Dynamic and Reactive Flight for Autonomous Aerial Vehicles

Tommy Dong
Missouri University of Science and Technology - Chief Executive Officer

Piotr Pogorzelski
Missouri University of Science and Technology - Chief Technical Officer

Cameron Falls
Missouri University of Science and Technology - Vision Subteam Lead

Michael Pieper
Missouri University of Science and Technology - Chief Software Engineer

June 1st, 2021

ABSTRACT

A strategy is presented for a dynamic and reactive flight for autonomous aerial vehicles in GPS-guided outdoor environments. The flight system, in addition to the use of computer vision and module interaction hardware, is intended to accomplish the tasks as required by the ninth mission of the International Aerial Robotics Competition. This strategy outlines the design of an aerial vehicle that will be capable of high speed, long-range flight, precise module manipulation, and on-board processing for the computer vision algorithms necessary for the successful completion of the mission.

Introduction

Statement of Problem

The ninth mission of the International Aerial Robotics Competition introduces several challenges for UAV design. The vehicle in practice must be capable of flying long range, at a fast rate of speed, have a moderate payload capacity, traverse using precise guided/unguided navigation, perform obstacle avoidance, implement efficient computer vision, perform precise manipulation on a dynamic target, be reliable, have redundant safety mechanisms, be cost efficient, and most importantly be fully autonomous in its actions.

Conceptual Solution to Solve the Problem

Figure 1. 3D model of competition aerial vehicle. The claws located in the front are meant to remove the existing antenna module from the mast.

Main Hardware Utilized

- Pixhawk Cube Orange Autopilot
- Pixhawk Kore Carrier Board
- Nvidia Xavier SOC
- Intel Realsense D415 Camera(s)
- T-Motor V605 210KV Motors
- T-Motor Flame 80A 12S ESCs
- T-Motor 22” CF Propellers
- Tattu 12S 22000 mAh LiPo Batteries
- Actuonix 300mm Linear Actuator
- RFD900x 915MHz Telemetry Radio
- FrSky R9M 900MHz Radio
- 25kg Electromagnet
- TE Aerospace 500A Contactor
- Here3 GNSS GPS w/ RTK

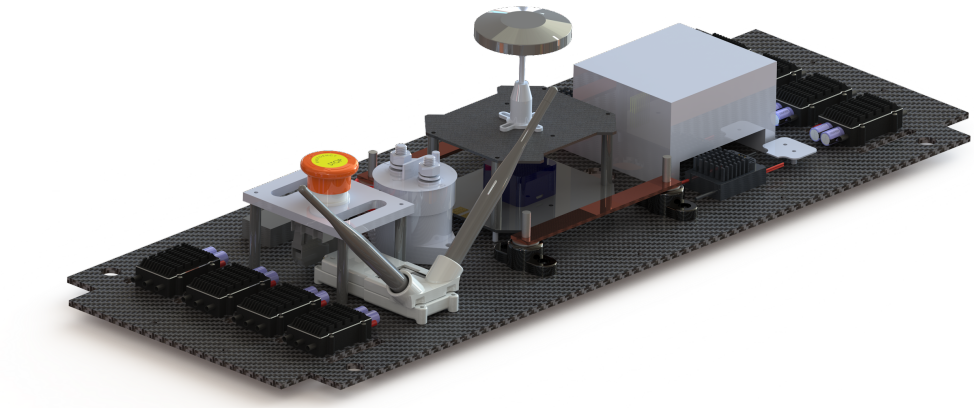

Figure 2. Model of layout of onboard electronics

Air Vehicle

Description of Configuration/Type

The platform designed by the Multirotor Design Team primarily takes form as a stretched H-frame style coaxial octocopter. The platform measures approximately 1.65m wide, 1.2m (retracted) / 1.5m (extended) in length, and .475m in height. The unloaded weight, no module or batteries, is 13.6kg and has the ability to accept either one or two 12S 22,000 mAh Lithium-Polymer batteries as its energy source. The single and dual battery configurations weigh 18.5kg and 22.5kg respectively. This, in combination with the 2kg payload module, brings the AUW (all up weight) of the vehicle to just under 25kg (55lb) in its heaviest configuration, which falls below the FAA maximum weight limit for UAV platforms without the need for special waivers.

