Holiday50AV Technical Paper

Christophe De Wagter, Matthijs Amelink
Delft University of Technology & D-CIS Lab,
Delft, The Netherlands

September 4, 2007

Abstract

As autonomous GPS waypoint tracking has been achieved for 50 cm 500 gram MAVs, this paper focuses on autonomous interaction with the world and higher level interaction with the operators.

Image Analysis is performed on down-streamed images in order to estimate altitude above ground, map the surroundings and detect targets of interest. Detected targets are collected, ordered by certainty and presented to the user for validation. Furthermore information from all levels of sophistication ranging from flight level to mission level are merged in a single intelligence map allowing the operator to chose the level of interaction.

The resulting system enables extremely easy task definitions such as search this area for, determine the location of this visual target very precisely identify target at this location. With no user intervention other than the launch of the vehicle this mission can be performed. However during tasks where the MAV has less certainty such as for instance during search missions, help of the operator is asked and user validation can speed up the search process or enhance the difficult maneuvers.

As MAV become more and more important, the number of MAVs per operator has to be increased. In order to achieve this, mission-level communication greatly reduces the cognitive workload of the operator. Autonomous interaction with the world allow the MAVs to operate on them selves while asking operator input only when useful to improve the mission performance.

Nomenclature

Acronyms	
AHRS	Attitude and Heading Reference System
EKF	Extended Kalman Filter
FOV	Field of View
GNC	Guidance Navigation and Control

GPS	Global Positioning System
HS	Horn & Schunck Optic Flow Technique
IMU	Inertial Measurement Unit
LK	Lucas & Kanade Optic Flow Technique
MAV	Micro Aerial Vehicle
MEMS	Micro Electro-Mechanical Sensors
PCB	Printed Circuit Board
SBAS	Satellite Based Augmentation System
UAV	Unmanned Aerial Vehicle

1 Introduction

Various micro aerial vehicle setups have been proposed over the past few years. Most of them [1] use a similar setup as they use GPS for guidance, infrared detectors for attitude determination, short-range high bandwidth video transmission and hard coded fixed gain controllers. The problem with raw video transmission is that only short range applications are possible. Infrared detectors can get in trouble over water, in clouds or in mountains. Finally small MAV are difficult to manufacture in a reproducible way and their characteristics differ from one vehicle to another, and sometimes even from one flight to another depending on the exact millimeter location of their battery pack for instance. Finally different gains can give better performance in different weather conditions.

Therefore this paper proposes a different approach. In order to make the attitude determination more robust, inertial MEMS sensors are used combined with GPS. When correctly calibrated against temperature these gyroscopes and accelerometers provide very constant and high bandwidth attitude information, ideal even for highly manoeuvrable small aircraft such as MAV. Moreover, to solve the short range video transmission problem, steps are taken to enable video processing onboard the micro aerial vehicle. After initial simple video processing, only interesting parts are transmitted over a long range low bandwidth digital link. Finally adaptive elements are included in the controller to accommodate for changes in the vehicle dynamics, remaining changes in the sensors and adapt to different weather conditions. The resulting autopilot has large computational capabilities, gyroscopes, accelerometers, GPS, USB for webcams and storage devices and ethernet access. The total weight of the onboard electronics is 185 grams, which enables a new range of MAV applications.

The ground control station runs custom UAV control software. The interface and automated features are designed using principles from Cognitive Systems Engineering [2] and Ecological Interface Design [3] around the operators cognitive processes. Current day systems require one or more human operators to control those aspects of UAV operation that aren't (reliably) automated and thereby the operator becomes part of the system. Providing he operator with a proper interface and transparent automation is necessary to effectively make him/her part of the system.

