Vulture [Prototype]

Overview

A Vulture core prototype is a prototype that would be capable of being controlled remotely and basic collision avoidance. The meaning of the word “core“ in the previous sentence refers to the modularity of the Vultures(i.e there would be 2 types of cores capable of supporting different payloads for a wide variety of tasks). This prototype would be useful for several reasons:

- test the capabilities of the system
- solve any potential issues that might arise during testing
- prove the capability of the team to build both the software and hardware required for this product to work
- use test data to rapidly iterate over different hardware configurations
- test the ability of the Vulture to use cell towers as a means of connectivity
- test the physical capabilities of the Vulture(e.g max flight time, wind resistance)
- figure out if the current internal architecture of the Vulture is capable of delivering the features that the Vulture will need to provide

Since the primary purpose of this prototype is experimenting with different hardware configurations, minimalizing its cost would be optimal.

Testing and Development

To prevent losing hardware as much as possible, the earliest Vulture prototypes will have a minimal sensor array in the first stages of testing. Hardware testing should be done in stages to validate the functionality of critical systems of the Vulture before starting to integrate additional hardware that is not required for the Vulture to be flown manually. The systems of the Vulture that are part of the first stage of testing and must successfully undergo integration tests are:

- propulsion
- power
- communication
- flight control

After the Vulture is proved to be stable, additional hardware can be integrated for real-world testing. The system with the highest priority after the first stage of tests has been completed is the autonomy system. Since one of the most important features of the Vulture is for it to be able to complete fully autonomous missions that require zero human interaction, it is optimal to start developing this feature as soon as possible.

Hardware

Ideally, the hardware used to build Vulture core prototypes would be similar if not the same as the hardware that is planned to be used in production. This would ensure that the integration and software developed would function as intended with minimal or no changes to the code base.

Integrated hardware so far

Sensor Array

Several sensors have already been integrated into a primitive sensor array of a Vulture prototype. These tests proved the ability of the Global Relay to relay the live telemetry to the browser/app. Some of the sensors are considered to be ready for flight while others (e.g the IMU) need additional calibration before they’d become usable by the Vulture. The following table will contain a list of all the sensors that have been attempted to be integrated with this sensor array:

> The hardware in the list below has been chosen for its local availability as well as compatibility with the Johnny-Five API used to test the prototype sensor array. All the parts listed in the table below are intended to be used in prototypes until they are replaced with potential production sensors. These sensors were also used to test the different features of the user interfaces that display the live telemetry coming from the Vulture.

Name	Type	Integration Status	Description
MPU9250	IMU	Calibration Required	The inertial measurement unit(i.e IMU) is used to provide the rotational and acceleration status across all the axis required for flight stability and full autonomy. Ideally, the production IMU used would have a built-in gyroscope, accelerometer, and magnetometer for better accuracy
AXDL345	Accelerometer	Completed	The accelerometer is used to counter the shift of the gyroscope included within the IMU as well as to provide a backup in case both IMUs would fail. The probability of both production-grade IMUs failing at the same time is extremely low but considering how critical it is for the Vulture to always have an accurate picture of its rotational status, it is considered a good idea to have a backup for the backup
NEO-6M	GPS	Completed	The GPS module is used to get the location of the Vulture as well as other navigation-related telemetry such as the heading, altitude, and speed of the Vulture
HC-SR04	Sonar	Completed	This sonar was used only for testing the UI and the production sensors that would provide the Vulture with the collision avoidance abilities are yet to be chosen(most likely some type of ToF sensors would be used for a good cost/accuracy ratio)
BMP280	Barometer(used as altimeter)	Completed	The barometer/altimeter is the sensor that provides the current altitude of the Vulture by measuring the ambient atmospheric pressure
QMC5883	Magnetometer	Failed	The magnetometer is used to get the magnetic heading of the Vulture for both manual flights and missions carried out fully autonomously

Propulsion System

Multiple attempts at a functional part(i.e a single motor, ESC, and propeller) of the propulsion system have partially failed so far. The only aspect of the system that has been fully tested and that's working as expected is the manual propulsion controls that can be used to test the motors individually on the ground. The partial failures that were encountered are due to compatibility issues between the parts in the propulsion system. The table below will contain the current hardware used and which part is most likely to be not compatible with the current setup.

Name	Type	Integration Status
Arduino Uno	Microcontroller	Completed
Tattu Fun Fly 3s 11.1V	Battery	Completed
Turnigy Multistar 3508-580	Brushless motor	Completed
Turnigy Multistar 40A	Electronic Speed Controller	Completed
Quanum Carbon Fiber T-style propeller 18x5.5	Propeller	Failed

The first attempt at a core propulsion test failed because of a compatibility issue between the motor and ESC. After the ESC was replaced, the motor was behaving as expected, until the propeller is mounted. The current working theory is the propeller used in these tests is too long for this motor to be able to spin at the expected speeds. Replacing the current propeller with a shorter one should fix this issue. The purpose of this propulsion test is to get real-world data regarding the performance metrics(especially thrust) of this system before buying the rest of the hardware required for a Vulture prototype to fly.

Software

Flight Computer and Controller

Two major components define the signal that controls the motors:

- the flight computer which is constantly computing the target pitch, roll, and yaw the Vulture should have based on user input and the autonomous system(this includes collision avoidance and other features such as target tracking)
- the flight controller which converts the target pitch, roll, and yaw into signals usable by the ESCs that control the motors

The flight computer will have to be custom developed since it requires tight integration with the autonomous system. This will most likely require some trial and error but it's achievable within an acceptable timeframe. The most important feature the flight computer must provide is redundancy. Even if the hardware will also have backups and be configured in a way to minimize the risk of a catastrophic failure, the flight computer also needs to be highly resilient to different disruptions that might affect the system. Such examples are:

- flight non-essential hardware failing mid-flight
- losing connectivity to the Global Relay/local radio emitter
- failure of the autonomy system
- erroneous telemetry coming from critical sensors(e.g IMU)

Testing for all different types of failures, even the ones that would never happen in real flights is necessary to prove the ability of this system to maintain safe and reliable flights.

It would be best to use a flight controller from a 3rd party considering the time and resources required to build one from scratch. This is considered optimal at least in the beginning especially since the input processing required to get to smooth and predictable control of the motors is fairly complex and the resources needed to get to that point are of far greater use if they're directed towards the autonomy system.

Status of the internal architecture

The current internal architecture of the Vulture is still in an early stage of development where the code used to interface with hardware is running within a single process instead of being split into different controllers that would interface with different parts. The idea is to have different processes for every primary microcontroller used between the actual hardware and the flight computer. This would prevent crashing the whole flight computer if a single process-ending error would occur. The only exception to this is the propulsion system that should always run within the same process to prevent de-synchronization issues. This would mean the process would need to be executed with something like `pm2` to take the downtime of this process as close to 0 as possible. Extensive software stability tests need to be completed before the official launch of the Vulture and the backup method that involves `pm2` exists only to provide a path to mid-flight recovery of the flight system if a flight computer crash would occur.

Flight-ready prototype tasks overview

> ⚠ The graph below is not an actual timeline, but rather a representation of the order the tasks required to get to a flight-ready Vulture prototype would be completed in.