KENNESAW STATE UNIVERSITY AERIAL ROBOTICS COMPETITION TEAM TECHNICAL PAPER

Alec Graves

Kennesaw State University | agrave15@students.kennesaw.edu

Nihad Kalathingal

Kennesaw State University | nkalathi@students.kennesaw.edu

Alex Resnick

Kennesaw State University | aresnick@students.kennesaw.edu

ABSTRACT

Small unmanned aerial systems are being deployed more frequently for many tasks involving collection of visual data. These tasks can include a range of fields including security operations, equipment and infrastructure inspection, and even professional photography and cinematography. Deployment of these systems often requires a trained operator that can provide constant supervision and control using a specialized electromechanical Human-Machine Interface. This work aims at developing stronger autonomy of these aerial systems along with a more natural human interface to direct these systems. Specifically, this work extends the existing capabilities of a low-cost, commercially available platform to include obstacle detection, target tracking, unknown environment navigation, swarm coordination, and real-time voice control with the aim of solving the International Aerial Robotics Competition Mission 8.

INTRODUCTION

Problem Overview

The International Aerial Robotics Competition (IARC) has the stated purpose of “mov[ing] the state-of-the-art in aerial robotics forward” [1]. It is also hoped that this objective will be achieved “for the benefit of the world” [1]. IARC’s creators, directors, volunteers, sponsors, and participants aim to achieve this goal by creating mission challenges which are difficult to solve at the time of their conception but will hopefully be solved through development of new technology by competition participants [1]. 2019 is the first year of IARC’s eighth mission and includes several new challenges that have not been tackled in prior missions [1].

This work details a system which combines existing technologies in a novel way with the intent of solving IARC Mission 8. The challenge presented by Mission 8 requires solutions in natural human-machine interface, useful swarm information feedback to a human operator, swarm coordination, target location and tracking, and adversarial swarm-to-swarm interaction [1].

Additionally, this competition has multiple requirements emphasizing safety of interaction in enclosed environments with a mobile human operator [1].

The primary objective of IARC Mission 8 is to have the human operator retrieve a "critical object" from four bins in the competition arena. These bins are locked and must be unlocked by entering the correct four-digit password. This password will be available to the human operator as a QR code which is split into four pieces and displayed on four separate displays located in the arena. All four displays must be viewed simultaneously for a valid QR code to be reconstructed and the password to be decoded.

To accomplish viewing all partial codes simultaneously, the human operator uses four helper aerial robots equipped with a camera. These helper robots must be capable of flying through the arena without crashing into obstacles and hovering stably over the displays long enough for the valid partial QR codes to be obtained.

As an additional constraint, while the helper robots move to the displays, the human operator must avoid contact with laser beams transmitted by four enemy aerial robots. If the human is hit 10 times by enemy aerial robot lasers, the mission will terminate in failure. Only one enemy robot can engage the human operator at a time, and the enemy robots can only fire a laser once every 5 seconds. Each of the friendly aerial robots may transmit their own healing beam to counter a single hit from an enemy robot one time.

To help the human operator avoid laser hits while friendly robots navigate to the partial QR displays, inflatable obstacles placed throughout the arena may be used as cover.

As additional constraints to the solution, the helper aerial robots described in this work must have enclosed propulsors such that a human is unable to injure themselves by physically interacting with the robot. The robot design must also include a remote safety shutoff mechanism.

Conceptual Solution

Overview

This work is an attempt to assimilate all required components of a system that conceivably solve Mission 8 with limited human and economic resources. As such, emphasis is placed on leveraging available software and hardware platforms while creating robust connections between these existing modules alongside simple and effective higher-level control systems.

