

Development of a 500 gram Vision-based Autonomous Quadrotor Vehicle Capable of Indoor Navigation

Stephen Haviland

shaviland3@gatech.edu

Graduate Research Assistant
Georgia Institute of Technology
Atlanta, GA, USA

Dmitry Bershadsky

dbershadsky@gatech.edu

Graduate Research Assistant
Georgia Institute of Technology
Atlanta, GA, USA

Daniel Magree

dmagree@gatech.edu

Graduate Research Assistant
Georgia Institute of Technology
Atlanta, GA, USA

Eric Johnson

eric.johnson@ae.gatech.edu

Associate Professor
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

This paper presents the work and related research done in preparation for the American Helicopter Society (AHS) Micro Aerial Vehicle (MAV) Student Challenge. The described MAV operates without human interaction in search of a ground target in an open indoor environment. The Georgia Tech Quadrotor-Mini (GTQ-Mini) weighs under 500 grams and was specifically sized to carry a high processing computer. The system platform also consists of a monocular camera, sonar, and an inertial measurement unit (IMU). All processing is done onboard the vehicle using a lightweight powerful computer. A vision navigation system generates vehicle state data and image feature estimates in a vision SLAM formation using a Bierman Thornton extended Kalman Filter (BTEKF). Simulation and flight tests have been performed to show and validate the systems performance.

INTRODUCTION

As UAV research progresses, much emphasis is being placed on smaller, lighter, agile, and more capable autonomous vehicles. These vehicles must be capable of operating in various environments, including cluttered and indoor, GPS-denied areas. This paper details the work done by the Unmanned Aerial Vehicle Research Facility (UAVRF) inspired partially by the 2014 AHS MAV Student Challenge in Montreal. The MAV Student Challenge tasked teams to locate a target in an open indoor environment. Vehicles were limited to 500 grams in weight and 450 mm in length in any dimension. The vehicle described here passes the competition's requirements; it is small, light, able to fly without external aide at speeds tested up to 8ft/s, and is computationally capable. It is a modular platform designed using a custom multirotor sizing tool, built, and flown within a span of a few weeks, demonstrating the ease of use and integration of its framework. This paper also

proposes an additional benchmark for use with UAVs which allows comparison of computational capability scaled by vehicle mass. This benchmark may be useful in certain situations in conjunction with others to assess the performance of UAV systems.

RELATED WORK

In the last decade much research has been done in place to further the autonomous capability of MAVs in indoor and outdoor settings. Outdoor exploration can sometimes rely on using GPS but in urban environments this does not always yield a usable result. Indoor exploration and navigation systems cannot use GPS but often have used external systems such as Vicon. Quadrotor vehicles have been even designed to be able to complete both indoor and outdoor missions (Ref. 1).

Smaller vehicles are often not able to carry LIDAR systems so systems may need to rely more heavily on lighter sensors such as vision-based ones. Vision-based navigation has been key for quadrotors to be able to fly in unknown indoor environments. Work was done that allowed vehicle to perform SLAM in an unknown environment without any external aides but needed off-board computing help for the SLAM problem (Ref. 2).

Quadrotor vehicles under 500 grams have also been able to complete vision-based navigation but did not have the weight and power budget to be able to complete all tasks onboard (Ref. 2), (Ref. 3).

When designing a MAV consideration has to take into account Size, Weight, and Power (SWAP). Typically if the vehicle is very lightweight it cannot carry enough computational power to do some missions that may more easily be possible with the greater payload allowances of larger vehicles. Research has been done in this subject (Ref. 4) for the design of small agile quadrotors. The main constraints when considering designing a vehicle are that battery and motors take up more than half of the mass distribution on average. Generally, these types of vehicles' low empty weight ratio limits the useful payload to something very small. This work also points

Presented at the AHS 71st Annual Forum, Virginia Beach, Virginia, May 5–7, 2015. Copyright © 2015 by the American Helicopter Society International, Inc. All rights reserved.

out that to achieve high figure of merit on such a vehicle, the proper pairing of motors and propellers becomes critical.

