AIAA Aviation Forum and ASCEND co-located Conference Proceedings
29 July - 2 August 2024, Las Vegas, Nevada
AIAA AVIATION FORUM AND ASCEND 2024
10.2514/6.2024-3896

Lighter-Than-Air Vehicle Design for Target Scoring in Adversarial Conditions

Caeden Taylor[*] and Or D. Dantsker[†]
Indiana University, Bloomington, IN 47408

At the Defend The Republic (DTR) competition, collegiate teams strive to use their autonomous lighter-than-air vehicle (LTA) vehicles (i.e., autonomous blimps) to autonomously capture helium balloons and score them into the opponent's goals——effectively Robotic Quidditch. The challenge involves designing and controlling robot vehicles with physical constraints that can operate in challenging environments with limited sensing, actuation, and computation capabilities. This paper will describe the methodologies used by the Indiana University (IU) DTR team to develop their 2 target scoring LTA vehicles used to win the Fall 2023 DTR tournament. Specifically, the paper provides details regarding the DTR competition including background, team structure, tournament rules, and vehicle restrictions. Then analysis of several competing team vehicles are presented. Afterwards, design methodology used to develop the scoring vehicles are presented, followed by detail on the two scoring vehicle designs developed over the course the Fall 2023 semester leading up to the DTR competition. A discussion of system architecture and integration is given. The paper concludes with lessons learned, and planned future work.

I. Introduction

At the Defend The Republic (DTR) competition, collegiate teams strive to use their autonomous lighter-than-air vehicle (LTA) vehicles (i.e., autonomous blimps) to autonomously capture helium balloons and score them into the opponent's goals——effectively Robotic Quidditch. Fig. 1 shows an example of the Indiana University's LTA vehicles navigating the game space during a DTR match. The challenge involves designing and controlling robot vehicles with physical constraints that can operate in challenging environments with limited sensing, actuation, and computation capabilities. Before the competition, teams perform vehicle design, fabrication, prototyping, embedded development, systems engineering, feedback control, motion planning, computer vision, among many other research and development tasks. Teams typically meet at a hosting university twice per year to compete in a round robin or bracket tournament.

The Fall 2023 DTR competition was hosted by Lehigh University at their Mountaintop Campus from November 13 to 17. Participating teams included Baylor University, Drexel University, George Mason University, Indiana University, Lehigh University, University of Florida, Virginia Tech University, and West Virginia University. Our Indiana University (IU) team won the Fall 2023 DTR tournament by applying strong foundational principles for autonomous unmanned aerial vehicle design to gain the competitive edge, which also serves as a strong platform for future LTA vehicle designs. Fig. 1 shows the 3 IU LTA vehicles used during the competition (2 target scoring and 1 defensive[1]).

This paper will describe the methodologies used by the IU DTR team to develop their 2 target scoring LTA vehicles used to win the Fall 2023 tournament. The paper starts by describing the DTR competition, including the competition background, team structure, tournament rules, and vehicle restrictions. Then the paper presents an analysis performed of competing team vehicles. Afterwards, design methodology used to develop the scoring vehicles will be presented, including ideation, vehicle design concepts, propulsion and control concepts, and system integration and computer vision. Then the paper will detail the scoring vehicle designs developed over the course the Fall 2023 semester leading up to the DTR competition. A discussion of system architecture and integration will then be given. The paper will conclude with lessons learned, and planned future work.

*Undergraduate Student, Department of Intelligent Systems Engineering, caedtayl@iu.edu
†Assistant Professor, Department of Intelligent Systems Engineering, odantske@iu.edu

Figure 1. Indiana University DTR team's LTA vehicles (blue) navigating the game space during an match against Baylor University (red) with goal posts (yellow) and game balls (green and purple) in the background.

II. DTR Competition Background

A. Competition

The Defend The Republic (DTR) Competition consists of collegiate teams fielding their autonomous LTA vehicles to autonomously capture green and purple helium balloons and score them into the opponent's yellow or orange goals. The scope of the challenge includes the physical architecture, sensor payload and design, software implementation, cyber-physical development of the entire system. Designs are constrained by total buoyancy, limiting complexity, sensing capabilities, and overall computational power. Typical implementations include camera-based computer vision navigation with brush-less motor propulsion. Basic nets are used for ball capturing and mylar balloons provide the lift for the vehicles.

Teams participating in DTR include Baylor University, Drexel University, George Mason University, Indiana University, Lehigh University, University of Florida, Virginia Tech University, and West Virginia University. Teams are typically structured as either faculty-led research laboratories within universities and/or as continuing senior design capstone projects for students. Teams typically compete at a hosting university twice per year to compete in a round robin or bracket tournament and demonstrate their LTA vehicles and technologies. The Fall 2023 DTR competition was hosted by Lehigh University at their Mountaintop Campus from November 13 to 17.

Indiana University's participation in the DTR competition has been part of the ENGR-E 399/599 "Autonomous Sports" class.[2,3] The competition represents the practical application and testing of the final vehicles the class developed.

B. Tournament Structure

The competition is structured with a predefined schedule for each team with several games. Each team is then paired with their closest competitor for a final match to determine the final rankings. In the case of a tie, the team with the most total points across all three games wins.

C. Game Rules

Each game has two 30 minute halves and 30 minute halftime. During each 30 minute half, there are six 5 minute intervals, with 30 seconds of manual blimp control, and 4.5 minutes for autonomous flight. In order to score points, teams must capture and score balls through goals placed on opposite ends of the field. To qualify for competition, teams must prove they are capable of scoring unopposed, during a single 30 half of a standard game. Teams advance by scoring the most points during a game, with different point values based on the level of autonomy used to complete a task. Capturing and scoring the ball with uninterrupted autonomy is worth 10 points. Autonomous capture of a game ball, without scoring is worth 1 point. Scoring a goal manually is worth 1 points. And intentionally passing a ball between vehicles is worth 10 points.

