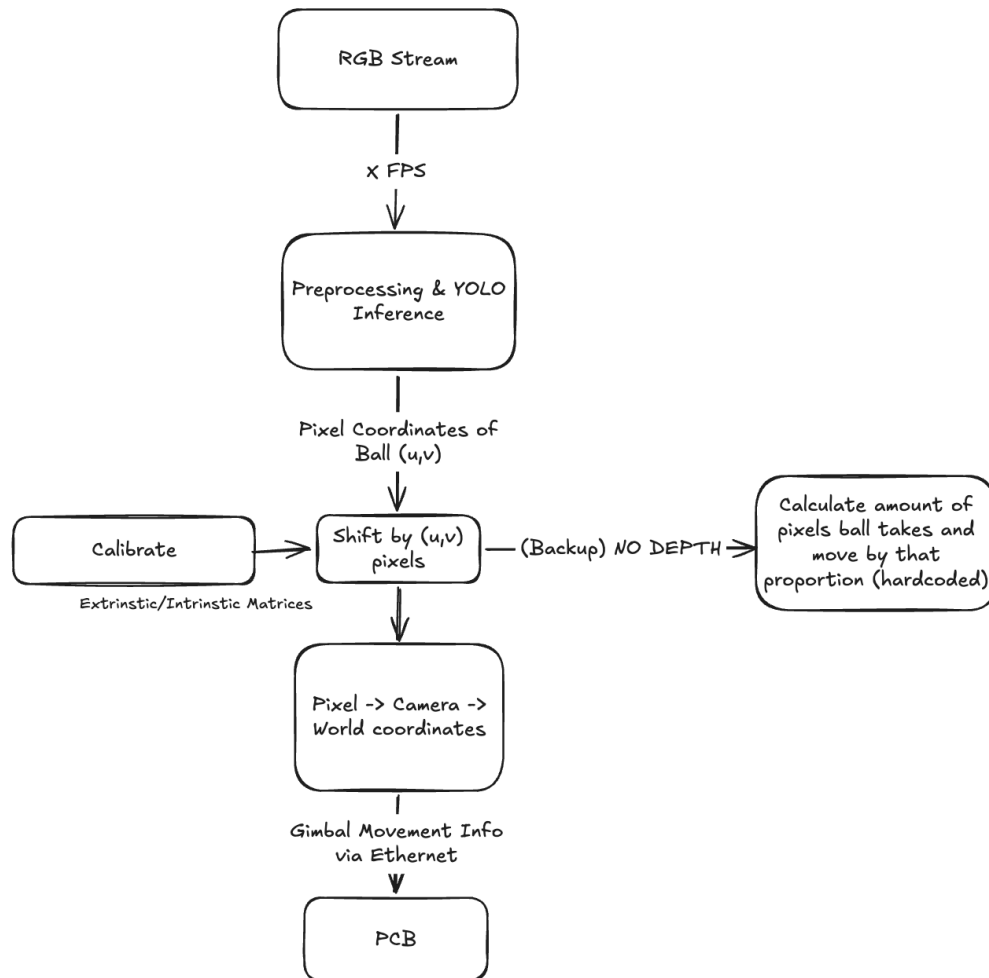
- While the ping pong ball is moving, the system should keep the ball visibly within the center 80% of the screen's ROI (Region of Interest) for at least 30 seconds.
- The system should detect the ping pong ball and update the gimbal's position at least 5 times per second (200ms).
- After the ping pong ball initially re-enters the frame, the camera system should lock on and center the ball within the frame within 1 second.

2 Design

2.1 Hardware



2.1 Vision



2.2 Subsystem Overview

2.2.1 Power

The power subsystem will be responsible for taking in 24V DC from an external power supply and serve as the main point of power conversion, distribution, and filtering within the PCB. There are two main loads on the 24V rail being the 24V inverter stage for controlling the two brushless DC motors and the 24V to 5V buck converter and its downstream loads. The 24V input should have proper capacitance for switching relatively higher power inductive loads like the two motors as well as a PI filter to prevent switching noise from the inverter from propagating back onto the 24V bus and into sensitive analog signals. The 24V rail should also have a schottky diode for reverse battery protection and a TVS diode for ESD protection. In addition to the 24V to 5V buck converter to generate a 5V rail, there will also be a 5V to 3.3V LDO for IC and microcontroller power. Depending on noise seen on the 3.3V rail, there may be a ferrite bead added to filter noise before feeding any reference voltages on the microcontroller.