The Unmanned Aerial Vehicle Research Facility (UAVRF) has done much work in the past with vision-based navigation on vehicles ranging from Yamaha RMAX(200-lb helicopter), quadrotors (1.5kg and up) and power- and data-tethered ducted fan vehicles[(Ref. 5) , (Ref. 6) , (Ref. 7)]. Research done on the RMAX platform has shown that having a capable CPU allows for a more stable navigation solution from vision systems. The RMAX platform is capable of flying two large Intel i7 machines which besides the benefits in computation speed there is benefits in setup times as this allows for standard software installation that can plague ARM-based systems sometimes.

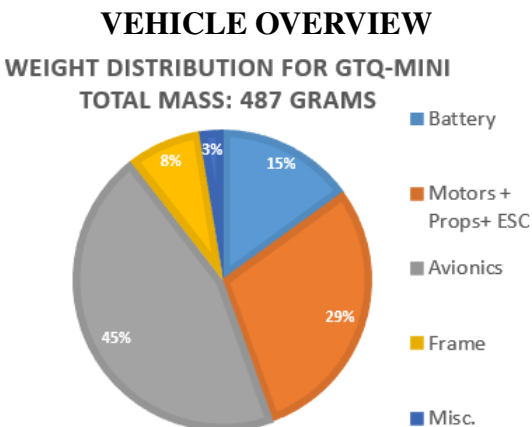


Fig. 1. Weight distribution for the GTQ-Mini quadrotor

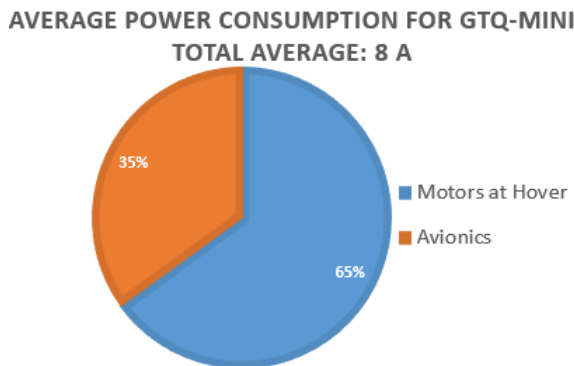


Fig. 2. Power distribution for the GTQ-Mini

The GTQ-mini is a sub-500 gram vehicle capable of indoor or outdoor GPS-denied navigation. The vehicle was designed for the AHS MAV Student challenge which requires vehicles be capable of vision navigation while restricted to 500 grams or less.

The UAVRF’s Electronic Multirotor Sizing Tool (EMST) optimizer is used to select the drive system for the vehicle. Information on the tool is not yet publicly available, although the tool’s validator is available¹. The tool was in an early

¹<http://controls.ae.gatech.edu/dbershad/EMSTAirTimeCalculator.html>

phase at the time of the design of the vehicle but it allowed us to size the motors, ESCs, battery, and propellers to achieve our desired operation time. One unexpected result is that the outputs pushed us to use a 4 cell battery on the motors, even though they are rated only for a 3 cell maximum. The vehicle performs better in all relevant performance categories on 4 cell, as does the main computer, as 16.8 V is closer to its normal 19 V rating. See Figure 1 for weight breakdown of the vehicle. As can be seen from the Figure, the weight savings from the frame and drive system allowed for a larger percentage of weight to be utilized for the electronics.

To keep frame weight down the vehicle utilizes lightweight carbon fiber rods for the frame which keep rigidity while saving on weight. Custom motor mounts were made using a 3D printer that were also optimized to reduce weight. For vision a downward facing Firefly-MV USB monocular camera² was used. In the past for quadrotor vehicles custom stability augmentation system (SAS) boards were developed with higher fidelity INS sensors. The open-source Ardupilot board³ was used. The open-source nature of the project allowed us to create custom firmware which allowed our Georgia Tech UAV Simulator Tool (GUST) to take in the sensor data and send motor commands back. This allowed for rapid development and integration of our system. Others groups have utilized the Ardupilot for ease in integration such as (Ref. 8). For altitude data we used a MB1040 LV MaxSonar EZ⁴ which provides distances up to 6.45 m away with a resolution of 25.4mm. Although the Ardupilot can take in the sonar data over analog, we found that taking the data in over the digital pin of the sonar provided much cleaner data. So a custom USB cable was made to allow the sonar to connect to the Gigabyte Brix. Table 1 provides data on our vehicle.

