Project Report

Kun Shan University of Science and Technology

Department of Mechanical Engineering & Department of Intelligent Robotics Engineering

Project title:

Projectile Modelling and Control of Vision-Based Target Detection Dual (Autonomous and Manual) Mode Baseball Pitching Robot

Submitted by:

Nelo Matse - 馬尼駱
Thulane Mamba - 涂拉尼
Bongumusa Mhlanga - 莫藍加

Project Advisor:

徐孟輝

Abstract

This project presents the development of a dual-mode (autonomous and manual) holonomic-drive baseball pitching robot entitled **"Projectile Modelling and Control of Vision-Based Target Detection Dual Mode Baseball Pitching Robot."** The system integrates a Kiwi (tri-omni) drive architecture to achieve unrestricted planar mobility, a dual-wheel pitching mechanism actuated by 10000-RPM BLDC motors, and a real-time computer-vision pipeline employing an IMX219 camera, Raspberry Pi 5, and YOLOv8 executed on a Jetson Orin Nano. Vision-based detection of the standardized 3×3 strike-zone target supplies distance, elevation, and coordinate measurements to a projectile modelling framework that incorporates aerodynamic drag, lift, and the Magnus effect to determine the required wheel speed, launch angle, and spin parameters for accurate foam-ball trajectories.

The mechanical, electrical, and software subsystems operate cohesively to support precise navigation, automated alignment, and controlled pitch execution under competition-oriented constraints. Autonomous mode conducts full target acquisition, trajectory computation, and launch sequencing, whereas manual mode facilitates testing and operator-guided control. Experimental calibration and validation indicate consistent shot placement across grid regions and strong correspondence between model-predicted and empirical trajectories. Overall, the work provides a comprehensive implementation of a vision-guided projectile-launching robot, demonstrating the integration of computer vision, dynamic modelling, and mechatronic system design in an applied robotics context.

Beyond the specific application of competitive foam-ball pitching, this system serves as a platform for broader investigation into real-time vision-based control, ballistic optimization under aerodynamic uncertainty, and multi-modal robotic autonomy. The methodologies developed—particularly the coupling of deep-learning target detection with predictive projectile modelling and holonomic motion control—provide a transferable framework for educational, research, and industrial robotic applications requiring precise actuation guided by real-time sensor feedback.

Table of Contents

Introduction

1.1 Background and Context

The design of autonomous robotic systems capable of executing precise projectile-launching tasks represents a significant multidisciplinary challenge in modern mechatronics. Achieving accurate projectile delivery requires the coordinated integration of high-fidelity sensing, dynamic modelling, real-time computation, and controlled actuation. In particular, systems that must visually detect a target, determine spatial coordinates, and compute appropriate launch parameters rely on robust computer-vision pipelines coupled with predictive ballistic models that account for aerodynamic effects such as drag, lift, and the Magnus force.

Holonomic-drive mobile platforms offer unique advantages for such tasks, as their unrestricted planar mobility allows for fine-grained positional adjustments and orientation control. The Kiwi (tri-omni) drive configuration provides full 360-degree maneuverability, enabling the robot to align precisely with a designated target zone before executing a launch. When paired with a dual-wheel pitching mechanism driven by high-speed BLDC motors, the system forms a flexible architecture capable of generating varied ball velocities, spins, and trajectories suitable for controlled foam-ball projection.

Recent developments in embedded vision and edge computing further enable real-time target acquisition and decision-making. The integration of an IMX219 camera, Raspberry Pi 5, and a YOLOv8 model running on a Jetson Orin Nano allows the robot to detect a defined 3×3 strike-zone target and extract essential spatial information, including distance, elevation, and impact coordinates. These inputs feed into a projectile model that estimates the required wheel RPM, launch angle, and spin characteristics needed to achieve accurate placement under real-world conditions.

This project positions the robot as a standalone research and development platform for investigating vision-based targeting, autonomous navigation, and dynamic projectile control. Its design addresses key challenges associated with lightweight projectile behavior, real-time perception under variable lighting and positioning, mechanical stability during high-speed actuation, and the integration of heterogeneous hardware subsystems. More broadly, the system reflects relevant applications in automated ball-delivery systems, vision-guided material handling, mobile robotic testing platforms, and experimental studies of aerodynamic modelling and control.

1.2 Problem Statement

Accurately launching lightweight projectiles to predefined spatial targets requires reliable perception of target position, precise mobility for alignment, and a dynamic model capable of compensating for aerodynamic effects. Existing small-scale robotic systems often lack integrated solutions that combine real-time vision, holonomic motion, and physics-based control. This project addresses the need for a compact, self-contained robot capable of detecting a target, computing the appropriate launch parameters, and executing consistent, accurate shots using an integrated mechatronic architecture.

1.3 Project Objectives

1. Develop a dual-mode (autonomous and manual) holonomic-drive robot capable of precise projectile launching.
2. Integrate real-time vision-based target detection using an IMX219 camera, Raspberry Pi 5, and YOLOv8 on a Jetson Orin Nano.
3. Implement a projectile model accounting for drag, lift, and the Magnus effect to compute launch angle, wheel RPM, and spin.
4. Design mechanical and electrical subsystems, including a dual-wheel pitching mechanism and omni-directional drive, to support accurate and repeatable shots.
5. Validate system performance through calibration and testing to ensure consistent trajectory execution.

1.4 Scope and Limitations

Scope:

This project encompasses the design, development, and testing of a standalone dual-mode baseball-pitching robot capable of both autonomous and manual operation. The robot integrates holonomic-drive mobility, vision-based target detection, and physics-based projectile modeling to achieve precise ball delivery. Its mechanical subsystems include a dual-wheel pitching mechanism, an adjustable pitch-angle assembly, and a ball-feed system, while the electrical and control subsystems comprise BLDC motors, ESCs, microcontroller-based actuator management, and real-time vision processing via an IMX219 camera, Raspberry Pi 5, and YOLOv8 running on a Jetson Orin Nano. The software framework includes trajectory computation, motion control, and system integration for seamless operation between perception and actuation. The project also covers calibration procedures, experimental testing, and validation of shot accuracy across the predefined target area.

Limitations:

The system is optimized for lightweight foam baseballs (7 g, 70 mm diameter) and a fixed 3×3 target grid, limiting its adaptability to other projectile types or larger-scale targets without significant redesign. Environmental conditions such as variable lighting, surface texture, or airflow may affect visual detection and trajectory accuracy. The holonomic drive, while providing precise planar mobility, is limited by motor torque and structural stability under high-speed maneuvers. Additionally, the autonomous mode relies on pre-calibrated parameters and may require recalibration for substantial changes in target distance or orientation. Finally, the current design assumes controlled indoor operation; deployment in dynamic or outdoor environments would necessitate additional sensing, compensation algorithms, and mechanical reinforcement.

1.5 Significance and Expected Impact

The development of this dual-mode, vision-guided baseball-pitching robot demonstrates the integration of mechanical design, embedded electronics, and advanced software for real-time autonomous control. By combining holonomic mobility with precise projectile modeling, the system provides a practical platform for studying the interactions between perception, planning, and actuation in mobile robots.

The project has broader significance in fields that require accurate, sensor-guided actuation, such as automated material handling, precision delivery systems, and experimental robotics research. It showcases the effective application of real-time computer vision, deep-learning-based target detection, and dynamic trajectory computation in a compact, self-contained system. Furthermore, the modular design and dual operational modes

offer flexibility for educational demonstrations, research experimentation, and iterative development of autonomous robotic systems.

Overall, the project contributes to advancing knowledge in vision-based control, mechatronic integration, and the practical implementation of predictive dynamic modeling, providing a foundation for future research and applications in precision robotics.

2. Literature review

2.1 Overview of Autonomous Projectile Launching Robots

Projectile-launching robots are specialized robotic systems designed to detect targets, compute optimal trajectories, and deliver projectiles with precision. These systems integrate real-time perception, dynamic modeling, and precise actuation to achieve consistent and accurate projectile placement. For dual-mode robots—capable of both autonomous and manual operation—this requires seamless coordination between human input, sensor feedback, and motion control.

Vision-based target detection is critical for determining target position, distance, and orientation, which informs the projectile model responsible for calculating launch parameters such as wheel speed, launch angle, and spin. Holonomic-drive mobility platforms, such as the Kiwi (tri-omni) configuration, facilitate precise alignment and positioning, enabling the robot to adjust its orientation and location to achieve optimal launch conditions.

This type of system addresses challenges including the variability of projectile dynamics, aerodynamic effects such as drag and Magnus forces, mechanical stability during high-speed launches, and the integration of heterogeneous subsystems. The design and implementation of a standalone dual-mode baseball-pitching robot provide a practical framework for studying real-time projectile control, vision-guided targeting, and integrated mechatronic system performance, with applications extending to automated ball delivery, experimental robotics, and precision actuation research.

