Outback Search and Rescue Challenge 2010

Team: Skylight

Deliverable 2: Proposed system design and safety considerations

Summary

Executive Summary (1 page)

This document outlines the Team Skylight's approach to solving ARCAA's UAV Challenge Outback Rescue 2010, including essential UAV design parameters and safety methodology.

Airframe design is based around an off-the-shelf RC aircraft, motorized by a gasoline engine. Permanent data communication is provided by a data link based on a 900MHz modem. Video telemetry is based on a 2.4GHz transmitter and a gimballed Sony block camera. Sensors include gyros, accelerometers and magnetometers for attitude determination, pressure sensors for altitude and airspeed and GPS for navigation. Sensor processing is done by custom circuitry, as well as actuator control and flight termination. The autopilot is a custom developed software running on a GumStix Overo computer. Operator control is done by a notebook running a custom made Ground Control Station based on STANAG-4586.

Safety is a primary concern, and will be addressed by safety measures that include independent flight termination circuitry, system monitoring, enhanced flight termination modes and safety oriented operational procedures. Several software and hardware features were implemented to achieve safe operations.

During Outback mission, the search path waypoints will be guided by the autopilot while two human operators monitor flight performance and video imagery looking for Joe. At any time the operators can change flight path mode in order to investigate a candidate point by loitering around a position while automatically pointing the camera to the candidate artifact position and/or changing speed and altitude. When the payload operator (human) finds Joe, it will request a drop authorization to the judges. If authorized, the ground station will calculate a trajectory for dropping the payload based on current wind speed and upload it to the vehicle so that it will automatically go to the drop point and release it.

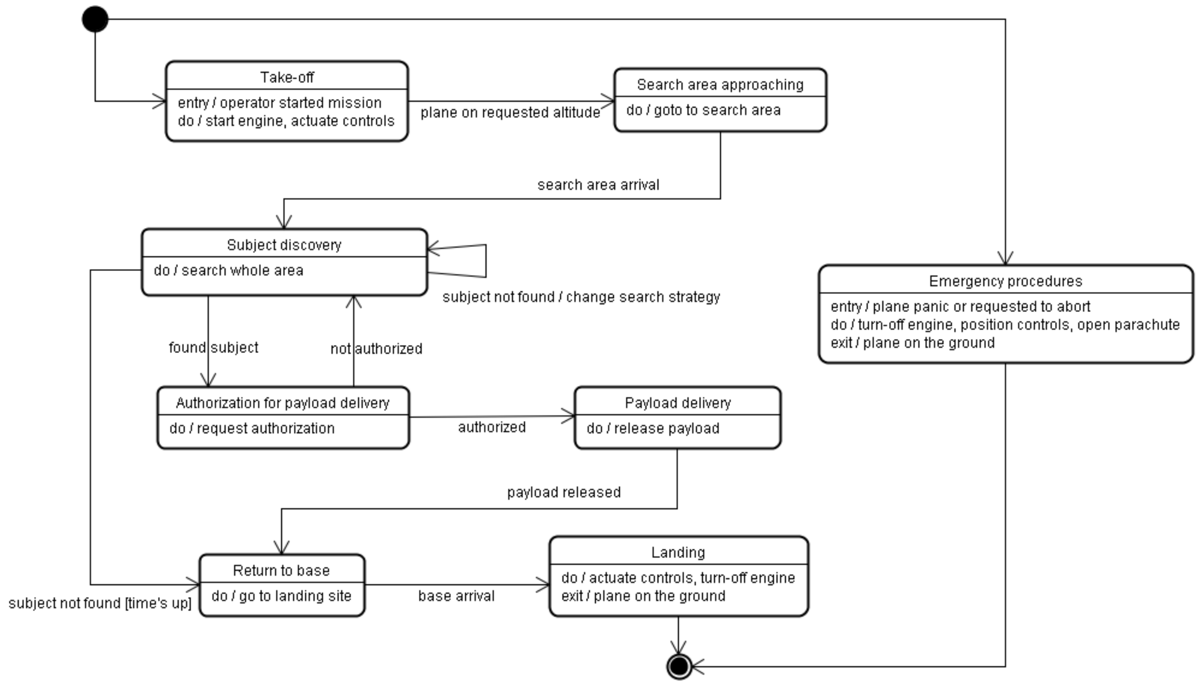


Fig 1: Mission profile for the UAV Outback Challenge

Introduction (1 page)

Team Skylight is located in Brasília, Brazil, and is comprised of two members, Eduardo Damasceno and Flávio Stutz, with somewhat different backgrounds but a common drive to solve the challenge.

