

Georgia Tech Team Entry for the 2013 AUVSI International Aerial Robotics Competition

Daniel Magree*, Dmitry Bershadsky†, Chris Costes‡, Stephen Haviland§,
David Sanz¶, Eric Kim||, Pierre Valdez**, Timothy Dyer††, Eric N. Johnson‡‡

Georgia Institute of Technology, Atlanta, GA, 30332-0152, USA

This paper describes the details of a Quadrotor Unmanned Aerial Vehicle capable of exploring cluttered indoor areas without relying on any external navigational aids. A Simultaneous Localization and Mapping (SLAM) algorithm is used to fuse information from a laser range sensor, an inertial measurement unit, and an altitude sonar to provide relative position, velocity, and attitude information. A wall avoidance and guidance system is implemented to ensure that the vehicle explores maximum indoor area. A semantic guidance algorithm identifies “rooms” and thoroughly explores their interior with an efficient path. Finally, an object detection system is implemented to identify target objects for retrieval and interaction. The vehicle is intended to be Georgia Tech Aerial Robotic Team’s entry for the 2013 International Aerial Robotics Competition.

I. Introduction

The Army, Navy, and the Air Force have identified indoor reconnaissance and surveillance capability as a top research priority due to the changing nature of the battlefield. Miniature air vehicles are ideal candidates for such missions as they can use three dimensional maneuvers to overcome obstacles that cannot be overcome by ground vehicles. However, significant technological challenges exist in order to ensure reliable operation in such environments. Most current algorithms for Unmanned Aerial System (UAS) Guidance Navigation and Control rely heavily on GPS signals,¹⁻³ and hence are not suitable for indoor navigation where GPS signal is normally not available. Furthermore, an indoor UAS must be sufficiently small in order to successfully navigate cluttered indoor environments, consequently limiting the amount of computational and sensory power that can be carried onboard the UAS. Finally, the UAS should be designed to be expendable due to the dangerous environments it needs to operate in, hence low-cost, low-weight designs need to be explored. These restrictions pose significant technological challenges for the design of reliable Miniature Air Vehicle (MAV) platforms capable of navigating cluttered areas in a GPS denied environment.

A. Problem Statement

The sixth mission of the International Aerial Robotics competition requires that a MAV weighing less than 1.5kg have the ability to enter and navigate within an unknown confined environment in search of a specific marked target without being detected. The mission also requires that the MAV locate and bring back a USB flash drive.

*Graduate Research Assistant, School of Aerospace Engineering

†Graduate Research Assistant, School of Aerospace Engineering

‡Undergraduate, School of Aerospace Engineering

§Graduate Research Assistant, School of Aerospace Engineering

¶Researcher, Center for Robotics and Automation, Madrid

||Graduate Research Assistant, School of Aerospace Engineering

**Graduate Research Assistant, School of Aerospace Engineering

††Graduate Research Assistant, School of Aerospace Engineering

‡‡Lockheed Martin Professor of Avionics Integration, School of Aerospace Engineering

B. Conceptual Solution

The Georgia Tech Aerial Robotics Team (GTAR) has developed an indoor MAV Unmanned Aerial System (UAS) capable of exploring cluttered indoor areas without relying on external navigational aids such as GPS. The MAV uses a custom-designed quadrotor platform and is equipped with off-the-shelf avionics and sensor packages. We use an elaborate navigation algorithm that fuses information from a laser range sensor, inertial measurement unit, and sonar altitude sensor to form an accurate estimate of the vehicle attitude, velocity, and position relative to indoor structures. We leverage the fact that all indoor structures have walls to design a guidance algorithm that detects and avoids walls as well as ensures that the navigation solution maintains its fidelity and maximum indoor area is explored in a reasonable amount of time. We use a control architecture that augments a proven baseline proportional-derivative controller with an optional adaptive element that aids in mitigating modeling error and other system uncertainties.

C. Yearly Milestones

The GTAR Team aims to continue to develop a stable MAV capable of carrying an avionics payload for completing the 6th mission. The vehicles will have the ability to navigate and explore the indoor environment, detect and extract the flash drive. Subsequent yearly milestones include exploring the indoor environment without being detected and improving guidance efficiency. A simple velocity field method was employed in previous years and is now used as a backup guidance mode. A frontier-based guidance system was used in 2010 which forms the basis of the graph-based exploration system used in 2011 and 2012.

The following improvements have been made for the year of 2013:

- Continued improvement of the graph based guidance system.
- A semantic guidance system is able to identify “rooms” and thoroughly and efficiently explore these areas for targets.
- Several improvements to the SLAM navigation system have been made which improve map consistency.
- A new aircraft frame has been designed which has removed 200 grams of mass.
- An upgraded avionics package has more than quadrupled the available processing power onboard the aircraft.
- A novel target retrieval mechanism based on a deployable rover has been developed which will allow greater flexibility and reliability.