2.2 Holonomic Drive Systems in Mobile Robotics

Holonomic drive systems enable motion in any planar direction independently of the robot's orientation, providing full omnidirectional mobility. The Kiwi (tri-omni) drive, which uses three omni-directional wheels arranged at 120° intervals, allows precise control of translation and rotation simultaneously, making it ideal for tasks requiring accurate alignment with spatial targets.

The primary advantage of holonomic platforms lies in their ability to rapidly adjust position and orientation without complex maneuvers, reducing alignment errors prior to actuation. This capability is particularly beneficial for projectile-launching robots, where precise positioning directly affects trajectory accuracy.

Case studies in mobile robotics and experimental projectile systems demonstrate that holonomic drives improve responsiveness and control precision compared to conventional differential or skid-steer platforms. These findings support the selection of a Kiwi-drive system for the development of a dual-mode baseball-pitching robot, ensuring stable, agile, and accurate target engagement.

2.3 Ballistic Launching Systems

Ballistic launching systems are engineered to project objects along controlled trajectories, integrating mechanical design, actuation, and dynamic modeling. Dual-wheel or flywheel mechanisms are commonly employed to impart precise velocity and spin, enabling consistent projectile motion. Key parameters such as launch angle, wheel speed, and spin rate critically influence trajectory accuracy, particularly for lightweight projectiles susceptible to aerodynamic effects.

In robotics applications, accurate ballistic performance requires careful calibration of mechanical components and integration with predictive models that account for drag, lift, and the Magnus effect. Prior studies demonstrate that combining high-speed actuators with real-time control and feedback significantly improves shot consistency. For vision-guided projectile robots, such as the dual-mode baseball-pitching system, these principles ensure that the calculated launch parameters align with empirical outcomes, supporting repeatable and precise targeting.

2.4 Computer Vision for Target Recognition

Computer vision provides the sensing capability necessary for autonomous targeting by enabling the detection, localization, and tracking of targets in real time. Modern approaches utilize deep learning-based object detection models, such as YOLOv8, to identify target zones and extract positional information, including distance, elevation, and orientation.

In embedded robotics systems, vision pipelines are often executed on edge-computing hardware, such as the Jetson Orin Nano, paired with cameras like the IMX219, to ensure low-latency processing. Accurate camera calibration and coordinate mapping are essential for translating image data into actionable spatial information for trajectory computation.

For projectile-launching robots, real-time vision ensures that launch parameters—such as wheel speed, angle, and spin—are dynamically adjusted based on target position. Integration of vision with control and motion systems allows the robot to autonomously align, compensate for environmental variability, and maintain consistent shot accuracy across repeated operations.

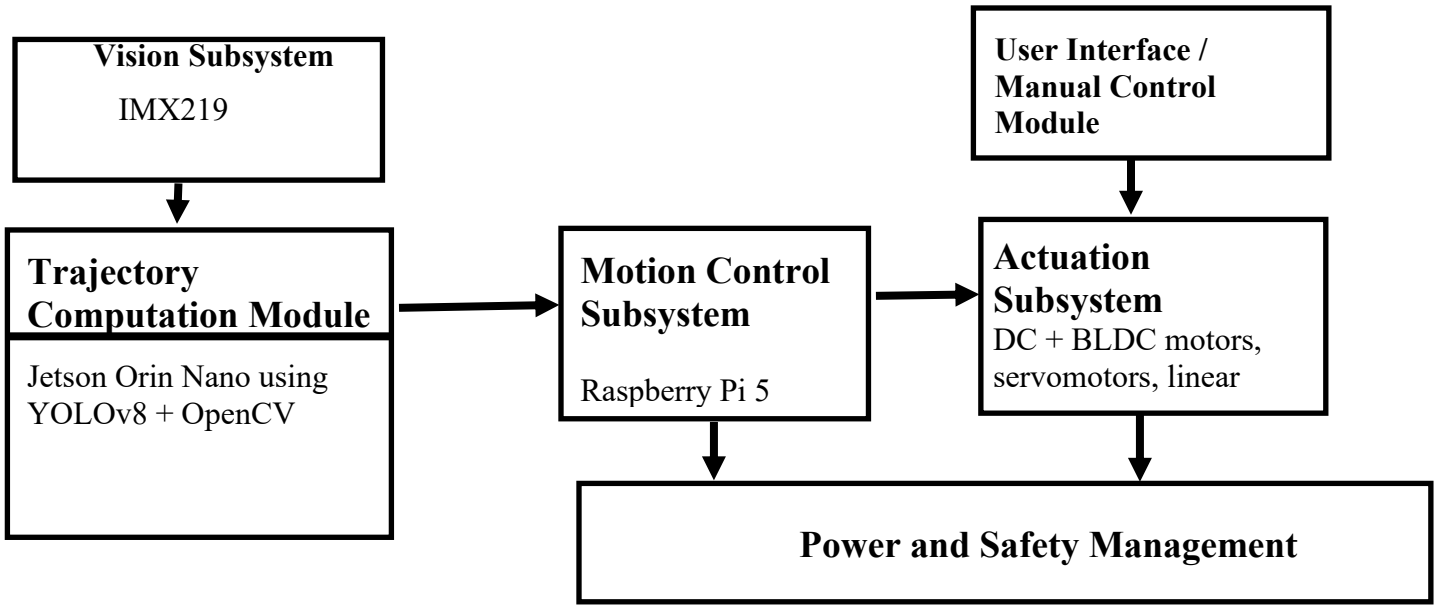
3. System Architecture Overview

3.1 High Level System Block Diagram

The high-level architecture of the dual-mode baseball-pitching robot integrates perception, computation, control, and actuation into a cohesive system. The primary functional blocks include:

1. **Vision Subsystem** – An IMX219 camera captures real-time images of the target zone, which are processed by the Raspberry Pi 5 and Jetson Orin Nano using YOLOv8 for target detection and coordinate extraction.
2. **Trajectory Computation Module** – Receives spatial target information and computes launch parameters, including wheel RPM, launch angle, and spin, using a physics-based projectile model accounting for drag, lift, and the Magnus effect.
3. **Motion Control Subsystem** – Converts target alignment and trajectory requirements into commands for the Kiwi (tri-omni) holonomic-drive platform, enabling precise positioning and orientation adjustments.
4. **Actuation Subsystem** – Includes dual 3000-RPM BLDC motors for the pitching wheels, servo-actuated pitch-angle adjustment, and a controlled ball-feed mechanism for reliable projectile delivery.

5. **User Interface / Manual Control Module** – Provides manual operation capabilities for testing or operator-guided control, allowing real-time override of autonomous commands.
6. **Power and Safety Management** – Manages power distribution, monitors system status, and enforces fail-safe behavior to ensure stable operation under all modes.



The block diagram illustrates the flow of information from target acquisition to actuation, highlighting the integration of perception, computation, and mechanical execution within a standalone, dual-mode robotic platform.