Eduardo is a software architect by profession, and a business manager by schooling. An aviation enthusiast, Eduardo has been for years both an R/C hobbyist and a skydiver. Alongside the aviation passion, he’s been fiddling with electronics for a long time, having developed several microcontroller based devices. For him, the opportunity to design and build an UAV for the challenge was just too wonderful to pass.

Flávio is also a software architect by profession, but a physicist by schooling. Sharing the same passion for challenge, Flávio grew interested in the possibility of applying both his knowledge of physics and software development on the tricky task of developing an autonomous flying machine, capable of finding and rescuing a lost person in distress. As he dove into the world of real-time software development, machine vision, control theory and aeronautical engineering, his interest and admiration for the challenges involved grew stronger.

The team has been working towards the UAV outback challenge since mid 2007. The approach chosen for solving the challenge is largely based on the concepts of incremental and iterative development, by tackling the larger risks first and building testable, functional parts in small increments. This kind of approach has helped the development of a whole autopilot and ground control system, from hardware to software. This is the 3rd version of the Skylight platform.

Based on the progress so far, the team is confident on their capacity to achieve the proposed goals.

The development of the platform has provided a lot of useful experiences on communications, imagery, power plant, hardware and software systems for the team members. See below an example of how some systems have evolved:

System	**Version 1 (2008)**	**Version 2 (2009)**	**Version 3 (2010)**
Main Data Link	3G/GSM based	WiFi with high gain antennas	900MHz modem with omni antennas
EO sensor	Nokia N95 camera	Fixed camera	Custom made georeferenced gimballed camera
Image transmission	Photos over 3G	2.4Ghz Video transmitter	2.4Ghz Video transmitter
UAV Control	Custom	Custom	STANAG-4586
Sensor/actuator processing hardware	Single microcontroller	Single microcontroller	Two 8 core microcontroller on 4 layered PCB
Onboard computer	Nokia N95 with MIDP autopilot	VIA EPIA PX x86	Gumstix Overo ARM
IMU	Thermopiles/pressure sensors and 5Hz GPS	Thermopiles/pressure sensors and 5Hz GPS	Gyro/accel/magn/pressure sensors and 10Hz GPS
Watchdogs	Hardware	Hardware	Hardware and all software elements
Power plant			3W with muffler (?)

Design Approach and Rationale (2 pages)

UAV Control System

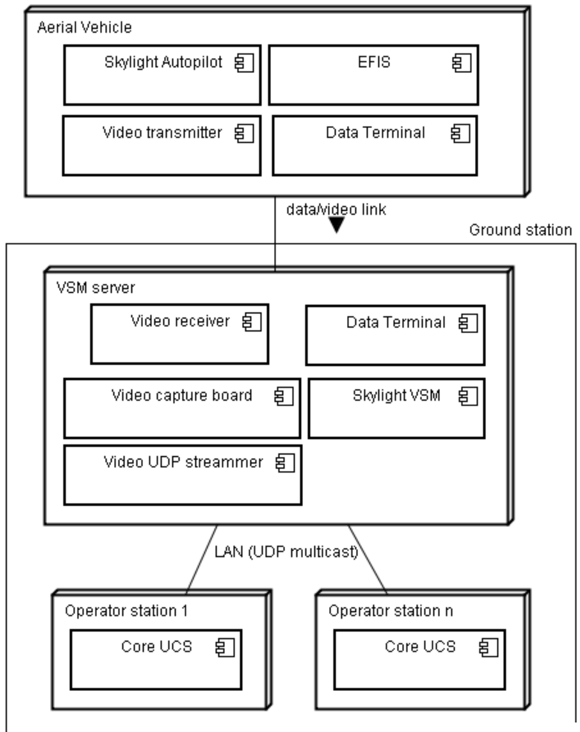


Fig 2: Overall UAV Control architecture

The entire UAV Control System was built from scratch by the team based on STANAG-4586. We were glad to find this standard because it proved to be based on industry experiencies and provides a common way to control UAVs.