2.2.2 Microcontroller

The microcontroller subsystem contains the microcontroller to run firmware and control the PCB, a flashing connector, a non-volatile memory IC, and some auxiliary buttons and potentiometers. The microcontroller subsystem interfaces with all other subsystems being the vision, communications, power, and motor + motor controls subsystems. The microcontroller will run the pcb state machine, controlling information flow, commanding the motor driver via PWM controls, and taking in 3.3V power from the power subsystem. The flashing connector is for uploading firmware to the microcontroller and debugging, the non-volatile memory IC is for storing motor position and calibration data, the auxiliary buttons are for boot modes and reset, and the potentiometer will be for initial testing of the motors for speed or position control without the communications in the loop.

2.2.3 Communications

The communications subsystem is responsible for converting board-level information into a protocol suitable for longer distance and protected against any incoming interference and interfacing with the laptop which runs the vision and will command the PCB to move the motors to a certain position. The communications subsystem currently consists of both ethernet and CAN protocol hardware. The CAN hardware serves as a backup in case the ethernet hardware is not successful. Ethernet could enable higher bandwidth and lower latency in the system. The communications subsystem interfaces between the microcontroller and vision subsystems.

2.2.4 Motor + Motor Controls

The Motor + Motor controls subsystem is responsible for converting DC input from the power subsystem into AC signals to control the external motors which will drive the position of the camera for vision tracking. The subsystem consists of a smart gate driver, inverter stage, and several sensing circuits: phase voltage and current, inverter temperature, dc-link voltage and current, as well as an encoder interface for mechanical position tracking of the motors. The subsystem interfaces with the microcontroller which sends inputs for the gate driver and takes in analog signals from the sensing circuits and I2C from the motor encoder to create a closed-loop feedback system.

2.2.5 Vision

The vision detection model is responsible for detecting the ball and providing a bounding box, then using camera intrinsic/extrinsic matrices to map from pixel coordinates to real world coordinates. These real world coordinates will then map to positional information which will be fed to the PCB from the laptop. In order to map pixel coordinates to real world coordinates, we have an option to either use different methods of pose estimation models due the monocular vision constraint, or to estimate the depth based on the proportion between how many pixels the ball takes up on the screen versus the actual diameter of the ball.

2.3 Subsystem Requirements

2.3.1 Power

High level requirement: Reliably deliver power to downstream loads; successful power conversion (24V, 5V, 3.3V rails), filtering, and capacitance design.

Design Contribution: The Power subsystem allows the PCB responsible for motor controls to be powered reliably and for the camera to constantly be tracking the ping pong ball.

Interfaces: The Power subsystem interfaces with the microcontroller and communications subsystems via 3.3V and 5V rails, and the motor and motor controls subsystem via 3.3V, 5V, and 24V rails.

Requirements

- The power subsystem should be tolerant to a 1kV ESD event
- The power subsystem should have reverse battery protection up to -50V
- The power subsystem should supply at least 500mA of current on the 3.3V rail
- The power subsystem should supply at least 500mA of current on the 5V rail
- The power subsystem should support at least 8A of current on the 24V rail
- The power subsystem should have a layout with polygons and traces sized to prevent >50C temperature rise from ambient temperature

2.3.2 Motor and Motor Controls

High level requirement: Control the two-axis motors with enough precision (within 5% accuracy) and response to meet object tracking requirements (run a ~20kHz control loop)

Design contribution: The motor and motor controls subsystem allows the system to move the camera's intended field of view to keep the ping pong ball within our specified window. The control loop of the motors will also contribute to the gimbal position frequency.