The playing field consists of two sets of three uniquely shaped goals hung from the roof of the competition space at either end. Each set of goals have one triangle, square, and circle in either yellow or orange. The circle has an interior diameter of 36.5 inches, and an outer diameter of 44.5 inches. The square has an inside leg length of 38 inches, with there outside length being 46 inches. The triangular goal is an equilateral triangle hung upside down, with the total height from base to tip being 55 inches. The goals are made from plywood and retro-reflective tape in order to enhance visibility. The following diagram is a two-dimensional birds-eye view of a DTR match.

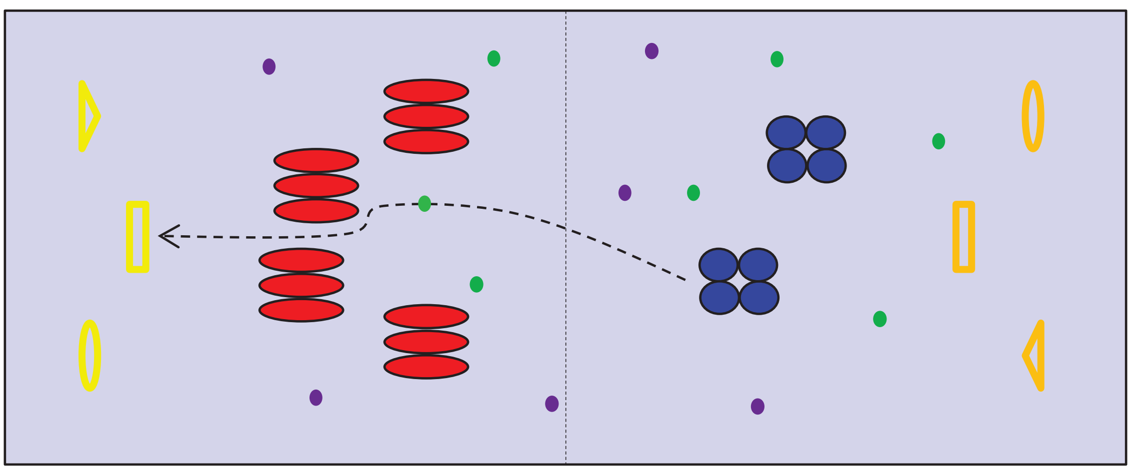

Figure 2. Birds-eye view representation of a DTR match layout.

D. Vehicle Restrictions

Each LTA must follow a strict set of rules, constraining helium usage, overall weight, and capture mechanisms. The team is allotted 200 cubic feet of helium to inflate their entire fleet for the week. Each vehicle while at rest must weigh no more than 100 grams, including helium, which may not be more than 50 cubic feet. Finally, all vehicles cannot use outwardly sticky adhesives for capturing objects in the game environment. All signals affecting the LTA must originate from within the control tent of the respective team, and the vehicle cannot sit at rest on the ground during autonomy.

III. Competitor Analysis

The DTR competition included teams from 8 universities across the United States, including Baylor University, Drexel University, George Mason University, Indiana University, Lehigh University, University of Florida, Virginia Tech University, and West Virginia University. Vehicles brought to the event fell into two categories, scoring and interference vehicles. Scoring vehicles utilize a ”basket” style capturing mechanism to catch balloon objectives, and continuously attempt to score. Interference vehicles use nets, ”spears”, strings, or direct contact in order to prevent the opposing teams vehicles from completing their autonomous objectives. Several of the competing team’s vehicles are presented below and were used for reference during the development of the IU DTR scoring vehicles.

A. Baylor University

Baylor University’s vehicle, which is shown in Fig. 3, was composed of a custom mylar envelope, with a square structured capture mechanism. The design’s purpose is to capture and score as many balloons as possible during both the manual and autonomous periods. The vehicle was highly stable during flight, and performed well throughout the competition. The potential limitation of the design is barrier to entry to fabricate custom mylar balloons.

Figure 3. Baylor University’s scoring vehicle [screenshot taken from YouTube[4]].

B. George Mason University

George Mason University utilized both an interference and 2 different scoring vehicle designs during the 2023 Fall DTR competition. Fig. 4 shows George Mason University’s primary scoring vehicle performing an autonomous capture. The vehicle uses a passive net design to capture balls and a large custom mylar balloon for buoyancy. Meanwhile, George Mason University’s ”Flappy” experimental vehicle, shown in Fig. 5, utilized a bio-mimicry propulsion system with wings. Bio-mimicry-based designs hope to achieve higher efficiency over conventional propulsion systems.

The main interference vehicle demonstrated by George Mason University was a broad sweeping net vehicle, designed to capture objective balls and the opposing team’s vehicles. Fig. 6 depicts a sweeping net interfering with another competing vehicle (light blue ”pill”-shaped balloon). The sweeping-net design proved successful, with multiple opponent captures and interference's throughout the course of the competition week. One major limitation of the design is the high design complexity and development time, with construction requiring dozens of hours and multiple students.

While George Mason University's' vehicles successfully demonstrated their capabilities, maintaining a large and unique fleet requires a large workforce, additional materials for each design, and specific knowledge on repair and use, overall reducing fielding capacity during the competition. This issue is further compounded by the use of custom sealed mylar balloons on all vehicles.

