Autonomous Quadrotor for the 2017 International Aerial Robotics Competition

Nicholas Eckardt
B.S.E. Computer Engineering, 2019, University of Michigan

Sasawat Prankprakma
B.S.E. Computer Science Engineering, 2018, University of Michigan

Nigel Swenson
B.S.E. Mechanical Engineering, 2018, University of Michigan

Cheng Jiang
B.S.E. Computer Science, 2019, University of Michigan

Steven Schulte
B.S.E. Computer Engineering, 2019, University of Michigan

Sajan Patel
M.S. Robotics, 2018, University of Michigan

ABSTRACT

Unmanned Aerial Vehicles (UAV) are becoming more popular for both professional and casual uses, but are restricted to open areas and require GPS for navigation. Vehicles capable of flying in environments without relying on GPS will pave the way toward redefining currently outdated and expensive methods of structural inspection, search and rescue, and law enforcement operations that often take place in areas with limited GPS availability. Michigan Autonomous Aerial Vehicles (MAAV) designs and builds lightweight quadrotor UAVs capable of stable, autonomous flight without GPS. MAAV's vehicle will compete in the 2017 International Aerial Robotics Competition (IARC) where it will demonstrate its ability to autonomously manage a herd of ground vehicles in an open environment. Using a combination of control, computer vision, and path planning algorithms, it will herd ground robots over the goal line in the required time.

1. INTRODUCTION

The 2017 International Aerial Robotics Competition will be held in Atlanta, Georgia. We have designed a solution to the IARC challenge. This document presents the MAAV system designed and fabricated for the IARC.

1.1 Problem Statement

To further advance unmanned aerial vehicle technology, the International Aerial Robotics Competition has put forth a mission that involves open-area navigation and inter-robot coordination. Competing teams must present a vehicle that can navigate indoors without the use of external localization devices, using only visual cues to navigate. The vehicle must be unmanned and operate autonomously. The arena itself is a 20m x 20m grid, with lines every 1m, and is populated by autonomous ground vehicles traversing semi-random paths. The UAV must be able to herd these vehicles across the one designated goal line without letting more than 3 of the ground vehicles escape across the 3 designated out-of-bounds lines. The presented vehicle must also be able to avoid dynamic obstacles.

1.2 MAAV Goals

MAAV has designed, built and tested a quadrotor UAV to complete the mission. The vehicle will be able to determine its location from the ground lines, locate and track ground robots and take off autonomously.

1.3 Yearly Milestones

MAAV is entering its eighth year as a competitor in the IARC. This year we made major changes to the structure and algorithms of the vehicle, as well as our simulations. We completed a full redesign of the chassis, which allows our vehicle to better land on and interact with the ground robots. Our localization algorithm has been improved to use an Iterative Kalman Filter to better predict our state. Our vision algorithm has been adapted to use line detection rather than corner detection, which is more robust and provides more information. We have also developed a vision simulation which generates images of the competition, and compares the accuracy of our algorithm to the ideal case.

2. DESIGN DESCRIPTION

MAAV designed a fully custom quadrotor to best fit the challenges posed at the IARC competition. It features two custom designed circuit boards, both of which are assembled and built by the team. The quadrotor utilizes five cameras, a 4m laser rangefinder, and an IMU to observe the arena. The algorithms use information from these sensors to predict the vehicle's position and relative location of the ground robots and obstacles, creating a map of the arena. Next, a strategy algorithm chooses which ground robot to interact with, after which a path planning software navigates the vehicles around the obstacles to the desired ground robot. A set of waypoints, along with IMU data are passed to TI DK-TM4C TIVA, which sends signals to the motors after passing through the kill-switch. *Figure 1* shows a diagram of MAAV system architecture.

