First Steps Towards Developing an Autonomous Quadrotor for Mission 7 of the 2014 International Aerial Robotics Competition

Jean-Alexandre Barszcz,
B. S. E. Computer Engineering 2017
Alexis Cabana-Loriaux,
B. S. E. Computer Engineering 2017
Quentin Gilli,
M. S. E. Industrial Engineering 2014
Antoine Mignon,
M. S. E. Electrical Engineering 2015
Andre Phu-Van Nguyen,
B. S. E. Computer Engineering 2015
Constant Riestch,
M. S. E. Industrial Engineering 2015
Marc-André Ruel,
B. S. E. Software Engineering 2015
Antonio Sanniravong,
B. S. E. Electrical Engineering 2017

ABSTRACT

Although remote controlled piloting of multi-rotor vehicles has become prevalent and largely accessible to the general public and research institutions, these systems are largely dependent on manual control and external navigation aids such as GPS, high speed motion capture systems and SLAM algorithms relying on walls as a reference point. Team Elikos from Polytechnique Montréal presents the first steps towards solving mission 7a of the International Aerial Robotics Competition including design, development methodology and results. Through rapid prototyping using off-the-shelf components, theoretical analysis and experimental data, we demonstrate a partial solution wherein our quadrotor is capable of autonomous takeoff, position control by optical flow and computer vision of the arena.

INTRODUCTION

Statement of the Problem

Mission 7a of the IARC involves the creation of an aerial vehicle, no larger than 1.25 meters in its longest dimension, able to interact with ground robots, dodge moving obstacles and navigate within a GPS denied arena. A set of preprogrammed ground robots roam the arena each carrying Hall Effect sensors and pivot whenever this sensor is triggered. Teams must redirect the robots towards a zone marked with green adhesive tape by approaching a magnet to trigger the sensors all the while dodging vertical PVC tubes mounted on similar robots hereinafter referred to as obstacle robots.

Conceptual Approach

This year's objective was to have a vehicle capable of autonomously taking off, hovering while doing image analysis and landing at the end of the competition's allotted 10 minutes. Obstacle robots are to be ignored and taken out of the equation by maintaining an altitude between 2 and 3 meters.

A quadrotor design was chosen due to the overwhelming amount of documentation and open source projects surrounding it. The vehicle is equipped with an autopilot for flight navigation, a camera for arena line and robot detection, a second camera for optical flow calculations and an embedded computer for wireless communications and superficial image processing. The autopilot has an isolated wireless link to a laptop on the ground which transmits attitude and position commands. Because of the high bandwidth requirements of the video stream, the embedded computer sends the stream over a separate Wi-Fi signal to a ground station for analysis. At the beginning of the challenge the vehicle is hard coded to liftoff, enter the arena and hover at the same position until the allotted time is up. The following figure presents our current solution to the mission.

