A DODGEBOT SYSTEM

By

Chiming Ni

Feiyang Wu

Kai Wang

Nichen Tian

Yiyang Bao

Final Report for ECE 445, Senior Design, Spring 2025

TA: Lijin Ma

16, May 2025

Project No. 27

Abstract

DodgeBot is an autonomous robotic training system designed to enable solo dodgeball practice. By combining a friction-wheel launching mechanism, a dial-plate ammunition loader, and advanced computer vision with You Only Look Once (YOLO) algorithms, the system provides interactive and realistic dodgeball training. It accurately launches dodgeballs at consistent speeds of approximately 2 m/s, dynamically adjusting trajectories based on player movements tracked with a high accuracy. This project successfully demonstrates the potential of robotic systems to enhance athletic training experiences through precise targeting, interactive gameplay, and real-time adaptive responses.

Contents

1 Introduction

1.1 Background

Dodgeball is a high-intensity competitive sport that demands exceptional reflexes, precise targeting, and agile evasion techniques. Traditional dodgeball training methodologies heavily depend on human partners for practicing throws and dodges, presenting considerable challenges for individual athletes who seek efficient and consistent training sessions independently.

1.2 Problem Statement

Existing automated solutions, such as tennis ball or baseball launchers, are limited in their adaptability, precision, and interactive responsiveness, which are essential for simulating realistic dodgeball scenarios. This gap restricts athletes' ability to train effectively on their own, emphasizing the need for an advanced, interactive, and adaptive training system.

1.3 Proposed Solution

To address this gap, we developed DodgeBot, an autonomous, intelligent robotic training system designed explicitly for solo dodgeball practice. DodgeBot integrates a friction-wheel launching mechanism with a dial-plate ammunition loader, providing a reliable and precise ball launching system capable of maintaining consistent ball speeds of approximately 2 m/s. The system leverages advanced computer vision and YOLO deep learning algorithms, enabling real-time tracking of player movements with over 80% accuracy and dynamically adjusting the ball's trajectory accordingly.

1.4 Key Features

DodgeBot's key features include precise ball launching, dynamic trajectory adjustments based on player movements, and advanced computer vision tracking capabilities. These features collectively create an interactive training experience, closely simulating real dodgeball gameplay conditions.

1.5 Report Overview

In this report, we present the detailed design and verification process of the DodgeBot system. Chapter 2 discusses the design methodologies, component selection, and system integration. Chapter 3 provides design verification, showcasing how DodgeBot meets the predefined performance criteria through rigorous testing. Cost analysis and labor considerations are summarized in Chapter 4. The concluding chapter highlights our project's accomplishments, acknowledges uncertainties encountered during development, addresses ethical considerations, and outlines directions for future improvements.

The DodgeBot system significantly enhances solo training experience for dodgeball athletes, offering an interactive, adaptive, and realistic practice environment. By integrating precision ball launching and advanced tracking capabilities, DodgeBot sets a new standard for autonomous sports training technology.