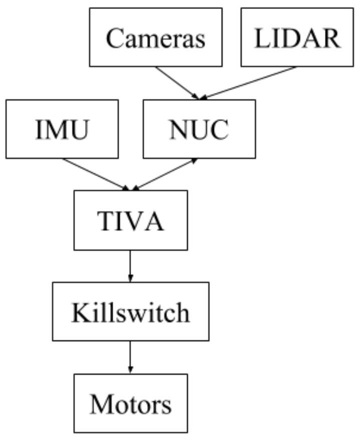

Figure 1: MAAV System Architecture

2.1 Physical System

The entire quadrotor design was conceived using CATIA V5. The model was designed and assembled to ensure proper placement of all components which allowed the team to predict the physical properties (i.e. moment of inertia, center of gravity) of the vehicle for use in simulations. CATIA was also used to generate the tool paths for machining custom parts. All parts including the carbon fiber airframe, carbon fiber plate center body, PCBs, sensor mounts, and motor mounts were custom designed and fabricated for this vehicle. An image of a prototype CAD model is shown in *Figure 2*.

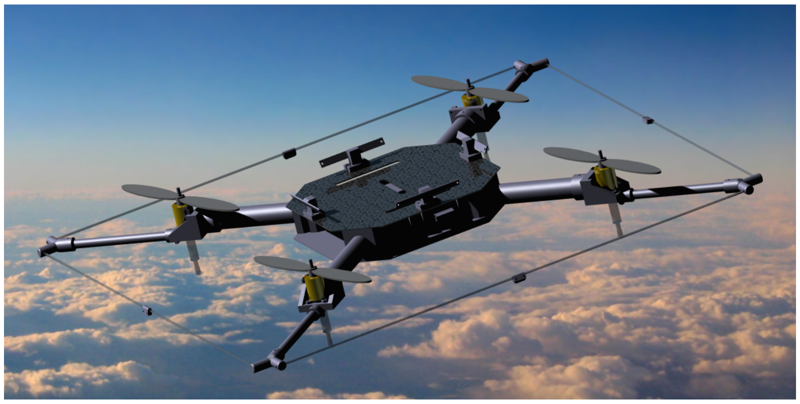
Figure 2: A prototype model in CATIA V5.

The MAAV quadrotor weighs approximately 3.5 kg, spans 105 cm diagonally from blade tip to blade tip, has a height of 20 cm, and has a vertical thrust of ~73N.

2.1.1 Structural Design

The overall structure of the vehicle is designed to interact with the ground robots and be robust enough to survive falls and crashes. The vehicle has four motors on the end of carbon fiber arms that connect to a wide center body which houses the batteries, sensors, and boards. The motors sit atop springs and have a wide fiberglass ring around the outside to make colliding with ground robots easy. In addition, the vehicle has four wide points of contact with the ground, allowing it to land on top of a ground robot without unbalancing it.

To prevent damage, the vehicle is supported by gas spring dampers to reduce the shock experienced upon impact and these are supported by material on the inside edge to prevent damage in both a straight down and a sideways fall. Furthermore, the sensors and sensitive components are located above the center body, ensuring that the first part to make contact with the ground after the legs is the carbon fiber center, instead of a more fragile part. The fiberglass ring also prevents ground robots or obstacles from harming the sensors and propellers housed closer towards the center of the vehicle. The batteries are contained in the center of the design to house and protect them, as well as to contain them in case of failure.

2.1.2 Propulsion and Lift Systems

The quadrotor is lifted by four 40 cm, two-blade propellers mounted on T-Motor MN4012 motors. These produce approximately 73 N of lift for a lift-to-weight ratio of 2.1, which is necessary for agile flight. The quadrotor is equipped with two 3700mA-hr lithium polymer (LiPo) battery which power the motors and allow for a flight time of roughly 12 minutes under competition conditions.

LiPo batteries maintain a constant voltage for most of their charge and thus it is important to have a method for monitoring battery charge. MAAV monitors battery status on a custom circuit board to maintain safe flight conditions.

MAAV chose propellers by testing on a thrust stand with the chosen motor and choosing the one with the highest maximum thrust that could produce 8.6 N of thrust, or enough to allow the drone to hover, for at least 12 minutes from a full set of batteries. These tests also provide relationships between thrust and battery power to help the vehicle fly better when batteries are low on power.

