Quadrotor Unmanned Aerial Vehicle for the International Aerial Robotics Competition

Thomas Brady
Undergraduate – Aerospace Engineering, 2011
Nona Ebrahimi
Undergraduate – Computer Engineering, 2011
Matt Edelman
Undergraduate – Electrical Engineering, 2011
Daniel Ellis
M.S.E. – Aerospace Engineering, 2012
Thomas Manville
Undergraduate – Computer Engineering, 2011
Rohan Mehta
Undergraduate – Aerospace Engineering, 2011
Ryan Moore
Undergraduate – Computer Engineering, 2011
Anthony Smith
Undergraduate – Aerospace Engineering, 2010
Alex Valencourt
M.S.E – Aerospace Engineering, 2011
Brandon Wagoner
Undergraduate – Aerospace Engineering, 2011
Jeff Walls
Ph.D. – Mechanical Engineering

ABSTRACT

The University of Michigan's Autonomous Aerial Vehicle Team (MAAV) will compete in the 2010 International Aerial Robotics Competition using a student-developed quadrotor helicopter called MAAVerick. MAAV has implemented mapping techniques, decision logic, and control algorithms to navigate through an unknown facility to locate and retrieve a flash drive.

1 INTRODUCTION

1.1 Problem Statement

The International Aerial Robotics Competition (IARC) serves as an annual platform for engineers to showcase their ideas for pushing the envelope of aerial vehicles and related technology. The 2010 competition requires entrants to develop an autonomous vehicle that has the ability to navigate an unknown Nari-controlled compound to locate and extract a flash drive containing sensitive data. Our design employs a quadrotor platform using continuous laser scanning and image recognition to create a map of the traveled path of the vehicle. This data, combined with optimized control algorithms, allows the vehicle to fly in through the compound entry point and move from room to room while searching for the flash drive. Image recognition software identifies signs posted outside each room and locates the flash drive when the correct room has been determined. In the final stage of the mission, the flash drive is collected by MAAVerick's retrieval device and a decoy flash drive is left in its place. Finally, the drive is delivered to the handler within 10 minutes.

1.2 MAAV History

Founded in September of 2009, MAAV began with a group of four aerospace engineers and soon expanded to a multidisciplinary team of nearly 20 members. The group was started with the intent to compete in the IARC and to enhance students' hands-on experience in engineering. Within a few months, MAAV developed and manufactured a custom airframe specifically designed for all of the necessary flight hardware. MAAV also implemented image recognition software, mapping and navigation algorithms, and a PID controller for the vehicle.

1.3 Conceptual Solution

MAAVerick is a quadrotor UAV designed for autonomous navigation through unknown territory and retrieval of a flash drive. Its processing power is distributed among three integral system components: an Atmega 128 processor that handles stability control and maintaining vehicle attitude setpoints, a custom built ground station that handles memory intensive algorithms, vehicle trajectory optimization and image processing, and a Gumstix Overo Fire, which handles communication among the three. MAAVerick's attitude is determined by a Microstrain 8V Inertial Measurement Unit (IMU), while position in three-dimensional space is determined from a Hokoyu laser range finder and a Senscomp ultrasonic height sensor. Cameras located on the MAAVerick's extremities transmit pictures through the Gumstix to the ground station to be scanned for building signs and the flash drive.