For the main avionics computer running our GUST software it was desired to have a very lightweight computer that was capable of running (Ref. 9) that performed similar to its performance on desktop machines or RMAX. From our testing the main bottleneck is in the vision processing. On our older ARM-based vehicles, the navigation module could only process feature points up to 7-8 Hz (Ref. 7). The Gigabyte Brix i7-4700⁵ was very close to what we needed. The Brix houses an 3.2/3.9 GHz quad-core processor and after modifications weighed 140 grams. Downsides to using a computer like this is losing input/output protocol like GPIO/I2c and UART that most ARM-based computers have. However, due to the amount of sensors used, we are able to use USB-TTL dongles to interface with serial devices. Another as expected affect of such a powerful computer is the increase in power consumption of the vehicle as seen in Figure 2.

The GTQ-Mini avionics are designed to not need any external aide from the vehicle’s ground station computer. All processing is performed onboard the vehicles computer which reduces the need for a strong wireless communications link.

²<http://www.ptgrey.com/>

³<http://3drobotics.com/>

⁴<http://www.maxbotix.com/>

⁵<http://www.gigabyte.us/>

Table 1. GTQ-Mini Specifications

Specs	Value
Size(motor-motor) (cm)	25
Mass(g)	487
Power Requirement (W)	120
Specific Power (W/kg)	247
Flight Time (min.)	8-10
Max Payload(g)	200
Power loading (kg/W)	0.004
Nominal Hover RPM	11460
Motors	Turnigy Multistar 1900 KV
Propeller: Dia x Pitch (in)	5x3
Processor	i7 3.90GHz(max)
Tip Mach Number (Propeller)	0.22
Sensors	IMU, Camera, & Sonar
Battery	1300 mAh/4S LiPo

Table 2. Clock loading benchmark

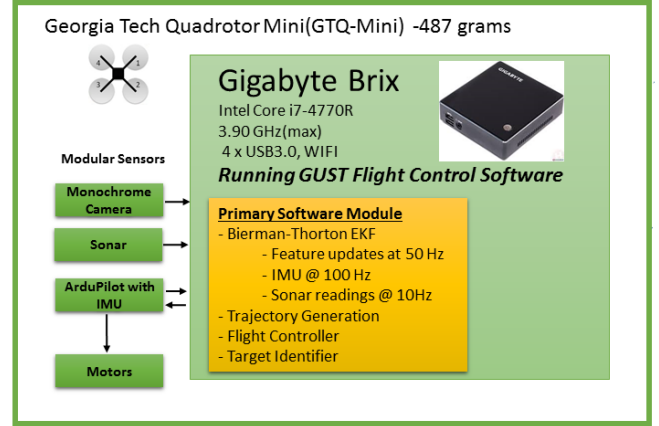
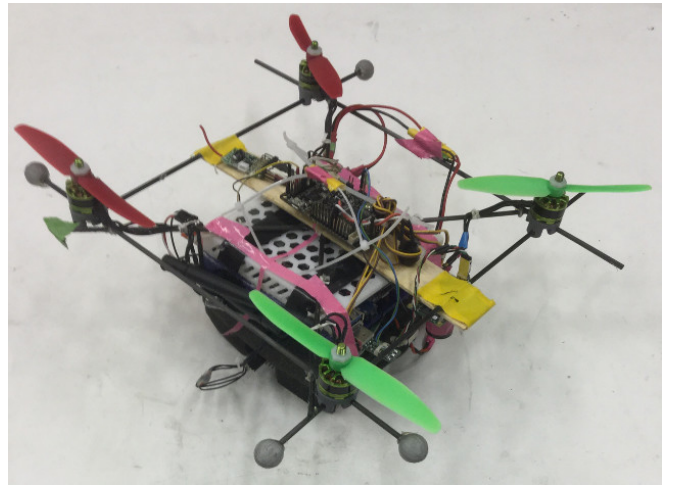
Vehicle	Clock (GHz)	Mass (kg)	Clock loading (GHz/kg)
GTQ Mini	3.9	0.487	8.00
Parrot 2.0	1	0.38	2.63
Firefly	3.1	1.6	1.94
Parrot Bebop	1.0	0.41	2.44
Pelican	3.1	1.65	1.88
Hummingbird	1.6	0.71	2.25
Parrot 1.0	0.486	0.38	1.28
GTQ3	1.86	1.498	1.24
Erle Hexacopter	1	0.878	1.14
GTQ2	0.72	1.5	0.48
GTLama	0.016	0.46	0.03