2.1.3 Circuit Design

The vehicle's electrical components are powered via a set of custom boards that are designed in house with Autodesk Eagle. There are two main boards, the power distribution board and the signal board. The power distribution board takes power from the batteries at 14.8 V and sends it to the motors. It also steps the voltage down to allow it to be used by the sensors and the Intel NUC. The signal board holds the TIVA and takes input from the sensors and the kill switch to send them to the NUC and the TIVA. It also sends the signals to another microcontroller, which are sent to the electronic speed controls to determine how much power is sent to the motors. It also records the data from flights on a micro SD card to be used later for testing code and troubleshooting the results of flights.

2.1.4 Sensor Suite

The vehicle uses a plethora of sensors, as shown in *Figure 2*, to understand the environment and give feedback to the algorithms to determine its own location and the location of the ground robots.

Microstrain 3DM-GX3-25 AHRS: The Microstrain attitude and heading reference system (AHRS) returns the roll, pitch, and yaw angles as well as the roll, pitch, and yaw angular rates in the form of radians and radians per second. These values are already filtered and are used by the localization filter as odometry input.

Figure 2: From left to right, The Microstrain 3DM-GX3-25, Logitech C920 Webcam, Hokuyo URG-04LG-UG01 Laser Rangefinder, and Pulsed Light Lidar-Lite Laser Module

Hokuyo URG-04LG-UG01 Laser Rangefinder: A horizontally mounted laser rangefinder returns a point cloud of 540 points over a 270 degree sweep. The sensor has a 4 meter range surrounding the vehicle and operates at a rate of 10 scans per second. This laser has been mounted horizontally to provide feedback for obstacle avoidance algorithms.

Logitech C920 HD Webcam: Five cameras provide visual feedback of the arena. Four cameras are mounted at 30-degree angles on the sides of the vehicle, and an additional one camera is mounted facing directly downwards. The images are used to detect the arena grid for localization, and to detect ground robots and their location with respect to the arena.

Pulsed Light Lidar-Lite Laser Module: One Lidar-Lite laser rangefinder will be mounted on the vehicle pointing directly downward to provide altitude feedback.

2.1.5 Shock/Vibration Isolation
MAAV tested the sensors under flight conditions and found that the cameras and IMU were sensitive to vibrations while the motors were running at high power settings. To counter this, the cameras have been mounted with fittings that incorporate vibration dampening material. The IMU is mounted to the vehicle on a bed of sorbothane to isolate it.

2.1.6 EMI/RFI Solutions
Circuitry is prone to electromagnetic and radio frequency interference. Fortunately, our data and video streams are transmitted over UDP where the communication protocol checks to make sure all data is successfully sent. In the case of interference, checksums and other error checking procedures invalidate the flawed message.

Electromagnetic interference can also be problematic for an IMU. Magnetometers inside the IMU measure the magnetic field of the earth to determine the IMU's orientation. However, the magnetic field becomes too corrupted by the EMI from the motors for this data to be useful. We eliminated this issue by combining integrated gyroscope data with the output of scan-matching from the laser rangefinder. Both the gyro and the laser devices are unaffected by EMI.

2.1.7 Communication
The communications system consists of a 5GHz WiFi channel. All WiFi communications are through a wireless protocol known as Zero Communications and Marshalling (ZCM). ZCM allows for low-latency multi-process communication.

2.2 Software System
The navigation software is optimized for the known IARC arena and a sparse environment. This section describes the architecture of MAAV's control architecture and navigation system.