These and other improvements are detailed in this report.

II. Description of Vehicle

A. Aerial Platform

Quadrotors have become a very popular choice for MAV’s due to their relatively high payload capacity and high maneuverability.⁴ Furthermore, unlike helicopters, quadrotors avoid the use of mechanical parts for exerting moments and forces required for maneuvering. We had in previous years selected the AscTec Pelican Quadrotor made by Ascending Technologies GmbH as the base airframe (see Fig.2(a)). The vehicle structure, motors, and rotors of AscTec Pelican were used without modification. Two such vehicles (GTQ1 and GTQ2) are being used for development of the 2013 competition system. The vehicle generates lift using four fixed pitch propellers driven by electric motors. Control is achieved by creating a relative thrust offset between the propellers. Quadrotors can either be flown with diamond configuration (front, right, back, left motors are used to effect pitching, rolling, and yawing motion) or square configuration (front-right, back-right, back-left, front-left motors are used to effect pitching, rolling, and yawing motion). Although many Quadrotors in aerial robotics community fly with diamond configuration,⁴⁻⁶ we selected the square configuration (shown in Fig 2(b)) to allow for more flexibility in sensor mounting locations.

The intended entry for the 2013 competition is built around a custom-designed frame using AscTec motors and motor controllers and is identified as GTQ3. At the 2012 competition, the vehicle had an AUW of 1.498 kg. This did not allow for addition of a more sophisticated pickup mechanism or a faster computer. The

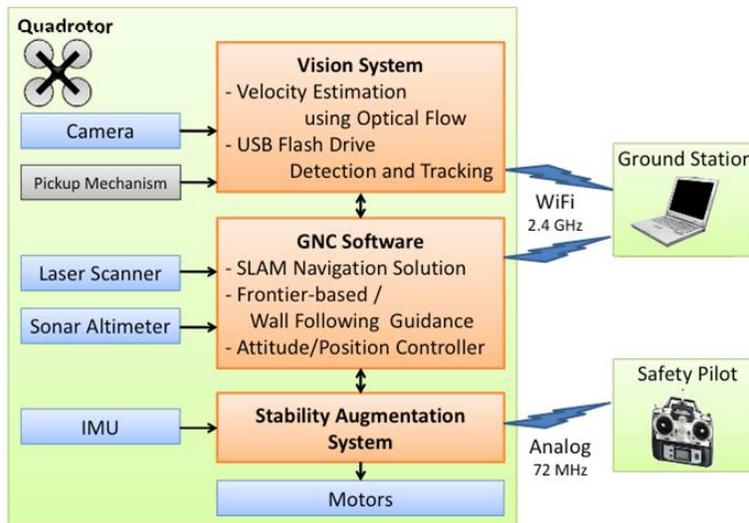
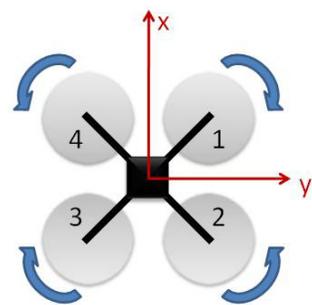


Figure 1. GTAR System Architecture



(a) AscTec Pelican Platform



(b) Quadrotor Flying in Square Configuration

Figure 2. The 2013 GTAR Team chose to use the Ascending Technologies Pelican for software development.

new GTQ3 frame has been designed for a more economical cost and weight budget. Swapping components and maintenance on GTQ3 has eased as well due to the more simple, open design. The drive system is replicated from that of the GTQ2. The AUV of GTQ3 is about 40% less (and up to 50% lower depending on pending equipment choices) than that of the GTQ2 and its cost is an order of magnitude lower. This allows for the addition of new avionics and a pickup rover. The GTQ3 frame is made from a combination of printed plastic, laser cut wood, and carbon rods.

B. Avionics

1. Flight Computers

The GTQ avionics are designed to be entirely independent of the vehicle ground station. All processing is performed onboard the vehicle which eliminates the dependency on a strong wireless communication link to a remote computer. Key considerations when the onboard computer was selected were physical size, weight, and power consumption. The onboard computer selected was the Hardkernel ODROID-X, a system based on a 1.4 Ghz quad core ARM A9 CPU in a 90 mm \times 95 mm format. An image of the ODROID can be seen in Figure 3(a). This computer handles both the vision system and GNC software (see Fig. 1). The ODROID runs on an ArchLinux operating system which allows leveraging of our previous work in rotorcraft control. This includes the ability to fully utilize the GUST software suite of the GeorgiaTech UAV Research Facility.⁷

The ODROID-X flight computer communicates via serial interface with a stability augmentation system (SAS) running on a custom designed board. An image of the SAS can be seen in Figure 3(b). The SAS is designed around an ATmega128 microcontroller, which sends and receives data from the flight computer. The board also communicates with the IMU, motors and RC receiver. To allow manual control of the vehicle by a safety pilot, the SAS board provides rate damping using the data from the IMU. All software handling the vehicle kill switch and the auto/manual switch is processed in the SAS.