2 Hardware

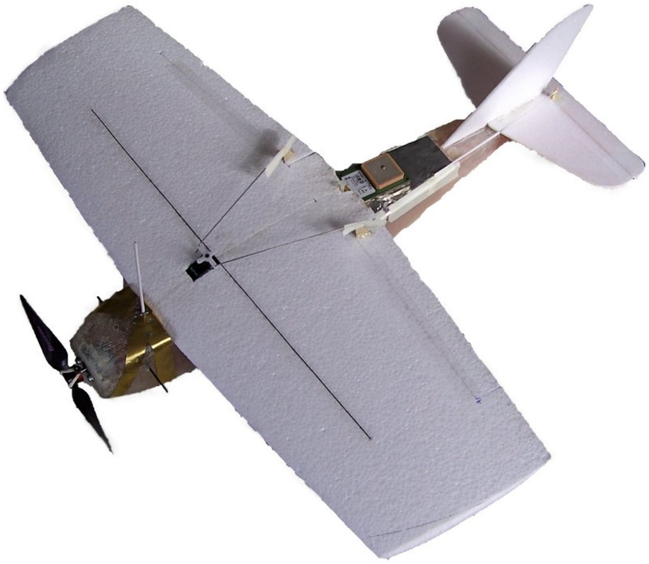


Figure 1: Holiday50av with widened fuselage to accommodate on board electronics and clipped wings.

The design of the Holiday50av is derived from the successful Holiday60 model airplane that has superior aerodynamic properties in its flight regime. Some adaption was necessary to meet the requirements for the MAV07 competition. The wings were clipped in order to meet the required maximum dimension of 50 cm. The fuselage was widened to accommodate the electronics taken on board.

2.1 Autopilot: MAVPilot3

Custom on board electronics have been developed because the capabilities of other autopilot boards such as the micropilot were insufficient and the ease of use for research purposes was not fulfilling. A board fitting in a MAV with a maximum dimension of 50cm, but capable of processing images onboard and running advanced guidance, navigation and control algorithms poses so many specific constraints that a custom board was designed and manufactured. The current version of the autopilot is MAVPilot3 which is the result the collaborative effort of the D-CIS lab and Delft University of Technology.

The MAVPilot3 autopilot consist of commercially available computer board, long range modem, custom designed MAVPilot3 board that carries an IO processor, flight data sensors and the servo center. The computer board and long range modem are piggybacked on the custom MAVPilot3 board which integrates all functions described below. Figure 2 shows the main functional blocks of the autopilot.

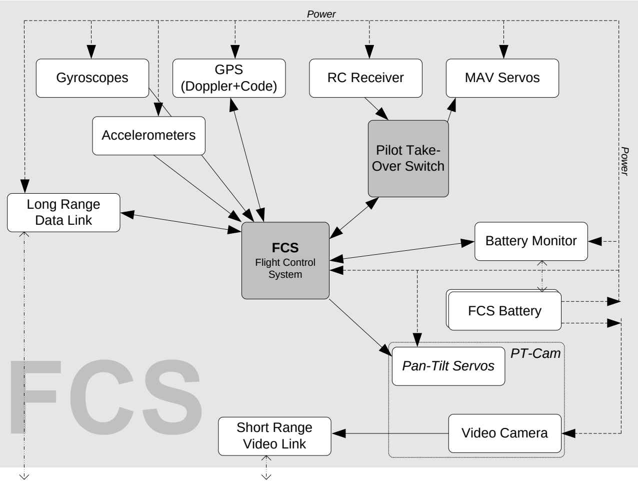


Figure 2: Functional overview of the flight control system (FCS). The main items are an inertial module, gps, the servo interface and the processor.

2.1.1 Computer Board

The autopilot described in this paper uses the DIL64-NET pc boards supplied by the SSV company (Figure: 4). These complete Linux boards come with various serial ports, USB host and device ports, full ethernet support, SPI bus and many digital IO. Two versions of these processor boards are looked at. The 180MHz 32bit AT91C9200 ARM processor from ATMEL uses about 1 Watt of power and is largely sufficient for autonomous flight including Kalman Filter and adaptive controller. But in order to be able to run more vision code onboard, a 400MHz XScale ARM board can also be accommodated, both having the same form factor and pin-out.