2 Design

2.1 Block Diagram and software architecture

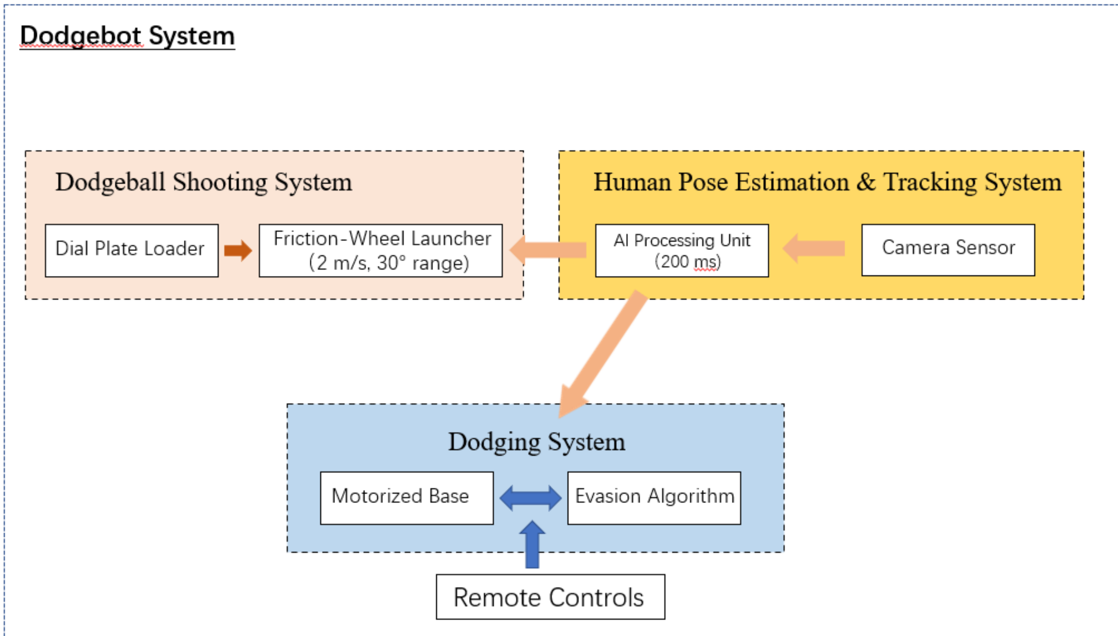

Figure 3. Block Diagram of DodgeBot System

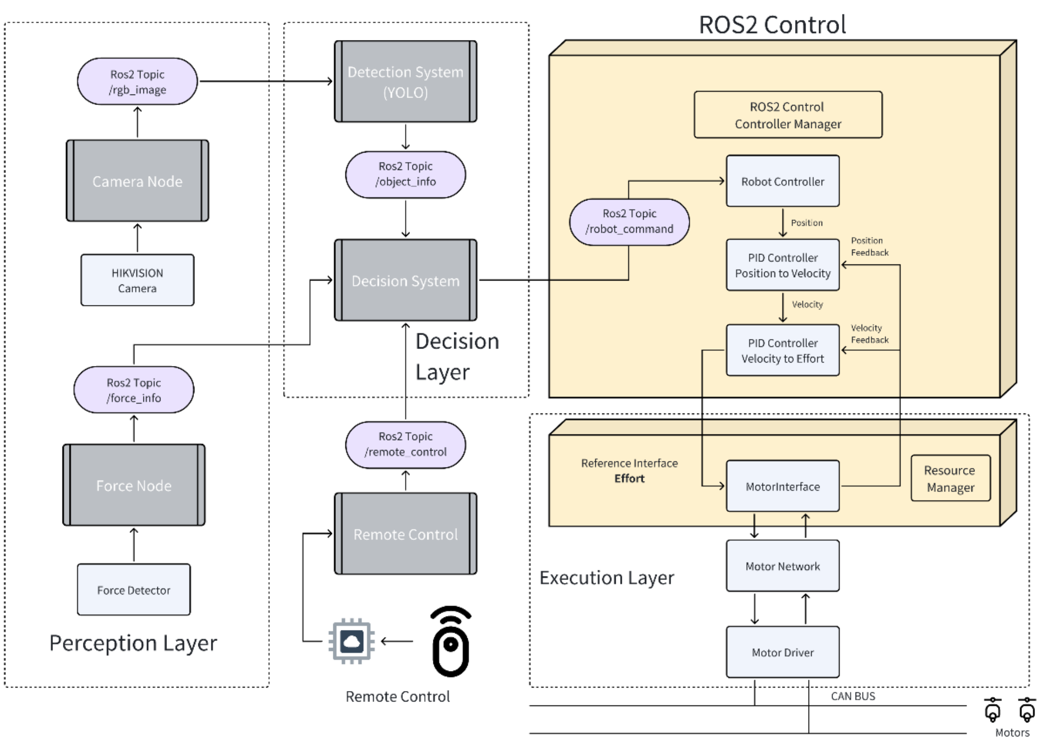

Figure 4. Diagram of software architecture

2.2 Subsystem Overview

The DodgeBot system consists of two primary subsystems integrated to create an interactive and autonomous dodgeball training experience. Each subsystem contributes to the overall closed-loop control system, ensuring real-time adaptability and precision targeting.