2.2.1 Control
The quadrotor maintains stable flight by using a two-loop controller architecture that alternates power to each motor. The inner control loop is implemented by a DJI Naza-M Lite quadcopter controller (which will herein be referred to as the DJI) and directly controls each motor to change the quadrotor's roll, pitch, yaw rate, and overall body-frame thrust. The outer control loop uses the vehicle's system dynamics to control its x, y, yaw, and height through a PID control algorithm and directly feeds the DJI its desired inputs of pitch, roll, yaw rate, and thrust. The outer loop is implemented on the NUC, which communicates to a TI TM4C TIVA microprocessor. The TIVA also communicate with the vehicle's various onboard sensors. The vehicle's roll, pitch, and yaw are monitored through a Microstrain inertial measurement unit (IMU), while height is tracked by using a laser rangefinder. Four cameras allow the vehicle to build a map of the surrounding environment and track ground robots. Once the vehicle is stable, it can traverse waypoints in 3D space determined by the navigation algorithm running on an Intel i7 processor.

2.2.2 Stability Augmentation System

As an inherently unstable and under-actuated system, a quadrotor requires a well-tuned, robust controller to stay aloft. MAAV uses a cascaded PID controller with nonlinear terms that are derived from vehicle dynamics. For each degree of freedom in the inertial or global state of the quadrotor (*x, y, z*), the controller has a PID loop that converts from value error to desired rate. A second PID algorithm then converts from rate error to force, and inner-loop controller inputs for roll, pitch, and thrust. This architecture allows for incremental tuning thus expediting the testing process. For controlling the yaw of the vehicle, only a value error to rate error PID is used as the inner-loop DJI implements yaw rate error to yaw moment control. The controller maintains stability of the quadrotor in a large range of states while rejecting external disturbances.

2.2.3 Control Architecture

The general control architecture of the system involves a multi-processor and multi-step feedback control loop. The navigation software and outer loop controller outputs net forces on the vehicle that are used to calculate the roll, pitch, yaw rate, and thrust set points. These set points are sent to the microcontroller, which relays the set points to the DJI inner-loop controller. The inner-loop attitude controller calculates the final force and torque resultants, balances them across the vehicle's four motors, and ultimately controls each motor. The flight controls microcontroller then gathers all sensor data from the IMU and laser rangefinder, and sends it back to the NUC which filters the measurements using an Extended Kalman Filter. The filtered state feedback is sent to the strategy algorithm to decide what action to take from there.

2.2.4 On-board Planning

The on-board planner can take off, populate its map of the arena, generate and follow waypoints while avoiding obstacles, and safely land. The primary task of the on-board planner is to continuously populate the map, select a target ground robot, and generate waypoints. The target ground robot is chosen by a combination of vehicle location, obstacle location, and ground robot location, as well as probabilistic estimates of future obstacle and ground robot positions.

Three strategies were proposed for benchmarking and testing: a random strategy is a base-line approach to the problem, a greedy strategy that priorities the ground robots closest to cross the green line, and a defensive strategy which interact with the ground robots closest to cross the red line first.

2.2.5 Obstacle avoidance

For safety and reliability, the planner has been programmed to avoid all obstacles at any cost. The straight-line path planning was augmented with a simple height-based obstacle avoidance algorithm, using a dynamic point cloud generated with a laser range finder. MAAV has chosen to employ a cost-map approach to avoid obstacles. The laser provides information on where any obstacles may be within a two-meter bubble around the vehicle. Paths that require the vehicle to go near any obstacles will have a higher cost than paths that do not, and paths that intersect with any obstacles will have an infinite cost. The avoidance algorithm is highly efficient, with a linear complexity of $O(n)$, where n is the number of obstacles in the path.

2.2.6 Landmark Detection

The cameras feeds provide information of the location of the landmarks, including lines and ground robots. Line detection consists of Canny edge detector [1], Hough line transform [2], and K-means clustering [3]. Canny edge detector uses the magnitude and orientation of the image gradient, along with non-maximum suppression to find a single pixel width edge on the image. Hough line transform is a voting scheme used to find the lines in polar coordinates. The K-means clustering combines repeated detections. Ground robot detection uses a customized blob detector based on color and shape. The blob detector filters the image based on hue, and produces a binary image. Erosion and Mean blur are applied to reduce noise. A separate binary image for white pixels is also created. The algorithm matches the centroids of the colored blobs to the white blobs. The combined blobs that does not contain satisfactory dimension or ratio of colored pixels to the blob dimensions are then rejected.