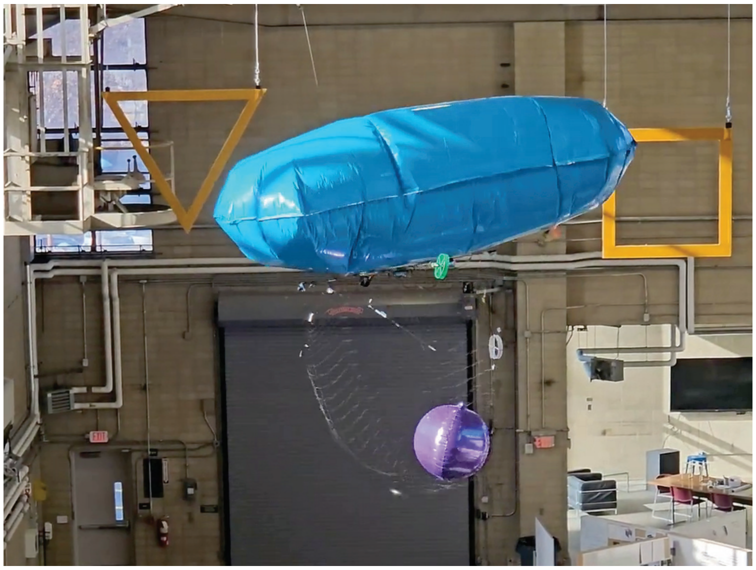

Figure 4. George Mason University's scoring vehicle [screenshot taken from YouTube[4]].

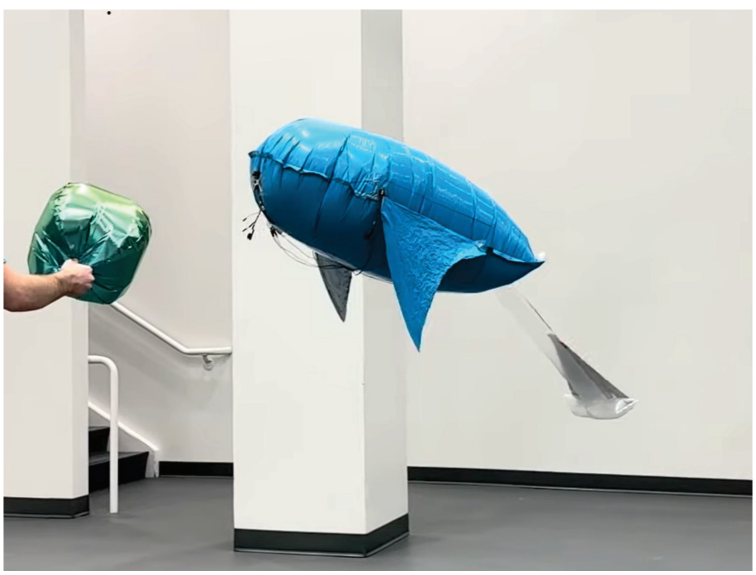

Figure 5. George Mason University's "Flappy" autonomous ball capture vehicle [screenshot taken from YouTube[5]].

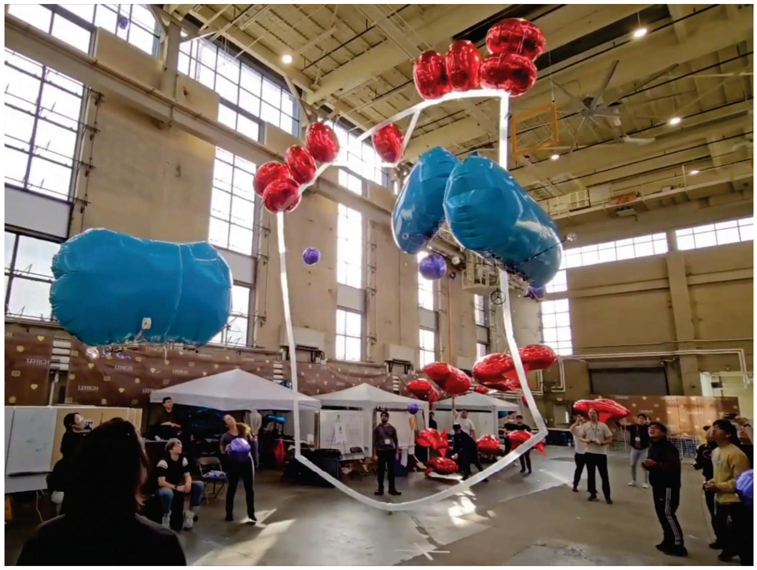

Figure 6. George Mason University's "Skynet" autonomous interference vehicle (red) capturing University of Florida scoring vehicles (blue) [screenshot taken from YouTube[6]].

C. Lehigh University

Lehigh University utilizes many low-cost vehicles to fill the playing field. Several designs are shown in Fig. 7 and 8. By engineering a small, highly agile vehicles their designs require less helium than many competing vehicles, increasing the number of total usable vehicles during a competition. The quantity of opposing vehicles prevents opposing teams from interfering with all vehicles in play, and work to effectively maximize the odds of scoring. Due to the lightweight design, these vehicles may struggle to deal with relatively weak environmental forces, such as temperature-based drafts of air conditioning, or opposing interference vehicles. The Fig. 8 shows the Lehigh University Vehicle capturing a game ball during testing, displaying the small scale of their vehicles.

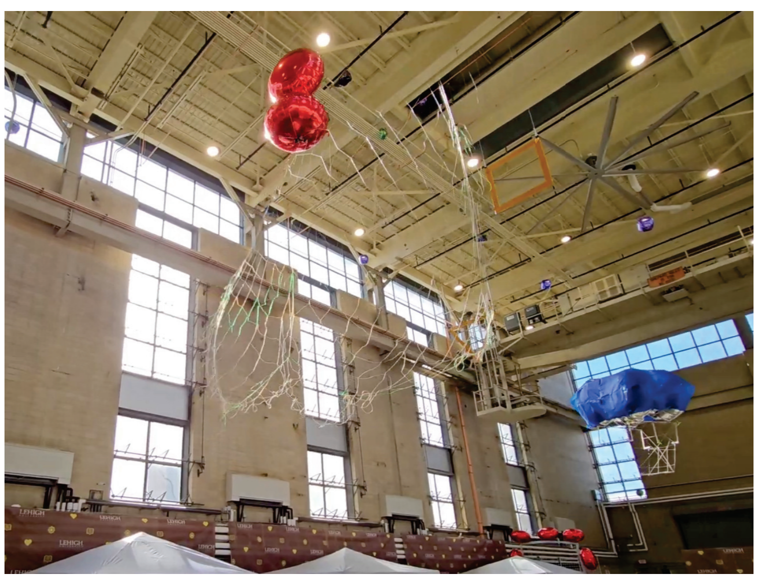

Figure 7. Lehigh University's spinning netted ball-capture and scoring vehicle [screenshot taken from YouTube[7]].