Interfaces: The motor and motor controls subsystem interfaces with the microcontroller subsystem via 6 PWM signals, a SPI interface, and 3 analog signals for phase current sensing.

Requirements

- The subsystem must have inverter MOSFETs sized for 6A continuous current draw with less than a 50C temperature rise above ambient temperature
- The subsystem must position each gimbal independently with less than 5% error of each axis' operating ranges
- The motor control loop should execute at a frequency of 20kHz
- The subsystem should accept and process new position commands at >=5Hz
- The subsystem should provide three phase current measurement, three phase voltage measurement, inverter temperature, and motor encoder position feedback for closed-loop control of each motor

2.3.3 Communications (Laptop to PCB)

High level requirement: Relay information with low latency (~5ms) between the microcontroller and laptop.

Design Contribution: The firmware converts board level logic into a differential, transformer-isolated physical layer suitable for a multi-meter cable run. It carries tracking error from the laptop to the STM32H743 on the PCB, and returns telemetry in the reverse direction.

Interfaces: The Communications subsystem interfaces with the Vision and Microcontroller subsystems via differential CAN or Ethernet MDI and either CAN Tx/Rx or Ethernet MII.

Requirements

- The subsystem must deliver tracking commands from the laptop to the microcontroller with latency of <5 ms
- The subsystem must transmit tracking commands to the microcontroller and return motor position, operating status, and fault telemetry to the laptop
- The subsystem must detect and reject messages that fail the selected protocol integrity checks and prevent rejected messages from being applied as motor commands
- The ethernet should run at $\geq 1\text{Mbps}$ or CAN FD should run at $\geq 1\text{Mbps}$

2.3.4 Microcontroller

High level requirement: Control the pcb; Run state machine, motor control loop, communicate with laptop for position information. The entire loop should run in under 200 ms to match the speed of the ball.

Design Contribution: The microcontroller converts the tracking error received from the laptop into an axis position using the camera, then closes the position and current loops against the encoder.

Interfaces: The microcontroller subsystem interfaces with the power, motor and motor controls, and communications subsystem. It should be able to Tx/Rx to the communications subsystem over RMII. It will interact with the motor and motor controls subsystem by outputting six complementary PWM signals. From the power subsystem it takes 3.3V for VDD.

Requirements

- The selected microcontroller should have an Ethernet MAC and CAN FD peripherals
- The selected microcontroller should have >2 SPI channels, 1x I2C, >16 ADC inputs, and high speed timer channels specifically for motor controls
- The microcontroller must generate six complementary PWM signals at 20 kHz with at least 10 bits of duty resolution and hardware dead-time insertion to prevent shoot-through on the inverter half-bridges
- The microcontroller must read the AS5600 angle register over I2C at 400 kHz and update the position loop at a minimum of 1 kHz, nested inside the 20 kHz current loop

- The microcontroller must store encoder zero offsets and calibration constants in non-volatile memory and validate them by checksum on boot
- The microcontroller must sample both phase-current channels simultaneously on the ADC, triggered from the PWM timer, with conversion complete within 2 μ s of the trigger so samples land in the low-side conduction window

2.3.5 Vision

High level requirement: Detect the ball and produce a gimbal position command from each frame, with total latency from image capture to command transmission under 65ms.

Design Contribution: The vision subsystem has to capture frames from the camera, detect the ball, and compute the offset between the ball's bounding-box center and the camera frame center. This is the part that makes the system autonomous.

Interfaces: The vision subsystem interfaces with the PCB subsystem via a USB connection to relay position commands from the laptop to the PCB and motor feedback and telemetry information from the PCB to the laptop.

Requirements

- Detection Inference: Detect object with >95% accuracy at 30 FPS within 50 ms of inference latency
- Preprocessing and Postprocessing: Image pre and post processing should be less than 10 ms
- Communication (Camera to Laptop): Gimbal positional movement information latency should be < 5ms when communicating from the laptop to PCB via ethernet.