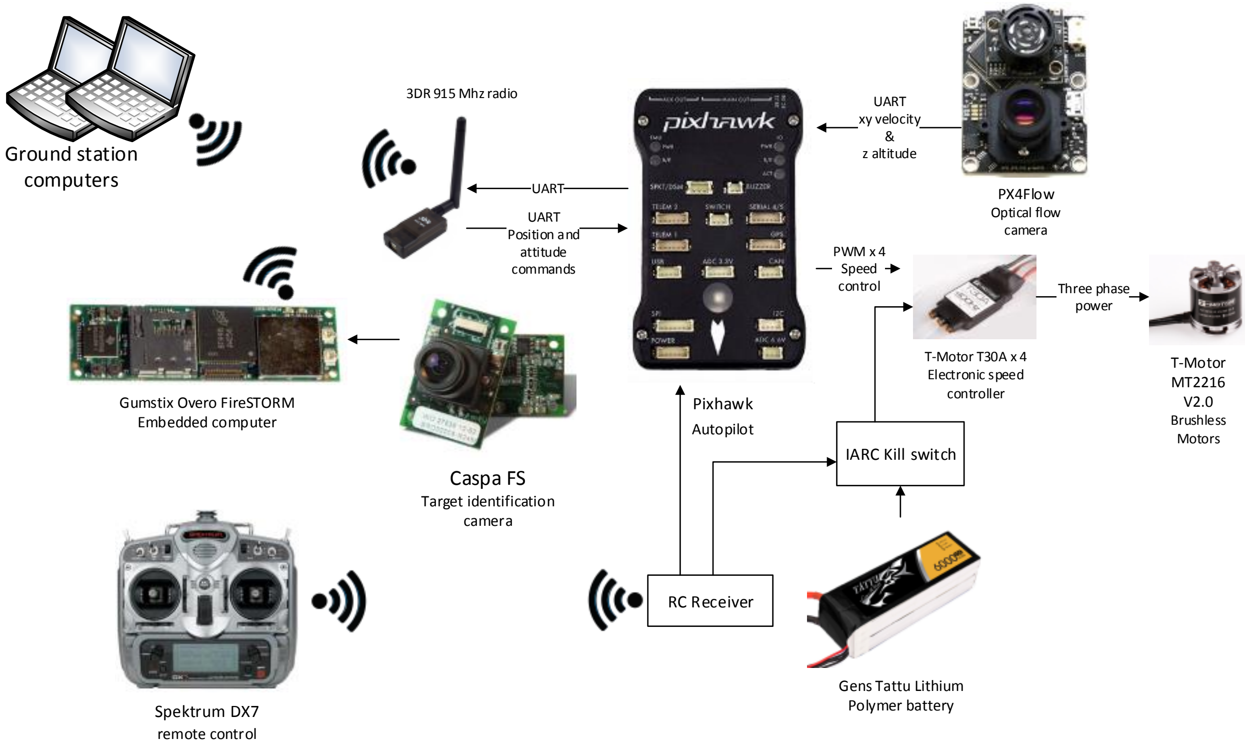

Figure 1 Hardware interactions within the Elikos system

Yearly Milestones

Elikos entered its first year of operations in November of 2013 and devised a plan to quickly get up to speed with teams having multiple years of experience under their belt. Since the IARC has decided of a minimum of three years before moving to mission 7b, the first year was focused on assembling a prototype following tried-and-true concepts and technologies as well as ensuring the safety of the development team through safety procedures and secure testing environments. The following figure presents our annual plan.

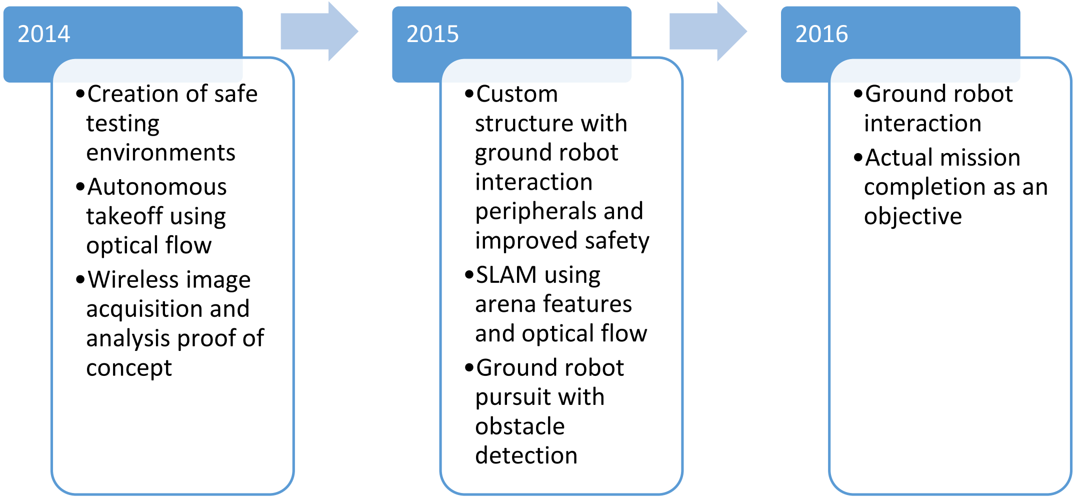

Figure 2 Annual milestone plan

AERIAL VEHICLE

Structure of the UAV

Following our vision of rapid prototyping using off-the-shelf components, the Hobbyking X650F was chosen for its aluminium arms capable of surviving crash landing during flight testing. The choice stems from a three step design process wherein we first determine the required payload including flight controller, embedded computer and cameras. Once an initial weight estimate is obtained, the battery, speed controller, motor and propeller quadruplet is chosen to accommodate the required payload and flight time. Finally the propeller size finalizes the requirements for arm length and gives a rough estimate of which frame can be purchased.