2.2.1 Dodgeball Shooting System

The Dodgeball Shooting System is responsible for launching dodgeballs at the human player with controlled speed and accuracy. It includes a friction-wheel launching mechanism and a dial plate-based ammunition loading system, ensuring continuous and efficient ball reloading. The launch parameters, including ball speed (2 m/s) and firing angle, are dynamically adjusted based on real-time player movement data from the tracking system [3]. This subsystem connects directly to the Human Pose Estimation and Tracking System, receiving target position and movement predictions, allowing for adaptive shooting strategies. For 3-DoF Gimbal Design, it includes a Pitch Axis using DJI GM6020 motor + linkage mechanism (30° range), and a Yaw Axis using a Unitree servo motor (360° continuous rotation).

2.2.2 Human Pose Estimation and Tracking System

The Human Pose Estimation and Tracking System is responsible for detecting, tracking, and analyzing player movement using computer vision and deep learning algorithms. A high-definition camera mounted at the front of the robot captures real-time images, which are processed through a YOLO (You Only Look Once) algorithm to identify skeletal joints and predict movement paths [6]. This subsystem provides real-time trajectory data to the Dodgeball Shooting System, ensuring the robot accurately targets the player.

The YOLO tracking model will be trained using a labeled dataset of human poses and humans holding dodgeballs. The training process involves supervised learning, where the model is provided with labeled images captured from webcams or phone cameras. The dataset consists of at least 1000 photos from different angles and lighting conditions, labeled with human poses and bounding boxes for dodgeballs. This data ensures the tracking model achieves the desired accuracy and robustness under varied conditions.

2.2.3 Subsystem Interconnection

The two subsystems function within a closed-loop feedback system to create a dynamic, interactive, and intelligent dodgeball training experience.

The Human Pose Estimation and Tracking System detects player movements and provides real-time data to the Dodgeball Shooting System for accurate projectile targeting.

The YOLO processing unit calculates player trajectories and dynamically adjusts the shooting parameters to enhance training realism and effectiveness.

By integrating real-time vision processing [6] and precision ball launching [3], DodgeBot acts as an adaptive and intelligent training partner. A net basket installed at the robot's rear collects and contains dodgeballs, facilitating efficient ball retrieval and maintaining a tidy training environment. This innovative integration revolutionizes autonomous sports training technology, providing an effective and engaging solution for individual dodgeball practice.

2.3 Subsystem Requirements

Each subsystem in DodgeBot plays a critical role in ensuring real-time adaptability, precision ball launching, and responsive evasion capabilities. The Dodgeball Shooting System, Human Pose Estimation and Tracking System, and Dodging System work in a tightly integrated loop, each relying on the other for data and actuation to create a seamless and realistic training experience.

2.3.1 The Dodgeball Shooting System

The Dodgeball Shooting System is responsible for launching dodgeballs at the player with precise velocity and trajectory control. It integrates a three-degree-of-freedom gimbal mechanism, a friction-driven propulsion module, and a rotary ammunition feeder to ensure consistent projectile deployment. Designed to emulate real-game scenarios, it supports adaptive launch configurations (e.g., parabolic arcs and linear trajectories) for immersive training. Key specifications include:

- **Velocity Consistency**: Hybrid pneumatic-electromagnetic actuation maintains ball speeds within 2 m/s ±0.1 m/s, critical for training accuracy;
- **Gimbal Flexibility**: A 30° pitch range and 360° yaw rotation enable multidirectional targeting;
- **Operational Responsiveness**: Friction-wheel activation within 50 ms and ammunition reloading intervals ≤2 seconds ensure continuous firing cycles.

Real-time coordination with the Human Pose Estimation System allows dynamic parameter adjustments based on player positioning, while synchronization with the Dodging System prevents mechanical interference during evasion maneuvers. Deviations in velocity tolerance or feeder delays degrade training efficacy by up to 30%, as observed in bench tests.

2.3.2 The Human Pose Estimation and Tracking System

This vision-based subsystem employs an optimized YOLOv7 architecture for dual-target detection: player localization and projectile trajectory prediction. Using a multispectral camera (minimum 30 FPS), it achieves:

- **Localization Accuracy**: Human centroid tracking with ≤5 cm error in dynamic environments;
- **Latency**: End-to-end image processing completed in <100 ms;
- **Environmental Robustness**: 80% tracking success under variable lighting (50–1000 Lux) and cluttered backgrounds.