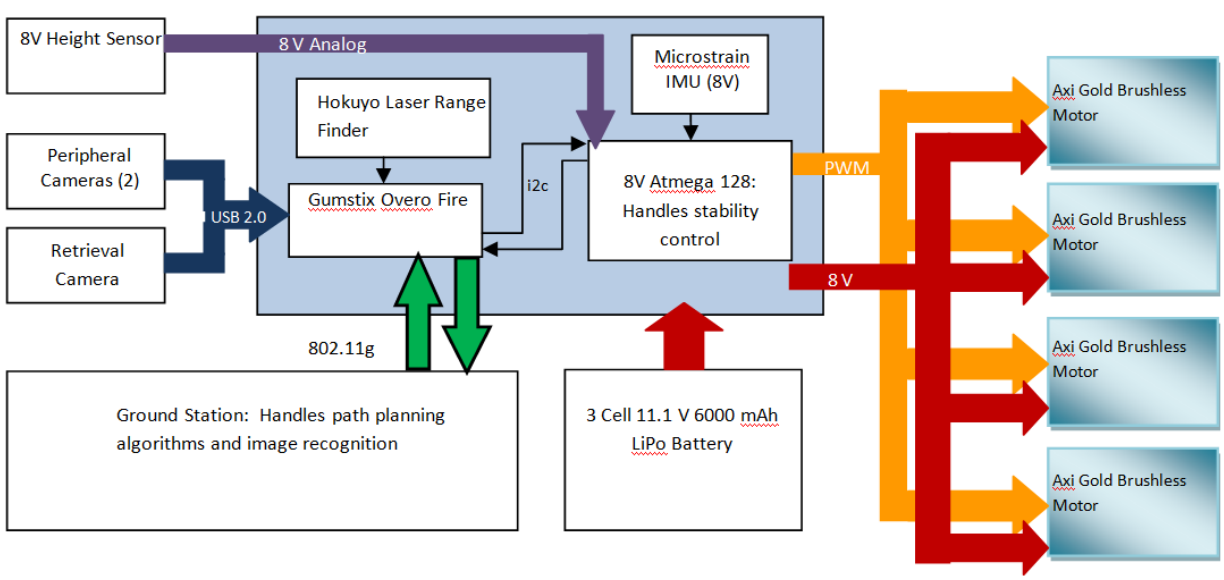


Figure 1 - MAAVerick System Architecture

Figure 2 - MAAVerick Structure and Electronics

2 AIR VEHICLE

2.1 Structure

In order to meet the requirements of the mission, MAAVerick's structure was designed to support four motors, withstand high stress loads, stay contained in a 30" x 30" x 30" volume, and maintain a low mass, while supporting all sensory and electronic devices. The structure is comprised of two concentric circles with circular cross sections, connected by radial bars at the motors. The structure has a foam core with a carbon fiber shell. Four carbon fiber propeller guards are attached to the structure. The battery is mounted under the circuit board, with an attachment holding the ultrasonic height sensor. The molds for the structure were designed using CATIA V5 and the foam core was machined using a CNC router. The carbon fiber was wrapped around the foam core and then placed between the two female molds to cure.

2.2 MAAVerick Control System

For MAAVerick, a PID controller was chosen due to its proven capability in similar systems and its ease of implementation. Because the vehicle was a new design, it was immediately apparent that the time to develop, build, test, and refine the design of this vehicle would be substantial. Thus, an effective yet relatively simple control scheme was ideal, because it minimized the time required to solve this important and complex problem. A number of similar applications and vehicles which used a PID controller were found and it was discovered that the performance of this type of controller would be adequate to complete the competition objectives [1][2][3]. Additionally, this type of controller allowed retuning based on observable behavior. Once the gains were set near the optimal values it was possible to tune the vehicle empirically, which reduced the reliance on the (potentially incomplete) dynamic model and provides a measure of robustness.

In any PID controller, the system input is comprised of three terms which are related to the difference (error) between the current system state and a desired state. The proportional (P) term multiplies the error by a gain value and uses this as an input. This increases the influence of this term when the current state is further away from the desired state. The integral (I) term uses the integral of the error over time multiplied by a gain value to influence the input. This generally has a gradual effect that reduces steady state error. The derivative (D) term, takes the derivative of the error and multiplies this by a gain value to influence the input based on how the error of the state is changing. This has a dampening effect which decreases oscillations in the system response. MAAVerick use four PID controllers in each of its roll, pitch, yaw, and height maneuvers.

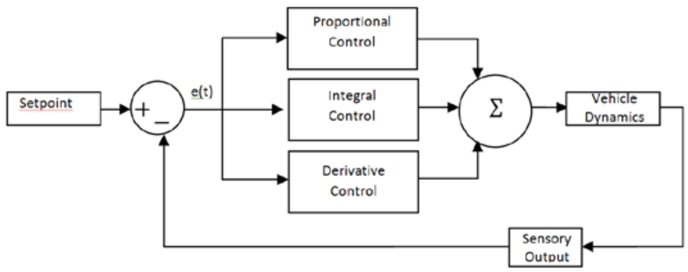


Figure 3 - Control Loop Architecture

2.3 Navigation

We have implemented a scan matching algorithm in order to obtain position information on the horizontal plane throughout the competition course. The scan matcher approximates the rigid body transformation between adjacent discrete time poses represented by the output of a single planar laser scanner. Once the transformation is known between neighboring poses, determining our global position is accomplished by applying the history of transformations to our known initial pose. Furthermore, the robot uses these transformations to build an occupancy-grid map from the raw laser scans in order to perform higher level tasks such as collision avoidance and goal oriented path-planning.