Aerial Robot

A low-cost, cheap, consumer-grade system is used as the basis for our platform to resource-efficiently leverage already existing technology. This platform, the DJI/Ryze Tello, is available at $100 per unit for the base unit and $130 per unit for the "EDU" edition which allows for control from an external Wi-Fi access point [2] [3]. This work uses the Tello EDU is designed in such a way that it is fully compatible with the base Tello. The Tello contains an IMU, forward-facing camera, laser altimeter, optical flow camera, and software for stabilization and control. Additionally, the Tello boasts Wi-Fi connectivity with a lightweight UDP messaging protocol from which control signals can be issued and real-time data from the platform's sensors and software can be acquired. This UDP protocol is well-documented in the Tello SDK [4] .

To fit competition requirements, the propulsor guards that come with the Tello are modified to further protect humans that contact the robot from injuring themselves. This is done in such a way that a human grasping the drone is reasonably unlikely to touch the spinning propulsor. The largest issue with this modification is the added weight to the system.

Networking and Computing

As mentioned, the hardware is controlled via a Wi-Fi connection. Due to the inability of the Tello EDU to transmit video over an external access point, each robot acts as its own Wi-Fi host which is joined by that robot's Control Computer (CC). All Control Computers are additionally networked together via a wired DHCP server. A GPU Processing Computer (GPC) is also connected to the server, and it acts as centralized processing and control node that interprets visual data from all four drones, interprets human operator inputs, provides feedback to control computers on the network and makes decisions about higher-level swarm behavior. This configuration is visualized in Figure 1.

The Control Computers are LattePanda embedded computers with 4GB of RAM and an Intel Atom x86_64 CPU [5]. They are each equipped with a high-gain USB Wi-Fi adapter for connecting to a Tello, an Ethernet port for connecting to the wired DCHP server, and an Intel Movidius Neural Compute Stick v1 for running time-critical neural networks [6].

Each Control Computer (CC) is responsible for providing all autonomy and control for a single Tello in real-time. This control involves monitoring and acting on battery and other diagnostic information from the robot, doing limited visual processing for real-time detection tasks, forwarding the robot's camera data and status information to the GPC, commanding the robot to the desired setpoints issued by the GPC, avoiding obstacles using information provided by the GPC, and issuing the kill signal when an emergency shutdown is required.

The GPU Processing Computer (GPC) is equipped with an x86_64 CPU, 16GB of RAM, and Nvidia GeForce GTX GPUs. This computer is responsible for processing audio input from the human operator's Pi-Boy and sending appropriate swarm commands to the CCs. It is also responsible for processing visual data provided from the CCs with larger neural models which cannot be run by the CCs because of their limited hardware.