2. Sensor Suite

The GTAR entry uses three primary measurement sensors for navigation, stability and control. These devices are a laser range finder and a sonar altimeter for the GNC software, and an inertial measurement unit (IMU) for stability augmentation. The IMU employed by the GTAR team is the ADIS-16365-BMLZ built by Analog Devices Inc. It consists of a tri-axis digital gyroscope and tri-axis accelerometer that can measure forces up to ± 18 g. The laser range finder used is the Hokuyo URG-04LX-UG01. It is capable of measuring distances up to 4 m and has a maximum detection area of 240 degrees, with a resolution of 1 mm and 0.36 degrees respectively. The sonar altimeter used is the MB1040 LV MaxSonar EZ4 high performance ultrasonic range finder. It is capable of measuring distances up to 6.45 m away with resolution of 25.4 mm. In addition to the navigational sensors, a USB camera is mounted on the vehicle for use in target identification and tracking. The camera is a PointGrey Firefly.

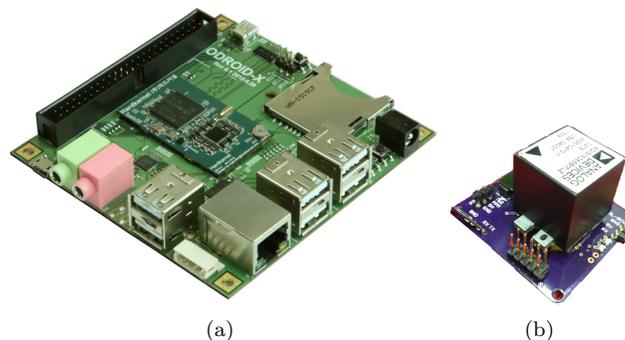


Figure 3. ODROID-X flight computer (left) and stability augmentation system custom-designed circuit board (right).

3. Communications

The onboard computer can communicate via 802.11g and Bluetooth wireless links. The computer will communicate with a ground computer using a wireless Local Area Network (LAN) link.

4. Power Management system

The system uses off-the-shelf battery packs which have a track record of proven safety. A single battery powers all onboard electronics. The motors and motor controllers are powered directly from the battery, while power for avionics is first regulated to 5V by a switching regulator.

III. Guidance, Navigation, and Control System

A. Exploratory Guidance

The developed guidance system uses only locally available information gathered through the onboard sensors, which includes a laser range scanner. The system maintains previous tree-based guidance systems as backup for the newly developed graph based frontier exploration system. Scan frontier points are used to add nodes to the undirected exploration graph at each scan point. A node is added (blue empty points, Fig.4(b)) as the midpoint of each independent frontier of at least a particular arc length. Each new node is checked against existing nodes. Should a node exist within a specified Cartesian distance from the new node, a node will not be added, but a graph connection (brown lines Fig.4(b)) will be made between the existing node and the node at the current scan point.

As the vehicle captures a node (blue, green-filled points Fig.4(b)), a new scan is performed and the next node is chosen based on a weighting system. Distance to the node, the arc length of the frontier used to create it, and a directional bias are used to select the next target highlighted in red. The intended graph path is highlighted as well. The directional bias is used should multiple vehicles be operating at the same time as an exploration efficiency enhancement. Simulated truth results are shown in Fig.4(b) where the yellow path traces the vehicle's position over time. The vehicle has traveled approximately 120' in the figure, showing about 60 seconds of exploration.

Should the vehicle not reach the target node in a specified amount of time, or should the GNC software detect that the vehicle is stuck in an unfavorable position, the vehicle will be commanded to backtrack to the last reached node. Stuck detection works by applying three criteria, all of which increase the confidence in the stuck detector output. Firstly, the mean velocity of the vehicle over a rolling time window is calculated and checked against a threshold set in the software. If the velocity is below this value, the probability of a stuck detection is increased. Secondly, in that same time window, should the velocity of the vehicle flip on itself more than a number of allowable times, the probability is yet again increased. Lastly, if the vehicle does not leave a circle of a specified radius within the time window but has traveled a specified distance, the stuck probability is again increased. Timeout is forced should probability reach a specified value.

B. Room Guidance

Room detection is based on OpenCV routines. The SLAM maps obstacles are dilated to close off the doorways. Then, OpenCV searches for closed contours of a certain area, bounding box area fraction, and aspect ratio.

Once one space has been defined as a room, the following task is to decide how to explore it. Considering that the 30cm-wide box containing the USB stick could be located close to the wall, the problem could be understood as a CPP one, where a full area coverage analysis is required. Several decomposition taxonomies are present in the literature: from cellular or polygonal approaches⁸ to graph based ones,⁹ every taxonomy have different advantages and cons. In this case, taking into account the size and regularity of the spaces, a grid-pattern approximation has been chosen due to its simplicity.