Although the X650F provides a great testing platform, its weight has proven problematic for flight times and is being replaced by the Turnigy Talon V2 with a better size to weight ratio. The V2's simplicity has also allowed us to replace part of it with custom sheets of carbon allowing freedom in component placement to respect security constraints and spacing margins relative to the ground and propellers despite the large quantity of components onboard.

Propulsion and Lift System

The vehicle is lifted by four MT2216 V2.0 brushless motors from T-motor mounted with 12 inch 2-blade propellers. Two kinds of propellers are used depending on the circumstances. Landing Products has provided the team with composite propellers integrating fiberglass with nylon allowing for greater tensile and flexural strength compared to regular nylon [1] thus these are the ones used for testing purposes. T-motor has also provided carbon fiber propellers with the same length and a similar pitch, however the potential danger of the strength of these propellers as well as their high cost has put off their use until competition time.

Guidance, Navigation and Control

Stability Augmentation System

The Pixhawk autopilot was chosen because of the advanced state of the inertial navigation system it provides and because of the open-source and open-hardware nature of the platform. The latter, which is known as PX4, makes it a possibility to develop new custom features relatively easily as the need arises. This is mostly thanks to the embedded real time operating system (RTOS) NuttX, which is also open-source and POSIX compliant. NuttX allows for low and high level programming with its integrated C and C++ libraries and for high interfaceability with peripherals with its versatile device driver library. Altogether, the PX4 environment proves itself to be an interesting development platform for the current state of technology in unmanned aerial vehicle control.

Navigation

A quadrotor possesses 6 degrees of freedom: 3 in translation and 3 in rotation. As such a quadrotor can move in altitude (z) and in position (x,y) according to a configuration in roll (φ), pitch (θ) and yaw (ψ) representing its attitude in space. Internally the North-East-Down coordinate system is also used to represent x, y and z. Vehicle displacement and attitude control depend on rotation speed of the motors.

Our UAV is configured in an X where all motors participate in the displacement motion contrary to a classic + configuration. Displacement in altitude (z) is done by applying the same variations of speed to each motor while translation within the plane (x, y) are done when a pair of motors has a different angular speed than that of another. This same principle is applied for yaw variations.

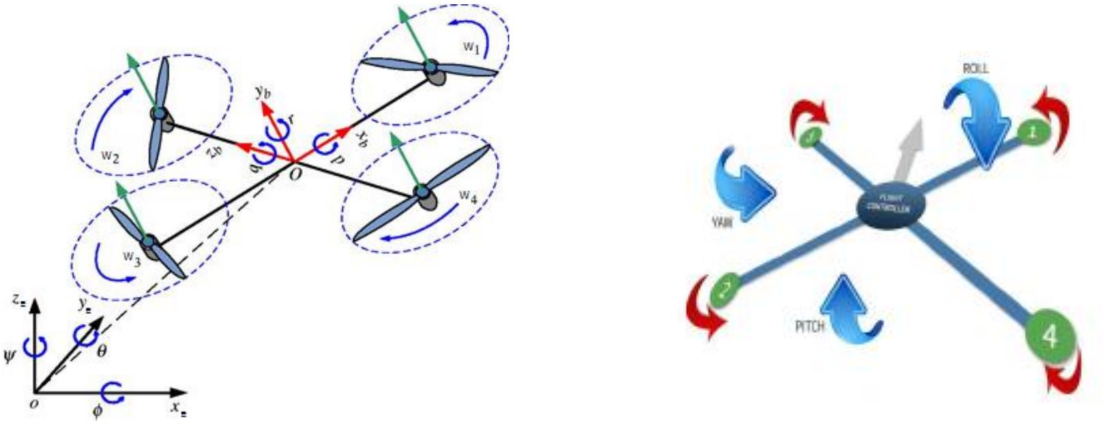

Figure 3 (Left) A quadcopter's possible reference points. (Right) A quadcopter in an X configuration