2.1.2 Power Supply

Very high efficiency power conversion from a 3-cell Lithium-Polymer battery pack of 11.1 Volts forms the basis of the power system. Efficiency is specified to be up to 98 percent, but measurements show typical values of > 95 percent efficient power conversion. 3.3V Volts is provided to the computer board and low noise 5.0 volts to the long range modem and servo controller. Ultra low-noise 5.0 Volt is generated for the inertial modules and analog digital conversion. Very low noise 3.3 Volts is made for the GPS receiver. Finally a backup battery provides tension for the volatile memory of the system clock and the

GPS receiver. A separate battery pack is used for propulsion in order to isolate EMI/ESI interference caused by the brushless electric motor on the battery power. On the PCB, all tracks are kept as short as possible and a ground mask is used to improve the shielding.

2.1.3 Inertial Measurement Unit

Gyroscope and accelerometer MEMS are integrated on the MAVPilot3 board. The gyroscopes and accelerometers signals are run through operational amplifiers before passing into an eight channel analog-digital convertor connected to the SPI bus of the processor. The analog circuitry is put as far away as possible from the switching power regulators and from the processor. The AD convertor also samples the battery voltage and temperature from the thermocouple inside one of the gyroscopes.

2.1.4 Servo control

The ServoCenter® servo-controller was chosen to reduce the processor load of creating PWM pulses. This preprogrammed ATMEL chip from Yost Engineering can drive up to 16 servos and includes rate control, minimum/maximum protection and many other features.

USB and Ethernet ports are added for logging on memory sticks, reprogramming, telnet and ftp access, hardware in the loop tests and webcam or frame-grabber connectivity. This ensures easy of programming and development.

2.1.5 GPS

A uBlox Smart Antenna Module is used providing 4Hz GPS updates in the efficient uBlox protocol. Code-based position and doppler-based velocity navigation solutions in Earth Centered Earth Fixed (ECEF) coordinates are retrieved, while *Satellite Based Augmentation System* (SBAS) information from EGNOS space vehicles 126 and 130 is used when available to overcome ionosphere signal degradation. The uBlox modules have a configurable dynamical model that can allow up to 4g dynamic manoeuvres. It accepts differential GPS information and can stream raw carrier and phase data to smooth the code solution.

2.1.6 Long Range Modem

An AeroComm wireless smart modem (Figure 5) in the 868MHz band is chosen for digital data transmission. This 250 milliWatt transmitter can send up to distances of 15 kilometers. A 4m high mast is used in the ground in order not to shield the Fresnel zone and thereby reduce the range. The onboard antenna is a ground plane antenna located on the nose of the airframe. The ground plane is formed by 6 strips of copper foil adhered to the fiberglass hull. Both antennas are oriented vertically having good antenna properties over larger distances and the worst properties when the aircraft is exactly above the ground control station at small distances.

2.1.7 Wireless Camera

To augment the onboard video processing, a short range but high bandwidth analog video link is used. When longer range than the video link supports is required images are captured by the onboard Linux board, processed, compressed and slowly sent over the long range modem providing either relatively fast update rates with very low resolution or very slow updates with better images. This becomes possible thanks to the large onboard processing capabilities of the Linux board.

2.1.8 Pilot-Take-Over Switch

Before full autonomous flight can be achieved, many test flights are necessary. Therefore a pilot-take-over switch that allows in flight switching between manual steering inputs or autopilot control was built. The manual takeover switch is activated over the RC transmitter. Although necessary for development it also poses a safety threat. At long ranges and present interference can cause the take over switch to switch to manual without notification. This especially dangerous when the MAV is operated beyond visual control range. As an extra safety measure the onboard processor monitors the RC signal and determines whether it is safe to listen to the manual take-over channel, giving override authority to the autopilot.

2.2 Ground Control Computer

A functional overview of the ground control station can be found in Figure 7 while Figure **??** shows a picture during flight testing. Simply put, a laptop running the ground control station code interfaces with the autopilot over a 15 km long range 868MHz smart modem and interfaces with a differential GPS. When on-ground video processing is required a second laptop is used that receives video from a short range video link when available, and interfaces with the ground control software using an ethernet connection.