The aircraft is powered by eight electric 210KV 7040 brushless motors, each paired to a carbon fiber 22x7.2 (bottom) or 22x6.6 (top) propeller. The 12S LiPo batteries allow for a maximum system thrust of 90kg (198lb) which allows the platform to maintain a very efficient ~4:1 thrust to weight ratio, even in its heaviest configuration. This enables the aircraft to fly for approximately 18 minutes (single battery) and 25 minutes (dual battery) under moderate load. The maximum speed achieved with this motor, frame, propeller, battery combination was ~55mph, with the theoretical maximum being 70mph.

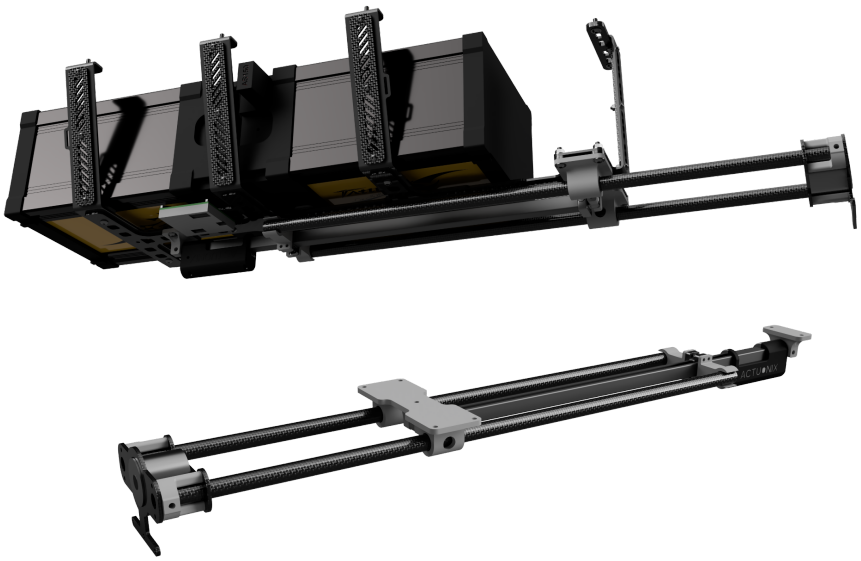

Figure 3. Model of linear actuator and battery mounting system (located below electronics bay)

The frame itself is primarily carbon fiber composite with a mixture of aluminum and steel as necessary. High modulus glass-fiber reinforced square carbon fiber tubing (1”x1”) creates the main underlying structure of the frame and allows for easy assembly when compared to round tubing. These lengths of tubing are held together using CNC machined 6061 aluminum slip-joints and carbon fiber gusset plates, which are then fastened by varying combinations of 5mm/ 6mm steel shoulder bolts, rivnuts, and locknuts throughout the chassis as needed.

All of the main avionics are housed within the upper shelf which is protected by a carbon-fiber canopy with air-ducting to promote cooling for the high-current ESCs, on-board computing, power management, and flight control systems. The electrical system is also designed to be as modular as possible to allow for simple replacement of parts in the event of faults or failure. This is accomplished by using properly rated keyed connectors throughout the wiring harness which have been placed purposefully for easy access.

Flight Control System

Navigation/State Estimation System

The navigation system relies on the Pixhawk Cube flight controller for the lower level control of aerial vehicle flight. Higher level autonomous guidance is provided using an onboard computer (either the Intel Up Board or Nvidia Jetson TX2) using the Python-MAVSDK library that is easily accessible to those who are new to autonomous

flight programming. The two computers communicate with each other through a serial connection for maximum integration and reliability.