2.2.7 Localization

In order for the vehicle to navigate, it must have some internal concept of where it is in space. MAAV's solution to the localization problem is based on a multi-sensor fusion approach [4] by Lynen, et. al (2013).

The core engine of MAAV's localization solution is an Extended Kalman Filter (EKF). The EKF can produce a best estimate of the pose of the vehicle given sensor inputs. Specifically, we use updates from three sources: an IMU, a single-shot Lidar for height measurements, and position and yaw data taken from data collected from cameras. These measurements are mostly independent, allowing for updates to be separate, except for the IMU. The EKF requires storing the pose and its associated covariances as a vector and a matrix, respectively. The pose information is represented in equation 1:

$$\mathbf{x} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \phi \quad \theta \quad \psi]^T, \tag{1}$$

where x, y, z are positions of the vehicle, $\dot{x}, \dot{y}, \dot{z}$ are velocities, and $\phi, \theta,$ and ψ are roll, pitch and yaw, respectively. Similarly, the output from IMU can be kept track using a vector as described in equation 2:

$$\mathbf{u} = [\ddot{x}_B \quad \ddot{y}_B \quad \ddot{z}_B \quad \omega_x \quad \omega_y \quad \omega_z]^T, \tag{2}$$

where $\ddot{x}_B, \ddot{y}_B, \ddot{z}_B$ are the accelerations in x, y, z directions of the body frame, respectively, and $\omega_x, \omega_y, \omega_z$ are angular velocities. Lidar output can be represented as

$$\mathbf{z}_{\text{Lidar}} = [z \quad \dot{z}]^T. \tag{3}$$

The following equations are used to calculate predictions and corrections for the state:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_{\mathrm{t}}, \mathbf{u}_t), \tag{4}$$

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T, \tag{5}$$

where $f(\mathbf{x_t}, \mathbf{u}_t)$ is a nonlinear dynamics-based prediction function, and $\mathbf{A} = \frac{\partial f}{\partial \mathbf{x}}$, $\mathbf{G} = \frac{\partial f}{\partial \mathbf{u}}$, and $\mathbf{Q}$ is the process noise matrix.

Most updates to the state are reasonably standard, except for the camera update. It is not a simple update – for each landmark (i.e. line intersection), corrections to the current vehicle state are performed, as described in [5]. The algorithm implemented is shown below:

```
Algorithm 1.
    $\mathbf{z}_c^{[i]} = \begin{bmatrix} x \\ y \end{bmatrix}, \boldsymbol{R}_c =$ camera noise
    $\mathbf{m}^{[i]} = \begin{bmatrix} x \\ y \end{bmatrix} =$ actual coordinates from map
    $\mathbf{p}^{[i]} = \begin{bmatrix} x \\ y \end{bmatrix} =$ pixel coords
    $\mathbf{H}_c =$ state mapping
    for $i = 1, \dots,$ number of deleted landmarks
        $\mathbf{z}_c^{[i]} = \text{cameraProjection}(\mathbf{p}^{[i]}, \mathbf{x}_t)$
        $\mathbf{K} = \mathbf{P}_t \mathbf{H}_c^T (\mathbf{H}_c \mathbf{P}_t \mathbf{H}_c^T + \mathbf{R}_c)^{-1}$
        $\mathbf{x}_t = \mathbf{x}_t + \mathbf{K}(\mathbf{z}_c^{[i]} - \mathbf{m}^{[i]})$
        $\mathbf{P}_t = (\mathbf{I} - \mathbf{K}\mathbf{H}_c)\mathbf{P}_t$
    end for
```

However, the model that has been described so far does not account for an added complexity in the design: due to sensor latency and required computation time for processing images, the model needs to be able to process delayed asynchronous sensor updates. This is done through a variation of the state buffering technique described by Lynen, et. al (2013). The vehicle stores a buffer of states, covariances, and sensor readings. Every time a sensor with any latency updates, the vehicle rolls back the state to the most recent measurement that took place before the time of the sensor update, and a new entry is added to the buffer. The vehicle then re-applies all calculations for each state, up to the present time. As a result, when a delayed sensor update arrives, the calculations are re-performed as if they have been known the whole time. This technique allows the model to account for delayed sensor update problem at no cost to accuracy.