Figure 8. Lehigh University's cooperative scoring vehicles: netted ball-capture (right) and scoring (left) [screenshot taken from YouTube[8]].

IV. Design Methodology

The Indiana University LTA vehicles were designed based on the vehicle requirements and gameplay rules described in Section II, including limitations for helium use, maximum mass on scale, and capture mechanisms. The LTA vehicles were designed by undergraduate students, utilizing common materials and readily-available hardware modules. These constraints prioritized finding the solution with the lowest overall system mass.

The design of the vehicle was divided into three major categories: physical design, propulsion and control, and integration/computer vision. Each category has a corresponding team assigned to develop and research the given field. Each team has a secondary priority to assure that the corresponding interfaces between categories is functional and well defined for overall integration.

A. Design Ideation

Each sub-team developed a list of ideas to complete objectives, such as controlled flight, or successful ball detection and reproducible design. The following list includes examples of ideas explored by students during the semester in the development categories.

- Vehicle Design
  - Passive net capturing mechanism
  - Active mechanical arm grapple
  - Suction-based capture and ejection.
- Propulsion and Control
  - Flapping wings
  - Static electric motors
  - Dynamic servo-mounted electric motors

- Air density control (hot-air balloon style)

- Computer Vision/Integration
  - Raspberry Pi control
  - Blob ball detection
  - Arduino motor control
  - WiFi communication
  - Object Oriented Python implementation
  - OpenMV
  - ESP32CAM

B. Vehicle Design Concepts

An analysis of the previous Indiana University's team design revealed carbon fiber tubes took a majority of the vehicles weight budget, with unnecessary strength. Additionally, the system was difficult to develop due to a loss of code for the Pix-racer control unit. Multiple revisions led to a complete, independent redesign of the frame.

The final class-made vehicle design is unique from previous vehicle designs, with little custom material work or computer aided design. Carbon fiber rods on the previous vehicle were substituted with balsa wood beams to build the primary frame due to their cost, malleability and light weight. Plastic netting and cloth string was used to line the capture basket. Four 30 inch diameter cylindrical balloons are used to lift the vehicle from the outermost corners. The following figure depicts the construction of the vehicle. From the initial brainstorming sessions, the vehicle design team settled on the implementation of an active propeller based solution for capturing and ejecting the game balls.

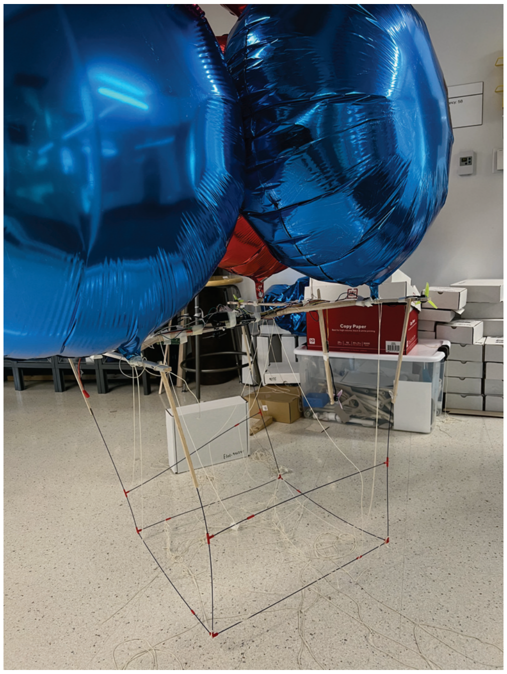

Figure 9. Photo of the student-led vehicle design prototype.

C. Propulsion and Control Concepts

The propulsion and control team went through many iterations to develop the most efficient and effective solution for autonomous motion. Air-density based height control resulted in weak, unstable lift and high power draw. Mechanical flapping wing designs proved to be heavy and difficult to control, and dynamic servo mounted motors required complex control systems, additional PWM pins, and weight.

In order to maneuver the vehicle, electric motors with propellers are statically placed on the left and right sides of the vehicle for differential control. An additional motor is placed in the center for upward thrust, and a fourth motor is placed in the rear for capturing and ejecting balloons. Differential control has a simple control matrix for converting control vectors to motor commands, simplifying software level control schemes.

The vehicle is powered by a two cell lithium ion battery, due to it's dense energy capacity of 2000 milliamp hours and high voltage. The cells are connected in series to double the nominal voltage output, driving the motors at thousands of rotations per minute.

Each motor connects to an electronic speed controller that was controlled by a multiplexer. The multiplexer allows a single signal to control the output to all of the motors. The multiplexer was wired to both a pulse-width-modulated control board that is controlled by a Raspberry Pi Zero 2W (RPi) single board computer, and a four channel hobby-grade remote control receiver.

D. Integration and Computer Vision

For computer vision, micro-controller solutions were found to be too slow for real-time applications, and required less intuitive coding languages. The team utilizes a RPi to control the vehicle, with onboard blob detection for autonomous control written with python. The RPi takes input from a Picamera sensor connected to the camera serial interface (CSI) of the RPi. The multiplexer's channel select is wired to a switch on the remote control radio controller, allowing the operator to control autonomous versus manual flight. Autonomous flight is controlled by the RPi with the camera, with a simple state machine for autonomous scoring.

V. Vehicle Designs Developed

Two LTA scoring vehicles were used in the DTR competition and are described below.

A. Class Prototype Vehicle Design

The Autonomous Sports class student team leveraged low-cost and and commercially available materials to quickly design, build and validate the developed vehicle design concept. Some low-cost approaches, including the use of balsa wood with adhesive reinforcement for the primary framework, influenced the final competition-ready vehicle. The balsa wood's weight to strength required excessive reinforcement, negatively impacting the overall system weight and reducing the maximize battery size for the system. Fig. 11 shows shows the class prototype vehicle while a students replaces the battery with adhesive.