The autonomous flight software is modeled by a finite state machine that splits the competition tasks into various states for the aerial vehicle. This structure is advantageous because it allows common behavior to be shared by multiple states, while also providing a self-contained set of instructions for each individual state.

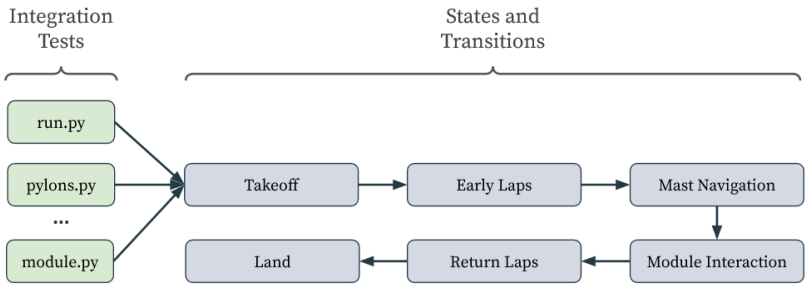

Figure 4. State machine diagram

Each integration test has control over the operations performed in each state. This enables rapid testing of specific segments of the mission, while maintaining consistent state transitions across all tests.

Multiple flight operations utilize GPS navigation and altitude detection with margins of error for arriving at the target. These include taking off, moving to the pylons, flying to the mast, and returning to land. The system for making laps around the pylons uses a combination of GPS and rotation calculations. GPS guides the drone along a straight path between the pylons with a five-meter offset. Once the vehicle reaches the offset position from the pylon, it begins its 180-degree turn at a constant radius. To achieve this, the drone first subscribes to attitude updates to continuously get the vehicle's yaw from the telemetry throughout the turn. The vehicle is then given a body velocity and yaw angular rate, which allows it to maintain a constant radius. The regular telemetry updates allow for repeated calculation of the drone's current angle in the turn. Once that angle is sufficiently close to the goal of 180 degrees, the turn terminates, and the drone flies to the other pylon via GPS. This process repeats for each of the laps around the pylons.

Attitude/Position Control System

A large portion of the position control system is achieved with the Pixhawk Cube autopilot system and GPS navigation due to the outdoor nature of the competition. In

order to minimize GPS drift, an RTK (Realtime Kinematic) system can be optionally used in order to get sub-centimeter accuracy by functioning as a ground beacon to assist the GPS positioning. This system is quite complex to set up, is only supported by the newest of the Pixhawk autopilot hardware, and in practice only guarantees about 15 centimeters of accuracy after 8 hours of surveying, so the team may not end up using that system in competition.

Flight Termination System

In terms of safety, the system features a three-stage kill procedure. If the aircraft is on the ground then the operator may opt to push an E-STOP (emergency stop) switch which is clearly marked near the rear of the top shelf protruding through the canopy. This directly triggers the 500A rated contactor which effectively disconnects the positive lead of the battery from the main motor supply circuit. This safely and quickly disables the propulsion system while also leaving other electronics still online to continue logging and maintain backup control if needed. The operator may also invoke a software kill switch which will disable the propulsion system until the switch is returned to safe-state and then the aircraft is rearmed, this is handled through the flight controller firmware. However, the main method for disabling the aircraft remotely is through the use of a separate long range 900MHz transmitter, with the sole responsibility of triggering a relay which will then trigger the same 500A main contactor mentioned earlier, resulting in safe termination of flight in the event of an emergency. These three methods can be used in varying combinations depending on the situation and the degree of severity and provide multiple independent redundant methods of effectively terminating a flight or otherwise.

Mission Package

Perception System

The main perception system involves two Intel D415 Realsense cameras: the main camera, which is mounted to the front of the vehicle, and a secondary camera mounted on the linear actuator of the vehicle. The main camera will be used for identification of the mast as the vehicle approaches the general location, while the secondary camera will be used to position the vehicle more accurately to the mast. Computer vision algorithms will be run on the onboard computers to detect the presence of the defining features of the mast and module.