3 Software Environment

In order to make various UAV applications possible, a flexible software project was started that incorporated the possibility to simulate everything in software and hardware in the loop mode before flight-testing it. Hardware-in-the loop testing to validate the onboard computing power and remove bugs was also particularly important. Finally the software was designed to be highly customizable and modular for research purposes. The result is a workspace consisting of different libraries that can be interconnected in various ways. It is called *SmartUAV*.

3.1 Flexible Data Channel Library

Every hardware interfacing is hidden behind a standardized structure further called *Flexible Data Channel* or (FDC). A Flexible Data Channel can be a serial port, an Ethernet port with UDP or TCP/IP, a Serial Peripheral Interface SPI bus device, a virtual data port, a file or any other thing that might get or provide data. Moreover, Linux and Windows have slightly different code to access the hardware listed above. These operating system differences are also hidden in the FDC library so that all above-laying code can be identical for any hardware or operating system.

The power of this setup lies in the fact that for instance the flight controller code can either be run together with a simulator exchanging data through a virtual data port (Figure 8 Top Row), or can be run on the autopilot board getting sensor data from the simulator through UDP or even serial cables (HIL), or can interface with its own GPS over a serial port and its inertial data through an SPI bus (Figure 8 Bottom Row). The code does not even have to be recompiled, but only has to be set up for a particular condition at boot-time.

This Flexible Data Channel setup makes it possible to use the very same code for simulation tests, hardware-in-the-loop tests and during real flights.

3.2 Flight Control Library

The flight control computer chosen is an AT91C9200 running the Linux 2.4 Kernel. A good C++ cross compiler is delivered with this product enabling standard C++ code to be executed with great ease. While writing every letter of flight control code, great importance is put in minimizing every computational effort since onboard processing power is expensive in many ways.

The flight control code is divided into two threads. One thread runs the actual guidance navigation and control (GNC) algorithms, while the other thread can run additional code such as data communication with the ground, simple vision algorithms or other data processing and reasoning.

The GNC loop starts by reading all sensors and running the drivers on all acquired data. This data is put through a Kalman Filter tuned for optimal speed. Then the navigation loop and control inner and outer-loops are run, finishing by sending the control commands to the actuators. The stress here is on minimal latency between the sensor data and the actuator command, since every delay of even a millisecond reduces the stability margin of the control loop. Then the control thread is put to sleep allowing the reasoning thread to execute for a limited amount of time.

3.3 Ground Control Station Library

The ground control station has basically three functions. First it forms the interface between operator(s) and the system, second it runs automated processes that aid the control of the air vehicles, and third, it enables communication with the flying vehicles.

The design of the first two functions is based on levels of sophistication identified by Amelink et al.[4]. This UAV system supports four levels of sophistication: flight level, aviation level, navigation level and the mission level. These levels define the support the system gives to the operator to function on that level. The operator can choose the level that is interacted with leaving the levels below to the automation. In a highly autonomous mode the operator will process information on a mission level, leaving the other levels to the automation. This is beneficial because it allows the operator to abstract away from the flight needs and focus on the mission needs that define the goals of the operation lowering the cognitive workload. When required by the situation e.g., during unanticipated events, the system support the operator to take control on a lower level, e.g., aviating or even flight level.

The ground control station has various functions as listed below. These function have been incorporated into several displays shown in Figure 9. The basis of all windows is the Fast Light Toolkit (FLTK) library[5] which runs under windows and Linux.

- Show status of the flight vehicle for security reasons.
- Collect differential GPS data to enhance the onboard position estimation.
- Edit, Show and Upload flight plans as lists of waypoints.
- Send joystick commands over the long range datalink for manual control.
- Give situation awareness to the operator.
- During the development phase, allow in-flight reconfigurations such as gain changes.
- Control vehicles on four levels of sphistication.

3.3.1 Instruments

The instruments window (Figure 9) shows the details of the onboard status for one vehicle. It includes a primary flight display, heading indicator and datatickers that show the frequency of the arrival of updated vehicle data. The instruments window has the following FCS control modes:

- commanded speed
- altitude hold
- vertical speed
- turn rate
- heading hold

These modes give full control over the flight of a stabilized air vehicle and represent the aviation level of sophistication. As an extra aid, when needed the main control mode can be switched to manual mode which gives stick control over the MAV. Joystick commands can be sent over the long range digital datalink to control the MAV manually on the flight level of sophistication.