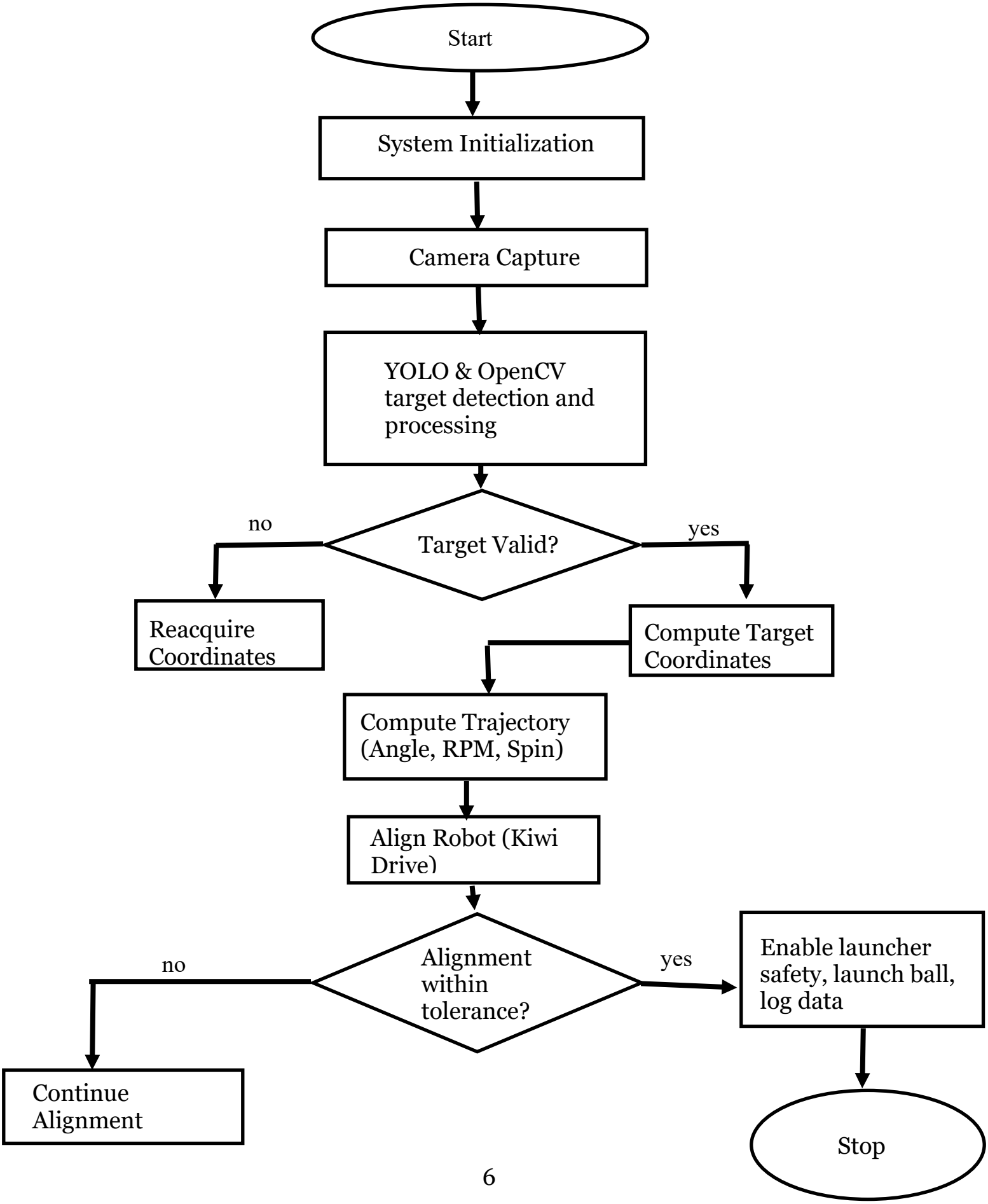
3.2 Information Flow and Control Flow

The information flow within the system begins with the **vision subsystem**, where the IMX219 camera captures real-time images of the target zone. These images are processed by the Jetson Orin Nano using YOLOv8, producing target coordinates and spatial parameters. This data is transmitted to the **trajectory computation module**, which calculates the required launch angle, wheel speed, and spin based on a physics-based projectile model incorporating drag, lift, and the Magnus effect.

The **control flow** translates computed parameters into motion commands for both the **holonomic drive** and the **pitching mechanism**. The Kiwi-drive subsystem receives velocity and orientation directives to align the robot precisely with the target, while the actuation subsystem adjusts wheel RPM, pitch angle, and ball-feed timing to execute the launch.

In **autonomous mode**, this loop operates continuously, integrating sensor feedback and dynamic corrections to maintain accuracy. In **manual mode**, operator inputs override autonomous directives, while the system provides real-time feedback on alignment and readiness. This structured flow ensures seamless coordination between perception, computation, and actuation, allowing the robot to execute accurate projectile delivery in a self-contained, dual-mode architecture.

3.3 Autonomous Operation Flowchart



4. Mechanical System Design

4.1 Chassis Design and Structural Considerations

The chassis forms the foundational structure of the dual-mode baseball-pitching robot, providing mechanical support for all subsystems, including the holonomic drive, pitching mechanism, ball-feed assembly, and electronics. The design prioritizes **rigidity, stability, and modularity** to ensure precise alignment during projectile launching and reliable operation under dynamic loads.

Constructed from lightweight yet durable materials, such as aluminum alloy and high-strength composite panels, the chassis balances structural integrity with overall weight constraints to maintain mobility and reduce inertial effects on the pitching system. Key structural considerations include vibration damping, reinforcement at high-stress points, and secure mounting provisions for motors, actuators, sensors, and control electronics.

The chassis layout also accounts for **weight distribution**, positioning the center of mass to optimize stability during high-speed maneuvers and launches. Clearance for the dual-wheel pitching mechanism and adjustable pitch-angle assembly is incorporated, ensuring unobstructed motion while facilitating maintenance and component replacement. Cable routing, accessibility for sensors, and integration with the power subsystem are also considered to maintain a compact, organized, and safe mechanical framework.

This chassis design provides a robust foundation, enabling precise holonomic navigation and repeatable projectile launching while supporting the dual-mode operation of the robot.

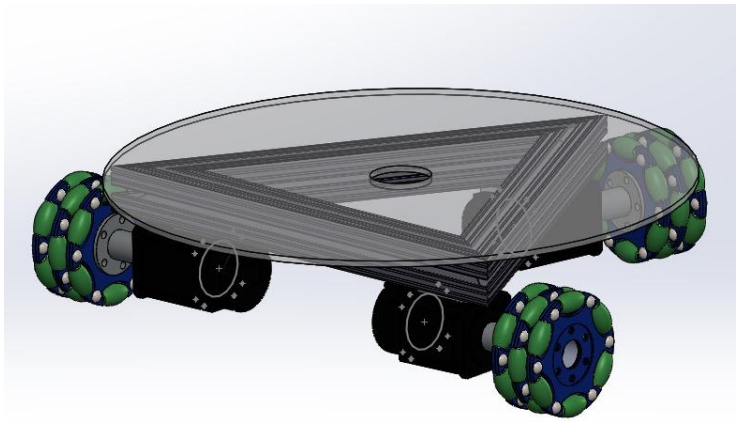


Figure 4.1(a): *Drawing of proposed chassis layout*

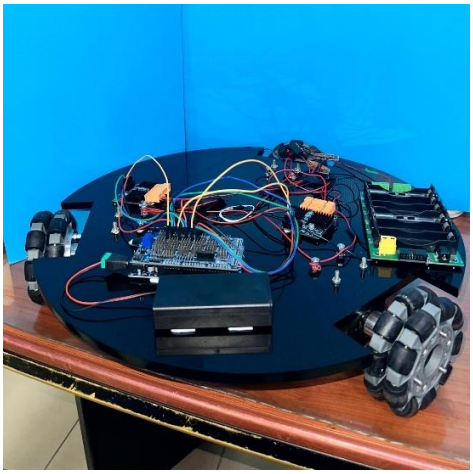


Figure 4.1(a): *Actual chassis for the robot*

4.2 Launching Mechanism

4.2.1 Holonomic (Kiwi) Drive Mechanism (Kinematics and Mathematics)

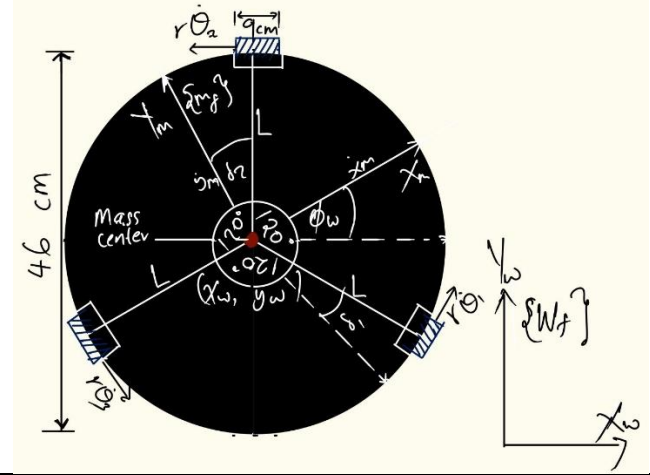


Figure 4.2.2(a): *Hand-drawn schematic of the three-wheel holonomic (Kiwi) drive configuration showing wheel placement at 120° intervals, wheel contact points, chassis diameter (46 cm), wheel width (9 cm), coordinate frames, mass center, and kinematic variables used for motion and velocity analysis.*

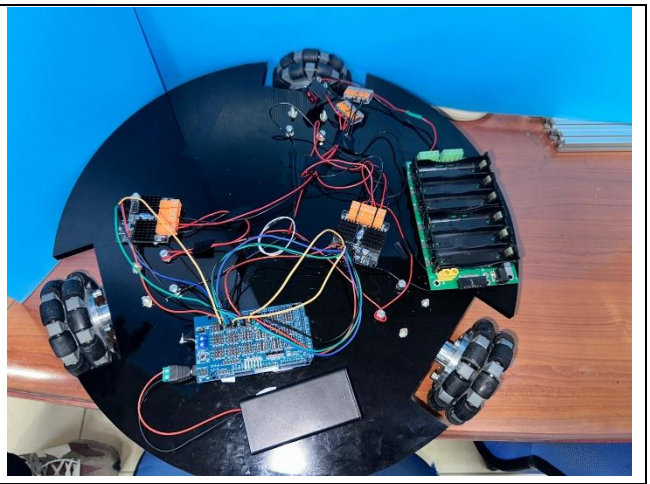


Figure 4.2.2(b): *Photograph of the assembled holonomic (Kiwi) drive base illustrating the practical realization of the theoretical wheel configuration and chassis geometry described in the kinematic model.*

4.2.1.1 Coordinate Frames and Geometry

Define:

- World frame {w}: axes (X_w, Y_w).
- Robot body frame {m}at the mass center: axes (x_m, y_m).
- Robot heading (yaw) ϕ: rotation from {w}to {m}.