The onboard Gigabyte computer runs on Ubuntu 14.04 Operating System which allows for easy integration of the UAVRF GUST software. GUST is a software framework that the UAVRF has used since its inception that allows for developing and testing of different vehicles. GUST provides hardware-in-the-loop(HITL), software-in-the-loop(SITL) and ground station software.

The Gigabyte Brix flight computer communicates via USB interface with an Ardupilot autopilot. The Ardupilot is designed around an Atmega 2560 processor, which sends and receives data from the Gigabyte computer. The Ardupilot provides inertial measurement unit (IMU) sensor readings to the Brix for processing. The Ardupilot is capable of sending raw sensor readings at up to 100 Hz. Motor commands are sent from the GUST software to the Ardupilot. This allows for more computationally intense controls to run on Intel i7 which could not be performed on conventional autopilots. If manual control is needed the safety pilot simply can switch to manual which removes control from Gigabyte and uses the stabilized mode that the Ardupilot comes standard with. Figure 3 shows architecture of the GTQ-mini and what is running on i7.

Some studies (Ref. 10), (Ref. 11) have proposed different benchmarks for vehicles, but none appear to address processing power on vehicles in this class. We propose a benchmark

known as the clock loading, measured in maximum clock speed of the vehicle normalized by the GTOW, in units of GHz/kg. Table 2 shows the clock loading for several relevant vehicles in the class under 2 kg.

**Fig. 3. GTQ-Mini Architecture****Fig. 4. GTQ-Mini Vehicle**

VEHICLE AND SENSOR MODELS

The basis of the navigation system is a Bierman-Thornton extended Kalman filter (BTEKF), composed of 15 vehicle states and 16 3-dimensional feature states. The navigation system requires models of the vehicle dynamics and sensors. The following section describes the models used.

The minimal vehicle state vector is

$$\hat{x}_v = [\hat{p}^i \quad \hat{v}^i \quad \hat{R} \quad \hat{s}_b \quad \hat{\omega}_b]^T \quad (1)$$

where p , v , q , is the vehicle position, velocity and attitude quaternion, respectively, s_b and ω_b are the acceleration/gyro biases. The Superscript i denotes the inertial frame and estimated quantities are hatted.

The full state vector of the system is composed of the vehicle state and the feature states:

$$\hat{x} = \begin{bmatrix} \hat{p}^i & \hat{v}^i & \hat{R} & \hat{s}_b & \hat{\omega}_b & \hat{p}_{f_1} & \dots & \hat{p}_{f_{N_f}} \end{bmatrix}^T \quad (2)$$

where N_f is the number of feature states. The feature states are assumed to be static, and so the process model may be ignored, though a feature state process noise term Q_f may be applied. The covariance of the state vector is not explicitly tracked in the Bierman-Thornton EKF. The process noise of the filter is given by $Q = \text{diag}(0, Q_a, Q_\omega, 0, 0, Q_f, \dots, Q_f)$.

The vehicle model is based on the specific force and angular velocity input from an IMU. The non-linear dynamics of the vehicle are driven by raw IMU input, which is assumed to have a static or slowly evolving bias and corrupted by white Gaussian noise.

Sensor measurements from the IMU are corrupted by noise and bias as follows:

$$s_{raw} = a + s_b + L_{bi}g + \eta_a, \quad (3)$$

$$\omega_{raw} = \omega_t + \omega_b + \eta_\omega. \quad (4)$$

where a and ω_t are the true acceleration and angular velocity, g is the acceleration due to gravity. The noise is assumed to be white Gaussian and zero mean. The rotation matrix from body to inertial is denoted $L_{ib} = L_{bi}^T$. The vehicle state is propagated by integrating data from the IMU.