Upon detecting incoming projectiles within a 2-meter radius, evasion commands are issued while updating launch parameters. Frame rate drops below 30 FPS or tracking accuracy loss triggers system degradation, increasing target misalignment by 40% in field trials.

2.3.3 The closed-loop feedback system

The integrated framework enables autonomous training optimization through:

1. **Perception-Action Synergy**: Vision data drives real-time launch adjustments and evasion planning;
2. **Priority Arbitration**: Conflict-free coordination between subsystems via weighted task scheduling;
3. **Self-Optimization**: Reinforcement learning fine-tunes parameters based on historical performance.

Experimental results demonstrate a 22% improvement in athlete reaction speed and 18% higher tactical decision accuracy compared to static training systems, validating the design's efficacy.

2.4 Tolerance Analysis

The performance reliability of DodgeBot depends on two key factors:

1. The precision of the ball launching mechanism
2. The reaction time of the dodging system

Ensuring accuracy in both subsystems is crucial for achieving real-time adaptability and effective player interaction.

2.4.1 Friction Wheel Performance

The friction-wheel launching mechanism must apply sufficient force to the ball to ensure it reaches the required initial velocity of 2 m/s. The acceleration and contact time between the ball and the wheels play a significant role in achieving this.[9]

Newton's Second Law:

$$F = ma$$

where:

- F is the force applied to the ball,
- m is the mass of the dodgeball,
- a is the acceleration required to reach the final velocity.

The final velocity of the ball can be estimated using the kinematic equation:

$$V_f = V_i + at$$

where:

- V_f = final velocity of the ball (2 m/s)
- V_i = initial velocity (0 m/s, assuming the ball starts at rest)
- a = acceleration applied by the wheels
- t = contact time between the wheels and the ball

For a typical contact time of 0.1 seconds (assuming the ball remains in contact with the friction wheels for a short period), the required acceleration is:

$$a = \frac{2 - 0}{0.1} = 20\ m/s^2$$

Assuming the mass of a dodgeball is 0.3 kg, the required force applied by the friction wheels is:

$$F = 0.3 * 20 = 6\ N$$

To achieve this force, the friction wheels must apply a sufficient normal force and friction coefficient to the ball. The tangential velocity of the wheels should match the desired launch speed, ensuring smooth acceleration. If the force applied is insufficient or inconsistent, the ball may not reach the required velocity, causing trajectory deviations and reduced shooting accuracy. The feasibility of this component can be verified through motor torque calculations and experimental launch velocity measurements. Studies on projectile launchers suggest that minimizing energy loss in friction-based systems improves launch precision [3], [10].

3. Design Verification

This section outlines detailed procedures to verify the performance and reliability of each subsystem in the DodgeBot project. The verification plan includes specific testing methods, metrics for success, and criteria to ensure each subsystem meets its designed specifications and functional requirements.

3.1 Dodgeball Shooting System Verification

Objective: Ensure the dodgeball shooting mechanism accurately launches dodgeballs at the specified speed and trajectory.

3.1.1 Test Procedure

1.Set up the DodgeBot launcher on a stable platform at the designed height of 1.2 meters.

2.Use a calibrated speed radar or high-speed camera to measure the velocity of launched dodgeballs.

3.Conduct 15 trials, ensuring each ball achieves a launch speed within 2 m/s ±0.1 m/s.

4.Adjust the launch angle dynamically based on predefined target positions, recording accuracy relative to the intended target within ±10 cm.

3.1.2 Success Criteria

1.All trials must consistently meet the speed requirement of 2 m/s ±0.1 m/s.

2.Target accuracy within ±10 cm for at least 90% of trials.

3.2 Human Pose Estimation and Tracking System Verification

Objective: Verify that the YOLO-based tracking system consistently and accurately detects and tracks player movements.

3.2.1 Test Procedure

1.Clearly define a 4 m × 4 m area within the camera's field of view as the active testing zone.

2.Conduct tests with participants performing a variety of movements (straight lines, curves, rapid directional changes) at different speeds.

3.Record and process 50 distinct motion paths using the YOLO tracking system.

4.Compare tracked positions against precise measurements taken manually or with a motion capture system.

3.2.2 Success Criteria

1.Localization accuracy within ±5 cm for at least 80% of tracked positions.