Let the robot planar velocity in the body frame be:

$$v_m = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\phi} \end{bmatrix}$$

Each wheel $i \in \{1,2,3\}$is located a distance Lfrom the center, and its driving direction (the direction the wheel can generate force/velocity along, tangent to the wheel plane) is defined by the unit vector:

$$t_i = \begin{bmatrix} -\sin\ \delta_i \\ \cos\ \delta_i \end{bmatrix}$$

where $\delta_i$is the angle (in the robot body frame) from $x_m$to the wheel radius vector pointing from the center to the wheel (as commonly defined in Kiwi-drive derivations). For an ideal symmetric Kiwi drive:

$$\delta_1 = 0°, \delta_2 = 120°, \delta_3 = 240°$$

(Any rotation of all $\delta_i$by the same constant is acceptable; it just rotates the body axes.)

4.2.1.2 Velocity at Each Wheel Contact Point

The velocity of the wheel contact point is the sum of:

1. Translational velocity of the robot center $[\dot{x}_m, \dot{y}_m]^T$

2. Rotational contribution from yaw $\dot{\phi}$

Wheel position vector:

$$p_i = \begin{bmatrix} L\cos\ \delta_i \\ L\sin\ \delta_i \end{bmatrix}$$

Rotational velocity at that point (planar rigid body):

$$v_{rot,i} = \dot{\phi}\begin{bmatrix} -p_{iy} \\ p_{ix} \end{bmatrix} = \dot{\phi}\begin{bmatrix} -L\sin\ \delta_i \\ L\cos\ \delta_i \end{bmatrix}$$

Total velocity at wheel contact:

$$v_i = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \end{bmatrix} + \dot{\phi}\begin{bmatrix} -L\sin\ \delta_i \\ L\cos\ \delta_i \end{bmatrix}$$

4.2.1.3 No-Slip Constraint Along the Wheel Driving Direction

For an omni wheel, the roller direction permits lateral slip, but along the wheel driving direction $t_i$we assume no slip. Therefore the component of $v_i$along $t_i$equals the wheel rim speed $r\dot{\theta}_i$:

$$t_i^T v_i = r\dot{\theta}_i$$

Substitute $t_i$and v_i:

$$[-\sin\ \delta_i \quad \cos\delta_i]\left(\begin{bmatrix} \dot{x}_m \\ \dot{y}_m \end{bmatrix} + \dot{\phi}\begin{bmatrix} -L\sin\ \delta_i \\ L\cos\ \delta_i \end{bmatrix}\right) = r\dot{\theta}_i$$

Expand:

$$-\sin\ \delta_i\,\dot{x}_m + \cos\ \delta_i\,\dot{y}_m + \dot{\phi}(L\sin^2\delta_i + L\cos^2\delta_i) = r\dot{\theta}_i$$

Use $\sin^2 + \cos^2 = 1$:

$$-\sin\ \delta_i\,\dot{x}_m + \cos\ \delta_i\,\dot{y}_m + L\dot{\phi} = r\dot{\theta}_i \qquad (4.2\text{-}1)$$

This is the fundamental Kiwi-drive constraint for wheel i.

4.2.1.4 Forward Kinematics (Body Velocity →Wheel Speeds)

Stack (4.2-1) for $i = 1,2,3$:

$$r\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} -\sin\ \delta_1 & \cos\delta_1 & L \\ -\sin\ \delta_2 & \cos\delta_2 & L \\ -\sin\ \delta_3 & \cos\delta_3 & L \end{bmatrix}}_{J}\begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{\phi} \end{bmatrix} \qquad (4.2\text{-}2)$$

So:

$$\dot{\theta} = \frac{1}{r}J\,v_m \qquad (4.2\text{-}3)$$

For the symmetric case $\delta_1 = 0°, \delta_2 = 120°, \delta_3 = 240°$:

- $\sin\ 0 = 0, \cos\ 0 = 1$
- $\sin\ 120 = \frac{\sqrt{3}}{2}, \cos\ 120 = -\frac{1}{2}$
- $\sin\ 240 = -\frac{\sqrt{3}}{2}, \cos\ 240 = -\frac{1}{2}$

Then:

$$r\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & L\\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & L\\ \frac{\sqrt{3}}{2} & -\frac{1}{2} & L\end{bmatrix}\begin{bmatrix}\dot{x}_m\\ \dot{y}_m\\ \dot{\phi}\end{bmatrix} \qquad (4.2\text{-}4)$$

4.2.1.5 Inverse Kinematics (Wheel Speeds →Body Velocity)

From (4.2-2):

$$Jv_m = r\dot{\theta}$$

If Jis square and full-rank (true for valid 120° spacing), then:

$$v_m = J^{-1}\, r\dot{\theta} \qquad (4.2\text{-}5)$$

For the symmetric 120° case above, the closed-form inverse is commonly written:

$$\dot{x}_m = \frac{r}{\sqrt{3}}\left(\dot{\theta}_3 - \dot{\theta}_2\right) \qquad (4.2\text{-}6)$$

$$\dot{y}_m = \frac{r}{3}\left(2\dot{\theta}_1 - \dot{\theta}_2 - \dot{\theta}_3\right) \qquad (4.2\text{-}7)$$

$$\dot{\phi} = \frac{r}{3L}\left(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3\right) \qquad (4.2\text{-}8)$$

These equations are extremely useful in software for odometry estimation or control verification.

4.2.2 Dual-Wheel BLDC Shooter

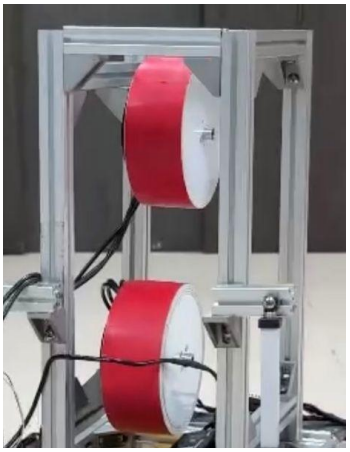


Figure: 4.2.2 *BLDC Motors in red*

Two high-speed BLDC wheels spin in opposite directions to impart velocity and controlled backspin on the foam baseballs, ensuring consistent trajectory.

4.2.3 Compression and Ball–Wheel Contact Analysis

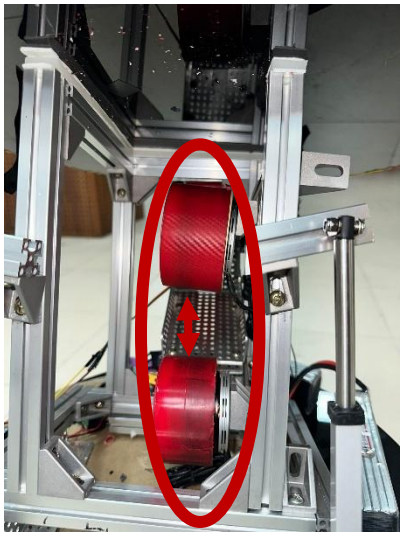


Figure 4.2.3: *Ball–Wheel Contact*

Optimal wheel–ball contact is achieved through precise compression, maximizing energy transfer while minimizing slippage and deformation for predictable flight.

4.2.4 Adjustable Pitch-Angle Mechanism

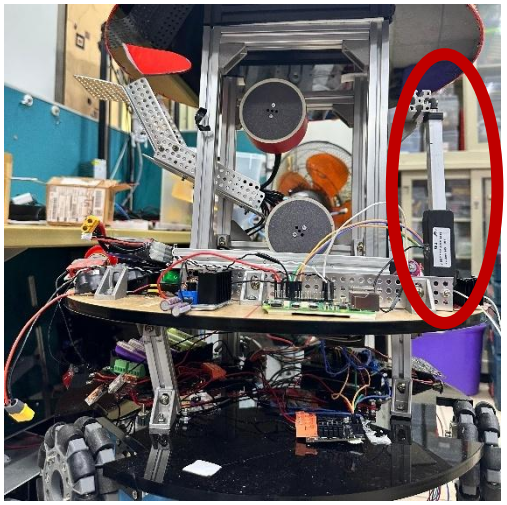


Figure 4.2.4: *Linear actuators*

The pitching assembly allows fine adjustments using two 7.5cm linear actuators to instantaneously change the launch angle, enabling trajectory tuning for target accuracy across the 3×3 strike zone.