Figure of System Architecture

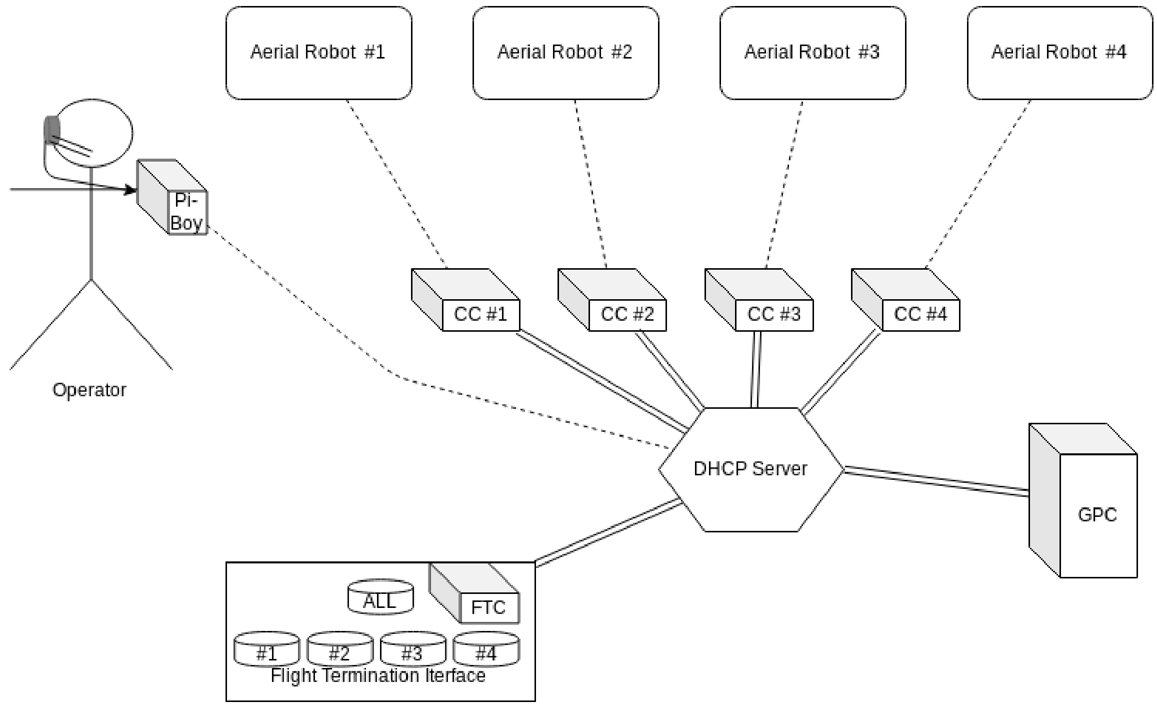

Figure 1. System Communication Architecture

Yearly Milestones

In order to ensure a fast development process, milestones were decided into primary and secondary milestones. The primary milestones pertain to the fundamental technology required to complete the mission:

- Safety
- Computer vision methodology to allow the drones to have:
    - Target identification and behavior.
    - Threat identification and behavior.
    - Gesture identification and behavior.
- Swarm behavior which allocates tasks to individual drones based on commands given by the user.
- Speech recognition to allow for the user to send commands to the drone.
- Networking solutions for all components.

The secondary milestones pertain to the technology needed to implement the primary goals:

- Architecture and organization of the drone.
- Design and manufacturing of the drone.
- Networking and final communications interface.

AIR VEHICLE

Description of Configuration/Type

The Tello EDU is a simple quadcopter/multi-rotor UAS that comfortably fits the team's budget for procuring or building drones. The drone's unmodified, compact body can fit inside of a 10 cm x 10 cm x 4.5 cm box and weighs only 80 grams with the battery installed. The flight time is advertised as 13 minutes, and in-lab testing has confirmed the flight time to be within a reasonable realm of error of the advertised time. The Tello EDU has a range of 100 m in the current form and therefore meets the range requirements for the arena. Additional modifications to the prop guards will be performed so that the drone will meet the covered prop requirements.

Flight Control System

Navigation and State Estimation Control System

Each Control Computer (CC) is responsible for maintaining state and executing navigation behaviors for a single aerial robot. The CC uses information from the robot's sensors along with processing results from the GPU Processing Computer (GPC) to make decisions. Additionally, the GPC provides higher-level setpoint goals for coordinating swarm behavior. A graphical overview of higher-level states, their inputs, and their transitions is shown in Figure 2.

There are four primary states a robot can be in: search, navigate, evade, and manual. In the search state, each robot maintains operating height of approximately 1.5 meters while generally following a search path specified by the GPC. The GPC attempts to provide a set of paths to the robots that will allow location of all four bins and partial QR displays. Obstacles are avoided in this state, and collision threats trigger a transition into the evade state.

In the navigate state, the robots attempt to reach a specified target or setpoint position. This can be provided by the human or by the GPC after all four QR displays have been located from a search. Static obstacles are still avoided in this state, and collision threats will cause the robot to transition into the evade state. If this state is triggered with the human operator as the target, the robot will follow the human operator at an attempted distance of three feet.

In the evade state, the robot will visually alert the human that evasive action is being taken using the Pi-Boy, and the robot will attempt to fly upwards to a safer altitude of 3m for a period of time until a safe decent location without any obstacles is located. The robot will then approach the safe decent location and return to standard operating altitude. Finally, the robot will exit this state, returning to the prior state from which evade was triggered.