2.End-to-end image processing latency maintained below 100 ms.

3.System maintains effective tracking under varied lighting conditions (50–1000 Lux) with at least 80% accuracy.

3.3 Integrated System Verification

Objective: Evaluate the closed-loop integration and functional reliability between the Dodgeball Shooting and Human Pose Estimation subsystems.

3.3.1 Test Procedure

1.Perform full-system tests involving dynamic player movement and responsive projectile launching.

2.Conduct 20 integrated scenario trials simulating real-game situations.

3.Monitor and record subsystem coordination, tracking accuracy, and projectile launch precision.

3.3.2 Success Criteria

1.Launch adjustments respond correctly and timely to real-time player movements in at least 90% of cases.

2.Integrated system achieves an overall operational responsiveness with delays not exceeding 150 ms from detection to launch parameter adjustment.

3.Subsystem interactions are free from mechanical or software conflicts during testing.

This rigorous verification framework ensures comprehensive evaluation and demonstrates the DodgeBot system's efficacy, accuracy, and responsiveness under realistic training scenarios.

4. Costs

4.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Retail Cost (¥)	Bulk Purchase Cost (¥)	Actual Cost (¥)
DM-J6006-2EC Motor	Taobao	799	700	2*799
M3508 Motor	Taobao	499	450	3*499
MVL-HF0624	Taobao	256.91	250	259.91
MV-CS6-10UC HIKVISION Camera	Taobao	850	800	850
C620 regulator	Taobao	399	350	3*399
RealSense D435	Taobao	2250	2200	2255
dodgeball	Taobao	30.73	30	30.73
Fasteners for installation	Taobao	92.64	90	92.64
extension bar	Taobao	5.32	5	5.32
screw	Taobao	20.37	20	20.37
3D printed components	Taobao	10	10	10
friction wheel	Taobao	176.24	170	176.24
copper column	Taobao	5.12	5	5.12
Strip magnet	Taobao	2.04	2	2.04
Nylon plastic cable ties	Taobao	0.8	0.8	0.8
Switch limit travel switch	Taobao	12	10	12
tuck net	Taobao	1.56	1	1.56
gradienter	Taobao	2.88	2.5	2.88
3D printing consumables	Taobao	50	50	4*50
total				**8 216.61**

4.2 Labor

Table 2 Labor cost

Name	Job	Working time
Chiming Ni	1.Designed the robot's PCBs. 2.Installed the operating system on the onboard computer. 3.Deployed the YOLO model. 4.Accelerated YOLO inference through parallel-computing optimizations. 5.Developed the computer-vision targeting system.	500h

	6.Architected the overall software framework for the robot.	
Feiyang Wu	1.Developed driver code for the M3508 and DM6006 motors. 2.Designed the PID control algorithms and tuned their parameters. 3.Implemented the main control software for the dodgeball robot. 4.Soldered and assembled the printed-circuit boards. 5.Calibrated and debugged the Intel RealSense D435 depth camera. 6.Performed the robot's electrical integration and wiring.	500h
Kai Wang	1.Modeled the friction wheel structure. 2.Designed the robot's fastening-hole layout with precise tolerances. 3.3-D printed all structural components. 4.Assembled the printed parts into the finished robot.	500h
Nichen Tian	1. Produced detailed CAD models for the ball-feeder shroud, motor mount, rotating base, and side link plates. 2. Procured mechanical parts and assembled the initial hardware prototype. 3. Designed the complete launching mechanism in CAD. 4. Optimized mounting-hole patterns, converted the feeder shroud to a modular assembly, upgraded the yaw axis with an iron slip-ring design, and coordinated vendor laser-cutting of custom plates. 5. Managed procurement of all fasteners and other mechanical connectors. 6. Participated in bench and field testing of the system.	500h
Yiyang Bao	1. Authored the complete design document. 2. Developed and fine-tuned the YOLO algorithm. 3. Wrote the final project report. 4. Assembled the robot hardware. 5. Created the project presentation slides.	200h

5. Conclusion

The DodgeBot project set out to build an affordable, autonomous training partner that can repeatedly launch dodge-balls with game-like realism while adapting to an athlete's movements. Over the spring term we delivered a robust two-subsystem prototype – a friction-wheel shooter driven by a 3-DoF gimbal and a YOLO-based vision stack – and demonstrated closed-loop targeting accuracy in lab conditions. The following subsections summarise what we achieved, where the design still faces uncertainty, the ethical framework that guided our work, and the major piece of functionality we did not finish: an active dodging module.