A Sukharev's decomposition approach has been selected, centering the the focus in the center of each cell.¹⁰ The distance among these future waypoints -and hence the cell size- bases on the flight height, the resolution required and the field of view of the camera. Equation 1 presents the formula for obtaining the cell dimension (in mm)),

$$CD = \frac{H * FoV}{F} \quad (1)$$

where CD corresponds with the Cell Dimension, H refers to the flight altitude, FoV with the Field Of View (another way to represent the angle of view of the lens, specified by the manufacturer) and F with the Focal Length -the distance from the optical center of the lens to the sensor, when the lens is focused on an object at infinity.

This discretization provides a cellular decomposition of the space previously extracted by the SLAM algorithm. It defines a grid, where -as said before- the center of each cell is a waypoint that should be visited. Furthermore, also the start position and the exit should be defined: in cases where there are multiple possible exits/alternatives, one of them is chosen randomly (as far as we have not *a priori* information about the location of the target). In this sense, the problem is defined as a collection of WP, an starting point (the actual one) and a unique exit.

The decision about how to explore this space could be understood as a particularization of the Traveling Salesman Problem (TSP): several nodes/places to visit, where the best path among them to be found. Nevertheless, despite of its simple appearance, it is well known that this is not a banal problem. The TSP is defined as NP-complete (more precisely, it is complete for the complexity class FP^{NP}), so no algorithms are known for solve the problem optimally without checking all the alternatives. It implies $O(n!)$ order and huge computing times when the number of nodes increase.

It could be formally expressed as:

$$P = n_0, n_1, n_2, ..n_{N-1} \quad (2)$$

$$\min(dist_p) = \min\left(\sum_{i=0}^{N-1} dist[n_i, n_{i+1*mod(N)}]\right) \quad (3)$$

where n_x are the nodes or WP. The permutation P of the n_x should be that that sum of the distances among each pair would be minimum ($\min(dist_p)$).

For the quadrotor, a simpler heuristic algorithm Christofides-based has been implemented. In it, the weights of the nodes are assigned according to the number of node's instances in the same position. It tries to maximize the dispersion of the solutions, providing a wider range of them. Table 1 provides the results obtained (in simulation) in for one of the rooms according to the number of iterations of the algorithm.

Num. Iterations	Mean computing time (ms)	Mean distance to cover (m)
1	86.5	42.47
7	169.8	35.85
35	665.0	33.74
70	1289.4	31.35
150	2736.9	27.36
350	6398.1	28.37
700	12514.1	24.40
7000	1.210 ⁵	22.56

Table 1. Metaheuristics performance comparison

Different space configurations were tested in order to define the balance value. The assessment of these results has allowed to define the operational threshold for this application in 100 iterations. It takes around 1.8s in provide the solution, with a quality of 7% (12% maximum deviation from the optimum).

The best solution found for every space is directly send to the drone as a waypoint list to cover.

C. Stability Augmentation System (SAS)

The quadrotor platform is inherently unstable, that is, without control inputs, the platform would enter an uncontrolled drift in velocity and angular rates and collide with the ground or nearby obstacles. Quadrotors are also known to be notoriously hard to control even for human pilots, particularly because the relationship between thrust and stick deflection is nonlinear and because attitude is coupled heavily with velocity. Hence, it is desirable to integrate angular rate damping to aid the pilot in controlling the quadrotor. Let \hat{p} , \hat{q} , and \hat{r} denote the gyroscope measurements of the quadrotor roll, pitch, and yaw rates, and δ_{ϕ_p} , δ_{θ_p} , and δ_{ψ_p} denote the pilot roll, pitch, and yaw stick deflections, then the actual stick deflection commands are assigned

using the following proportional control logic:

$$\delta_\phi = \delta_{\phi_p} - K_p \hat{p}, \quad (4)$$

$$\delta_\theta = \delta_{\theta_p} - K_q \hat{q}, \quad (5)$$

$$\delta_\psi = \delta_{\psi_p} - K_r \hat{r}. \quad (6)$$

In equation 4, K_p , K_q , and K_r denote the linear gains chosen to provide appropriate rate damping.

D. Control Algorithm

The complexity of the control system depends not only on the quantities being controlled, but also on the dynamics of the system itself. Unlike ground vehicles, unstable air vehicles are susceptible to oscillation and divergent flight when the control system is not properly tuned. Even for stable flying vehicles, coupling between lateral and longitudinal motion as well as aerodynamic interaction with the environment must be considered. The control architecture used since the GTAR 2011 team leverages the proven Model Reference Adaptive Control architecture developed for control of VTOL UAS throughout their flight envelop by Georgia Tech UAV Research Facility.¹¹⁻¹³ In this architecture, a position control loop generates a velocity command, a velocity control loop generates an attitude command, and an attitude control loop generates servo commands to stabilize the vehicle by controlling the angular rate. Kannan has shown that such nested and cascaded control loop architecture with actuator saturation can indeed be used to control VTOL UAS.¹² This system of nested control loops requires that the vehicle maintain an estimate of its position, velocity, attitude, and angular rate. For details the reader is referred to references 11, 12, 14, 15 for further details.