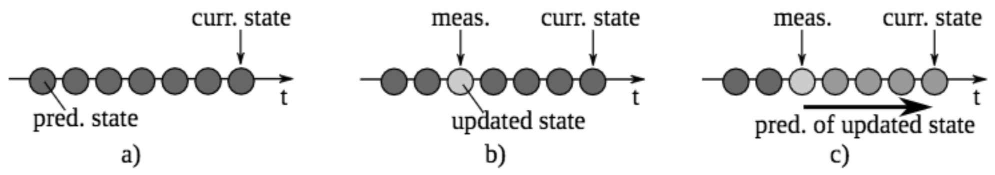

Figure 3: a) A current state based on IMU input used for vehicle control. b) Delayed measurement arrives, it is applied to the corresponding state in the buffer. c) From the updated state, IMU prediction is applied to update the current state.

Using the modified state-buffering technique, an up-to-date record of the best estimate for the vehicle pose can be kept track, which can then be given to other vehicle systems, such as assisting vision algorithms to detect ground robots, path planning, and vehicle control.

3. METHODS

To ensure that code and structure function properly, MAAV performs multiple different tests and simulations. Since the flying of an autonomous quadrotor is potentially dangerous, there are many safety features that are implemented.

3.1 Safety

MAAV performs flight tests in a hangar in Willow Run Airport to ensure that no one who isn't trained is at risk.

3.1.1 General Flight Safety

While flying, all nearby members are equipped with safety glasses and stay 2 meters away from the vehicle at all times. In addition, the vehicle is always tethered with two ropes to prevent it from harming anyone or from crashing.

3.1.2 Flight Termination System

A backup kill switch is implemented as a last resort. In the event of a complete computer meltdown that causes the quadrotor to enter an unresponsive and dangerous state, a human-operated kill switch disables all power to the motors. The source of the kill switch signal originates from a common RC controller supplied, and operated by IARC judges. This standardization guarantees that the kill switch operates on a reliable frequency, separate from the communication frequencies used by the vehicle for data and video transmission. The signal from the kill switch receiver is a PWM signal that is processed by a microcontroller independent of the main system. We chose to use a microcontroller instead of the suggested design to give added flexibility to how the vehicle responds to the receiver's signal. The added complexity is justified because it allows us to add important features like noise immunity, and fail safe functionality without sacrificing response time.

3.2 Testing

Testing is broken into two stages: calibration and free flight testing.

The vehicle is initially tested on a steel test stand that isolates a single axis for tuning controller gains while keeping the vehicle restrained. After tuning the control loops on the test stand, the vehicle is tested with safety ropes and finally in free flight. In all cases the vehicle is subject to two separate kill switches: one in the normal flight software and one external, dedicated kill switch that operates on a separate frequency to circumvent the dangers of a loss of WiFi connection.

Battery voltage is checked to be at operating level and the propellers are securely tightened to the motors. The vehicle is powered on and automatically connects to the configured WiFi network and communications are initialized. The enable signal is sent and the vehicle is ready for flight.

3.2.1 Calibration

Calibration is required for each motor/speed-controller/propeller triad. Motor/speed-controller/propeller calibration curves mapping RPM to force are calculated using the motor test cell shown in *Figure 4*. The test cell is equipped with an air bearing, force and torque transducers, and a data acquisition system (DAQ). The test cell automatically collects relevant data for each motor/speed-controller/propeller combination.