Due to its speed, simplicity, and power we have modeled our scan matcher on Generalized-ICP (GICP) [4]. This algorithm combines the classic Iterative Closest Point (ICP) and the recently more popular point-to-plane methods under a probabilistic framework. GICP mirrors standard ICP in computing point correspondences between two scans using Euclidean distance. The algorithm differs by using a probabilistic model in the minimization step of ICP. We consider that each scan is drawn from independent Gaussians so that the distribution of the distance between corresponding points is described by a zero-mean Gaussian with covariance given by a function of the sample covariances of the scans and the transformation. Finally, we use a Maximum Likelihood Estimator (MLE) to compute the transformation that maximizes the probability of the distance between corresponding points in the scans.

2.4 Decision Making

The navigation has several components after the generation of the map by the scan matcher. The vehicle must utilize the laser scanner and camera inputs in conjunction, weighing the information which each provides. Additionally, it keeps track of its mission state and prioritizes the inputs based on the current objective.

2.5 Mission States

The current objective of the vehicle is given by its mission state. As each objective is completed, the vehicle moves into the next mission state. Listed chronologically, the mission states of the vehicle are: Facility Entry, Exploration, Door Entry, Door Entry, Flash drive Search, and Exit.

2.5.1 Facility Entry

MAAVerick searches for the window designated for facility entry and attempts to enter through the center of the window.

2.5.2 Exploration

Information from imaging and mapping is used to create specific locations of interest on the map. In general, these locations are the lowest cost point on a frontier, which is a consecutive set of unexplored, but reachable points. Once a goal is set, the lowest cost path from the location of interest to the vehicle is traced, and the vehicle moves along the path.

2.5.3 Door Entry

The vehicle sweeps the outer edge of the door frame, using its cameras to search for the laser sensor indicators. The vehicle then moves to the center of the largest gap between the lasers and the boundries before moving into the next room and continuing its exploration.

2.5.4 Flash Drive Search

Once the target room has been found and entered, the vehicle explores the area to determine its shape then begins a lawnmowing strategy to find the flash drive in an efficient manner. In order to give the onboard cameras the best chance to locate the flash drive, the vehicle optimizes its height for imaging. Once located, the vehicle utilizes camera data to position the vehicle over the flash drive for pick-up.

2.5.5 Exit

With the flash drive onboard, the vehicle plots the lowest cost path through explored regions to the original window. Following the same procedures for doors and windows as it did on the way in, the vehicle exits the building.

2.6 Retrieval Strategy

In order to keep the mass of the vehicle below 1.5 kg and keep the number of moving parts on the MAAVerick to a minimum, our retrieval strategy utilizes industrial strength double-sided tape to stick the flash drive to the vehicle. Once MAAVerick has located the flash drive, it will land on top of the flash drive. Simultaneously, our vehicle will release the decoy flash drive by running a current through a wire that passes through two nylon hooks. These hooks hold a trap door in place, which secures the decoy flash drive onboard. When the nylon hooks melt, the weight of the flash drive will swing the door open and the flash drive will fall to the table.

2.7 Cost Map Generation

The determination of the vehicle's path comes from a grid-based cost map generated from the map passed by the scan matching algorithm. The cost map is the same size and has the same resolution as the original map and each grid square is assigned a cost for traversing through it.

Something that is impassible, such as a wall, will have an infinite cost. The closer a square is to a wall, the higher the cost. From this, the system can generate what it perceives to be the safest path to a destination by finding the path that provides the lowest cost. Once a path has been selected, it is transformed into directional vectors which are sent to the onboard controls.