2.4 Tolerance Analysis

Assuming a 640x480 image resolution at 30 FPS, the YOLOv26 object detection is running on an Intel Core Ultra 7 16 Core, which is estimated to run inference and postprocessing in around 20 ms with around a 10 ms latency from the camera through USB to the laptop. From the laptop to the MCU via ethernet, there's an estimated 2-3 ms of latency with around 20 ms of latency for the motor to receive and start moving from the data. Furthermore, there should be around a 10-15ms delay to account for additional latency bottlenecks such as waiting for frames or having an idle CPU throughput. In total, this accumulates to roughly 65 ms from image capture to visible gimbal response. This is significantly faster than our required baseline for around 200ms. However, for us to be able to capture a ping pong rally well we should assume the ball is traveling at around 4 m/s, which roughly corresponds to around 75 ms in latency. With 65 ms as a minimum and not fully being conservative, it can easily exceed this 75 ms threshold. Therefore, a primary bottleneck in this is the image capturing and image detection stage where the CPU isn't fully streaming data as efficiently as possible. To address this, we can avoid using YOLOv26 model detection as a whole, and entirely rely on classical computer vision techniques such as

contour detection where we can base it off of recognizing a circular white or yellow object, which would run in only a few ms.

3. Design Verification

3.1 Hardware

The hardware will be verified through bench testing of the power, microcontroller, communications, and motor control subsystems. The 24 V, 5 V, and 3.3 V rails will be measured under idle and loaded conditions to verify they can meet the power requirements. An oscilloscope and logic analyzer will be used to verify PWM timing, gate-drive signals, and communication interfaces, while sensing circuits will be checked against their respective ranges. Motor testing will evaluate continuous current capability, inverter temperature rise, and positioning accuracy with the camera payload installed.

3.2 Software

The vision subsystem will be verified using a labeled dataset containing soccer-ball images used to measure detection performance, with timestamps to measure inference latency and pre/post processing latency. Detection accuracy will be evaluated using intersection-over-union (IoU), and the system will be tested on live video to measure end-to-end latency from image capture to gimbal command generation.

3.2.1 Vision Detection Verification

At least 50 test frames will be collected across different distances, ball positions, and backgrounds. For each frame, the IoU will be calculated by finding the area of prediction intersected with the area of the ground-truth bounding box, divided by the union of the predicted and ground-truth bounding boxes. A detection is correct if its IoU is ≥ 0.5 . The system is considered complete if it achieves at least 95% precision, processes a video at a minimum of 5 updates per second, and produces each result within 50 ms.

4. Costs

Make sure that any tables of costs are numbered, given titles, and cited directly in the text.

4.1 Parts

Listing parts that would not already be included in available parts, would either be necessary to complete the project or useful for firmware development and integration work.

Table X Parts Costs

Part	Manufacturer	Retail Cost (\$)	Bulk Purchase Cost (\$)	Actual Cost (\$)
Motor Eval Board	TI	\$110	\$110	\$110
Motors	DFRobot	\$30	\$60	\$60
Total				\$170

4.2 Labor

Aaron Zhang	Jayden Kim	Colin Smithsuvan
Vision object detection and high-level firmware logic between the camera, laptop, and PCB. Assist with mechanical design.	Low-level firmware logic, including drivers for all hardware components. Assist with mechanical design.	Electrical hardware design and integration for PCB, motors, etc. Assist with mechanical design.

5. Conclusion

The laser tracking gimbal combines computer vision, embedded control, motor control, power electronics, and communication into one system that is capable of automatically tracking a moving ping pong ball. The camera and laptop will be able to determine the ball's location relative to the the center of the image, and the STM32 microcontroller will be able to use this information to control the motor and two-axis gimbal.

5.1 Accomplishments

The design will establish a complete system architecture and divide the project into five major subsystems which include vision, power, motors, communications, and the microcontroller. The design also defines the communication path between the laptop and the PCB, encoder based motor feedback,

and the firmware state machine needed for initialization, calibration, searching, tracking, idle, and fault handling.