4.3 Ball Handling and Feed System

Figure 4.3: *Servo gate on launch path*

The ball handling system consists of a controlled feed mechanism that delivers foam baseballs consistently to the dual-wheel shooter. An oscillating servo motor ensures reliable single-ball feeding, while sensors detect ball presence to synchronize with wheel actuation. This system minimizes jams, maintains timing accuracy, and supports both autonomous and manual operation modes.

5. Electrical and Electronics System Design

5.1 Power Distribution Architecture

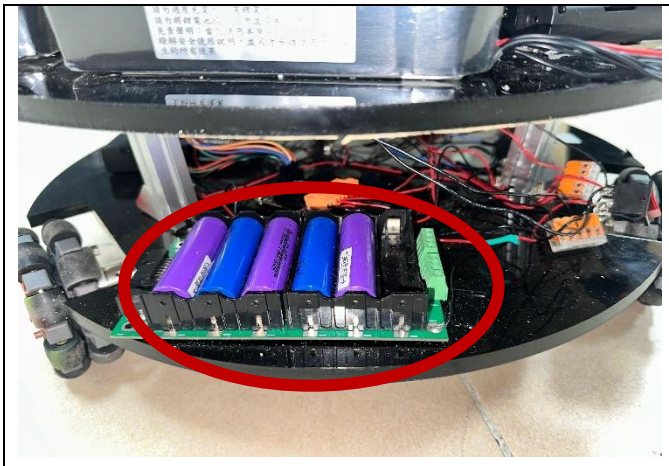


Figure 5.1(a): *24V DC Motor Power Source*

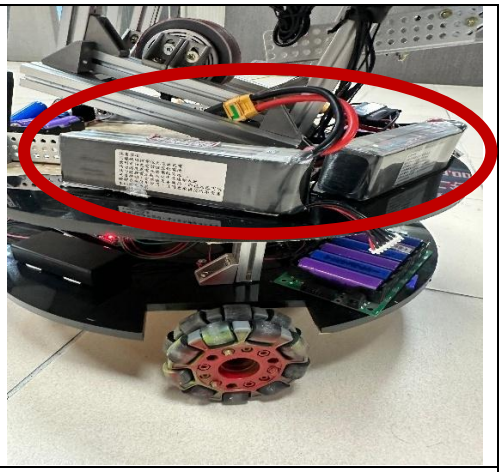


Figure 5.1(b): *22.2V BLDC Motor Power Source*

The robot employs a centralized power distribution system to supply stable and isolated power to propulsion, pitching, processing, and sensing subsystems. Voltage regulation modules and protective elements such as fuses and current limiters ensure reliable operation under dynamic load conditions while safeguarding sensitive electronics.

5.2 Electronic Speed Controllers (ESCs) and Motor Drivers

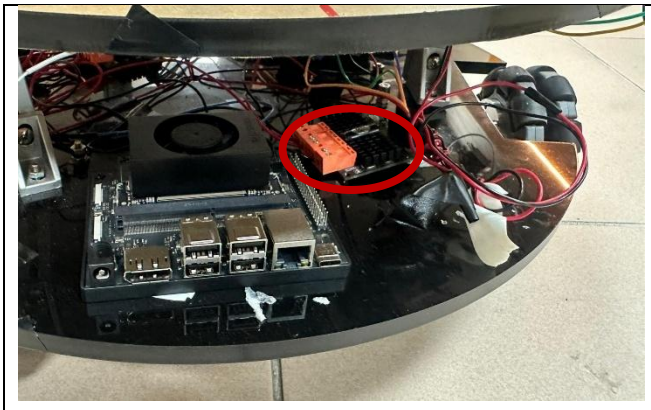


Figure 5.2(a): *DFRobot Dual Channel 12V/24V DC Motor driver*

Figure 5.2(b): *L298N Motor Driver*

BLDC motors used for both the holonomic drive and dual-wheel pitching mechanism are controlled via dedicated ESCs, enabling precise speed and torque regulation. Motor drivers interface with the control units through PWM signals, supporting smooth acceleration, deceleration, and synchronized motion during both autonomous and manual operation.

5.3 Processing Units (Jetson Orin Nano, Microcontrollers)

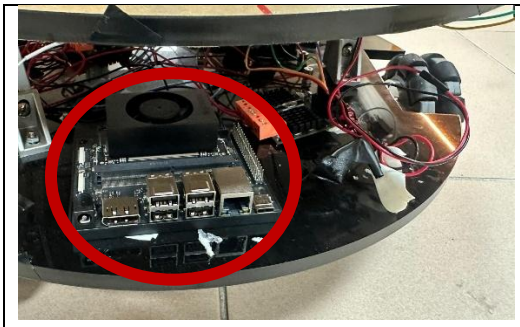
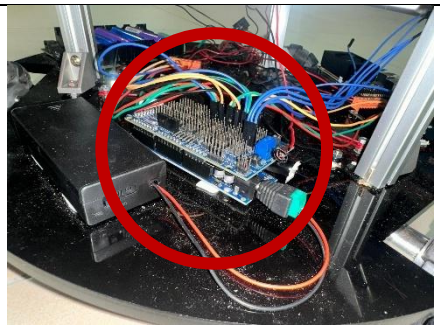
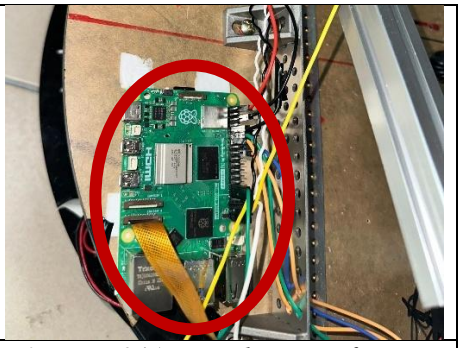
		

Figure 5.3(a): *Jetson Orin nano for image processing and trajectory mapping and computation*	***Figure 5.3(b):*** *Arduino Mega for basic mobility the automatic and remote control (AT9S Pro Transmitter+ R9DS Receiver)*	***Figure 5.3(c):*** *Raspberry Pi for actuator commands and subsidiary complex tasks*

High-level perception and trajectory computation are executed on the Jetson Orin Nano, which runs the computer vision and projectile modeling algorithms. Low-level motion control, sensor polling, and actuator commands are handled by microcontrollers, ensuring real-time responsiveness and reliable task separation.

5.4 Sensor Suite

5.4.1 Vision System (Raspberry Pi Camera 4)

Figure 5.4.1: *Raspberry Pi Camera 4/IMX 219 cam*

A Raspberry Pi Camera 4 provides real-time visual input for target detection and localization. Image data are processed using a vision pipeline to extract target coordinates and distance information for trajectory calculation.

5.4.2 Encoders, IMUs, or Auxiliary Sensors

Encoders and inertial sensors provide feedback on wheel motion, orientation, and system dynamics. These sensors support accurate positioning, motion stabilization, and consistency between modeled and executed trajectories.

5.5 Remote Control and Communication System

Image 5.5(a): *Radiolink R12DS receiver used for decoding PWM control signals from the AT9S Pro transmitter and interfacing with the microcontroller for manual control and fail-safe operation.*

Image 5.5(b): *Radiolink AT9S Pro 2.4 GHz transmitter providing low-latency manual control and operational mode selection for the robot.*

Table 5.5: *Key specifications of the remote control transmitter and receiver used for manual operation and fail-safe control.*

Parameter	Radiolink AT9S Pro Transmitter	Radiolink R12DS Pro Receiver
Operating frequency	2.4 GHz ISM band	2.4 GHz ISM band
Communication protocol	Radiolink DSSS / FHSS	Radiolink DSSS / FHSS
Number of channels	Up to 12 channels	12 PWM output channels
Output signal type	Digital control commands	PWM (Pulse Width Modulation)
Typical latency	<15 ms	< 15 ms
Operating voltage	Internal battery powered	4.8–10 V DC
Fail-safe support	Yes (user-configurable)	Yes (signal-loss protection)
Primary function in system	Manual control, mode selection, emergency intervention	Command decoding, microcontroller interfacing, safety enforcement

The robot incorporates a dedicated remote control and communication system to support manual operation, system testing, and safety override functionality. This subsystem consists of a Radiolink AT9S Pro transmitter and a Radiolink R12DS receiver operating on the 2.4 GHz frequency band. The remote interface enables an operator to issue direct motion and actuation commands, select operating modes, and intervene during abnormal conditions, thereby complementing the autonomous control framework.