E. Navigation Algorithm

A variety of SLAM algorithm implementations are available for free use at the web site OpenSLAM.org. The algorithm used for the preliminary research, called CoreSLAM,¹⁶ was chosen primarily because it is simple, easy to implement, and it uses integer math where possible to improve computational speed.¹⁷ There are two main parts to any SLAM routine. The first task is to measure distance to obstacles or landmarks in the environment, and to map them given the vehicle's position and orientation (i.e. mapping). The second task is to determine the best estimate of the vehicle's position and orientation based on the latest scan (or series of scans) given a stored map (i.e. localization). The mapping and localization tasks are performed together to maintain the most current map and position estimate.

Several improvements have been made to the original CoreSLAM algorithm that have greatly enhanced its efficiency and reliability. The original algorithm was modified to allow interpolation of obstacle locations during mapping and scan matching. This has allowed in ideal conditions a position accuracy of less than the grid size of the of the map, currently set to 4 in. This change also allowed a smaller obstacle width to be represented in the map, resulting in cleaner maps. In addition, the confidence in laser hits was made greater than laser miss confidence. This change prioritized obstacles over free space, and generally matches the characteristics of the laser scanner better. The laser scanner is more likely to return a false miss than a false hit. Figure 4(a) shows the trajectory of the quadrotor and the SLAM map generated during the simulated flight. Note the minimal angular or translational drift during the 3 minute flight.

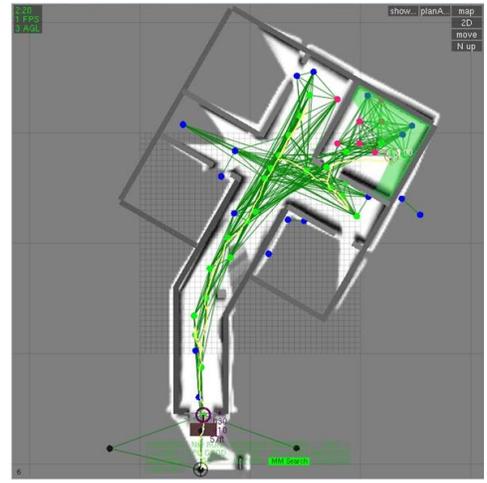
The inherent nonlinearities in the vehicle dynamics and the measurement sensors are handled through the use of an Extended Kalman Filter (EKF). An existing EKF-based navigation filter architecture developed at the Georgia Tech UAV Research Facility is utilized as part of the indoor navigation system.² The navigation algorithm developed by the GTAR team augments the existing EKF based architecture to function without GPS signals by using the range information obtained from the laser range scanner and the sonar altimeter.¹⁸

F. Switch Button Approach and Collision

Approaching to alarm-disabling switch button located on corridor wall requires the quadrotor to know both the three-dimensional switch button location and the wall location information. Thus, pressing switch button guidance system utilizes both camera for visual tracking algorithm and LIDAR for SLAM wall avoidance guidance system. LBP cascade classifier from OpenCV is used to detect switch button sign, where state of image location and size is used to estimate target location as explained previously on memory stick detection.



(a) map



(b) guidance

Figure 4. On the left, an image of the resulting slam map and trajectory of the quadrotor while exploring a simulated arena. On the right, the structure of the graph-based exploratory guidance system is visible

The wall avoidance algorithm, explained on exploratory guidance system, is added on top of the tracking command to avoid hitting other wall or obstacle during approach.

Pressing a switch button while in air causes unpredictable momentum and instability on aircrafts dynamics. To avoid this issue, guidance algorithm is developed to first fly parallel to the wall until switch button is located in minimum distance, collide straight to the detected object, and recover back to the center of corridor before going back to exploratory guidance mode.

G. Memory Stick Detection and Retrieval

Memory stick detection and retrieval is achieved with a combination of open-source software and an original tracking algorithm optimized for use on computers with low computational power. The object detection algorithm is part of the open-source software package OpenCV. The detector first is trained using a series of images of the target object, called the positive image set. The positive image set can be created from multiple images of the object, or from a single image that is artificially distorted to simulate viewing the object from many angles. Approximately 1000 images were used. Additionally, a set of 1000 images without the object, called the negative set, is passed to the function. The output of the training algorithm is a cascade of Haar classifiers that can be used to efficiently identify the object in a sample image. Generating the classifier typically took 8 hours of processing time on a standard desktop computer with a Core 2 Duo processor. Processing an image with the classifier could be performed in 0.2 seconds on the GTQ onboard computer.