B. Refined Composite Vehicle Design

The final revision of the student-led design incorporated the same design concept, however, applied more exotic materials further decreased system weight and increase performance. Carbon and Kevlar were glued onto the balsa

Figure 10. Photo of IU Autonomous Sports class prototype LTA vehicle while a student replaces the battery.

structure to create a strong yet flexible structure that could maintain a basket shape. Additionally, the carbon rod capture basket acted as a spring to dampen the force from quick descents.

Further weight optimizations were performed including adhesive balloon mounting, which brought the total vehicle weight significantly down to approximately 250 gr. With each of the 4 mylar balloons producing 85 gr of lift, the vehicle needed a 95-100 gr of ballast to maintain negative buoyancy, which in the case of power failure would return the vehicle to the ground. These strong fundamental vehicle design developments improved overall maneuverability and reduced total vehicle cost.

Figure 11. Photo of IU Team refined composite LTA vehicle during scoring practice.

VI. System Architecture and Integration

The system architecture must include factors of modularity, academic ease of use, and performance to create the best solution for autonomous control of the vehicle. Off the shelf solutions, as well as custom solutions, were evaluated in order to decide upon the ideal platform for software development. The following diagram visualizes the hardware components that the architecture must be capable of interfacing with, in order to retain full functionality.

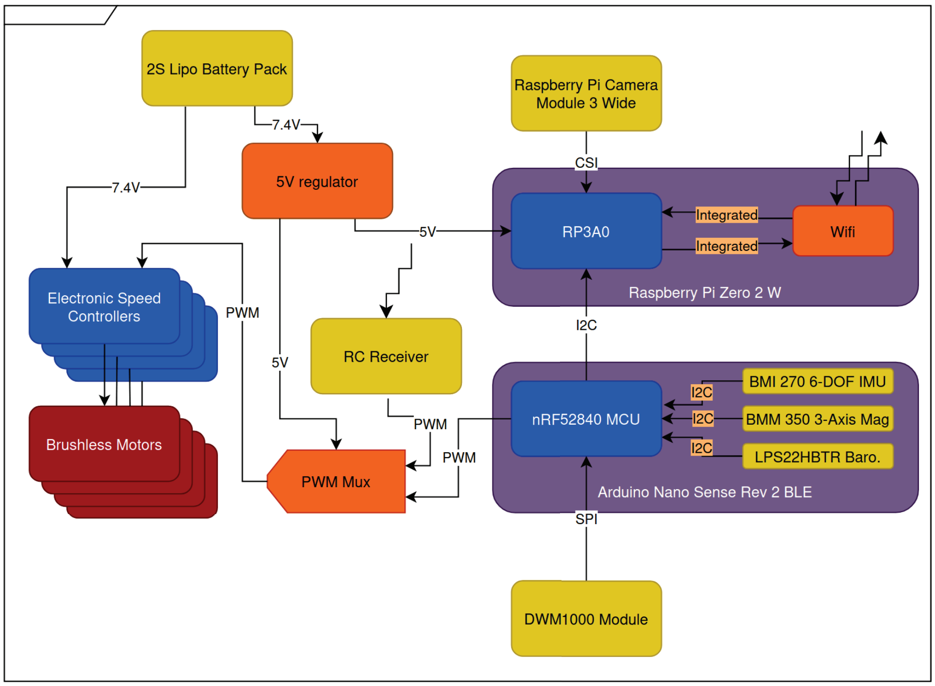

Figure 12. Hardware system diagram for the design of Indiana University's blimp.

Based on previous competitions, the system architecture team developed the following state diagram as a rudimentary approach for consistent goal scoring attempts. By approaching the goal opposite from the main target, the vehicle can consistently approach the target from a straight on approach, eliminating edge cases such as scoring while parallel with the face of the goal.

A. Software Control Architecture

Software architectures were evaluated and compared for best hardware compatibility, modularity, and sufficient abstraction. Potential architectures investigated were Robot Operating System 2 (ROS2), Object Oriented Python, and ArduPilot.

ROS2 is the standard software solution used for aggregating autonomous robotics sensors and platforms, with a standardized modular framework utilizing nodes and a publisher/subscriber architecture. ROS2 contains robust support for sensor packages, and abstracts low level distribution of hardware resources. Abstractions and seamless package communication requires large overhead, causing the RPi to struggle with real time tasks. ROS2 is better suited for larger autonomous systems with higher computational power and excels at high throughput, high traffic message routing.

ArduPilot prioritizes robust autonomous navigation and sensor fusion capabilities. ArduPilot utilizes three dimensional way-points and predefined vehicle dynamics to translate high-level actions to discrete vehicle movements.