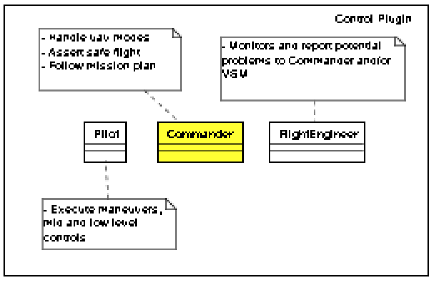
The main architecture elements described by the NATO standard were implemented by the team. See fig 2 for more details.

Control modes are:

- Flight Director: Direct commanding of Speed, Altitude and/or heading
- Waypoint: Follow an uploaded mission plan
- Loiter Director: Loiter around a center coordinate, allowing realtime change in altitude and speed
- Safety Procedures: A custom mode that we created to handle all kind of safety procedures that the autopilot or operator wants to activate at any time (not part of STANAG).

Autopilot

Autopilot software is based on plugins and is heavily based on OO and realtime programming techniques. The core control system employs a "real plane crew" metaphor for its organization. A great attention to software cohesion and testing was taken in order to easy the development of such a complex and critical system. As the autopilot evolved in complexity and number of services, it had to be refactored in order to handle the new requirements. A lot of work, but it was fun. See fig 2.



The navigation controls were implemented using state machines and the lower controls were implemented using PID theory. All vehicle configuration and PID parameters can be set during flights from the ground station with the help of setpoint/feedback charts.

Control Station

Our Control Station was developed to deliver good situation wareness to the operator and a good level of view customization according to operator role in mission. By this way, a payload operator may choose to show only payload specific views, while a commander may choose to see only mission and system health statuses. All messages from/to the control station are based on multicast UDP packets over an ethernet network, so it is possible for multiple stations on the network with different levels of grants to be connected to the same VSM/Vehicle.

Basic features:

• Configuration of airframe characteristics for	• Mission creation, validation and uploading

Autopilot ○ Operational limits and optimum parameters ○ Calibration of PID parameters on-the-fly with the help of setpoint/feedback graphs	○ Rules of safety definition: min/max altitudes, communication hold location, manual recovery location, safety procedures configuration, mission boundaries etc ○ Mission planning by clicks on a georeferentiated map ○ Validation of missions with invalid parameters (path loops warnings, waypoints outside ROS limits etc) ○ Flight termination button
• Overall telemetry ○ Artificial horizon ○ Subsystems status ○ Actuators position ○ Sensors readings ○ Georeferenced map ○ Subsystem alerts	• Flight path modes ○ Loiter Director control integrated to control map ○ Loiter and stare with camera an specific location by clicking on control map ○ Flight Director/Manual override controls ○ Waypoint mode control

Sensor Processing

The set of sensors installed on the aircraft is comprised of the following elements, all integrated in a custom designed board:

- Static pressure sensor for barometric altitude and climb rate calculation (BMP085)
- Differential pressure sensor for airspeed calculation (MPXV5004DP)
- 3 axis gyroscope, 3 axis accelerometer and 3 axis magnetometer for attitude determination
- 10 Hz L1 GPS receiver for navigation and ground speed determination
- Temperature sensor for cylinder head temperature measurement (LM34)

All sensors will be interfaced with custom designed circuitry, developed around Parallax´s Propeller multi-core microcontroller. All analog sensors will be interfaced with a 12 bit SPI ADC, while all digital sensors are interfaced via an I2C bus.

Payload & Payload Delivery System

The chosen payload, based on the specified challenge rules, will be a 500 ml bottle of water suitable for drinking. It will be attached to a small round parachute, which will provide a slower descent to the ground.

Payload is attached near to the plane’s center of gravity, to avoid its dislocation after payload release. Attachment will be provided by single point release system, made of monofilament nylon line, similar to fishing line, of sufficient tensional strength. The release device was custom developed, and consists of nichrome wire attached to the single point release line, that once activated becomes incandescent and immediately melts the nylon line, releasing the payload. Tests proved the system to be very effective.