2.8 Interface with Imaging

Cameras send information to the path planning algorithms in the form of flags. Information from camera is treated with high priority and may change the mission state depending on the analysis results:

1. Event: Sign seen, but not recognized
   Reaction: Attempt to move into position to read sign

2. Event: Sign read, found room with USB key
   Reaction: Enter door, change state to usb search

3. Event: Sign read, found non-critical room
   Reaction: Ignore effect of sign

4. Event: Sign read, found direction to room with USB key
   Reaction: Continue exploration, but with a huge increase to cost to move in the opposite direction

2.9 Image Processing

The e-CAM20_USB camera has been implemented on MAAVerick for use in image processing. Several factors were considered in the choice of camera, including size, weight, connection type, capture resolution, compatibility, and price. Weight was especially important as the mass budget was one of the most restricting criteria during design. The camera uses a USB 2.0 connection which was chosen to avoid connection issues with the Gumstix.

Size	38 x 9.8 x 0.2 mm
Weight	6.0741g
USB 2.0	yes
Capture Resolution	640 x 480
Linux	yes
Price	$69

Figure 4: e-CAM20_USB specifications

The e-CAM20_USB is small and lightweight, and integrates easily with the electronics board.

2.10 Operation

The camera is operated through the use of OpenCV, a library of image processing algorithms. In order to increase the flexibility of the image processing algorithm, a program was written which sets the number of frames per second of video capture along with the total number of stored frames. This flexibility is crucial in order to minimize the delay from the cameras and to maximize the efficiency of the object detection algorithm. Individual frames from the stream of each camera are analyzed to check for recognizable objects (see Training for details).

2.11 Object Detection

After acquiring the captured image from the camera using the algorithm mentioned above, MAAVerick must be able to distinguish whether or not each image contains an object of interest. Our system utilizes OpenCV's Haar training method to accomplish this. Haar training uses thousands of images designated as “positive” or “negative” images to learn the distinguishing features of an object of interest. In our case, positive images are pictures of the sign and flash drive in different lighting and at various angles, and negative images are images of anything else. For good results, many positive and negative images need to be processed by the Haar training program.

2.12 Testing and Optimization

Once the program has been trained to recognize an image, OpenCV tests recognition percentage by overlaying positive and negative images. The Haar training performance program is then used to determine how many positive images the computer is able to detect. If training is done correctly, the computer should be able to detect and point out over 99% of positive images. Once a sign has been recognized, the image processing code sends a flag to the navigation software for a proper response.

2.13 Joint Architecture for Unmanned Systems (JAUS)

Based on competition guidelines, the MAAV vehicle will provide a series of time tagged status and position messages to the Judges’ monitoring station using JAUS compliant messaging. The purpose of this requirement is to familiarize competitors with this premiere standard of communication which is quickly becoming necessary in the world of UAVs.

2.14 MAAVerick Manual Control

In order to prove that MAAVerick would behave properly during autonomous flight, we first demonstrated that the vehicle could respond correctly to manually provided commands. Manual control is achieved with a six degree-of-freedom (DOF) Space Explorer mouse provided by 3DConnexion (Figure 5). Our system dynamics require a four DOF control system for height, pitch, roll, and yaw. Software provided in the 3DConnexion Developers Kit allows us to restrict control to these four cardinal commands, filtering out commands for translation in the lateral directions. The Space Explorer is programmed such that its buttons are able to enable

Figure 5 - Space Explorer by 3DConnexion used to control the vehicle remotely

MAAVerick's motors and PID Control Loops as well as reboot the onboard Atmega and cut vehicle power. Packets are transmitted from the joystick to the ground station via a USB connection and then from the ground station to the vehicle using LCM. The packets contain 4 integers corresponding to vehicle control setpoints and also 4 boolean variables corresponding to button being pressed.

3 PAYLOAD

3.1 Hardware

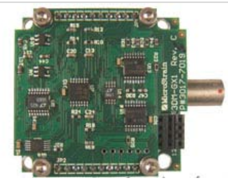
The onboard processing is performed by two units: the Gumstix Overo Fire and an Atmega 128 microprocessor. The Gumstix serves as wireless communication between the onboard sensors and the ground stations which handle much of the decision logic and image processing. The Atmega processes the inner control loops. The two devices communicate over a bidirectional I2C interface.