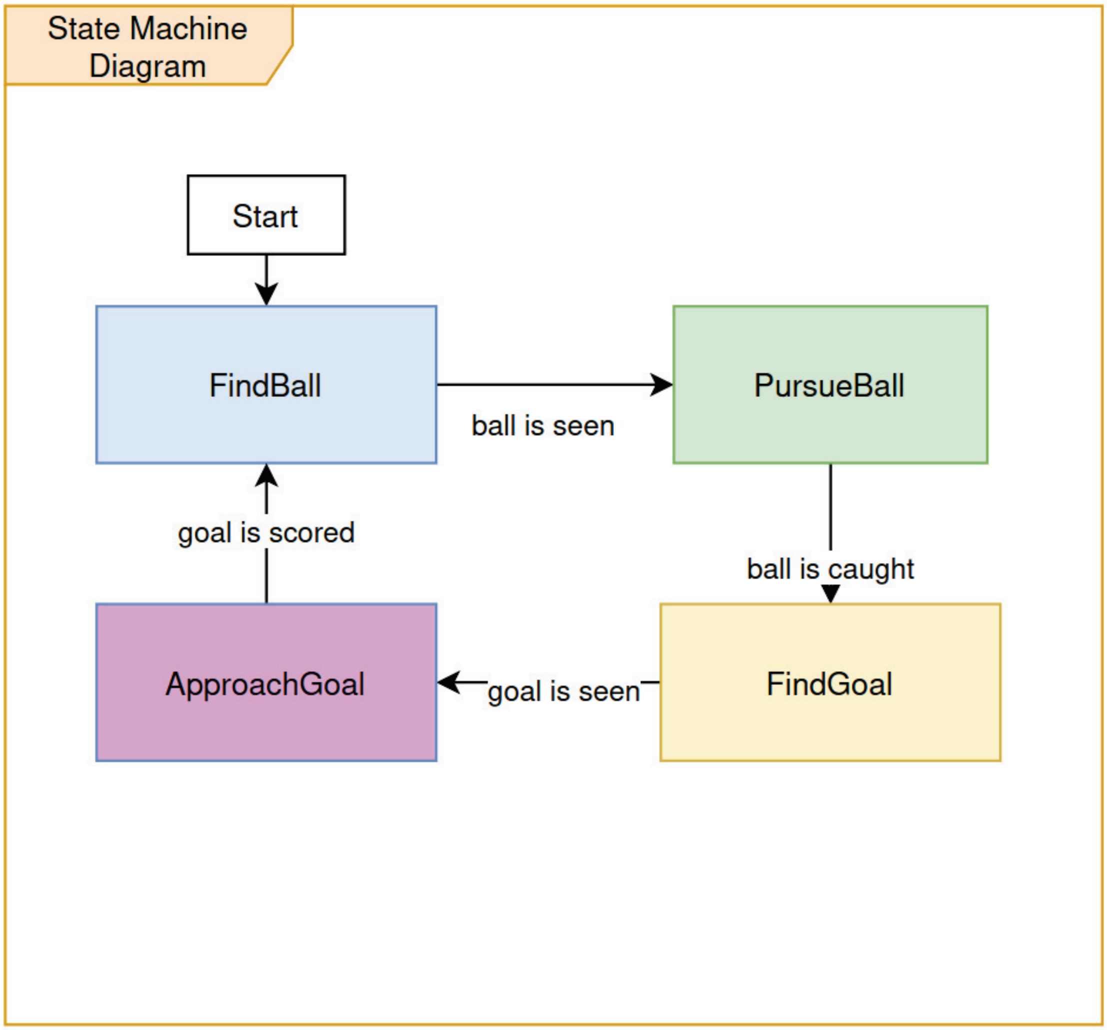

Figure 13. A flow chart diagram of the discrete state machine for the blimp.

ArduPilot communicates with MAVLink messages to enter input, control the vehicle, and output data. Ardupilot requires an additional translation layer to parse MAVLink messages to non-Ardupilot modules, and does not handle sub-meter precision way points effectively, such as capturing a slow moving balloon.

Custom object oriented Python code does not restrict communication methods, and by the nature of object oriented coding languages, provides abstraction and modularity. A key difference of custom python implementations compared to ROS2 or ArduPilot is the non-standardized method for passing information across modules. Developing and defining syntax for communication creates additional development overhead for system architects and developers.

The computational overhead of ROS2 constrains the performance of the vehicle to be inadequate for real-time computation. ArduPilot, while much faster than ROS2, obscures and abstracts access to lower level functionality for custom movement sequences. Custom object-oriented Python is used in order to maximize student productivity and education, due to it's familiarity in coursework.

B. Software Implementation Modules

Software implementation was written with five major custom pythonic objects, composing the "blimp" object, including the camera, state, sensors, and network objects. Configuration for each module was controlled by a set of Tom's Obvious Minimal Language (TOML) files to consolidate frequently changed variable values.

The module types were explicitly chosen to best mirror the modularity of the hardware counterparts, further enhancing the system's modularity. For instance, when a camera module was swapped for a higher performance variation, only the camera python script would need to be changed, preserving development time and reducing inter-module communication technical debt.

The sensor object primarily functions to send rudimentary motor control values to an Arduino-based subsystem for continuous PWM control, and returns basic altimeter data.

The state machine is a software system using the PyTransitions library for decision making. First, the vehicle uses a rudimentary search pattern to randomly navigate the space to locate a game ball. Second, once a ball was seen, the vehicle then uses the camera and sensor modules to continuously navigate towards the ball until it reaches a predefined trigger distance. Once the vehicle was within it's trigger distance, the vehicle "lunges" towards the ball, quickly moving forward to pull the ball into the net and out of the camera's field of view. This process was repeated for autonomously scoring a goal, with the capturing lunge replaced with a forward movement along with blowing the balloon out.

The network module asynchronously transmitted annotated images from the RPi over WiFi to a system on the ground for advanced diagnostics, development, and system presentation. The network module instantiates a TCP socket on the RPi as a server, and upon connection begins streaming compressed JPEG image byte streams. These byte streams are reassembled on the ground system and displayed through an additional python client.

C. Computer Vision

The camera object instantiated the physical camera device on the system, as well as providing methods for object detection. Object detection is the primary driver of the system state machine based on objects present within the frame.

In order to detect objects with the camera, the OpenCV library is used to parse images using blob-based detection methods. First, a mask is applied to the image with the "inRange()" function,[9] preserving all pixels within a specified hue, saturation, and value (HSV) range, and shifting all other pixels to black. HSV is independently determined for four color ranges that correspond to green balloons, purple balloons, yellow goals, and orange goals. All detection methods leverage the contour methods provided by OpenCV to calculate common parameters such as surface area, bounding boxes, and shape approximation. After detection, the pixel coordinates for the center of the object, width, height, angle to surface, and distance are all stored for further processing. The following figure depicts an example of a frame during competition filtered for the HSV range of a purple balloon.

Figure 14. Filtered image from the blob detection algorithm for purple ball detection.