In the manual state, the human operator verbally specifies commands for the robot to execute. During execution of these commands, no evasive or higher-level autonomous behavior will occur. The robot will enter the emergency state when a specifically formatted vocal command is issued, and the robot can only exit this state by another vocal command. The manual state can be triggered while the robot is in any prior state. When the manual state is exited, it will resume the last non-evade state that the craft was in.

Attitude/Position Control System

The DJI/Ryze Tello EDU is used in this work because of the reliability and the stability of the integrated control system. The control software of the robot estimates and tracks its full position from multiple data sources including an infrared light altimeter, multiple cameras, and an integrated IMU with barometer. In testing, the Tello was able to track position with high reliability while moving multiple meters over a several minute period. Additionally, the control system integrated into the Tello can maintain an accurate, position-holding hover for the duration of the robot's battery life even when subjected to external disturbances from nearby obstacles and light collision.

Overall Control Architecture

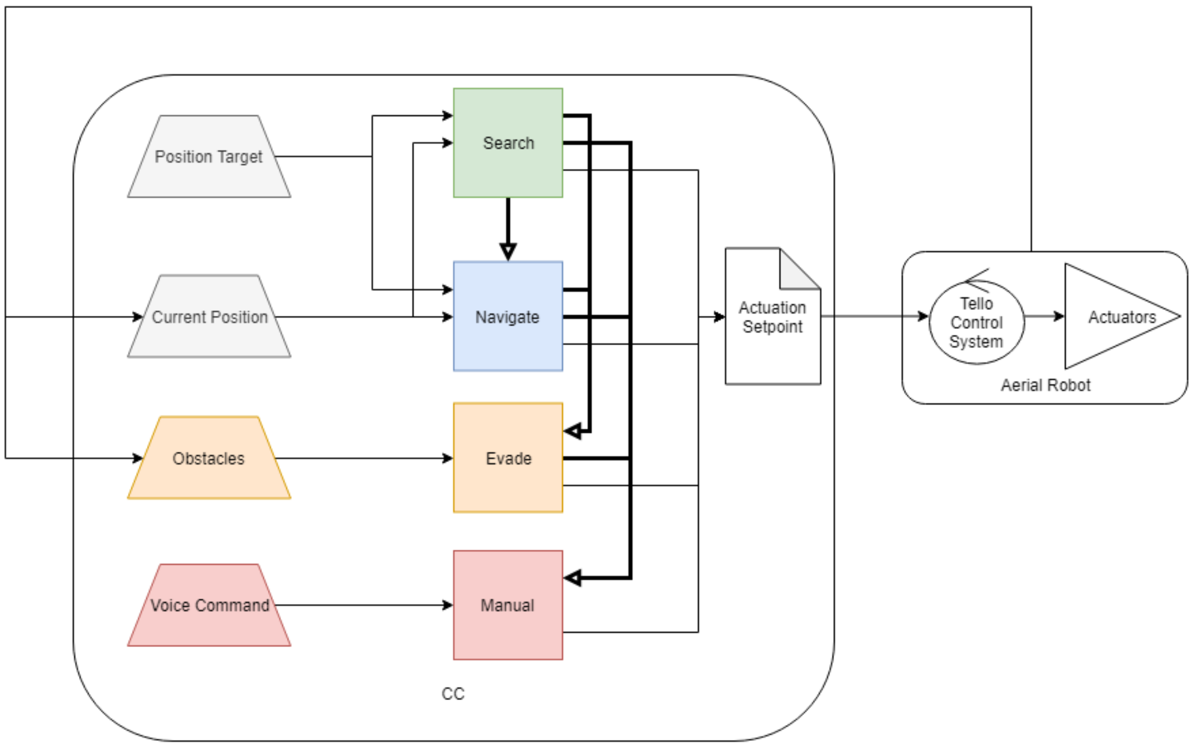

Figure 2. High-Level Controller Architecture

Flight Termination System

The flight termination system is a set of hardware buttons connected to the Flight Termination Computer (FTC) which is in turn hardwired to the central DHCP server. When a button is pressed, the FTC registers this event and sends a shutdown message to the associated Control Computer(s), which sends the emergency shutdown command to its Tello via the Wi-Fi connection. After the emergency shutdown command is received by the Tello, its propulsors lose power and the system falls from the sky.