Risk Management Approach (5 pages)

Spectrum Management

Our UAV System uses 3 wireless links:

- **Vehicle ⇔ VSM data link**
  A 900MHz Aerocomm AC4790 modem designed for long range operations.
  The transmitter has a maximum power of 1W and is connected to a 2dBi omni antenna by cable.
  Considering max transmitter power, antenna amplification and losses in connectors/cables, the total irradiated power will be near (but below) 1W. That power will be sufficient for direct communications to vehicle while in the farthest location of the rescue mission (~8km).
  The modem will be used in half duplex mode because at most of the time the link will be sending messages from vehicle to VSM.
  Field tests showed good performance with omni antennas.
- **Vehicle ⇔ VSM video link**
  A 2.4 GHz video transmitter used by FPV hobbyists.
  The transmitter has a maximum power of 1W and is coupled to a 1.5 dBi omni antenna.
  Considering max transmitter power, antenna amplification and losses in connectors, the total irradiated power will be near (but below) 1W. That power will be sufficient for sending video telemetry to vehicle while in the farthest location of the rescue mission (~8km).
  This device will be turn-on only when the autopilot is in auto mode and far from manual recovery position. Switch on/off procedures will be automatically executed by autopilot.
  Fields tests showed good performance with omni antennas.
- **Vehicle ⇔ RC link**
  A 2.4 GHz Futaba 6EX Digital Radio Controller compliant to FCC Part 15.
  This link will be used for take-off, landing and manual recovery procedures in near range.
  The radio controller uses digital technology and employs frequency hopping, what greatly reduces the chance for interferences with other devices.
  Although video link uses a frequency near RC spectrum (2 GHz), **they will never overlap** because the video link will be turn-on only on mid-far range operations, while RC radio will be used in near range operations.

Flight Termination

A completely independent microcontroller was used to handle Watchdog and Flight Termination procedures. All servo control lines are overridden by the Flight Termination microcontroller, even if manual RC control is activated. A secondary battery is used to guarantee that if the main battery dies, the Flight Termination will take place with no problems. Tests have proven the servos override capability as well as correct Flight Termination on power failure conditions.

The Flight Termination is activated when a heart beat originated by the sensors/actuators microcontroller is not detected by Watchdog, indicating that the autopilot hardware is in trouble. Additionally, the autopilot hardware may be commanded to stop sending watchdog kicks to force Flight Termination in the occurrence of the following events:

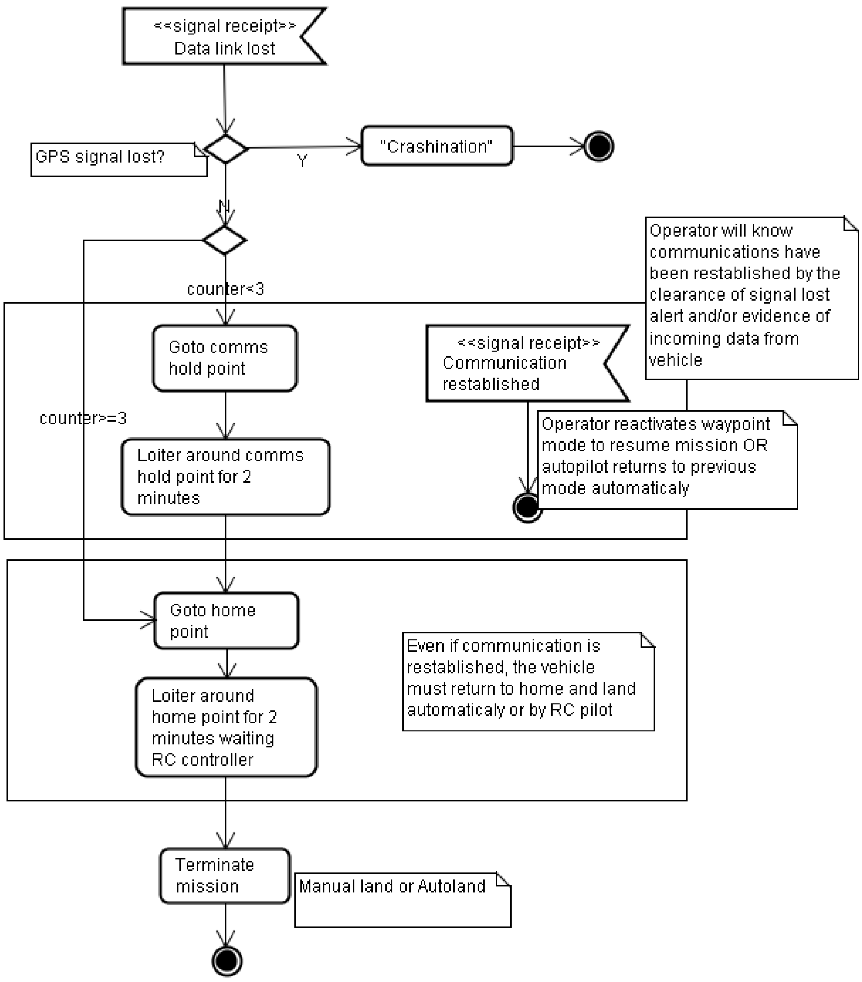
- Autopilot hardware stops receiving messages from autopilot software;