To detect balloon objective detection, a frame is masked with an HSV filter for it's respective balloon color. The OpenCV ”findContours”[10] method accepts the masked frames, and returns a list of contours representing the groupings of filtered non-zero pixels. The largest contour that fits an approximate circular representation is labelled to be the closest ball to the screen.

Goal detection employs the same HSV masking method as ball detection with an additional function to determine goal shape. The OpenCV python wrapper's ”approxPolyDP” function approximates the number of sides of a contour, enhancing the functionality of the goal detection to distinguish between all six goals by color and shape.

Due to the isometric properties of goal balls, and the square proportions of the goal surfaces, geometrical functions can be exploited to additionally estimate distance and relative angle to a goal's face. Using the field of view in degrees of the physical camera and the camera resolution, the width of a perceived object can be expressed in degrees relative to the image sensor.

The calculation for the distance to a given object can be visualized as a triangle, with the base representing the real world width of the object, and the apex of the triangle representing the perceived object width in degrees. By bisecting the triangle down the center, the result is a right triangle with the apex and real objects length being cut in half respectively, and the adjacent side length representing the distance to the object. The distance estimation method requires advanced understanding of objectives, as well as fixed properties for the image capture system. The following equation calculates distance based on the objects real world width and the camera's perceived width in degrees. Distance to objectives were used to determine the appropriate speed to approach, and could trigger potential actions.

distance = (objectWidth)/(2*tan(perceivedCameraWidth)

Calculation of the angle to the face of a goal capitalizes further on the fixed proportions of the goal shapes. By comparing the perceived width and height of a detected object, to the known ratio of the true goal, the perceived difference can correspond to an angle between zero and ninety. Specifically, by calculating the width over height, values approaching one would indicate the vehicle is directly facing the goal. Values approaching zero indicate the goal is disproportionately skinnier than it's true dimensions, representing the vehicle being nearly parallel with the face of the goal. Due to the symmetry of the goals, further computation is required for determining whether or not the goal is skewed to the left or right for positional correction. The angle relative to the goal is an important measure for determining whether or not to attempt to score.

Due to the noise-prone results of color based detection and inconsistent detection across frames, temporal object permanence is maintained with additional averaging and processing between captured frames. Once detected, an object's attributes are approximated by using a weighted moving average (EWMA) filter. EWMA is a simple geometric filtering method that weights new data against all previous entries. The following equation describes how the EWMA is computed on every iteration.

$$EWMA_x = a * x + (1 - a) * EWMA_{x-1}$$

The following visualization depicts a noisy step-function signal, filtered with various levels of EWMA smoothing. As the alpha value approaches one, the filtered signal further resembles the noisy signal. Lower alpha values approaching zero result in a cleaner signal, with an imparted delay.

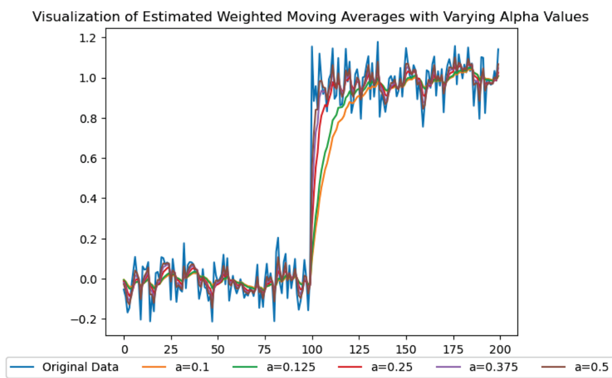

Figure 15. Graph displaying various levels of EWMA filtering for a noisy step function.

Using the EWMA filtering method additionally give the system a rudimentary indication of persistence. By applying the filtering to an input of ones and zeros, representing if the object is simply present in the frame, the intermittent detection can be used to gauge detection confidence.

Blob detection is computationally inexpensive, allowing the RPi to process over thirty frames per second for stable control of the vehicle. The following diagram depicts the cooperative integration of the various software modules during an autonomous goal scoring state.

VII. Lessons Learned

Over the course of the Autonomous Sports course and through November 2023 DTR competition, several important realizations were made:

A. Computer Vision

While blob detection is computationally performant, environmental factors heavily impact the accuracy and range of detection. Constant changes in lighting conditions due to windows in the competition space and the reflective nature of mylar balloons prevented fine-tuning of HSV ranges for a given game. To accommodate for the widened range of goal balloon color space, the HSV range for ball detection was widened much further than in artificial lighting, permitting

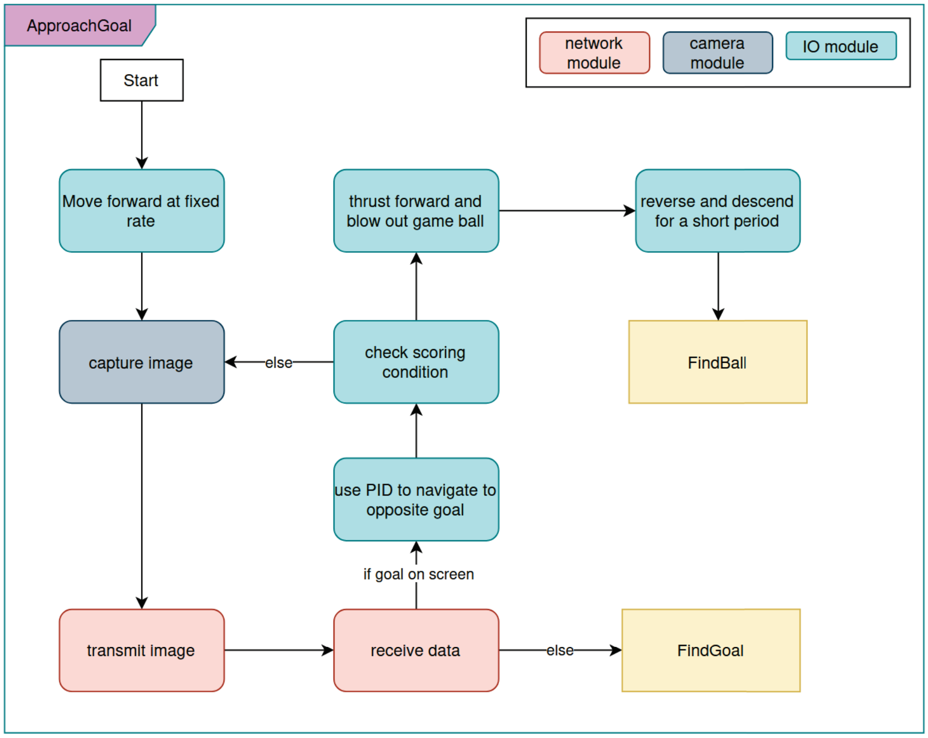

Figure 16. Flowchart for the actions and conditions during the "approachGoal" state in the state machine.