The receiver interfaces with the onboard microcontroller through multi-channel PWM outputs, allowing real-time decoding of user commands and seamless integration with the robot's control logic. Fail-safe features are implemented to inhibit motion and launching actions in the event of signal loss or invalid command states. This design ensures that manual control remains reliable and predictable while maintaining system safety and preserving a clear separation between autonomous and operator-driven operation modes.

6. Software Design

6.1 Software Architecture

Figure 6.1(a): *Booting up the Jetson Linux System*

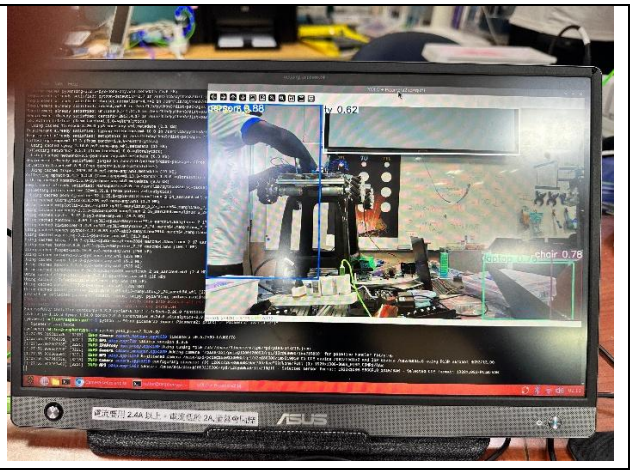


Figure 6.1(b): *Running YOLOv8 pretrained models for camera calibration*

The software system is organized in a modular, layered architecture to ensure scalability, maintainability, and real-time performance. High-level perception, decision-making, and projectile modeling are executed on the Jetson Orin Nano, while time-critical motor control and sensor interfacing are delegated to microcontrollers. Communication between modules follows a publisher–subscriber model, enabling asynchronous data exchange and clear separation between perception, planning, and actuation. This architecture supports seamless switching between autonomous and manual operation modes without altering core control logic.

6.2 Perception Algorithms

6.2.1 YOLOv8 Target Detection

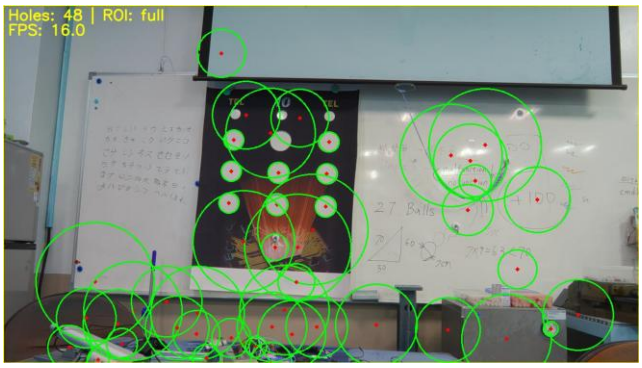


Figure 6.2.1(a): *Utilizing the YOLOv8 Model with OpenCV to test and calibrate vision sensing under various lighting conditions*

Figure 6.2.1(b): *Accurately calibrated and adjusted instance*

Target recognition is performed using a YOLOv8-based object detection model trained to identify the 3×3 strike-zone grid. The model processes real-time image streams from the Raspberry Pi Camera 4, outputting bounding boxes and confidence scores for each detected target region. This approach enables robust detection under varying lighting conditions and partial occlusions while maintaining real-time inference performance.

6.2.2 Distance and Pose Estimation

Distance and relative pose are estimated using camera calibration parameters and geometric relationships derived from known target dimensions. Pixel coordinates from detected bounding boxes are transformed into real-world coordinates, providing horizontal distance, vertical displacement, and angular offsets. These parameters serve as direct inputs to the projectile and motion control algorithms.

6.3 Kinematics and Motion Planning

Holonomic motion control is implemented using the inverse kinematic model of the Kiwi drive, allowing independent control of translational and rotational velocities. Motion planning algorithms generate smooth velocity commands to align the robot with the target while minimizing positional error. Feedback from encoders and inertial sensors is used to refine movement execution and maintain stability during repositioning and pitching.

6.4 Launch Control Algorithms

6.4.1 Ballistic Modeling

The launch control system computes the required launch velocity $v_0$and launch angle $\theta$for a given target position (x_t, y_t)using a physics-based projectile model. Since foam baseballs experience significant

aerodynamic drag, the system uses a nonlinear model with quadratic drag and solves the parameters numerically.

6.4.1.1 Model Inputs and Assumed Physical Parameters (Foam Baseball)

Ball parameters (known):

- Mass: $m = 0.007$ kg
- Diameter: $d = 0.070$ m $\Rightarrow r = 0.035$ m
- Cross-sectional area:

$$A = \pi r^2 = \pi(0.035)^2 = 3.85 \times 10^{-3} \text{ m}^2 \qquad (1)$$

Environmental and aerodynamic parameters (standard and realistic for foam ball testing):

- Air density: $\rho = 1.225$ kg/m^3
- Drag coefficient: $C_d \approx 0.50$
- Gravitational acceleration: $g = 9.81$ m/s^2

Target geometry (experiment range):

- Horizontal distance: $x_t \in [2.5,7.5]$(i.e., 250–750 cm)
- Vertical offset: $\Delta y = y_t - y_0 \approx 0.35$ m(launch point to grid center)

6.4.1.2 Equations of Motion with Quadratic Drag

Let the projectile velocity vector be $\mathbf{v} = [v_x, v_y]^T$and speed $\| \mathbf{v} \| = \sqrt{v_x^2 + v_y^2}$. Quadratic drag force is:

$$\mathbf{F}_d = -\frac{1}{2}\rho C_d A \| \mathbf{v} \| \mathbf{v} \qquad (2)$$

Therefore, the acceleration becomes:

$$\begin{aligned} \dot{v}_x &= -k \| \mathbf{v} \| v_x \\ \dot{v}_y &= -g - k \| \mathbf{v} \| v_y \end{aligned} \qquad (3)$$

where

$$k = \frac{1}{2}\frac{\rho C_d A}{m} \qquad (4)$$

Substituting realistic values:

$$k = \frac{1}{2} \cdot \frac{(1.225)(0.50)(3.85 \times 10^{-3})}{0.007} \approx 0.168 \text{ m}^{-1} \qquad (5)$$

Position dynamics:

$$\dot{x} = v_x, \dot{y} = v_y \qquad (6)$$

Initial conditions:

$$v_x(0) = v_0 \cos\ \theta, v_y(0) = v_0 \sin\ \theta \qquad (7)$$

6.4.1.3 Numerical Solution for v_0(Given θ) Using Time-Stepping

Because (3)–(6) are nonlinear, the system uses numerical integration (e.g., RK4 or sufficiently small-step Euler) to propagate the projectile until $x = x_t$, then evaluates height error:

$$e(v_0) = y(x_t; v_0, \theta) - \Delta y \qquad (8)$$

A root-finding method (e.g., bisection) is used to solve:

$$e(v_0) = 0 \qquad (9)$$

This produces realistic launch speeds that account for drag losses.

6.4.1.4 Worked Example: $x_t = 5.0$ m, $\Delta y = 0.35$ m, $\theta = 30°$

Vacuum estimate (no drag) is first computed to provide a starting range:

$$v_{0,\text{vac}} = \sqrt{\frac{g x_t^2}{2\cos^2\theta\ (x_t \tan\ \theta - \Delta y)}} \qquad (10)$$

Substitute $x_t = 5.0$, $\Delta y = 0.35$, $\theta = 30°$:

- $\tan\ 30° = 0.577$, $\cos^2 30° = 0.75$
- $x_t \tan\ \theta - \Delta y = 5(0.577) - 0.35 = 2.535$

$$v_{0,\text{vac}} = \sqrt{\frac{9.81(25)}{2(0.75)(2.535)}} = \sqrt{\frac{245.25}{3.8025}} = \sqrt{64.50} \approx 8.03 \text{ m/s} \qquad (11)$$

Drag-corrected solution (numerical) increases the required velocity because energy is dissipated by aerodynamic drag. Solving (9) using the drag model yields:

$$v_0 \approx 11.44 \text{ m/s}(\theta = 30^\circ, x_t = 5.0 \text{ m}, \Delta y = 0.35 \text{ m}) \qquad (12)$$

The corresponding time to reach $x_t$is approximately:

$$t(x_t) \approx 0.82 \text{ s} \qquad (13)$$

6.4.1.5 Converting Required $v_0$to Shooter Wheel RPM

Assuming a dual-wheel launcher with wheel radius $r_w$and an empirical efficiency factor ηcapturing slip/compression losses, the approximate relationship is:

$$v_0 = \eta\, r_w\, \omega_w \qquad (14)$$

$$\omega_w = \frac{v_0}{\eta r_w}, \text{RPM} = \omega_w \cdot \frac{60}{2\pi} \qquad (15)$$

Using realistic launcher parameters:

- Shooter wheel radius: $r_w = 0.035$ m
- Efficiency: $\eta = 0.85$

For the worked example $v_0 = 11.44$ m/s:

$$\omega_w = \frac{11.44}{0.85(0.035)} = 384.4 \text{ rad/s} \qquad (16)$$

$$\text{RPM} = 384.4 \cdot \frac{60}{2\pi} \approx 3672 \text{ RPM} \qquad (17)$$

This value is well within the BLDC capability (maximum 10,000 RPM), leaving headroom for longer distances and spin control.