- Autopilot software detects that the vehicle has crossed mission boundaries;
- Autopilot software detects simultaneous loss of GPS and Data link signals;
- Manual recovery has failed after data link signal loss recovery failure
- A Flight Termination command is armed and executed by the operator in ground station;

Loss of data link

The procedures described in Challenge Rules Document (section 5.5.2.1) for data link recovery was implemented for data link recovery.

The loss of datalink is detected when a heartbeat signal sent each 1 second from VSM is not received by the vehicle for more than 10s. When this situation is detected, the vehicle automatically enters in Safety Procedure Mode and performs the DataLinkLossRecovery action. In summary, the vehicle will go to a Rally Point where it will wait for data link recovery. If it doesn't detect incoming messages from VSM, it will go to Manual Recovery point so that the pilot can regain control of UAV. If the manual control cannot be recovered it will perform the Flight Termination by spinning to the ground. The implementation followed all procedures described in Challenge Rules Document.

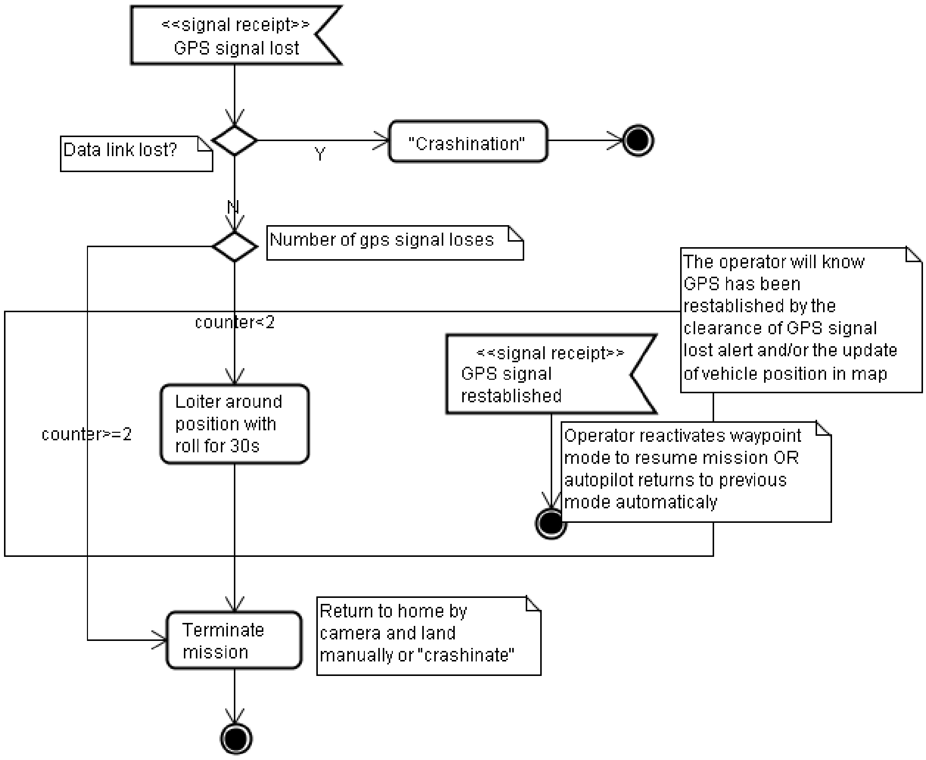


Loss of GPS

The procedures described in Challenge Rules Document (section 5.5.4) for GPS link recovery were implemented and tested in simulators.