Target Identification and Behavior

Once the vehicle approaches the approximate location of the competition mast, the onboard computer will run computer vision algorithms in order to detect the various defining features of the mast. Most notably, these algorithms detect the cyrllic text that is displayed on the mast, as well as the holes that are present on the antenna module. Several Python libraries assist in these computer vision algorithms, including OpenCV, NumPy, and PyTesseract.

In order to detect the cyrillic text located on the mast, Google's Tesseract OCR engine is used with the PyTesseract library for integration with Python. Tesseract's language libraries allow the vehicle to recognize the text in a frame and filter for the given string in order to eliminate noise. Since Tesseract can have difficulty detecting angled text, preprocessing is done on the image before attempting to detect the text. First, the frame is blurred and masked for the color blue, the color of the plate surrounding the text. Pixels that are not considered blue are set to white, while blue pixels are set to black. The depth data from the camera is then used to eliminate parts of the image that are far away. This is followed by edge detection to create an outline of the mast plate. Then, the algorithm finds the largest contour in the image and uses it to construct a rectangle, which covers the plate in the image. Next, the algorithm slices this rectangle from the image and rotates it so that the text is level. The algorithm then runs the Tesseract OCR[2] engine on this sliced image. Since the preprocessing makes the text horizontal instead of at an angle, it increases the likelihood of Tesseract detecting the text. The final stage of text detection is to iterate through the strings detected by Tesseract and filter for the given words. From the positional data given by Tesseract, the algorithm constructs bounding boxes for the words by remapping the vertices to the unsliced image, allowing the vehicle to identify the words in the original image.

Figure 5. Text detection example

A series of computer vision algorithms is run to identify the module, as well as its orientation in the frame. The first algorithm identifies the center of the module in a frame, consisting of two stages. The first stage determines whether the module is in the frame. The second stage calculates the center of the module in the frame. The main identifiable feature that these algorithms use to identify the module is the four holes located on its front face. Using circle detection, we can reliably recognize the holes and use their positions to calculate the location of the module. To ascertain whether the module is in the frame, the algorithm first performs preprocessing on the image. This preprocessing does a weighted brightness increase that pushes the median brightness to the middle of the color spectrum to make circle detection more accurate on dark images. Next, the image is grayscaled and blurred to eliminate noise. Finally, the algorithm conducts a Laplacian transformation for edge detection. The preprocessing is followed by Hough circle detection to identify the holes. During testing, it was discovered that Hough circle detection is sensitive to noise, so additionally post-processing of the detected circles is used to increase confidence that the holes have been identified.

This post-processing algorithm eliminates circles that are far away using the depth image and uses the text bounding boxes to eliminate the circles that are not below the text relative to the text's angle. Then, a clustering algorithm is used to group the circles based on position. Since the four holes on the module are close to each other, clustering can be used to eliminate noise circles and select circles as the holes. The algorithm performs repeated K-Means clustering until compactness is sufficiently low, meaning there is a low variance of distance within the clusters. Clusters that do not have at least four circles are discarded since these cannot be the holes. A final metric used to identify the holes is whether the circles are correctly aligned. Since the four holes are arranged in a square, the algorithm can determine the slopes between the centers of the holes to identify structure. Only a cluster that has a sufficient number of parallel slopes can be the holes. The algorithm gets the slopes between the centers of every circle in each cluster. Within each cluster, a bucket sort is performed on the slopes to determine which slopes are parallel. The Freedman-Diaconis Rule is used to calculate a width for each bucket, which can be used to determine the number of buckets over the range of slopes in a cluster. The most structured cluster is then selected as the cluster containing the four holes. At this point, if the best cluster contains a sufficient number of slopes, the module is considered to be in frame. If the module is in frame, the algorithm goes on to calculate the center of the module.