Before propagating the model the estimated bias is subtracted from the IMU data

$$s = s_{raw} - \hat{s}_b, \quad (5)$$

$$\omega = \omega_{raw} - \hat{\omega}_b. \quad (6)$$

The vehicle dynamics are given by the following:

$$\dot{\hat{p}}^i = v^i \quad (7)$$

$$\dot{\hat{v}}^i = L_{ib}s - g = L_{ib}(s_{raw} - \hat{s}_b) - g \quad (8)$$

$$\dot{\hat{q}}^i = \frac{1}{2}Q(\omega)\hat{q}^i = \frac{1}{2}Q(\omega_{raw} - \hat{\omega}_b)\hat{q}^i \quad (9)$$

$$\dot{\hat{s}}_b = 0 \quad (10)$$

$$\dot{\hat{\omega}}_b = 0 \quad (11)$$

where s and ω are the bias-corrected specific force and angular velocity. The function Q maps angular velocity to the quaternion derivative matrix coefficient.

Using the quaternion representation in the estimation algorithm causes the covariance matrix to become singular and requires careful accounting of the quaternion constraints. To avoid these difficulties, a minimal representation of the vehicle's attitude is used, which defines the vehicle's current attitude with respect to an arbitrary reference frame, in this case the attitude in the previous time step.

To keep the covariance matrix from becoming singular, a minimal representation of the vehicle's attitude is used. It is

defined by the current attitude with respect to the reference frame from the previous time step which is defined from the vehicle's attitude. It is assumed since the attitude changes over small time steps are small it can be defined to be an infinitesimal error quaternion.

$$\delta q = [1 \quad \hat{R}]^T \quad (12)$$

such that

$$\delta q = \hat{q}_{ref}^{-1} \otimes \hat{q}. \quad (13)$$

Additional details on this formulation can be found in (Ref. 12).

CONTROL

The complexity of the control system depends on dynamics of vehicle and the quantities being controlled. Air vehicles are susceptible to drastic oscillations and system divergent in flight if the control system is not properly tuned. Consideration of couple lateral and longitudinal motion along with aerodynamic interaction must be considered. GTQ-Mini uses a Model Reference Adaptive Control (MRAC) architecture (Ref. 13). A position control loop generates velocity commands, which is used in the velocity control loop to generate attitude commands. Attitude commands go into the most inner loop to generate servo commands which allow stabilized flight. This type of architecture has been shown in the past to control unmanned systems (Ref. 13).

NAVIGATION SYSTEM

For navigation a Bierman-Thornton Extended Kalman Filter (BTEKF) is used (Ref. 14) instead of the conventional EKF. When using an EKF one can run into diverging solutions with bad initial estimations of states and poorly modeled process model. The EKF also often under estimates the covariance. The benefits of the BTEKF are drastic improvements to numerical stability and similar computational costs to standard EKF. The BTEKF uses modified Cholesky factors U and D on the covariance matrix P .

$$P = UDU^T \quad (14)$$

where U is upper triangular with a unit diagonal, and D is diagonal. This requires that the update and propagation steps in the filter change to reflect U and D . The propagation of the state vector is carried out according to the nonlinear equations of motion given in section . The update gain K is found from the covariance update equations and is applied in the same way as the standard EKF formulation. For more information on update and propagation equations of the BTEKF see (Ref. 14).

Processing feature and vehicle states Visual SLAM can use the new update and propagate steps. The full covariance is encoded into the U and D factors from the Cholesky factorization. When flying feature points can come into and out of the field of view of the camera. This causes us to implement an initialization and removal step for feature points in the state vector. This step can be read about in (Ref. 9).

FLIGHT TEST RESULTS AND COMPARISONS

Flight tests were completed on several occasions to tune the vehicle and understand its capabilities. For all tests simulation work was done in GUST to have an idea on how the vehicle would perform. The hope was that the simulator performed on equal footing to confirm the simulation model.

For testing we had two locations at our disposal; the Indoor Flight Facility (IFF) at Georgia Tech and Peachtree room. The IFF has a series of vicon cameras that allow having reference data to compare the navigation results to. It also provides a better location for tuning the vehicle. The downside of the location is that the room is small and we cannot complete large tests. The Peachtree room is a banquet room with wood floors approximately 150' long by 50' wide. This gives a great location to test long trajectories and our mission manager for the competition. The downside being there is no vicon data.