Since the current state of the Pixhawk firmware does not fully support offboard control for navigation without GPS, we use a customized firmware version that can accept local position set points. The customization is done on the "Position Controller" process that originally cannot handle other instructions than manual input from a radio or GPS waypoints. Since we intend to have our quadrotor hover statically over a fixed point in the arena, an advanced navigation system will not be required. Only a basic approximation of the position obtained by integration of optical flow will be used to navigate to the destination point.

Control system architecture

In the following diagram, we can observe the Pixhawk's software main loop which runs at a frequency of 500Hz. Each part of the exploded pie chart represents an internal process. The closer a process is to the center, the higher its priority. All the processes from "Sensor Read" to "Actuator Output" are the most time critical processes as they represent the essential software counterpart of the IMU and thus need to be run at a higher priority to ensure uncompromised control on the vehicle's attitude.

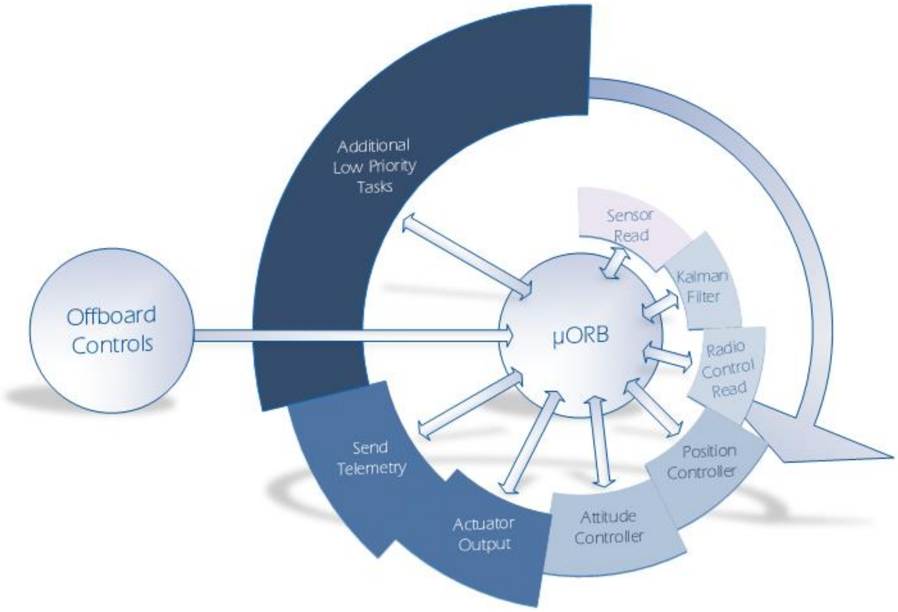

Figure 4 Graphical representation of the Pixhawk's various processes

As may be clear from the previous figure, all of the processes are interdependent. Each process' output is used to provide relevant input for one or many of the following processes either in the current or future iterations of the main loop. This is achieved with the Micro Object Request Broker (µORB) process which follows the simple publish-subscribe software design pattern. µORB makes it possible for processes to subscribe and publish data through different channels called topics. The µORB also takes into account the update rate of each topic with a timestamp system so that each process subscribed to these topics can verify the validity and usefulness of the available data.

Flight Termination System

The kill switch is based off the official IARC design with a slight modification to the choice of MOSFET to accommodate higher current ratings.

PAYLOAD

Sensor suite

Guidance, Navigation and Control Sensors

The hardware provided within the Pixhawk autopilot consists of an inertial measurement unit (IMU) coupled with an array of input and output ports for external peripherals. The following diagram illustrates the internal hardware components of the Pixhawk by function.