$$width = 2 * \frac{IQR}{\sqrt[3]{n}}$$

Figure 6. Freedman-Diaconis Rule[1]

To calculate the center of the module, the four holes are found from the slopes in the best cluster. The slope value with the most associated parallel slopes is used to find the circles in the cluster that contributed to it. These circles are identified as the module holes. Once the holes are found in the array of circles, the positions of these holes are averaged to calculate the coordinates of the center of the module. Then, the relative depth from the vehicle to the module is calculated using the center and the depth information in a radius around the center.

In the next phase of the module algorithm, the orientation and bounds of the module are determined. By using the calculated center, the given constant values of the actual dimensions of the module, and the relative depth value of the module, the algorithm calculates vertices for a region of interest. The region of interest is a sliced portion of the frame using an underestimate of the bounds of the module. Using the depth information in this region, the algorithm calculates x and y angles for orientation. The bounds of the module are found in the same way as the region of interest, except an overestimate of the bounds is used. These bounds are used to construct the module bounding box and to get the z orientation angle. The z angle is found using contours in a sliced image of the bounds. The angle of rectangle contours is used to find the z angle orientation. The complete module bounding box is constructed using the calculated module bounds and includes the x, y, and z orientation angles, as well as the relative depth of the module.

Threat Identification and Behavior

The primary threats of the Mission 9 competition are mostly restricted to the environment itself, where the two pylons are static, and the mast is the primary dynamic object in the scene. The GPS navigation should ensure that the vehicle does not come too close to the pylons, and the vehicle should be prepared to track the mast itself and maintain the necessary distance to do module interaction at a safe distance.

Communications System(s)

Due to one of the primary requirements of this competition to mostly use on-board computing, most of the wireless communication systems are for the name of safety

rather than for competition-related tasks. Listed below are the main communication systems.

- RFD900x long range telemetry radio w/ ppm passthrough for pilot control.
- FrSky R9M 900MHz long range radio for the kill switch
- WiFi transceiver for ground station to on-board computer
    - In order to start the vehicle or send an emergency landing command
- USB busses between other devices
    - In order for communication between the sensors, flight controller, and onboard computer

Risk Reduction

Vehicle Design

There are various mechanisms built into the aerial vehicle in order to maximize the safety between the vehicle and its operators. The first safety feature to note is the emergency stop button. It is located on the top level of the vehicle, and once pressed, the button will need to be twisted in a particular direction in order to be reset. This E-stop system kills power to the motors and can ensure that the blade will not spin up.

While the vehicle is in the air, the operator will be able to use a passthrough mode through the telemetry radio to manually take over the vehicle, as well as enable the kill switch through a seperate radio if manual takeover is not possible. There will also be a link from the onboard computer to a ground station for the sole purpose of executing a stop command to the onboard computer to cease autonomous flight and initiate landing mode.

Due to the experimental nature of the competition and the team, the autonomous software is split up into states, such that the team can specify and modify parameters to dictate whether or not the vehicle attempts to run certain procedures, such as the difficult module interaction procedure. This enables the team to be more flexible and safe in competition runs, even if it means forgoing certain competition tasks for this year.

EMI/RFI Solutions

The Pixhawk Cube provides EMI/RFI solutions for built in protection for basic sensors. This is achieved with metallic shielding and a high insertion loss filtering system onboard. Other than shielding, the other methods of EMI reduction the team uses are to separate the power system from the the signaling system, keep all wire runs as short

as possible, use grounding as necessary, include coupling / decoupling capacitors on all power/sensor inputs, and keep sensitive electronics as far away as possible from possible interfering sources.

Shock/Vibration Systems

The Pixhawk autopilot features great internally calibrated vibration damping and filtering to account for common vibrations. Extra vibration on the flight controller itself is not recommended nor is it very effective. However, mechanically, vibration on fasteners poses a serious safety hazard. This is mitigated by using nylon insert locknuts, loctite, and by properly torquing fasteners to ensure that they don't become loose during flight.