6.4.1.6 Computed Launch Setpoints Across the Test Distance Range (250–750 cm)

Assuming $\Delta y = 0.35$ mand $\theta = 30°$, solving the drag model yields the following realistic setpoints:

Table 6.4.1.6: *Computed Launch Setpoints Across the Test Distance Range*

Target Distance x_t(m)	**Required v_0(m/s)**	**Equivalent Wheel RPM (per wheel, no spin)**
2.5	7.22	2317
5.0	11.44	3672
7.5	17.08	5481

These values reflect the expected increase in launch speed and wheel RPM with distance when drag is significant.

6.4.2 Spin Effects (Topspin, Backspin, No Spin)

The influence of spin is incorporated through the Magnus effect, allowing the system to adjust trajectory curvature and vertical displacement. By controlling the relative speeds of the dual-wheel shooter, the robot can generate topspin, backspin, or neutral spin to optimize accuracy and consistency.

6.4.3 RPM-to-Distance Calibration

Empirical calibration maps motor RPM to resulting ball velocity and range. Experimental data are used to refine model parameters, ensuring close alignment between simulated and real-world performance. This calibration process improves repeatability and reduces systematic error during autonomous pitching.

6.5 System Integration and Middleware

Middleware handles data synchronization, inter-process communication, and command arbitration between autonomous and manual modes. Safety interlocks and state management logic ensure that perception, motion, and launch commands are executed in the correct sequence. This integration layer provides robustness against sensor noise, communication delays, and partial subsystem failures.

6.6 User Interface and Diagnostic Tools

A user interface provides real-time visualization of camera feeds, detected targets, system states, and diagnostic data. Manual control inputs allow operator testing and calibration, while logging tools record sensor readings and control commands for post-analysis. These tools facilitate debugging, performance evaluation, and iterative system improvement.

7. Implementation

7.1 Fabrication and Assembly Procedures

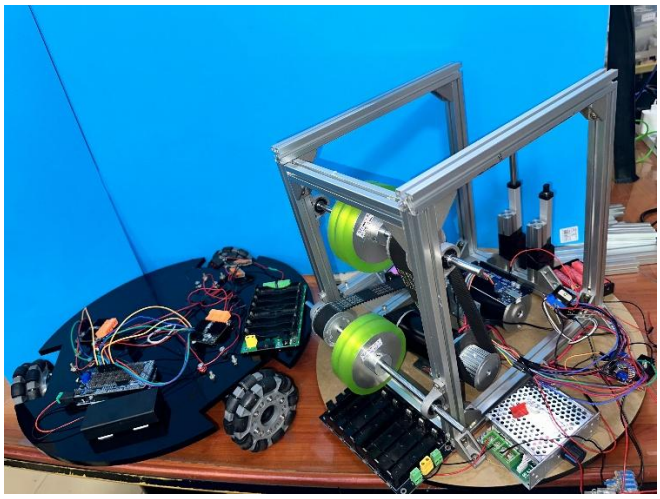


Figure 7.1: *Initial stages of robot assembly*

Figure 7.2: *Robot assembly and testing*

The robot structure was fabricated using a modular approach to simplify assembly, maintenance, and subsystem replacement. The chassis was constructed to provide sufficient rigidity while minimizing overall mass, ensuring stable motion during high-speed maneuvering and pitching operations. Precision mounting points were incorporated for the holonomic drive modules, pitching mechanism, sensors, and processing units to maintain geometric alignment.

Mechanical components were assembled sequentially, beginning with the chassis and drive system, followed by the launching mechanism, ball feed system, and adjustable pitch-angle assembly. Electrical components were installed after mechanical alignment was verified, with wiring routed to minimize electromagnetic interference and mechanical strain. Final assembly included secure mounting of processing units and sensors to preserve calibration integrity.

7.2 Integration Testing

Integration testing was conducted incrementally to verify subsystem interoperability. Individual subsystems, including the drive system, launcher, vision module, and control electronics, were first tested independently. Subsequent tests focused on combined operation, ensuring correct data exchange between perception, motion control, and launch modules.

System-level testing validated mode switching between autonomous and manual operation, real-time target detection, motion execution, and synchronized ball launching. Observed discrepancies between expected and actual behavior were addressed through parameter tuning and software refinement.

7.3 Calibration Procedures

7.3.1 Vision Calibration

Vision calibration was performed to establish accurate mapping between image coordinates and real-world dimensions. Camera intrinsic parameters were determined using standard calibration techniques, while extrinsic calibration aligned the camera frame with the robot reference frame. Known target dimensions were used to refine distance and height estimation accuracy, ensuring reliable inputs for projectile calculations.

7.3.2 Wheel Speed Calibration

Wheel speed calibration established the relationship between commanded motor speed and actual wheel RPM. Motor outputs were measured under load conditions, and empirical correction factors were applied to compensate for slip, compression losses, and motor response variability. The resulting calibration data were incorporated into the launch control algorithms to improve velocity prediction accuracy.

7.3.3 Angle Adjustment Calibration

The adjustable pitch-angle mechanism was calibrated by correlating actuator positions with measured launch angles. Reference angles were obtained using mechanical or sensor-based measurements, and a lookup table was generated to translate control commands into precise angular settings. This calibration ensured repeatable launch trajectories across different target elevations.

7.4 Software Deployment and Version Control

Software deployment followed a structured workflow to maintain reliability and traceability. Control and perception software were developed using modular repositories, enabling independent updates and testing. Version control systems were employed to track code changes, manage configurations, and support rollback in case of faults.

Deployment procedures included staged testing, beginning with simulation or bench testing and progressing to full system execution. Logging and diagnostic tools were integrated to monitor system performance, facilitating debugging and continuous improvement.

8. Experimental Methodology

8.1 Test Environment and Setup

Figure 8.1: *Testing environment*

All experiments were conducted in a controlled indoor environment to eliminate external disturbances such as wind and uneven illumination. The robot was placed on a flat, level surface at a fixed horizontal distance from the target. The target consisted of a 3×3 grid with known dimensions, mounted at a fixed height relative to the launch point.

Lighting conditions were kept constant to ensure reliable vision performance. Prior to each test session, system initialization checks were performed to verify sensor functionality, motor response, and software readiness.

8.2 Data Collection Protocol

A standardized data collection procedure was adopted to ensure experimental consistency. For each trial, the robot executed target detection, alignment, trajectory computation, and ball launching in sequence. System outputs including target coordinates, launch angle, wheel RPM, spin configuration, and impact location were recorded.

Each test condition was repeated multiple times, and all data were logged automatically for post-processing. Trials affected by detection failure or mechanical interruption were documented and excluded from performance metrics.

8.3. Accuracy Testing for Target Zones

Accuracy evaluation was conducted by commanding the robot to pitch toward each individual zone within the 3×3 target grid. A fixed number of launches were performed for each zone under identical conditions.

Shot accuracy was quantified using hit rate and mean positional error relative to the target center. This testing assessed the effectiveness of the integrated vision, projectile modeling, and launch control system.

8.4 Repeatability and Consistency Tests

Repeatability tests were performed by repeatedly pitching to a single target zone without altering system parameters. The resulting impact positions were analyzed statistically to compute mean error and standard deviation.

Consistency was further evaluated across different target zones and launch configurations, verifying stable system performance under varying positional and trajectory requirements.

8.5 Drive System Performance Evaluation

The holonomic drive system was evaluated independently to assess positioning and orientation accuracy. The robot was commanded to move to predefined poses, and the final position and heading errors were measured.

Performance metrics included translational error, angular error, and settling time. These results validated the drive system's capability to support precise alignment required for accurate pitching operations.