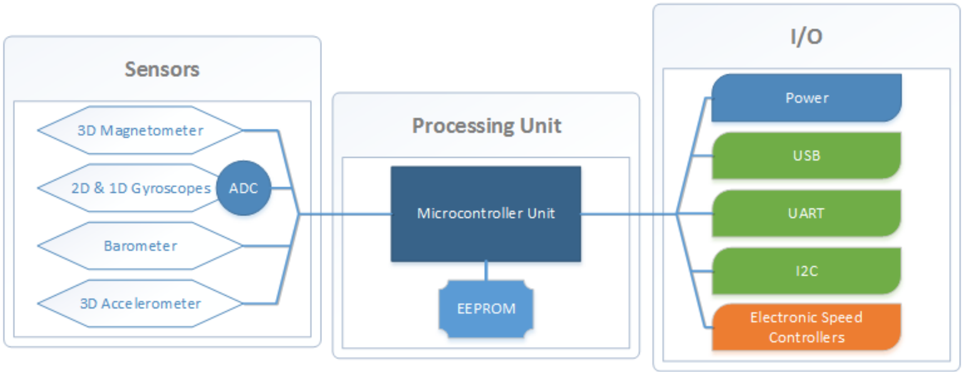

Figure 5 Stability sensors onboard the Pixhawk and I/Os for additional sensors

The leftmost part consists of the sensors readily available onboard. Together with the processing unit, they form the IMU which is the most critical part for the control of the attitude of the vehicle. A high priority software loop running on the MCU ensures that data collected from the sensors is treated in a timely manner for stabilization.

On the right hand side, we have the inputs and outputs available. Possibilities for serial communication include USB, UART and I^2C ports. These ports can be used freely to interface external peripherals to the processing unit. For our concerns, we have interfaced the PX4Flow smart camera through UART. The camera provides velocity and height information by doing frame by frame pixel analysis of the ground, i.e. optical flow calculations. This data is then integrated with respect to time within the Pixhawk to create a position estimation.

Mission Sensors

The only sensor strictly related to the mission objectives (i.e. interacting with ground robots) is the Gumstix Caspa camera which is attached to a Gumstix Overo FireSTORM (embedded computer). The camera acquires an image of the visible arena in front of the drone which is put through the embedded computer’s digital signal processor after which it is forwarded wirelessly to the ground station for analysis. Image analysis is done using the OpenCV library.

Obstacle avoidance is done by maintaining an altitude, acquired from the PX4flow’s sonar, greater than the maximum height of the obstacle robots.

Communications

Wireless communications are split onto the 915 Mhz and 2.4 Ghz frequency bands. The 915 Mhz band is operated by a set of HM-TRP transceivers which act as a simple serial bridge on which packets containing telemetry data, attitude commands and position commands are transmitted. These packets are encoded using the MAVLink protocol which is a very lightweight communication protocol optimized for micro air vehicles and suitable for over the air radio links. One end is connected to a ground station computer and the other end is connected directly to the Pixhawk autopilot to allow for minimal latency between commands and to a certain extent real time flight behaviour monitoring.

The 2.4 Ghz band hosts the Wi-Fi connection as well as the DSMX radio signals for manual overriding of controls and killswitch activation. Using TCP and UDP the ground station communicates with the Overo FireSTORM to transmit the camera feed. To simplify data exchanges and create a layer of abstraction on inter-node communications, we run the Robot Operating System (ROS) which contains interfaces for a variety of sensors including cameras and laser rangefinders. ROS allows us to offload image processing to the more powerful ground station computers while the embedded computer focuses on managing onboard peripherals.

Power management system

The energy source of autonomous aerial vehicles is a critical aspect of operations. Whereas traditional forms of RC vehicles have used gas engines and electricity for control systems, modern aerial vehicles, especially quadrotors, rely solely on lithium polymer batteries due to their high energy density. Since a brushless motor's rotation speed is related to its kV rating and the input voltage it is important to choose a battery allowing for a good capacity to weight ratio while providing an appropriate voltage. Furthermore, a higher voltage creates a higher rpm but a larger propeller requires a lower rpm. Though 2 and 3 cell batteries were tested we ultimately chose the GensTattu 6000 mAh 3 cell batteries for their longevity and because the ~12V they provide allow us to create the appropriate lift with our speed controller, motor and propeller triplet. If need be, a parallel connector was built allowing the use of two batteries simultaneously.