3.1.1 Electronics

Axi Gold 2212/26 Brushless Motor The Maaverick utilizes four brushless motors for propulsion.	 **Figure 6 – Axi Gold 2212/26 Brushless Motor**
Gumstix Overo Fire The main onboard processor, it handles all wireless communications with the ground station and relays commands to the Atmega processor onboard.	 **Figure 7 – Gumstix Overo Fire**
Thunder Power Pro Lite 3 Cell Lithium Polymer Battery The battery powers all components on the Maaverick and is rated for 6000 mAh and 11.1 V output.	 **Figure 8 – Thunder Power Pro Lite Lithium Polymer Battery**
Atmega 128 The Atmega handles the PID control loops and issues motor commands. It also controls various LED's on the board which can be used to troubleshoot.	 **Figure 9 – Atmega 128 Microprocessor**

e-CAM20_USB Three onboard cameras are used to recognize signs and find the flashdrive.	 **Figure 10 – e-CAM20_USB**

3.1.2 Sensors

Microstrain 3DM-GX1 IMU The IMU uses three angular rate gyros, three orthogonal accelerometers, and three magnetometers to output its orientation in dynamic environments. The IMU relays this information to the Atmega processor on the circuit board, which adjusts motor values accordingly.	 **Figure 11 – Microstrain 3DM-GX1 Inertial Measurement Unit**
Hokuyo URG-04LX-UG01 Scanning Laser Range Finder The Laser Range Finder scans a 270 degree area ahead of the Maaverick, and reports the angle and distance of obstacles in relation to it. The Maaverick sends this data to a path planning algorithm on the ground station, which analyzes it and returns a command.	 **Figure 12 – Hokuyo URG-04LX-UG01 Scanning Laser Range Finder**
SensComp MINI-A Ultrasonic Sensor The Ultrasonic Sensor is used for height measurements. The output is relayed to the Atmega processor, which factors height readings into the control loops.	 **Figure 13 – SensComp MINI-A Ultrasonic Sensor**

3.2 Lightweight Communications and Marshalling (LCM)

The Lightweight Communications and Marshalling library was originally developed as the message passing system for the Massachusetts Institute of Technology Darpa Urban Challenge Team. It is both platform and language independent, allowing for low latency message passing between the Gumstix on MAAVerick and ground stations via a Wi-fi network. LCM is built on a subscribe/publish structure where each device on the network can choose to receive or publish to any LCM channel, while ignoring irrelevant data, thus improving system efficiency.

4 OPERATIONS

4.1 Flight Preparation Checklist

MAAV members must perform a series of safety checks before each flight. Motors are only enabled after system checks are performed on all vital navigation sensors and programs.

- Ground station communicating successfully with Gumstix
- I2c between Gumstix and Atmega is functioning properly
- Laser scanner online
- Ultrasonic height sensor is operational
- Enable motors

4.2 Man/Machine Interface

MAAVerick's custom circuit board features eight LEDs that inform the team what the vehicle is doing at all times. With these LEDs, MAAV members are able to determine if the motors are armed, if the control loops are running, and if the onboard Atmega processor is functioning properly. It is also possible to program the LEDs to check for specific problems during troubleshooting.

5 RISK REDUCTION

5.1 Modeling and Simulation

MAAVerick was modeled in Catia V5 to determine its physical characteristics such as moments of inertia and center of mass. This data, along with motor thrust values were used in Simulink to predict vehicle response. Proper PID gain values were then empirically determined and fine-tuned from this simulation.

5.2 Flight Testing and Safety

Throughout the testing phases of our vehicle we have implemented the use of a tethering system to keep both the vehicle and team members safe. Carbon fiber propeller guards prevent the propellers from shattering on impact with a solid surface. Our system provides several fail-safe's to prevent vehicle damage and personal injury. In the case of an emergency, our motors can be killed in three independent ways: by joystick control, by a command issued from the ground station, and by a switch using JAUS Protocol.

5.3 Vehicle Status

MAAVerick vehicle status can be monitored through the graphical user interface (GUI). The GUI reports critical flight data such as vehicle altitude, battery life, and communication signal strength as well as the mission state. For example, we will know whether the vehicle is going to the next waypoint, reading a sign, or retrieving the flash drive. With this information, we can predict what issues could possibly arise allowing us to better react to a situation and reduce risk as much as possible.