new false positive cases to occur. Fig. 16displays an instance of a false positive ball detection being labelled by the system during the competition due to the blue sky through a nearby window.

Intersecting goals additionally disrupted the detection, as overlapping contours could not be processed separately. The blob edge counting methodology did not have redundancy to handle intersection effectively, resulting in a complete loss of detection during overlap.

B. Vehicle Design

Despite the robust vehicle design, weight optimization resulted in the loss of convenience features. The lack of tool-free battery mounting, requiring the operator to use single-use adhesive to attach lithium-ion batteries for flight, slightly altered the motion dynamics of the vehicle. Due to the diffusion properties of the mylar balloons, the center of gravity and lift on the system changed daily throughout the entire course of the competition. Constant vehicle adjustments cascaded into inconsistent autonomous vehicle control, with daily calibration being required.

VIII. Future Work

Several design improvements are planned in future work. To fix the design implementation issues, a major redesign of the vehicle's balloon assembly is necessary. Consistent battery mounting points would decrease the time, cost, and positional error of swapping the lithium-ion battery. Closed-loop control of the vehicles pose using an inertial motion unit (IMU) would accommodate for natural variation between vehicle degradation over time and variance in vehicle design. Additionally, investigation in to optimal propulsion system elements is desired.[11–14]

Figure 17. Visualized results of the detection algorithm from the real game-time environment with a false positive detection for a purple ball in the background window.

One solution to the computer vision problems presented is the introduction of image-based machine learning. Computer vision algorithms have enabled similar bounding box detection, with additional redundancy and stability. Ultralytics's You Only Look Once (YOLO)[15] architecture provides a robust ecosystem for training, configuration, and deployment in a python environment. During preliminary research, students have found a substantial increase in dynamic environments, but slower inference times, down to less than a single frame per second. Further exploration of such models may allow for computer vision implementations to produce reasonable inference capabilities and inference times on the edge for more robust detection.[16]

Acknowledgments

References

[1] Taylor, C. and Dantsker, O. D., "Lighter-Than-Air-Vehicle Design for Adversarial Defense," Submitted to AIAA SciTech Forum 2025, Orlando, Florida, Jan. 2025.

[2] Dantsker, O. D., "Design, Build, and Fly Autonomous Lighter-Than-Air Vehicles as a Project-Based Class," AIAA Aviation Forum 2024, Las Vegas, Nevada, Jul. 2024.

[3] Dantsker, O. D., "Integrating Unmanned Aerial Systems into the Intelligent Systems Engineering Curriculum," International Congress on the Aeronautical Sciences Systems, Florence, Italy, Sep. 2024.

[4] LehighBlimpsClub, "Robots at DTR Lehigh 2023 - Day 1," https://youtube.com/shorts/CCUiIsVeV3I.

[5] GMUBlimpSquad, "Happy Robotic Stingray," https://www.youtube.com/watch?v=bWzLGA4qwtQ.

[6] GMUBlimpSquad, "GMU Braitenberg Blimps with a giant skynet at DTR competition in Nov 2023." https://www.youtube.com/watch?v=rOjwytXUIUQ.

[7] LehighBlimpsClub, "Lehigh U Vs Indiana U - DTR at Lehigh 2023," https://www.youtube.com/watch?v=NoKMjggwmBs.

[8] LehighBlimpsClub, "Testing ball passing with detachable net," https://www.youtube.com/watch?v=qJtQlauN-eM.

[9] OpenCV, "Thresholding Operations using inRange," https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html.

[10] OpenCV, "Structural Analysis and Shape Descriptors," https://docs.opencv.org/3.4/d3/dc0/group$_{imgproc_{shape.htmlga17ed9f5d79ae97bd4c7cf18403e1689a.}}$

[11] Deters, R. W., Dantsker, O. D., Kleinke, S., Norman, N., and Selig, M. S., "Static Performance Results of Propellers Used on Nano, Micro, and Mini Quadrotors," AIAA Paper 2018-4122, AIAA Aviation Forum, Atlanta, Georgia, June 2018.

[12] Chang, A., Pik, E., Deters, R. W., and Dantsker, O. D., "Empirical Propeller Mass Sizing for Small-Scale Aircraft," AIAA Aviation Forum 2024, Las Vegas, Nevada, Jul. 2024.

[13] Cox, B., Dantsker, O. D., Anantha, K., and Deters, R. W., "Propulsion System Testing Instrumentation for Multi-Rotor and Lighter-than-Air UAVs," Submitted to AIAA SciTech Forum 2025, Orlando, Florida, Jan. 2025.

[14] Cox, B. and Dantsker, O. D., "Performance Testing of Fixed-Blade Multi-Rotor UAV Propellers," Submitted to AIAA SciTech Forum 2025, Orlando, Florida, Jan. 2025.

[15] Ultralytics, "Ultralytics YOLO Docs," https://docs.ultralytics.com/.

[16] Taylor, C. and Dantsker, O. D., "Remotely-Processed Vision-Based Control of Autonomous Lighter-Than-Air UAVs with Real-Time Constraints," Submitted to AIAA SciTech Forum 2025, Orlando, Florida, Jan. 2025.