The batteries first go through a breakout power module consisting of a simple pass-through to a power distribution board (PDB) and a DC-DC voltage converter to power the Pixhawk with 5 volts and allows for battery voltage monitoring. Once the current is at the power distribution board, it splits off to the four electronic speed controllers and another DC-DC to power the embedded computer. The entire system is connected using 10 AWG and 16 AWG wire, the former of which is used for the battery/kill switch/PDB connections and the latter from the PDB to the speed controllers.

OPERATIONS

Flight Preparations

The checklist that is used before every flight is as follows in this precise order:

1. Electrical inspection
    a. Battery voltage is high (~12.5 V).
    b. All connectors are plugged in, secure and of the appropriate polarity.
    c. No exposed connections are prone to shorting.
2. Mechanical inspection
    a. All structural components are undamaged.
    b. All screws are tightly secure after the last flight and any screws requiring thread locking adhesive have been attended to.
    c. All propellers are tightly secured and none are chipped or damaged in anyway.
3. All components are securely fastened (especially battery retention Velcro).

4. All wireless links are operational including Wi-Fi, remote control and 915 Mhz radio.
5. Surrounding is clear of loose objects and non-essential personnel
6. Arming switch is activated

Additionally, it is recommended to listen for strange sounds from the rotors as the UAV is taking off and to be prepared to stop the mission at any time.

Man/Machine Interface

There are two ways to interact with the quadrotor. Manually controlled flight is done through a radio controller mainly for testing purposes and in case of unintended program behavior during autonomous flight. If such an accident were to happen, a safety pilot could take control of the quadrotor to attempt safe landing or simply switch on the kill switch. The second man/machine interface is the ground station. Its role is to make reading of flight and mission data possible in real time.

RISK REDUCTION

Vehicle Status

The status of the quadrotor will be continually sent wirelessly to the ground station. The ground station will then save all that information for later analysis. Status information comprises the following: roll, pitch, yaw, position (x, y), height (z), battery voltage and state.

Shock/Vibration Isolation

Propeller Protection

Priority was put on protecting the propellers from contact with the obstacle robots since they are the only means of propulsion. A small plastic barrier protruding from the motor mounts was added to help with these conditions.

Landing

Following the scope of our hovering objective, we determined that the only reason to land was in case of flight termination whether intended or not. Hence, the landing gear was designed to absorb impacts from 1.5 meters without damaging the integrity of the frame. A few designs were tested such as the X650F's default plastic legs, deformable aluminum legs, cardboard boxes and blocks of packing foam. The X650F's plastic legs would bounce from high landings and flip the vehicle which lead to damaged propellers while the aluminum legs required too much maintenance and had to be reformed before every flight to ensure a flat takeoff. Though the cardboard boxes were a surprisingly good fit, we ultimately chose the blocks of packing foam because they are easier to tailor to our needs and because of their slightly greater energy absorption properties.

EMI/RFI Solutions

By design, the control link is on an isolated frequency (915Mhz) as we expect the 2.4Ghz band to be fully saturated from the video stream, remote control and operations from other teams. Be that as it may, the true weakness of the vehicle is the magnetometer's sensitivity to EMI and general indoor shielding from the earth's magnetic field creating a potential for unexpected yaw behaviour. To reduce EMI, proximity to high current wires and components was minimized, however, as of date a true solution has yet to be implemented. Nonetheless, it should be possible to externalize the magnetometer from the autopilot module and mount it further away from the quadrotor's electronics to reduce self-inflicted EMI.