When the autopilot detects that it is not receiving messages from GPS or that the GPS signal is poor (no satellite lock), it will enter in Safety Procedures Modes and execute the action GPSLinkLossRecovery. This action will command a loiter with roll of 20 degrees around current position and hold current altitude and airspeed. If it detects signal recovery, it will switch back to the previous mode before GPS loss, if recovery is not possible, the operator will activate Flight Director and go back to home using onboard video and assisted instruments holding for heading/altitude/speed controls. **If both GPS and DataLink are lost simultaneously, the Flight Termination procedure will take place at any time.**

The implementation followed all procedures described in Challenge Rules Document.



Lockup or failure of autopilot

The autopilot is composed of some elements that must be monitored in order to guarantee that all pieces are working well. We chose the "Watchdog" technique for monitoring those parts by creating a chain of watchdogs. Each autopilot element performs a self sanity check and kicks its related watchdog when everything is OK. When something is bad, it tries to recover itself by performing a self reset. If the reset isn't successful (or it takes too long to come alive), its watchdog won't receive a kick and won't kick the next watchdog in his chain, causing other Watchdogs to be timedout. By this way the Flight Termination is activated when any element detects a trouble.

The chain of watchdogs is:

Computer Watchdog => EFIS Watchdog => Flight Termination Watchdog

- **Computer Watchdog:** Monitors all software threads. If any thread seems to be lockup, it resets the autopilot software. If the autopilot doesn't come live within 1s, the Flight Termination device is activated by the autopilot hardware because it stops kicking the Flight Termination Watchdog.
- **EFIS Watchdog:** Monitors avionics sensors readings. If it detects a problem with avionics, it resets the EFIS hardware. If the reset doesn't solve the problem, the Flight Termination Watchdog won't be kicked, so Flight Termination will take place.
- **Flight Termination Watchdog:** Monitors kicks from EFIS hardware. If a kick is not received within 1s, it overrides any servo command from

Autopilot and starts the Flight Termination procedures as described in section 5.6 of Challenge Rules Document.