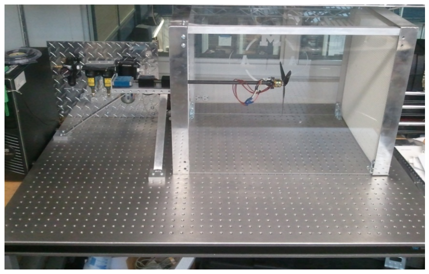

Figure 4: MAAV motor test cell

3.2.2 Free Flight Testing

After performing calibrations, the vehicle is tested in free flight. This is because the inner-loop DJI controller is a tested, off-the-shelf module that requires no tuning. However, the outer-loop controller still needs to be tested and its PID gains need to be tuned. For safety purposes, ropes are attached to the vehicle. Initially, the height control is removed from the system and the height setting is manually controlled from a joystick. The vehicle is raised roughly 30 cm off the ground to verify DJI functionality and tune x, y, and yaw stability in the outer-loop. Once stability is achieved at 30 cm off the ground, the vehicle is slowly raised to an operating altitude of 1.5 m. Slight adjustments are made to account for leaving the ground effect zone. Once this stability is achieved the PID gains for height control are tuned until they are stable. x, y, yaw, and height set points are then sent manually to the vehicle from the ground station. The set points are altered by moving the joystick. Movement in each direction is tested before autonomous movement is attempted. Once the outer control loops are stable, preprogrammed autonomous movement is tested. After verifying proper vehicle response, the onboard sensors are used to locate and map the surrounding environment. Next, the simulation GUI described below is used to examine the actions the quadrotor would take if set in autonomous mode. Finally, the exploration functions are enabled and the vehicle is ready to fly the mission.

3.3 Modeling and Simulation

To properly test the entire algorithm pipeline without conducting costly real-world flight tests, MAAV has developed a simulator capable of performing system-wide tests on the entire software stack. The simulator includes a raytracing render engine in Blender to generate photorealistic images with which to refine the computer vision algorithms. A separate dynamics simulation uses the open source Bullet Physics software library to model ground robots and obstacles as cylinders for simplicity. In contrast, the vehicle is composed of multiple rigid bodies allowing the physics engine to simulate forces on each rotor. This allows the simulated quadrotor to fly as it would in the real world. The inter-process communications system used in all MAAV software allows for easy replacement of the real-world interface with its simulated counterpart. These simulations allow for easy testing of multiple algorithms without the logistical overhead of real world flight tests. In the future, the University of Michigan's computing cluster, Flux, will be used to simulate hundreds of flight tests to tune algorithms used on the quadcopter.

4. CONCLUSION

MAAV has designed and constructed a small quadrotor UAV weighing around 2 kg that is capable of autonomous interaction with autonomous ground robots in order to control them. The vehicle is currently in the manual and autonomous testing phases. We expect the quadrotor to navigate the competition arena and complete the mission objectives in the allotted time. MAAV would like to thank Northrop Grumman Corporation, our title sponsor, as well as all our sponsors for their generous contributions.

5. REFERENCES

[1] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679-698, 1986.

[2] P. V. C. Hough, A method and means for recognizing complex patterns, U.S. Patent 3,069,654.

[3] Hartigan, J. A., and Wong, M. A. (1978), "Algorithm AS 136: A k-Means Clustering Algorithm," *Applied Statistics*, 28, 100–108.

[4] Lynen S, Achtelik MW, Weiss S, Chli M, Siegwart R (2013) "A robust and modular multi-sensor fusion approach applied to MAV navigation." In: *2013 IEEE/RSJ international conference on intelligent robots and systems (IROS).* IEEE, pp. 3923–3929.

[5] Thrun S, Burgard W, Fox D (2005) "Probabilistic Robotics." Cambridge, MA: MIT Press.

NORTHROP GRUMMAN

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

LOCKHEED MARTIN

COLLEGE OF ENGINEERING
AEROSPACE ENGINEERING
UNIVERSITY OF MICHIGAN

COLLEGE OF ENGINEERING
COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

Digi-Key® CORPORATION

SEC SATURN ELECTRONICS CORPORATION

RobotShop.com Robotics at Your Service!™

NEWWAY® air bearings

saleae

3DRobotics UAV TECHNOLOGY

DELPHI