SAFETY

In order to ensure safety of both flight personnel and bystanders and to minimize chances of damage to the quadcopter, it is important to take certain precautions and have knowledge of our equipment's safety features. For testing purposes and for demonstrations in public places, a flight cage has been built. It consists of a 5 x 5 meters area surrounded by 2.5 meter high walls made of a PVC frame with nets spanning all sides. The ground inside the flight cage is covered with foam to absorb shocks and can be removed for eventual testing with ground robots.

Figure 6 The flight cage being built

In terms of electronics, if the 915 Mhz wireless link is cut off the autopilot will fail to receive a command at the required 2Hz minimum frequency and thrust to the motors will be cut off. A similar mechanism happens if the kill switch loses radio signal. Furthermore, our ESCs have an over-current protection feature where they will stop functioning if they detect currents higher than their rating. Thus, situations where high currents are let through due to motor obstruction will not damage electronics.

Finally, battery under voltage is avoided by having voltage monitoring onboard and through telemetry.

MODELING AND SIMULATION

Since the Pixhawk's firmware is open source and relatively well documented, we sought to implement our own control module instead of the existing one. The design was done using Matlab/Simulink and Catia, the latter of which allowed us to create a 3D model of our quadrotor and calculate the center of mass (I_{xx}, I_{yy}, I_{zz}). We then used the Simulink module in Matlab to simulate the dynamics of a quadrotor within the Laplace domain.

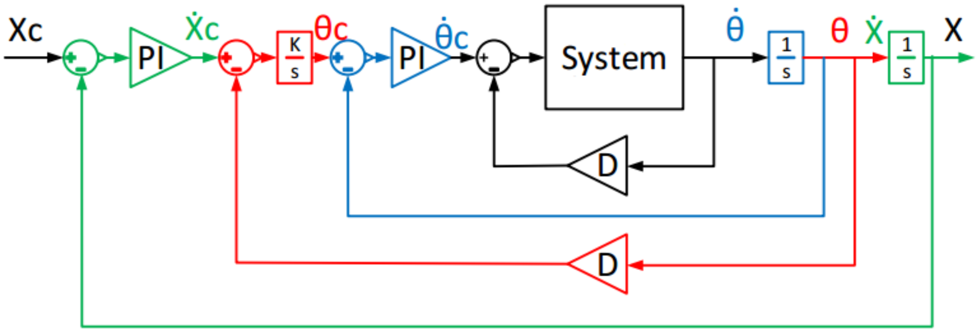

Figure 7 Graphical representation of the various control loops

The mathematical model obtained is then used to design cascaded control loops for angular speeds $(d\phi, d\theta, d\psi)$, attitude (ϕ, θ, ψ), displacement speeds (dx, dy, dz) and position (x, y, z). The design of the compensators was executed using a proportional-integral-derivative (PID) design for the speeds and a simple proportional design for attitude and position. The final design is represented in a simplified manner in the following figure.