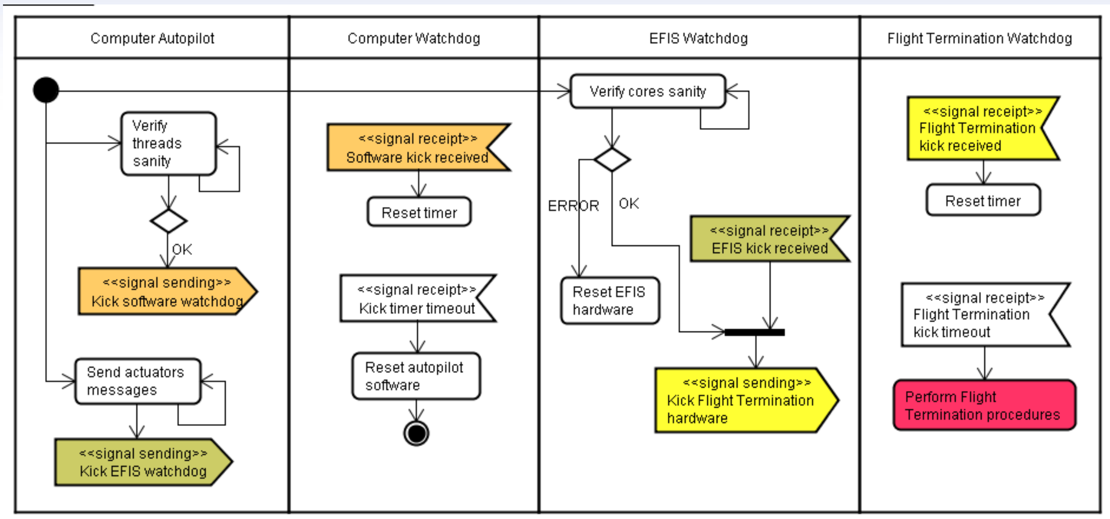


Fig 2: Watchdogs chain

Loss of engine power

The autopilot monitors various aspects of flight healthiness and sends alerts to ground operator when something is dangerous or has failed. One of the health checks is to verify if the RPM is above some minimal level. If it detects that the RPM is too low (probably because engine is cut), it will perform a safety action called LoiterWithRollDescending. This action will perform a loiter using a roll of 10 degrees while keeping airspeed slightly above stall speed using pitch. By this way the vehicle will begin descending while maintaining a safe attitude when touching ground.

Tests in simulator showed that the loiter will have 150m of radius and a vertical speed of 3ft/s down. If the vehicle crosses mission boundary at any time, the Flight Termination procedures will be executed. The touchdown will have very low kinetic impact on ground.

Li-Po battery management

Our main power lines are protected by fuses that will cut the current on the event of a short circuit, avoiding battery explosion. The battery packs will be protected by a cushion in order to avoid any kind of puncturing in the event of a crash.

We use a charger specially designed to charge Li-Po batteries. It employs the best charge/discharge techniques around. The power levels of the batteries are never allowed to reach a dangerous low level, because this would damage them too.

Checklists

To avoid problems during flights, there is a list of items to be verified before taking off in order to ensure correct UAV operation. This list represents attention points that were perceived as important and were developed during project tests. It has evolved during the project as new items were included to the overall system. See below:

- **Airframe checklist**
  - ☐ Overall airframe in good shape
  - ☐ Payload delivery system well fixed to airframe and armed
- **UAV systems checklist**
  - ☐ All systems responsive

- ☐ Internal system tests successful
- ☐ Sweep test has changed from min to full range the elevator, ailerons and rudder

- **Radio controller checklist**
  - ☐ Elevator, ailerons and rudder responding to controller at min/max ranges
  - ☐ Changing from manual to auto-pilot is working (and vice-versa)
- **Ground station checklist**
  - ☐ Mission “Outback Challenge” loaded and uploaded to UAV systems
  - ☐ Airplane configuration “Rascal” loaded and uploaded to UAV systems
  - ☐ Telemetry is being received on ground station
  - ☐ AGL altitude set and receiving 0 ft on telemetry
  - ☐ Plane located at take-off starting point on map
  - ☐ Link latency lower than 1s

Flight Tests Results and Discussion (2 pages)

Field tests organization

Our tests were organized as follows:

1. **Planning**: definition of test objectives, required resources, auto-pilot configuration for mission and execution of the mission plan in simulator (XPlane)
2. **Pre-flight**: UAV systems startup, ground station startup, start engine, checklists and aircraft positioning on runway
3. **Flight**: take-off, execution of test objectives, return to base and landing
4. **Post-flight**: engine stop, systems down, verification of overall aircraft state, discussion of lessons learnt, definition of next steps

Team roles/responsibilities

Role	Responsibilities
Pilot and commander	• Apply "airframe" and "radio controller" checklists • Control airplane by manual radio control when needed • Take decision on abnormal circumstances
Operator	• Apply "UAV systems" and "Ground Station" checklists • Control planned mission execution using ground station • Monitor overall mission status and report to pilot

First autonomous flight

Objective: Perform autonomous flight by holding level

We had various field tests before the first autonomous flight test. We've tested the autopilot hardware/software in hardware-in-the-loop a lot before that day. The moment the pilot gave control to the autopilot for the first time while flying was very exciting. The autopilot managed to fly our Rascal that day, but there was still work to do regarding to communications link that proved to be inefficient. At that time we were trying to use 3G/GSM links, but link latency issues and lack of signal coverage in remote areas proved it to be a bad idea.

The autopilot showed the capability to receive PID controllers parameters while flying and managed to hold level and performed a simple waypoint following.

During operations we had a problem with the engine, but our pilot managed to glide the vehicle to a safe location and land it safely.

Wifi based flights

Objective: Verify Wifi link viability for data link

We've got two high power Wifi modems along with a omni and a high gain antenna and performed various field tests looking for range capability. At first the Wifi idea would work, so we decided to test it on a flying vehicle.