Once the target memory stick is located a rover is lowered onto the table. A specialized guidance system generates a trajectory for the vehicle for picking up the USB drive. The guidance system estimates the three-dimensional location of the memory stick using a downward-facing camera mounted on the vehicle. A cascade filter identifies the pixel location and pixel area of the memory stick in each image from the camera, and an EKF estimates the three-dimensional location based on the pixel measurements. The rover vehicle is then commanded to push the USB drive into a wall in order for the USB drive to attach to the rover.

Let the camera frame coordinates of the memory stick be given by $\bar{x}_c = [XYZ]^T$. Measurements from the camera are the square-root of the pixel area, A_{sqrtp} , and pixel location, x_p and y_p , and define measurement

vector $\bar{y} = [A_{sqrtp} x_p y_p]^T$. The equations for the measurement model are given by

$$A_{sqrtp} = \frac{k}{X} \sqrt{A_{ft}} \quad (7)$$

$$x_p = \frac{k}{X} Y \quad (8)$$

$$y_p = \frac{k}{X} Z \quad (9)$$

where A_{ft} is the area of the memory stick in feet and k is the pixel to radian scale factor, both known constants. The linearized measurement model is generated by taking the first order Taylor series expansion of the previous equations.

$$\frac{\partial \bar{y}}{\partial \bar{x}_c} = \begin{bmatrix} -\frac{k}{X^2} \sqrt{A_{ft}} & 0 & 0 \\ -\frac{k}{X^2} Y & \frac{k}{X} & 0 \\ -\frac{k}{X^2} Z & 0 & \frac{k}{X} \end{bmatrix} \quad (10)$$

Converting to the inertial frame,

$$\frac{\partial \bar{y}}{\partial \bar{x}} = \frac{\partial \bar{y}}{\partial \bar{x}_c} \frac{\partial \bar{x}_c}{\partial \bar{x}} = \frac{\partial \bar{y}}{\partial \bar{x}_c} L_{ci} = \mathbf{C} \quad (11)$$

where L_{ci} is the rotation matrix converting from the cameraframe to the inertial frame. Inserting matrix \mathbf{C} into the EKF measurement update gives the corrected state.

$$\mathbf{K} = \mathbf{P}^- \mathbf{C}^T (\mathbf{C} \mathbf{P}^- \mathbf{C}^T + \mathbf{R})^{-1} \quad (12)$$

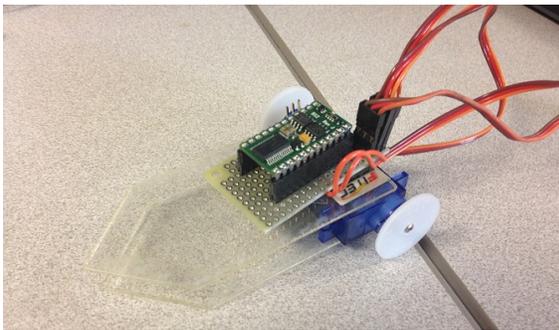
$$\hat{x} = \hat{x}^- + \mathbf{K} [\bar{y} - h(\hat{x}^-)] \quad (13)$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K} \mathbf{C}) \mathbf{P}^- \quad (14)$$

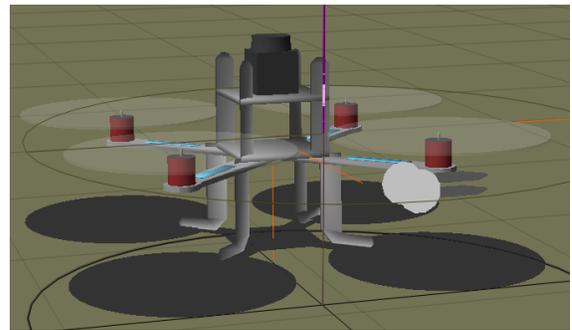
where \mathbf{R} is the measurement covariance matrix, \hat{x} is the inertial frame measurement estimate, and $h(\hat{x}^-)$ is the output of the nonlinear measurement model. This system estimates the three-dimensional location of the memory stick with sufficient accuracy and the ability to navigate the rover to grab the USB drive.

H. Rover Design

The rover consists of two thin acrylic sheets that are separated enough for the USB drive to fit inside as seen in Figure 5(a). The idea is to push the USB drive with the rover into a wall in order to push the USB drive into the gap between sheets. Between the acrylic sheets is a sticky residue that won't allow the USB drive to slip out of the rover when its pulled back up to the MAV. The rover has an onboard BASIC Stamp 2 computer which controls the servos. The Odroid sends commands via serial to the stamp computer on which direction to travel. The wheels will be made of rubber to ensure traction on any surface. To be able to lower the rover into a corner of a room the rover will be placed underneath a propeller as seen in Figure 5(b).