MISSION PACKAGE

Perception System

Target and threat identification are performed by multiple object-detection neural networks. The architectures used are different-sized variants of DarkNet-YOLO [7] [8]. The Control Computers (CCs) have more limited computational capacity, so they are constrained to run predictions using a much smaller and less accurate model. The GPU Processing Computer (GPC) uses a larger and more accurate network to verify information provided by each CC's model. The GPC's detection model is slower and must work with network-introduced latency, so it is constrained to providing supplementary information for the CC to consider when making real-time control decisions.

During system operation, the CC makes all real-time control decisions. The CC runs a fast neural network on its robot's video stream to guide these decisions. This network is primarily concerned with locating immediate collision threats including enemy robots and the human operator. These threats are tracked over time, and threat information is shared between CCs. When a nearby threat is detected over multiple video frames and confirmed by other CCs or the GPC, evasive action is taken. This action involves flying upwards to the altitude of 3 meters, flying to an area without a located obstacle, then flying back down to the operating altitude of 1.5 meters. The human operator may issue manual vocal control commands if a collision seems likely, preventing the CC from performing any autonomous obstacle avoidance maneuver.

The major objective of the aerial system is locating and reading from the code displays and navigating around obstacles and bins. Because these targets and obstacles are static, the GPC's network is used to locate them in a camera frame. The CCs receive this information and use it for navigation. The human operator can also issue vocal instructions, guiding the helper robots to general positions.

When the QR Code displays are located and a display is visible to each of the robots, the GPC will send frames to a custom-trained neural network which segments parts of the QR Code from each display. Then, the GPC attempts to stitch together the image. The GPC also displays this stitching attempt on the human's Pi-Boy for the human operator to vocally correct before decoding is attempted. After the numeric code is extracted from the QR image, the human player will unlock the bins and the friendly robots take on a more assistive stance, tracking the human and obstacles using the GPC and CC detection networks, and respond to vocal requests to heal.

Communications System(s)

Each Tello acts as its own Wi-Fi access point which is joined by a Control Computer. The Control Computer can send and receive commands and sensor data over this connection. The Robot Operating System (ROS) message communication library is used to transmit data between Control Computers (CCs) and the GPU Processing Computer (GPC) over the wired DHCP Network. The Human Operator's Pi-Boy is wirelessly connected to this DHCP server and transmits vocal audio data to the GPC using ROS. The Flight Termination Computer is hardwired to this server.

User Interface / Man-Machine Interface

The man-machine interface is composed of the following components:

- "The Pi-Boy:" A wrist-mounted raspberry pi with an attached display to relay video feed to the contestant.
- Headset: to send voice commands to the off-board computer
- GPU Processing Computer: to perform GPU intensive tasks that processes speech and sends commands to drones.

The main method of man-machine interface is voice. The user will be able to control the drones and change their behavior of the drones with abstract commands. The specific actions of each command are determined by the autonomous flight system of the drone. This will allow the user to request specific actions from the drone like “heal” and “search” (for QR codes).

The speech recognition is enabled using the pretrained DeepSpeech speech-to-text engine [9]. This allows for offline speech recognition with an error rate of 11% [9]. The other option was to use Google’s speech recognition API to implement this recognition. However, this was abandoned due to networking and reliability concerns.