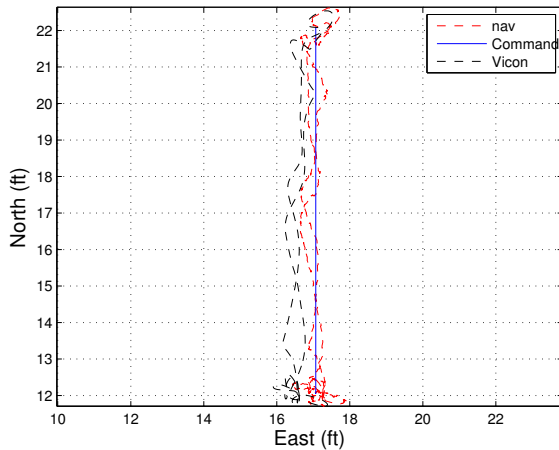


Fig. 5. Simulation of Quadrotor traversing two waypoints 10ft apart using Vision Slam. Vicon is only for reference

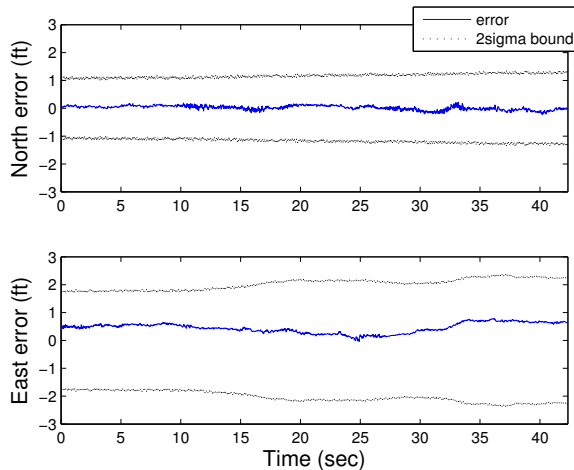


Fig. 6. Error in quadrotor traversing two waypoints 10ft apart using vision only for navigation. Drift can be seen in the error.



Fig. 7. GUST Quadrotor running Vision navigation simulation of track similar to vicon room. Two quadrotor vehicles can be seen. One is actual the other is where the navigation system thinks it is.

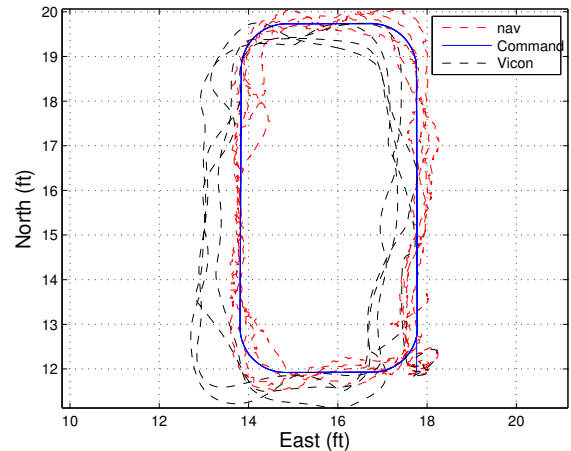


Fig. 8. Flight path of simulated track using Vision only with Vicon being a reference for comparison.

Figures 5 and 6 are for a simulated test in GUST of the Vision navigation. Two waypoints were placed ten feet apart and vehicle was to go there and back to see how much drift would accumulate. There is a slight east offset that can be seen in Figure 6 but the navigation tracks very well with respect to the vicon. A similar test was performed in the Peachtree room where the vehicle was commanded to go a waypoint 32.8 feet north (65.6 feet round trip). We marked with tape where we wanted the vehicle to end up and had the vehicle complete two roundtrips and measured the euclidean error from where it should have ended up. The average vehicle euclidean error was 1.96 feet of error per 65.6 feet round trip. A video of the flight test can be found at <http://youtu.be/GGqexQy-FgE>. The flights in the ballroom were done with additional battery load (1500 mAh from designed 1300 mAh) and foam for hard-landings which brought the vehicle up to 580 grams.