The flight dynamics made the wifi data link too instable due to antenna polarity issues and TCP/IP protocol characteristics. When the link was lost, there was about 4 seconds of blackouts while the connection was re-established. The range capability when moving the antennas was not very good and it was too susceptible to vehicle attitude.

At that time we've detected a strange behavior on pressure sensors: when sun light approached the sensors, the telemetry showed a change in readings. That was fixed by protecting sensors from any light.

Flight path performance tests

Objective: Test vehicle capability to follow predefined flight paths

On the first autonomous flight, it was possible for the vehicle to hold level and follow a simple flight path plan. After analysing the flight results we’ve managed to enhance some PID control aspects and the way flight path following would be performed. The flight tests showed autopilot capability to follow waypoints that were uploaded during flights successfully. We’ve tested the autopilot capability of following a line path between two waypoints, instead of only going direct to the second waypoint. That feature is important due to the need for covering the search area in a predicted way.

At each test the importance of good planning and time/objectives management was more evident. Some tests and data collections were forgotten due to the flight excitement.

Camera tests

Objective: Test a simple fixed camera for searching an area

We always tried to use things “Simplest as enough”, so before implementing a complex solution, we tried the simpler solution. Our first idea on image acquisition was to use a cellular camera for that task.

During our tests we realized that the camera FOV was too narrow and the flight dynamics changes the image very fast, especially on turns. At 400ft the image below the camera passes fast and it is hard for an operator to look for small objects using such a camera solution. Another problem was related to engine vibration. We determined a resonant RPM, but even at small resonances the MPEG compression made the image blurred and distorted.

From that point we decided to implement a gimballed camera solution so that the operator can inspect candidate artefacts without changing flight course and keep a target artefact in the center of the video automatically despite of vehicle attitude changes. By this way, the payload operator looking for Joe can focus on searching, instead of thinking about the flight itself. In real field and simulator tests we realized that it is very hard for the operator to relate vehicle course with camera imagery.

Payload drop tests

Objective: Drop a bottle of water coupled to a small parachute at 300ft

To avoid a change in center of gravity of the vehicle after dropping the bottle of water, the payload was fixed at the center of gravity of the vehicle. Fortunately our bottle+parachute system worked well and the bottle landed safety near target. Analying the video, we found that the parachute may be smaller to avoid too much travel between its release and its touchdown position.

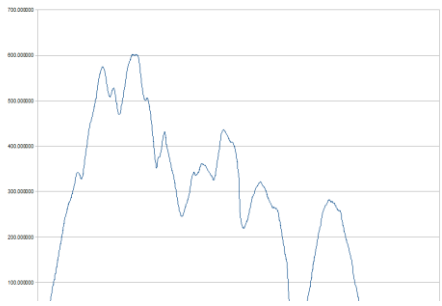


Fig 4: Altitude profile

Conclusions (1 page)

Through the use of safety checklists, good planning, careful constructions and lots of tests using software simulators the team has achieved a good level of safety on test flights. We were happy to see our autopilot flying Rascal right at our first real flight test. Severe software and hardware problems were detected in simulators before real take-off. The realistic airplane simulator XPlane was used for developing/testing autopilot controls, testing mission plans viability, to determine initial PID calibrations and for testing portions of the hardware using “hardware-in-the-loop” test techniques. We’ve used gimbal and camera simulators for developing the georeferenced camera gimbal and the video station control too.

An UAV System has many complex parts, from aerodynamics to hardware, software, mechanics, legal and personnel perspectives. The success of the platform depends on well trained pilots/operators using systems that were built with care and tested against different situations. When in field, the time for decisions may be very short, so during mission and rules of safety planning we can visualize mentally or in simulators, how the mission will be executed and what to do in case of unexpected situations.

Finding Joe is a great challenge, not only because the airplane has to fly autonomously, but because of the safety measures we have to think, implement and test in order to avoid accidents, not only during the competition itself, but during all kinds of field tests. We haven’t lost any airframe nor had anybody injured by any means until now, but we know that the safety procedures have always to be followed because the risk is always present.

Implementing the Skylight UAV System was fun, but not easy. We hope all this work worth finding the thirsty Joe lost in the Outback!