(a) Rover Mockup



(b) Rover Placement as seen in Simulation

Figure 5. Rover will be used to grab the USB drive from within the In-box on the desk.

I. Flight Termination System

A manual takeover switch is provided so that a human safety pilot can take over control when required. The system is also provided with a remotely controlled “kill-switch” that ensures power to the motors is killed when triggered.

IV. Operations

A. Flight Preparations

Before each autonomous flight test or competition trial, a checklist of preparations are to be followed (see table A).

Table 2. Flight Checklist

Steps completed days before flight session	Charge flight batteries, transmitter batteries Load new software onboard and ground station Complete hardware-in-the-loop (HITL) tests to ensure proper operation of any code changes
Steps completed day of flight session	Ensure all flight test equipment is present. Set up ground station.
Steps completed before each flight	Clearly brief safety pilot of intention of flight Check structural integrity of vehicle and ensure proper center-of-gravity position.
During flight test	Pilot has primary discretion on whether to take manual control if vehicle is in jeopardy. Besides this discretion, safety pilot will only obey judges or ground station operator. Once the low voltage warning tone is heard, safety pilot takes control and lands the aircraft.

B. Man/Machine Interface

A ground station based on the GIT GUST software environment will continuously monitor the flight vehicle and display health and status information during the flight.⁷ The flight vehicle will send its current estimated position/heading, obstacle locations, and battery voltage via a wireless LAN data link. In addition, a frame-grabber is used to retrieve images from the incoming video stream for processing. Instructions from the ground station, including the adjustment of system parameters during manual flight, are transmitted over a wireless LAN data link. A safety pilot link is included, which operates via a separate 2.4GHz radio uplink.

V. Risk Reduction

A. Vehicle Status Monitoring

The flight vehicle continually monitors its surroundings for potential hazards and obstacles using the onboard laser range scanner. The information about potential hazard can be transmitted to the ground station for monitoring.

1. Shock and vibration isolation

The chosen onboard electronics have inherent tolerance to shock and vibration. Further vibration reduction is achieved through careful mounting of the hardware. The avionics package is mounted close to the center of gravity to minimize motion induced due to body rotations. The IMU is mounted directly on the SAS board. The laser range scanner and the sonar altimeter are mounted using a low cost vibration isolation mechanism.

2. *Electromagnetic Interference (EMI)/Radio Frequency Interference (RFI) Solutions*

The chosen quadrotor platform has brushless motors, which has reduced EMI signature. Further EMI mitigation is achieved by mounting the avionics package at the center of the airframe, and thus spatially separating it from the motors. Proper electric grounding and additional capacitors are used to provide further protection against EMI. A 2.4GHz transmitter was chosen for the video link, the safety pilot radio control link, and the data link. This eliminates the typical “servo jitter” affecting UAVs operating with 900MHz transmitters nearby. Possible interference between the different 2.4GHz systems is reduced by proper shielding and location of antennas.

B. Safety to Bystanders

The quadrotor platform used in this work has a protective shroud that minimizes the risk of rotor strike and improves crash-worthiness. Further safety is incorporated by using off-the-shelf battery packs with a track record of proven safety. A manual takeover switch is provided so that a human safety pilot can take over control when required. Finally, the system is also provided with a remotely controlled “kill-switch” that ensures power to the motors is killed when triggered.

C. Simulation

The GTAR team utilizes existing simulation software developed for research projects at Georgia Tech UAV lab.²⁰ The simulation significantly reduces development time as the team can adopt navigation filter and controller already implemented in other UAVs. The simulation comes complete with modeling of uncertainties such as gusts, and modeling of indoor environments. All sensors are elaborately emulated and their noise properties are reproduced for testing purposes. Onboard code developed in simulation is directly used for autonomous flight. The setup is also capable of hardware-in-the-loop (HITL) test (simulating only vehicle dynamics and sensor readings).

VI. Testing

The GTAR system is being rigorously subjected to flight testing at the indoor test flight facility at Georgia Tech. The VICON camera based object tracking system is being used to validate the navigation algorithm. We are using the protocols developed by the GTAR team for the 2013 effort for ensuring safety, efficiency, and reliability in flight tests.

VII. Conclusion

We presented the details of a quadrotor Unmanned Aerial Vehicle intended for exploring indoor areas. The vehicle uses a custom-designed platform equipped with off-the-shelf avionics and sensor packages. Information from a scanning laser range sensor, inertial measurement unit, and a altitude measurement sonar are fused in an EKF-based navigation solution using Simultaneous Localization and Mapping (SLAM) methods. An important feature of this navigation architecture is that it does not rely on any external navigational aid, such as Global Positioning System signal. The output from the navigation system is used in an adaptive neural network control architecture to provide a robust and stable flight vehicle.