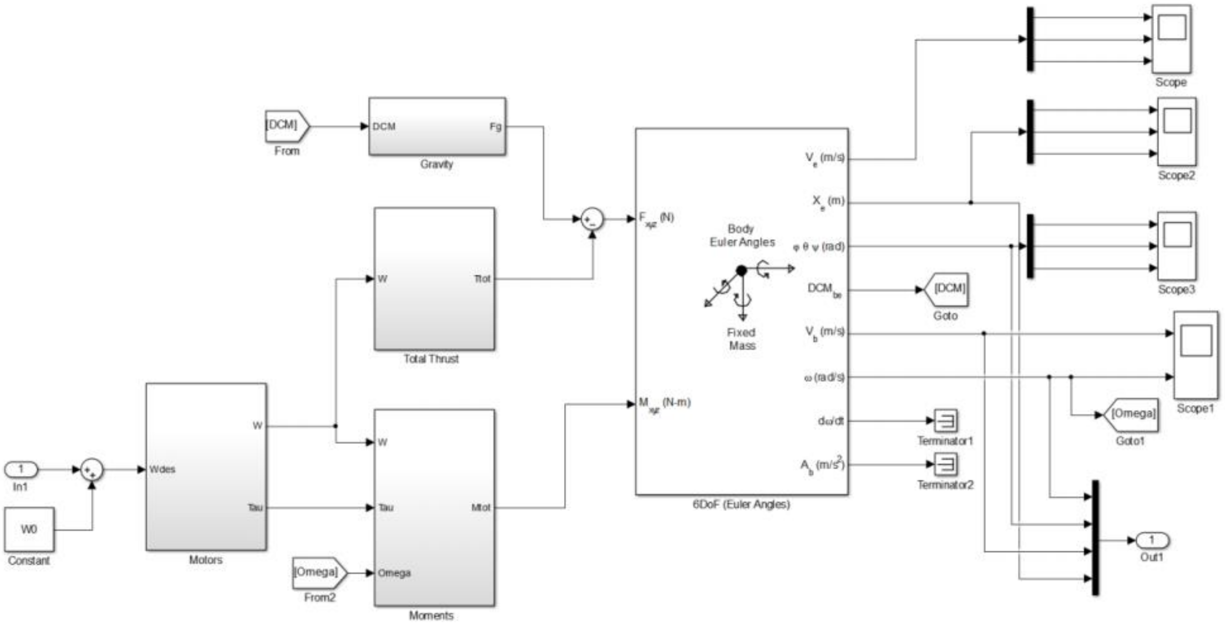

Figure 8 Simplified Simulink modelof the quadcopter

The implementation of our design was shown to be close to that of the Pixhawk. However, since the design had to ignore the actual firmware's implementation for simplicity, it was impossible to use the resulting control loop parameters without modifying the firmware. A heuristics approach was then used to determine these parameters. Finally, after reconsidering our initial control loop design by following the firmware, we were able to obtain theoretical values similar to our experimental values.

Testing

It is possible to distinguish between three types of tests: calibration tests, manual free flight, and autonomous free flight. All free flight tests take place inside our flight cage.

Calibration Testing

A test bench has been elaborated for PID calibration. Its role is to restrain movement to a single axis (either roll or pitch). It consists of a wooden square frame supported by a leg in each corner. Hooks have been screwed into its inner sides to allow for the quadrotor to be easily attached. Once it is tied in place, the quadrotor's PID coefficients are tuned and tried following a heuristic procedure.

Manual Flight Testing

Manually radio-controlled flight is often necessary to test various components and parameters. For instance, it has been useful in a finer adjustment of the PID coefficients, determination of battery life for a normal flight, verification of the behavior of our landing system, and reading free flight data from our sensors. It is done in the flight cage.

Autonomous Flight Testing

Implemented autonomous behavior consists of takeoff and position hold. It is tested in the flight cage and flight data is saved for analysis.

CONCLUSION

Elikos has shown that it is possible to rapidly create a UAV development platform using off-the-shelf components and will show on competition day a vehicle capable of autonomous take off and arena analysis through computer vision. The team looks forward to improving the state of the art in aerial robotics within the near future.

Future work

Elikos expects to have an implementation of the required behaviour within the year 2015 with significant improvements and customizations in structural, electronics and software design. Development of ground robot interaction algorithms will move forward from simulation to implementation. Additionally, it would be essential to make use of recent advances in Wi-Fi technology as recent consumer products supporting the latest 802.11ac specification allow for much higher bandwidths. We theorize that a Wi-Fi data link could be further improved by implementing UAV specific Quality of Service (packet prioritization) at the router level.

ACKNOWLEDGEMENTS

Team Elikos would like to thank the following companies, institutions and people for helping kick start this project into motion.

Bassem Srouji
David Saussié
Jérôme Le Ny

REFERENCES

[1] Landing Products, "APC Research," [Online]. Available: http://www.apcprop.com/v/Research/research.html. [Accessed 29 05 2014].

[2] "NuttX Real-Time Operating System," [Online]. Available: http://nuttx.org.

[3] "Electronics / IMU - ETH Pixhawk: MAV Computer Vision," [Online]. Available: https://pixhawk.ethz.ch/electronics/start.

[4] "IMU / Flight control Software - ETH PIXHAWK: MAV Computer Vision," [Online]. Available: https://pixhawk.ethz.ch/software/imu/start.