5.2 Uncertainties

The 65 ms latency mentioned in the section 2.4 is based on estimations, and USB buffering and OS scheduling delays in a typical laptop can reasonably account for about 10 ms of leeway in relation to the 75 ms threshold. The classical-vision fallback can get back that margin, but it has not yet been tested with appropriate backgrounds wherein floor markings and lighting make it impossible to obtain a solid signature from a white circular object. The performance of the detection system remains untested, as there is no labeled dataset available, while a model trained on COCO had soccer balls in use instead of a ping-pong ball which comprises few pixels and is blurred when rallying process takes place. Since the gimbal setup is not implemented, there is no way to prove that the required accuracy of 5% can be achieved under dynamic load. Finally, the whole problem of laser-camera parallax deals with going to a particular distance, while the issues of Ethernet latency, inverter noise influence, and occlusion are still in the air.

5.3 Ethical considerations

As prescribed by the IEEE Code of Ethics, we prioritize public safety in our design. The system uses a Class 2 visible laser with a power below 1 mW, in compliance with ANSI Z136.1 and IEC 60825-1. The principles of Section 1 prevent the laser beam from being pointed at any person. Mechanical halts on the tilt mechanism stop the beam from dropping down to head position, and the firmware set limits play the same role. The choice of testing location is made with respect to specular reflections. The pinch components of the gimbal are managed according to standard scientific practice and the fault condition disables the motors. The camera images are printed only when a consent is given and deleted after the work is done; no face recognition is done.

5.4 Future work

Possible future work includes increasing the degrees-of-freedom by extending it to an actual arm instead of a simple gimbal. Furthermore, we can improve the overall performance of the system and track faster-moving objects, such as ping pong balls or golf swings, which are significantly faster than passing a soccer ball. Furthermore, additional algorithms such as Kalman filters could be utilized to predict the trajectory of the ball to account for object occlusion when the object is blocked by players so it doesn't miss a single frame. Finally, the project can be expanded from simply tracking a ball between two players to an entire field at once while only needing one camera view.

References

- [1] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” in *Computer Vision—ECCV 2014*, Lecture Notes in Computer Science, vol. 8693, Springer, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48. Available: https://doi.org/10.1007/978-3-319-10602-1_48
- [2] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO Documentation,” Ultralytics. Available: <https://docs.ultralytics.com/>
- [3] G. Bradski, “The OpenCV Library,” *Dr. Dobbs’s Journal of Software Tools*, 2000. Available: <https://opencv.org/>
- [4] Z. Zhang, “A Flexible New Technique for Camera Calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. doi: 10.1109/34.888718. Available: <https://doi.org/10.1109/34.888718>
- [5] OpenCV, “Camera Calibration,” *OpenCV Documentation*. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Appendix A Requirement and Verification Table

Table of System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Reliably deliver power to downstream loads; successful power conversion (24V, 5V, 3.3V rails), filtering, and capacitance design.	1. Set up each power rail with an e-load and draw the max expected power for each rail, confirming that each can be supplied properly 2. Measure noise on filtered rails to confirm absence of high frequencies	N
2. Control the two-axis motors with enough precision (within 5% accuracy) and response to meet object tracking requirements (run a ~20kHz control loop)	3. Use potentiometer interface to control motors without communications in the loop 4. Confirm mapping between potentiometer positions read via ADC and relative position of motors independently	N
3. Detect the ball and produce a gimbal position command from each frame, with total latency from image capture to command transmission under 65ms.	5. Point the camera at a millisecond timer on a second screen and log the time read from the frame against the host clock	N
6. Relay information with low latency (~5ms) between the microcontroller and laptop.	7. Send dummy message from laptop to PCB via CAN or Ethernet to confirm communications link 8. Send dummy message from PCB to laptop via CAN or Ethernet to confirm communications link	N
9. Control the pcb; Run state machine, motor control loop, communicate with laptop for position information. The entire loop should run in under 200 ms to match the speed of the ball.	10. Verification is confirmed in individual bench tests between microcontroller and its interfaces like motor controls and communications 11. An integrated test will verify end to end success - use logging to the laptop to verify speed and latency of the system	N