Higher level trajectory commands are generated by multi-mode guidance system. The exploration guidance picks waypoints which take the vehicle to map frontiers. If, during exploration, a room is identified, an room guidance is engaged which efficiently traverses the space in search of the target. Once the target is identified, a target tracking and retrieval guidance system retrieves the target and the vehicle exits the arena.

A simulation model of the vehicle has been developed and the navigation and control algorithms have already been validated in simulation. Flight test of most of the aforementioned system and software components have already taken place and further combined testing is in progress. The Georgia Tech Aerial Robotics team intends to compete in the 2013 IARC competition with this vehicle.

Acknowledgments

The Georgia Tech Aerial Robotics team wishes to thank Jeong Hur, Dr. Suresh Kannan, Claus Christmann, Nimrod Rooz, Dr. Erwan Salaün, Stu Godlasky, Jeremy Montgomery, Xo Wang, and Eohan George for valuable contributions.

References

- ¹Kayton, M. and Fried, W. R., *Avionics Navigation Systems*, John Wiley and Sons, 1997.
- ²Christophersen, H. B., Pickell, W. R., Neidoefer, J. C., Koller, A. A., Kannan, S. K., and Johnson, E. N., "A compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles," *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, May 2006.
- ³Wendel, J., Maier, A., Metzger, J., and Trommer, G. F., "Comparison of Extended and Sigma-Point Kalman Filters for Tightly Coupled GPS/INS Integration," *AIAA Guidance Navigation and Control Conference*, San Francisco, CA, 2005.
- ⁴Bouabdallah, S., Noth, A., and R., S., "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.
- ⁵Portlock, J. and Cubero, S., "Dynamics and Control of a VTOL quad-thrustaerial robot," *Mechatronics and Machine Vision in Practice*, edited by J. Billingsley and R. Bradbeer, 2008.
- ⁶Guo, W. and Horn, J., "Modeling and simulation for the development of a quad-rotor UAV capable of indoor flight," *Modeling and Simulation Technologies Conference and Exhibit*, 2006.
- ⁷Johnson, Eric N, C. A. J., Watanabe, Y., Ha, J.-C., and Neidhoefer, J., "Vision Only Control and Guidance of Aircraft," *Journal of Aerospace Computing, Information, and Communication*, Vol. 23, No. 10, October 2006.
- ⁸Maza, I. and Ollero, A., "Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms," *Distributed Autonomous Robotic Systems 6*, Springer, 2007, pp. 221–230.
- ⁹Valente, J., Sanz Muñoz, D., Cerro Giner, J. d., Rossi, C., Garzon Oviedo, M., Hernandez Vega, J. D., and Barrientos Cruz, A., "Techniques for Area Discretization and Coverage in Aerial Photography for Precision Agriculture employing mini quad-rotors," 2011.
- ¹⁰LaValle, S. M., *Planning Algorithms*, Cambridge University Press, Cambridge, U.K., 2006.
- ¹¹Johnson, E. and Kannan, S., "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance Control and Dynamics*, Vol. 28, No. 3, May 2005, pp. 524–538.
- ¹²Kannan, S. K., *Adaptive Control of Systems in Cascade with Saturation*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2005.
- ¹³Chowdhary, G. and Johnson, E., "Flight Test Validation of Long Term Learning Adaptive Flight Controller," *Proceedings of the AIAA GNC Conference, held at Chicago, IL*, 2009.
- ¹⁴Johnson, E. N., *Limited Authority Adaptive Flight Control*, Ph.D. thesis, Georgia Institute of Technology, Atlanta Ga, 2000.
- ¹⁵Chowdhary, G. V. and Johnson, E. N., "Theory and Flight Test Validation of Long Term Learning Adaptive Flight Controller," *Proceedings of the AIAA Guidance Navigation and Control Conference*, Honolulu, HI, 2008.
- ¹⁶Steux, B. and Hamzaoui, O. E., "Coreslam on openslam.org," website.
- ¹⁷Steux, B. and Hamzaoui, O. E., "CoreSLAM : a SLAM Algorithm in less than 200 lines of C code," Tech. rep., Mines ParisTech, Center for Robotics, Paris, France, 2009.
- ¹⁸Sobers, M. J., *INDOOR NAVIGATION FOR UNMANNED AERIAL VEHICLES USING RANGE SENSORS*, Ph.D. thesis, Georgia Institute of Technology, 2010.
- ¹⁹Johnson, E. and Turbe, M., "Modeling, Control, and Flight Testing of a Small Ducted Fan Aircraft," *Journal of Guidance Control and Dynamics*, Vol. 29, No. 4, July/August 2006, pp. 769–779.
- ²⁰Johnson, E. N. and Schrage, D. P., "System Integration and Operation of a Research Unmanned Aerial Vehicle," *AIAA Journal of Aerospace Computing, Information and Communication*, Vol. 1, No. 1, Jan 2004, pp. 5–18.