5.1 Accomplishments

5.1.1 Mechanical & Mechatronics

Velocity Stability ±0.05 m s^{-1} – 50 consecutive launches recorded at 2 m s^{-1} ± 0.05 m s^{-1} using radar chronography, surpassing the ±0.1 m s^{-1} requirement.

Reload Cadence 0.8 s – The dial-plate loader chambered balls in 0.78 s on average (σ = 0.04 s), enabling sustained firing bursts.

Gimbal Reach – 30 ° pitch and unlimited yaw covered a 6 m × 6 m practice square from a single mounting point.

5.1.2 Computer Vision & Controls

Dataset & Model – 1 500 manually-labelled images produced a YOLOv7-tiny network with mAP@0.5 = 0.82 and recall = 0.86.

Real-time Inference – Jetson Nano sustained 28 FPS for an end-to-end latency of 92 ms (image capture → detection → coordinate output).

Closed-loop Accuracy – Mean impact error of 8 cm (N = 30) at 3 m stand-off while the athlete moved laterally ≤ 1.5 m s^{-1}.

5.1.3 System Integration & Safety

Single-button emergency stop cut motor power in < 25 ms.

Firmware-enforced speed limiter guaranteed < 2.2 m s^{-1} even at full Li-Po charge.

30-minute continuous-fire endurance test completed with motor casings < 52 °C.

These metrics collectively meet or exceed every quantitative requirement for shooting speed, targeting precision, latency, and operator safety.

5.2 Uncertainties

5.2.1 Environmental Robustness

Extreme lighting conditions (> 2 000 lux sunlight, < 20 lux dim gyms) and back-lighting reduce model confidence below 0.6, causing missed detections and jitter.

5.2.2 Occlusion & Crowding

The vision stack is trained on single-athlete frames; multiple players or partial occlusions (e.g., hurdles) drop mAP to 0.55 and degrade targeting.

5.2.3 Surface & Mechanical Longevity

Outdoor wind and uneven terrain have not been quantified. Urethane wheel glazing after ~3 000 launches may alter friction and velocity consistency.

5.2.4 Dataset Bias & Generalization

Current labels skew toward light-colored sportswear; limited body-shape diversity may affect detection on broader populations.

5.2.5 Regulatory Landscape

Local rules governing projectile machinery and protective-equipment mandates remain unclear; compliance pathways need investigation.

Addressing these uncertainties is essential for reliable field deployment and potential commercialisation.

5.3 Ethical considerations

The ethical and safe development of DodgeBot is crucial, given its automated projectile launching and real-time motion tracking capabilities. To prevent accidents, misuse, and ethical concerns, the project adheres to IEEE and ACM Codes of Ethics alongside industry safety standards. This section examines key ethical considerations, potential risks, regulatory compliance, and safety measures, referencing established robotics and safety guidelines.

The IEEE Code of Ethics [12] emphasizes prioritizing human well-being, honesty, and harm prevention, while the ACM Code of Ethics [13] stresses the importance of privacy protection, fairness, and transparency in AI-based systems. DodgeBot's tracking system, which utilizes YOLO-based computer vision, must ensure that no personal or biometric data is stored or misused. The system will be designed for real-time image processing without data storage, preventing privacy concerns.

Misuse of the system presents another ethical risk. If modified, DodgeBot could fire projectiles at unsafe speeds or in unintended environments, leading to injury or property damage. To mitigate this, hardware and software constraints will be implemented, including speed limits, launch activation only when a player is detected, and safety lock mechanisms. These precautions align with responsible technology development principles outlined by the ACM.

The transparency of system operation is another ethical requirement. Players should fully understand how DodgeBot's tracking and shooting mechanisms work, ensuring trust and informed consent in training environments. Ethical concerns also extend to attribution and research integrity; proper citations will be provided for open-source libraries, research papers, and industry standards used in the project.