The next test was to see how the vehicle performed in a track. This test was done as it is an easy to perform in the vicon room with our limited space and it allows for fine tuning of the navigation system. Figure 7 shows the GUST ground station where the purple is the commanded waypoints and flight path, yellow being where the vehicle actually traversed. Two quadrotor vehicles can be seen as one is the actual and the other being where it thinks it is. Figure 9 show how the navigation system error with respect to the vicon(not used by navigation). Some drift can be seen in both the north and east directions.

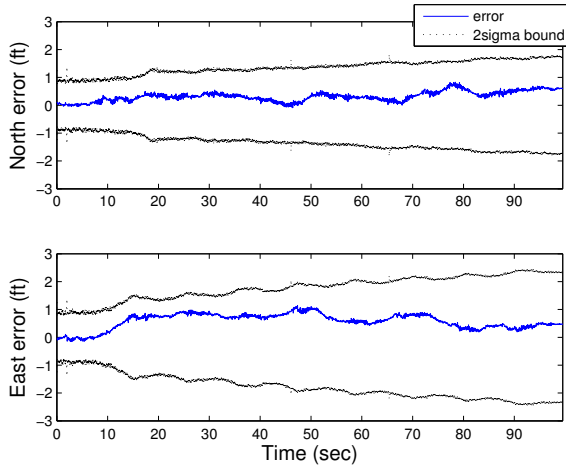


Fig. 9. Error of track flight path with respect to Vicon data

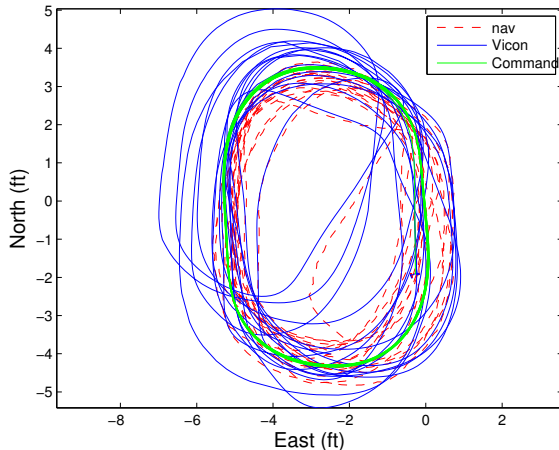


Fig. 10. Vicon used in navigation for certain sections of time.

We then completed a flight test in the Vicon room using a similar loop. The test consisted of completing several loops with vicon and vision being used within the navigation and turning off the vicon data for sections of time to see how the vehicle compared position and response-wise to using only vision. Figure 10 shows vehicle completing multiple loops and apparent drift can be seen between the navigation solution and vicon. Figure 11 shows how well the navigation system compares to the commanded trajectory and vicon. Figure 12 shows how the error builds up with vicon off and how it jumps back down with vicon on.

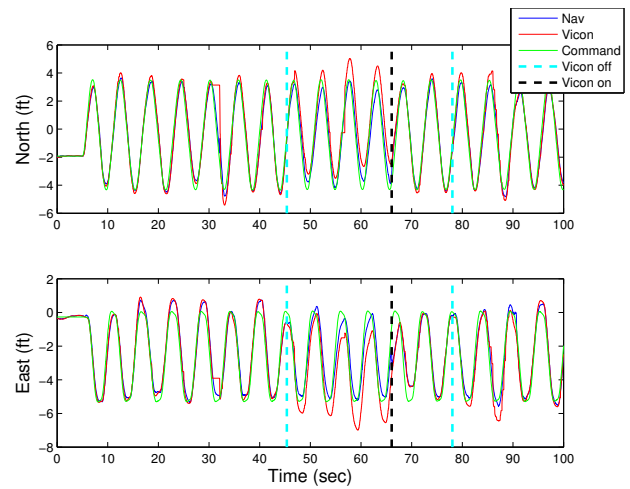


Fig. 11. North and East track plots where vicon is turned on in the navigation for sections showing how Vision only compares with vicon external aide.