Risk Reduction

There are a number of features that will reduce the risk associated with this project. When we went about designing the hardware, we built in bumper guards to reduce the damage done by the drone in the event of an accident. There is also, built in software, safeguards for this. The software will automatically steer the drone away from hitting any other objects. There is also a manual kill switch that can be activated in the event that a judge believes a drone poses a threat to others.

Further, risk has been reduced in the component-selection phase of this project. Off-the-shelf consumer-grade robots specifically designed for operation around humans in indoor and outdoor environments with a variety of physical conditions. In doing so, we ensure that reasonable safeguards are built into the

Vehicle Design

Consumer-grade aerial robots and computing and networking hardware were selected for use. Due to this, EMI/RFI was not deemed to be a significant problem, as Wi-Fi communication used for communication with the robots is relatively robust and high-gain radio receivers and transmitters are used. Similarly, the Tello is designed in such a way that it is rarely damaged during falls and collisions.

The Figure 3 (in millimeters) is the new ABS - 3D printed propeller guard that the Tello drone will use to meet the safety requirements as stated in the rules. The largest opening on the guard is 11 mm on the outside most end of the pattern. The pattern is an additional 6 mm wider in radium than it needs to be, which results in the actual smallest opening available to be struck by the propeller is roughly 9 mm. The team has considered this to meet the safety requirements as that width is smaller than the width of an average human’s index finger. The weight of these prop guards will add 20 grams to the drones, bringing the total drone weight up to 100 grams. This will adjust the aircraft flight time to roughly 10.5 minutes instead of 13 minutes.