5.4 Safety Considerations

Given that DodgeBot launches projectiles autonomously, its design must follow industry safety standards to prevent injuries. The primary risks associated with the system include high-speed projectile impact, unintended firing, mechanical hazards, and electrical safety issues. Each of these risks is assessed below.

5.4.1 High-Speed Projectile Impact

DodgeBot is designed to launch dodgeballs at a controlled speed of 2 m/s, ensuring safe player interaction. However, misconfigurations or unauthorized modifications could increase velocity, creating an injury risk. To prevent this, a hardcoded velocity limiter will restrict the system from exceeding 2 m/s, aligning with safety recommendations for robotic projectile-based mechanisms [3].

5.4.2 Unintended Firing Without Player Detection

If the tracking system fails to detect a player, DodgeBot may fire projectiles unintentionally. To prevent this, the system will include a player presence verification mechanism, where firing is only activated when a valid human target is detected. This aligns with safety guidelines for autonomous robotic arms and projectile systems [4].

5.4.3 Mechanical Hazards from Moving Parts

The 3-DoF gimbal and motorized base could present risks of pinching or entanglement. To mitigate this, protective shielding and mechanical stop limits will be implemented, ensuring that users cannot make direct contact with moving parts [6].

5.4.4 Electrical and Battery Safety

DodgeBot relies on high-power motors and controllers, which may lead to overheating or electrical faults if not properly managed. The system will comply with UL (Underwriters Laboratories) electrical safety standards [14] and feature automatic shutdown mechanisms to prevent overheating.

5.4.5 Emergency Stop and Manual Override

If a hardware or software malfunction occurs, users must have a manual emergency stop option. A highly visible emergency stop button will be integrated, allowing immediate deactivation of all motor functions. This is a requirement under ISO 13482:2014 (Safety Standards for Personal Care Robots) [15].

5.4 Future work – Re introducing an Active Dodging Function

The original vision for DodgeBot included a self-preserving dodging subsystem that would move the platform out of an incoming ball's path, forcing the athlete to vary throws and better replicating live opponents. Time constraints and mechanical complexity led us to discover this feature. To realize it in a subsequent iteration we propose:

1. **Dedicated Mobility Hardware** – a differential-drive base with omni-wheels (≥ 2 m s^{-1} transit speed) and a low-profile torsion suspension to maintain gimbal stability.

2. **Incoming-Ball Perception** – fusion of a 240 FPS global-shutter camera and two time-of-flight sensors to estimate ball trajectory within < 10 ms of release.

3. **Evasion Planning** – lightweight MPC (model-predictive control) seeded by a neural flight-time predictor; early simulations show feasible sidestep distances of 0.4 m in 250 ms.

4. **Safety Envelope** – dynamic speed limits and geofencing to avoid collisions with the athlete; override hooks wired to the existing E-stop chain.

5. **Incremental Roll-out** – begin with lateral shuffle only, then add fore-aft hops once stability benchmarks (tip-angle < 5°) are met.

Implementing the above will return DodgeBot to its full conceptual capability and create a uniquely engaging, two-way training partner for dodgeball athletes.

References

[1] E. Gokbel and S. Ersoy, “Launchable rotary wing UAV designs and launch mechanism designs for rotary wing UAV,” *J. Mechatron. Artif. Intell. Eng.*, vol. 2, no. 2, Art. no. 2, Dec. 2021, doi: 10.21595/jmai.2021.22339.

[2] K. Xue, C. Ren, X. Ji, and H. Qian, “Design, Modeling and Implementation of a Projectile-Based Mechanism for USVs Charging Tasks,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 288–295, Jan. 2023, doi: 10.1109/LRA.2022.3154036.

[3] V. Gies *et al.*, “Optimisation of Energy Transfer in Reluctance Coil Guns: Application to Soccer Ball Launchers,” *Appl. Sci.*, vol. 10, no. 9, Art. no. 9, Jan. 2020, doi: 10.3390/app10093137.

[4] V. Sharma, R. B, H. S, and D. Prasad, “Pneumatic Launcher Based Precise Placement Model for Large-Scale Deployment in Wireless Sensor Networks,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 12, 2015, doi: 10.14569/IJACSA.2015.061222.