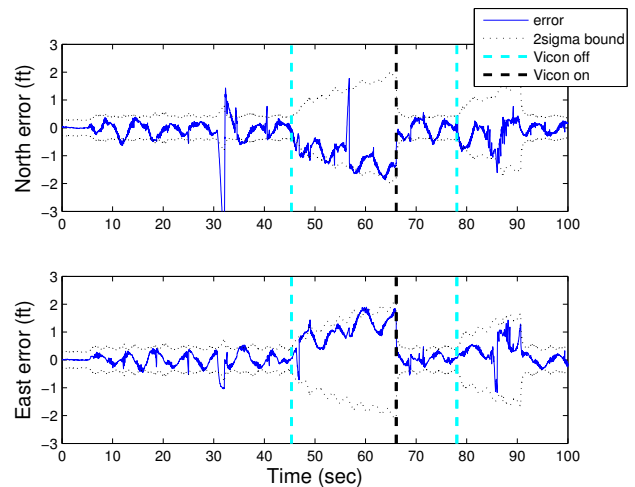


Fig. 12. North and East error with respect to vicon where vision is used only in certain time sections and vicon turned back on in sections. When vicon is turned on the error and covariance jump down.

CONCLUSIONS

This paper highlights the work done by the UAVRF in developing a powerful lightweight quadrotor vehicle capable of indoor navigation using vision only with no external aides. The navigation system uses a Bierman-Thornton extended Kalman filter which improves numerical stability. Through extensive simulation and flight test work, the vehicle has proven to be a viable platform for future indoor vision-based navigation work and hopefully a winning platform for the AHS MAV 2015 competition.

REFERENCES

- ¹Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V., “Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav,” Robotics and Automation (ICRA), 2014 IEEE International Conference on, 2014.
- ²Blosch, M., Weiss, S., Scaramuzza, D., and Siegwart, R., “Vision based MAV navigation in unknown and unstructured environments,” Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010.
doi: 10.1109/ROBOT.2010.5509920
- ³Engel, J., Sturm, J., and Cremers, D., “Camera-based navigation of a low-cost quadcopter,” Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, Oct 2012.
doi: 10.1109/IROS.2012.6385458
- ⁴Mulgaonkar, Y., Whitzer, M., Morgan, B., Kroninger, C. M., Harrington, A. M., and Kumar, V., “Power and weight considerations in small, agile quadrotors,” SPIE Defense+ Security, 2014.
- ⁵Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., and Shein, A., “GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft,” *Journal of Field Robotics*, Vol. 30, (3), 2013, pp. 415–438.
- ⁶Wu, A., *Vision-based navigation and mapping for flight in GPS-denied environments*, Ph.D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332, dec 2010.
- ⁷Magree, D. and Johnson, E. N., “Performance of a monocular vision-aided inertial navigation system for a small uav,” AIAA Guidance, Navigation, and Control (GNC) Conference. American Institute of Aeronautics and Astronautics, 2013.
- ⁸Barry, A. J. and Tedrake, R., “Pushbroom Stereo for High-Speed Navigation in Cluttered Environments,” *CoRR*, Vol. abs/1407.7091, 2014.
- ⁹Magree, D. and Johnson, E. N., “A Monocular Vision-aided Inertial Navigation System with Improved Numerical Stability,” *AIAA SciTech*, 2015.
doi: 10.2514/6.2015-0097
- ¹⁰Mettler, B., Kong, Z., Goerzen, C., and Whalley, M., “Benchmarking of obstacle field navigation algorithms for autonomous helicopters,” Proceedings of the American Helicopter Society 66th Annual Forum. Phoenix, AZ, 2010.
- ¹¹Bershadsky, D. and Johnson, E., “Benchmarking of UAV Guidance Systems in Nap of the Earth (NOE) Flight,” Proceedings of the American Helicopter Society 71st Annual Forum. Montreal, QC, 2014.
- ¹²Lefferts, E. J., Markley, F. L., and Shuster, M. D., “Kalman filtering for spacecraft attitude estimation,” *Journal of Guidance, Control, and Dynamics*, Vol. 5, (5), 1982, pp. 417–429.
- ¹³Johnson, E. N. and Kannan, S. K., “Adaptive trajectory control for autonomous helicopters,” *Journal of Guidance, Control, and Dynamics*, Vol. 28, (3), 2005, pp. 524–538.
- ¹⁴Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, 1977.