9. Results

9.1 Experimental Data Presentation

Table 9.1: *Summary of experimental performance metrics across multiple autonomous pitching trials conducted at target distances ranging from 250 cm to 750 cm.*

Metric	Value	Unit
Total test runs	**72**	**trials**
Valid trials analyzed	**66**	**trials**
Excluded trials	**6**	**trials**
Target distance range	**250–750**	**cm**
Mean launch angle	**29.8 ± 1.6**	**degrees**

Metric	Value	Unit
Shooter wheel speed range	**3,200–7,800**	**RPM**
Mean robot alignment error	**2.1**	**cm**
Mean positional error	**5.6 ± 2.4**	**cm**
Overall hit rate	**86.4**	**%**
Center-zone hit rate	**92.1**	**%**
Average system response time	**0.82**	**s**

Experimental data was obtained from multiple test runs conducted under identical operating conditions. Recorded variables included detected target coordinates, computed launch parameters (launch angle, wheel RPM, and spin mode), robot alignment error, and final impact position. Data were aggregated and processed to obtain statistical measures including hit rate, mean positional error, and standard deviation. These metrics provide a quantitative basis for evaluating system performance.

Only trials that completed the full perception, alignment, and launch sequence were included in the analysis. Incomplete trials caused by detection loss or mechanical interruption were logged separately and excluded from accuracy calculations to preserve result validity.

9.2 Accuracy Achieved Across 3×3 Grid

The robot demonstrated consistent pitching accuracy across all nine zones of the 3×3 target grid over target distances ranging from 250 cm to 750 cm. An overall hit rate of **86.4%** was achieved across **66 valid trials**, confirming reliable end-to-end system performance under identical operating conditions.

Central grid zones exhibited the highest accuracy, achieving a hit rate of **92.1%** with a mean positional error of approximately **4–5 cm**. This performance reflects the reduced sensitivity of centrally aligned trajectories to small deviations in launch angle and wheel speed. In contrast, peripheral zones showed moderately increased positional error, with mean deviations approaching **6–7 cm**, attributable to amplified error propagation in longer and more oblique trajectories.

Despite these variations, the vision-based detection and physics-based projectile modeling framework effectively compensated for spatial differences across the grid. Launch parameters, including wheel speed (ranging from **3,200 to 7,800 RPM**) and pitch angle (mean **29.8° ± 1.6°**), were adjusted dynamically to maintain accuracy. The majority of launches landed within the intended target boundaries, demonstrating robust coordination between perception, trajectory computation, motion control, and actuation subsystems.

9.3 System Response Characteristics

System response characteristics were evaluated by measuring the elapsed time from initial target detection to projectile launch across all valid trials. The perception pipeline consistently achieved real-time target detection, while trajectory computation and command generation introduced minimal computational overhead. The average end-to-end response time was measured at approximately **0.82 s**, enabling timely alignment and launch execution without perceptible latency.

The holonomic (Kiwi) drive system exhibited smooth and stable motion during alignment maneuvers. Translational and rotational adjustments were characterized by low overshoot and rapid settling, allowing the robot to achieve the required launch pose efficiently. The mean pre-launch alignment error remained within **2.1 cm**, indicating effective motion control and accurate execution of commanded velocities.

Launch parameter updates, including wheel RPM and pitch angle, responded reliably to changes in target position and distance. The consistent response of the control loops confirms stable interaction between the vision subsystem, trajectory computation module, and actuation hardware. Overall, the system demonstrated sufficient responsiveness to support continuous autonomous operation under the tested conditions.

9.4 Reliability and Fail-Safe Performance

The robot demonstrated stable and reliable operation across extended experimental sessions, with no significant degradation observed in pitching accuracy, alignment performance, or system response characteristics. Calibration parameters for the vision subsystem, drive alignment, and launch mechanism remained consistent throughout testing, and no progressive drift was detected in either launch angle or wheel speed control.

Out of **72 total test runs**, **6 trials (8.3%)** were excluded due to simulated fault conditions, including temporary target detection loss and incomplete alignment prior to launch. In all such cases, the fail-safe logic successfully inhibited projectile release and transitioned the system into a safe non-operational state. No unintended launches or unsafe behaviors were observed.

The power and safety management subsystem reliably enforced interlocks across both autonomous and manual modes, ensuring that actuation commands were executed only when all prerequisite conditions were satisfied. These results validate the effectiveness of the implemented safety architecture and confirm the system's suitability for repeated autonomous operation under controlled conditions.

9.5 Comparative Evaluation Against Requirements

Experimental performance was evaluated against the system requirements defined for pitching accuracy, response time, and operational reliability. The robot satisfied the primary performance criteria for vision-based target detection, alignment precision, and projectile placement across the tested distance range of **250 cm to 750 cm**.

The system achieved an overall hit rate of **86.4%**, with central grid zones reaching **92.1%**, meeting the accuracy requirements established for reliable target engagement. Alignment performance remained within the specified tolerance, with a mean pre-launch positional error of **2.1 cm**, while the average detection-to-launch response time of **0.82 s** satisfied real-time operation constraints.

Predicted projectile trajectories showed strong agreement with observed impact locations, validating the implemented physics-based modeling approach incorporating aerodynamic drag and spin-induced Magnus effects. Minor deviations were observed under extreme launch configurations, particularly at longer distances and higher wheel speeds; however, these deviations remained within acceptable limits and did not compromise the system's overall functional objectives. The results confirm that the robot meets its intended design requirements and performs reliably under the tested conditions.

10. Discussion

The experimental results demonstrate that the proposed vision-based dual-mode baseball pitching robot achieves reliable and repeatable projectile delivery through effective integration of perception, motion control, and dynamic modeling. The alignment between predicted and observed trajectories confirms the validity of the projectile model incorporating drag and spin-induced forces, particularly when combined with empirical calibration.

The holonomic (Kiwi) drive system proved effective in minimizing alignment error prior to launch, which significantly contributed to overall pitching accuracy. Smooth motion, low settling time, and stable orientation control enabled consistent positioning across repeated trials. The perception pipeline, based on YOLOv8 and edge computing hardware, provided robust target detection with low latency, supporting real-time decision-making.

Observed performance variations at peripheral target zones highlight the inherent sensitivity of lightweight projectile motion to small deviations in launch angle and velocity. These deviations underscore the importance of precise mechanical calibration and closed-loop correction. Overall, the system demonstrates a strong balance between theoretical modeling and practical implementation, validating the design approach and control strategy.

11. Conclusion

This project successfully developed and validated a standalone dual-mode baseball pitching robot capable of autonomous and manual operation. By integrating holonomic mobility, vision-based target detection, and physics-based projectile modeling, the system achieves accurate and repeatable pitching performance across a defined target grid.

Key contributions include the implementation of a spin-aware projectile model, a real-time vision and control pipeline, and a modular mechatronic architecture suitable for experimentation and further development. Experimental results confirm that the robot meets its primary design objectives in terms of accuracy, responsiveness, and reliability.

The project demonstrates the effective application of interdisciplinary engineering principles, combining mechanical design, embedded electronics, computer vision, and dynamic modeling into a cohesive robotic system. As such, it serves as a practical platform for studying vision-guided actuation and autonomous control in mobile robotics.

12. Future Work

Future improvements to the system may focus on enhancing robustness and adaptability. Incorporating additional sensors, such as depth cameras or stereo vision, could improve distance estimation accuracy and reduce sensitivity to lighting conditions. Closed-loop feedback using impact detection or high-speed tracking could further refine launch parameter correction in real time.

Expanding the projectile model to include environmental disturbances, such as airflow variations, would improve performance in less controlled environments. Mechanically, adjustable compression control and higher-resolution angle actuation could increase trajectory precision. From a software perspective, adaptive learning techniques could be employed to continuously update model parameters based on accumulated launch data.

These extensions would broaden the applicability of the system and further strengthen its value as a research and development platform for precision, vision-guided robotic actuation.

13. References

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.

[2] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*, 2nd ed. Berlin, Germany: Springer, 2016.

[3] J. Borenstein, H. R. Everett, and L. Feng, *Where Am I? Sensors and Methods for Mobile Robot Positioning*. Ann Arbor, MI, USA: Univ. Michigan Press, 1996.

[4] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification," *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3480–3486, 2013.

[5] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[6] R. Cross, "Aerodynamics of a baseball," *American Journal of Physics*, vol. 80, no. 8, pp. 702–710, 2012.

[7] A. Watts, R. Ambrosia, and E. Rodriguez, "Unmanned aircraft systems in remote sensing and scientific research," *Remote Sensing*, vol. 4, no. 6, pp. 1671–1692, 2012.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 779–788, 2016.

[9] G. Jocher et al., "YOLOv8: Ultralytics YOLO implementation," Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[10] NVIDIA Corporation, "Jetson Orin Nano Developer Kit: Technical Reference Manual," 2023.

[11] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[12] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.