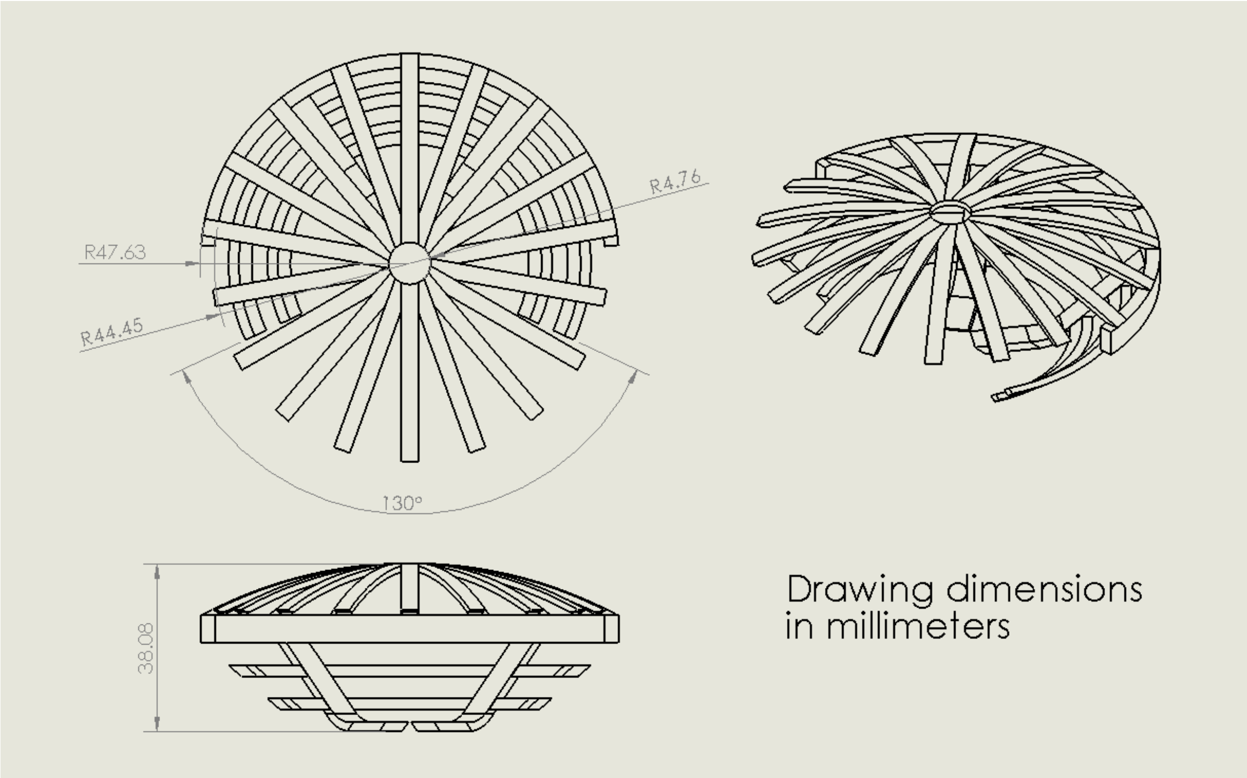

Figure 3. Custom Propulsor Guard

Safety

We selected and designed components of the physical and software system such that even if something does go wrong it is unlikely to cause physical harm. All of the blades are put behind bumper guards so that if the drone hits anyone it won't cut them with the propeller blades.

Modeling and Simulation

Figure 4 shows a cut plot of the flow velocity depicting the effect the propeller blade guards have on the free stream velocity simulated by a fan. The fan is providing an upward velocity of 2.5 m/s. It is noticeable that the largest impact zone that has detrimental effect on the air flow would be the outside if the motors. This would be correct as that area is the zone with the most protection.