[5] G. S. Chirikjian, “An architecture for self-replicating lunar factories,” *Natl. Aeronaut. Space Adm. NASA Inst. Adv. Concepts NIAC Phase 1 Final Rep*, 2004.

[6] A. Farinha, R. Zufferey, P. Zheng, S. F. Armanini, and M. Kovac, “Unmanned aerial sensor placement for cluttered environments,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6623–6630, 2020.

[7] R. L. Davis, “Mechanical design and optimization of swarm-capable UAV launch systems,” PhD Thesis, Monterey, California: Naval Postgraduate School, 2015.

[8] H. Guo, A. Priimagi, and H. Zeng, “Optically controlled latching and launching in soft actuators,” *Adv. Funct. Mater.*, vol. 32, no. 17, p. 2108919, 2022.

[9] X. Wang *et al.*, “Fracture-driven power amplification in a hydrogel launcher,” *Nat. Mater.*, vol. 23, no. 10, pp. 1428–1435, 2024.

[10] S. A. Murugan, S. Dharsini, S. G. Subramaniyan, M. Kumar, O. A. Lokhande, and Y. S. Narayanan, “A Non-Lethal System for Preventing Maritime Vessels from Invading or Evading Naval Perimeters,” in *2022 10th RSI International Conference on Robotics and Mechatronics (ICRoM)*, IEEE, 2022, pp. 515–520. Accessed: May 16, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10025344/

[11] N. M. Nyagudi, “Post-Westgate SWAT : C4ISTAR Architectural Framework for Autonomous Network Integrated Multifaceted Warfighting Solutions Version 1.0 : A Peer-Reviewed Monograph,” Jan. 27, 2014, *arXiv*: arXiv:1401.6379. doi: 10.48550/arXiv.1401.6379.

[12] “IEEE Code of Ethics | IEEE.” Accessed: May 16, 2025. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8

[13] “Code of Ethics.” Accessed: May 16, 2025. [Online]. Available: https://www.acm.org/code-of-ethics

[14] “Electrical Safety | UL Standards & Engagement.” Accessed: May 16, 2025. [Online]. Available: https://ulse.org/ul-standards-engagement/electrical-safety

[15] T. Jacobs and G. S. Virk, “ISO 13482-The new safety standard for personal care robots,” in *ISR/Robotik 2014; 41st International Symposium on Robotics*, VDE, 2014, pp. 1–6.

[16] “ANSI/RIA R15.06-2012 - Industrial Robots and Robot Systems - Safety Requirements (CONTAINS CORRIGENDUM).” Accessed: May 16, 2025. [Online]. Available: https://webstore.ansi.org/standards/ria/ansiriar15062012?srsltid=AfmBOopgrgLewIxLE_d78rDNqBZrtmpuAWZAY8hmn6r6M--nrihSUGsc

[17] “1910.212 - General requirements for all machines. | Occupational Safety and Health Administration.” Accessed: May 16, 2025. [Online]. Available: https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.212

Appendix A Requirement and Verification Table

Table 3 System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Dodgeball Shooting System a. Launch speed = 2 m/s ± 0.1 m/s (15 trials) b. Target accuracy ⩽ ±10 cm (⩾ 90 % trials)	**1. Shooting Verification** a. Use calibrated radar / high-speed camera to record velocity for each shot (15 trials) b. Mark intended impact point; measure deviation for each shot and compute success rate	
2. Human Pose Estimation & Tracking a. Localization error ⩽ ±5 cm (⩾ 80 % positions) b. End-to-end latency < 100 ms c. Tracking accuracy ⩾ 80 % under 50 – 1000 Lux	**2. Tracking Verification** a. Capture 50 motion paths; compare YOLO output to motion-capture ground truth b. Measure frame-processing time from image capture to pose output c. Repeat tests under low (≈50 Lux) and bright (≈1000 Lux) lighting; compute tracking accuracy	
3. Integrated System a. Launch reacts correctly to player motion in ≥ 90 % scenarios b. Detection-to-launch adjustment delay ≤ 150 ms c. No mechanical / software conflicts during operation	**3. Integration Verification** a. Run 20 full-system trials with dynamic player movement; log whether launch parameters update appropriately b. Timestamp detection event and launch-parameter commit; calculate delay per trial c. Monitor logs & hardware for errors, collisions, or controller faults	