5.4 Shock/Vibration Isolation

The outer ring of the airframe connects the motor attachment points and reduces the vibration generated at each location. Additionally, soft spacers have been installed between the electronics board and the airframe, protecting electronic components from unintentional vibration.

6 CONCLUSION

Michigan Autonomous Aerial Vehicles (MAAV) has made significant progress in developing a completely autonomous UAV capable of completing the 6^{th} IARC mission. The objective is to navigate an obstacle course, retrieve a flash drive containing sensitive information from an unknown location, leave a decoy, and exit the course in under 10 minutes while remaining undetected by security cameras. To complete the assigned IARC mission, a quadrotor helicopter called MAAVerick was designed and built to be optimized for payload and maneuvering capabilities. MAAVerick will use various sensors and an integrated system design capable of control, path planning, communication, and image processing. As this is the University of Michigan's first year entering the IARC competition, there are still many areas for significant improvement. In the future, continued testing and improvement will provide advances in efficiency and control in the hope of broadening the use of the vehicle in various applications.

7 REFERENCES

1. Yang, H. C., AbouSleimann, R., Sababha, B., Gjoni, E., Korff, D., & Rawashdeh, O. (2009). Implementation of an Autonomous Surveillance Quadrotor System. *American Institute of Aeronautics and Astronautics, 6*(9), 1-7.
2. Bresciani, T. (2008). *Modelling, Identification and Control of a Quadrotor Helicopter*. Lund: Deptartment of Automatic Control.
3. Hoffmann, G. M., Huang, H., Waslander, S. L., & Tomlin, C. J. (2007). Quadrotor Helicopter Flight Dynamics and Control Theory and Experiment. *AIAA Guidance, Navigation and Control Conference and Exhibit*, *20*(23), 1-20.
4. A. Segal, D. Haehnel, S. Thrun. "Generalized-ICP," Robotics: Science and Systems, 2009.
5. USB Camera Board - 2 Mega Pixel USB Camera : Autofocus USB Camera. (n.d.). *Low Power Computer on Module | Product Design Services | Embedded XScale PXA270 | Carrier Boards*. Retrieved June 1, 2010, from http://www.e-consystems.com/2mp_cameraboard_usb.asp
6. Point Grey - Imaging - Firefly MV CMOS USB 2.0 / FireWire Camera. (n.d.). *Point Grey CCD and CMOS Digital Cameras for Industrial, Machine, and Computer Vision*. Retrieved June 1, 2010, from http://www.ptgrey.com/products/fireflymv/index.asp
7. IDS: UI-1410SE USB 2.0 camera: datasheet - manual - software. (n.d.). *machine vision:firewire,usb,gige,cameralink,ccd,cmos cameras,framegrabbers,lenses,software*. Retrieved June 1, 2010, from http://www.1stvision.com/cameras/IDS/UI-1410SE.html
8. ICECAM Macally Peripherals. (n.d.). *Macally*. Retrieved June 1, 2010, from http://www.macally.com/EN/Product/ArticleShow.asp?ArticleID=119
9. Damiles. (n.d.). Damiles » OpenCV Seting and working with a camera. *Damiles*. Retrieved June 1, 2010, from http://blog.damiles.com/?p=61
10. OpenCV: Basic Structures and Operations Reference. (n.d.). *Batiment Nautibus (ex 710), Universite Claude Bernard Lyon 1*. Retrieved June 1, 2010, from http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef_BasicFuncs.htm
11. Tutorial: OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features) - Naotoshi Seo. (n.d.). *Naotoshi Seo*. Retrieved June 1, 2010, from http://note.sonots.com/SciSoftware/haartraining.html
12. Object Detection Using Haar-like Features with Cascade of BoostedClassifiers. (n.d.). *Naotoshi Seo*. Retrieved June 1, 2010, from http://note.sonots.com/SciSoftware/haartraining/document.html
13. lcm - Project Hosting on Google Code. (n.d.). *Google Code*. Retrieved June 1, 2010, from http://code.google.com/p/lcm/