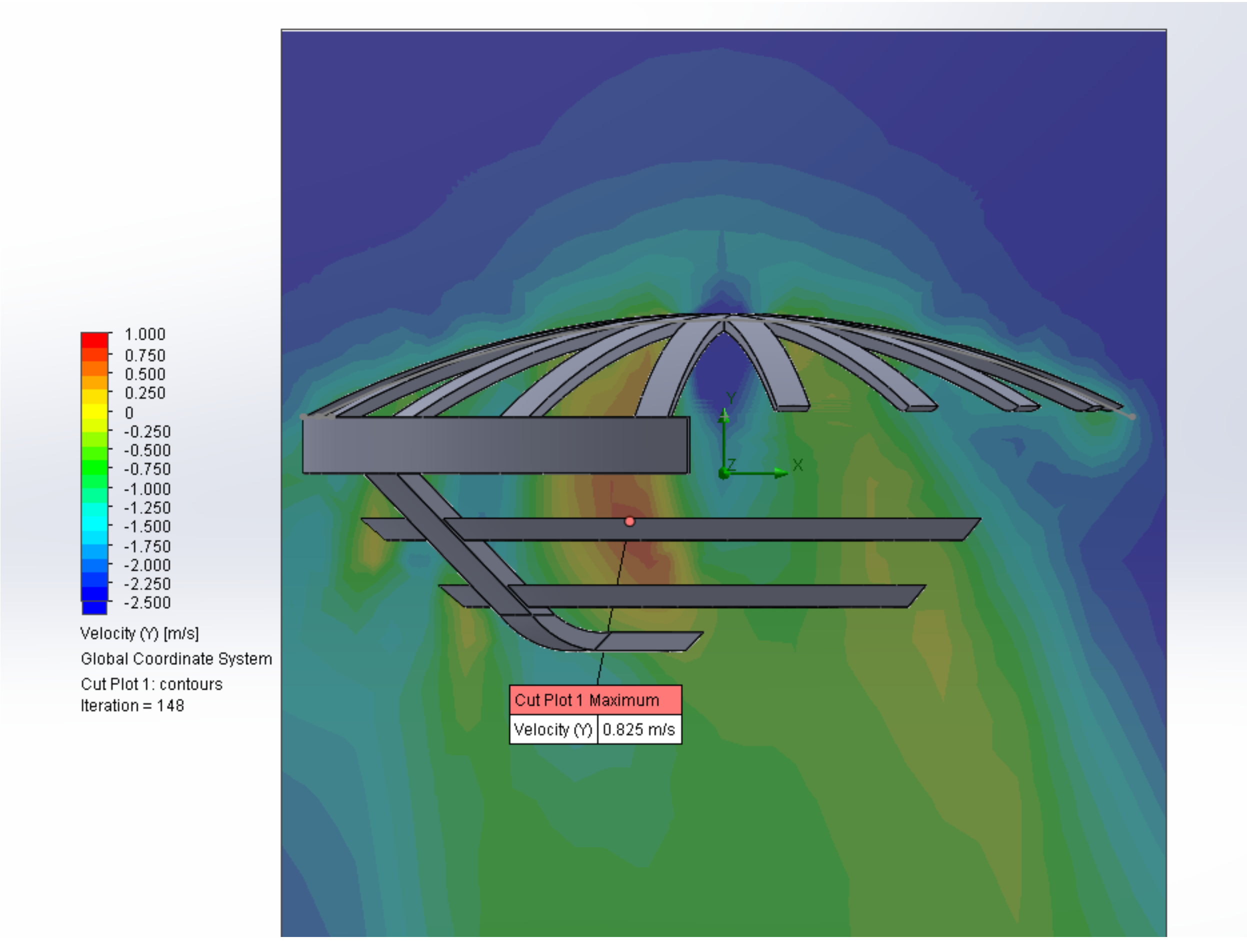

Figure 4. Propulsor Guard Flow Simulation

Physical Testing

The DJI/Ryze Tello EDU was tested for stability and drift in standard environments, and it was determined that this system had the necessary stability and accuracy to successfully execute IARC Mission 8. The networking library used to communicate between various components of the swarm has been tested for hundreds of hours without issue. The flight termination command for the Tellos has been tested extensively, and a robust method for ensuring the signal is received has been created.

The object detection networks require training on new data when the environment changes significantly, but a robust data-gathering and retraining process that can take place in less than 30 minutes has been developed and tested in multiple lighting conditions for multiple object types.

The DeepSpeech natural human-machine interface has been developed for several months now, and it has been tested for accuracy with multiple acoustic sensors and in multiple environments. A robust method for interpreting intended commands even in the presence of model inaccuracy and high word error rate has been developed.

CONCLUSION

The outline for a low-cost, modular, expandable, and safe guided-autonomous aerial robot swarm system capable of fulfilling IARC Mission 8 objectives is provided as part of this work. Objectives fulfilled in this system include natural human-machine interface, visual multi-object tracking, autonomous navigation and collision avoidance, and operation in a space occupied by an adversarial swarm.

The software architecture used is modular and can be directly integrated into aerial robot hardware as efficiency of computing hardware and algorithms improves. The software can easily be interfaced with custom hardware platforms with greater sensing capability in the future.

REFERENCES

[1] AUVSI Foundataion International Aerial Robotics Competition, "Official Rules for the International Aerial Robotics Competition," March 2019. [Online]. [Accessed 31 May 2019].

[2] DJI, "Tello," [Online]. Available: https://store.dji.com/product/tello. [Accessed 31 May 2019].

[3] DJI, "Tello EDU," [Online]. Available: https://store.dji.com/product/tello-edu. [Accessed 31 May 2019].

[4] Ryze Tech, "Tello SDK 1.3.0.0," [Online]. Available: https://www.ryzerobotics.com. [Accessed 31 May 2019].

[5] LattePanda, "LattePanda 4G/64G," [Online]. Available: https://www.lattepanda.com/products/3.html. [Accessed 31 May 2019].

[6] Intel Corporation, "Intel® Neural Compute Stick," [Online]. Available: https://software.intel.com/en-us/movidius-ncs. [Accessed 31 May 2019].

[7] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv CoRR,* vol. abs/1612.08242, 2016.

[8] allanzelener, "YAD2K," 14 February 2017. [Online]. Available: https://github.com/allanzelener/YAD2K. [Accessed 31 May 2019].

[9] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv,* vol. 2, 2014.