Safety

The design features mechanical safety margins in excess of 1.25 in order to maintain the balance between weight, performance, and structural integrity. The main safety features are handled within the flight termination operation and within software.

Modeling and Simulation

A significant portion of the structural modeling was created using SolidWorks and Fusion360. The two modeling softwares allowed the team to create base designs and consider approximate sizes, weights, and ease of construction, which was essential in creating the balance between power and weight. It also allowed the team to visualize and compare various designs. From there, the team can run simulations and structural analysis on the final models.

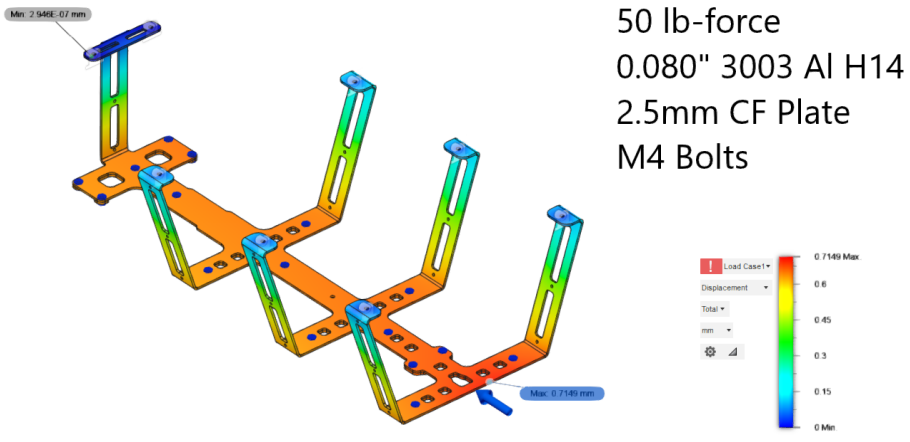

Figure 7. Finite element analysis for battery and linear actuator mount (for module interaction) under large forces to simulate forces at high speeds.

Further simulations of the autonomous flight were run under jMAVSim, which helped the team capture expected and unexpected behavior before running physical tasks. This ended up saving the team a lot of time due to the logistics required to travel to and secure the outdoor testing area. While the team this year did not have the knowledge and members to build a fully fledged simulation system for module interaction, full competition simulation is definitely one of the top priorities for the team moving forward.

Physical Testing

Due to the outdoor nature of the competition, and the relatively high speeds that the team hopes to achieve for the competition, it is essential to find an optimal testing location that was also out of harm's way for the general public. With approval from the Environmental Health and Safety division from the university, the team procured a large outdoor area that is grassy and mostly flat, with trees surrounding a large patch of open land.

The team continues to abide by the procedures that were set for the previous competition, ensuring that all the necessary members and part-107 certified pilots were at the testing scene to take over in times of emergency. With a decent amount of physical testing being done throughout the year on a testing platform, the team members will be prepared to respond to any significant and undefined behavior when flying the actual competition platform at IARC.

Conclusion

With the use of an improved autonomous flight algorithm, computer vision, and a custom-hardware design to support powerful motors, the Multirotor Design Team has a solid foundation and strategy to begin tackling the basics of Mission 9 of the International Aerial Robotics Competition. By creating a faster turning algorithm paired with powerful motors, the aerial vehicle should be able to travel a long distance with capable speeds around the competition pylons, and the computer vision algorithms will be essential to provide the necessary information required for the

References

[1] Wikimedia Foundation. (2021, March 21). Freedman–Diaconis rule. Wikipedia. https://en.wikipedia.org/wiki/Freedman%E2%80%93Diaconis_rule.
[2] Tesseract User Manual. tessdoc. (n.d.). https://tesseract-ocr.github.io/tessdoc/#introduction.