3.3.2 Mission View

The mission view forms the heart of the interface. It is a graphical interface for mission planning and editing, see figure **??**. Missions and flight plans can be loaded, modified and saved to disk. Two levels of sophistication are supported in this window: mission level and navigation level. On mission level the operator can design a mission using mission elements. Mission elements specifically designed for the MAV07 competition represent each of the assignments that are intended to be completed: launch, drop sensor, urban canyon, identify target, search and land. Once the mission has been designed it can be converted to a flight plan that consists of series of waypoints that can be uploaded to the autopilot over the long range data link. Each waypoint has a position and altitude and the following attributes:

- Force altitude: MAV circles waypoint until it reaches commanded altitude before continuing flight plan.
- Force along leg: fly to next waypoint along the leg to that waypoint.
- Climb: aggressive climb mode.
- Descent: aggressive descent mode.
- Landing: cut engine and glide to next waypoint.
- Next way point: specifies the number of the next waypoint to fly to.
- Speed: specifies the air speed to fly to next waypoint.

The flight plan can be compiled by the GCS from a mission plan or manually by the operator. In either case the operator is supported to edit the flight plan as necessary in a graphical manner. During flight the operator can switch between navigation and mission representation to intervene if necessary by controlling mission elements, way points or, by going back to the instruments window, FCS commands and stick commands.

3.3.3 GCS View

The GCS as a system has its own status window that shows battery status of the laptop, the differential GPS activity status (if present) and datalink status showing the amount of CRC checksum errors. It provides a pop-up window showing the status of the GPS system including a graphical display with space vehicle information at that location at that time.

3.3.4 GainChanger Window

During the development phase, many flight parameters need to be changed in-flight. The so-called gain-changer can upload new parameters during flight. This feature is only used in the presence of a safety pilot with pilot-take-over switch to manually control the vehicle, but it turns out to be very valuable to speed-up the tuning of a new vehicle considerably. It includes parameters for the Kalman Filter and the controller. Figure 10 shows a shorted list of adjustable control gains.

3.3.5 Other

A variety of other windows can be used showing warnings, loading and saving files. When many flight vehicles are used simultaneously, a Fleet Overview window appears on top, showing summarized information of each vehicle and allowing the instruments window to show details of the selected vehicle. The wind display visible shown in Figure 9 plots the vehicles heading against speed in a polar diagram producing a representation proven valuable for wind estimation.

3.4 Simulation Library

An important feature of the developed software package is that before flight everything can be tested on the ground. The simulation library consists of simulated vehicle dynamics and simulated sensors running in a simulated environment. The simulated vehicle can be equipped with various simulated sensors such as inertial sensors, different types of GPS and actuator controllers that all stream exactly the same bytes as the real sensors to FCS channels. The configurations are loaded from XML using the TinyXml library. Finally an OpenGL graphics generator is added providing simulated video that can also be used to check the vision algorithms.

3.5 Delft Workspace for Autonomous Vision Library and Editor

Vision based control is thought to be an important part of future autonomous vehicle operations. But Vision is also computationally expensive on one hand and mission-specific on the other hand. Therefore a *highly modular* software library (DWAV) is developed based on OpenCV[6] that can be interconnected easily using a module editor. Figure 11 shows a screenshot of this very important gallery of modules that can be interconnected to detect various objects ranging from lines, edges, circles, colors, landing pads to even human faces or the horizon line[7] (See Figure 11). This library is used to find the “circular objects clearly visible from above” (Figure 12) or many more tasks such as terrain height estimation from optic flow, orientation estimation, etc . . .

3.6 Advanced Flight Management System Library

All flight code that does not just fly the vehicle along predefined waypoints is grouped into a library called *Advanced Flight Management System Library* (AFMS). This library uses de latter vision gallery and has extra navigation code to perform search missions, plan optimal routes, interact with other vehicles, check air traffic control zones and so on. This code can either be run onboard or on the ground and has the same modularity and easy of reconfiguration without loss of computation power as DWAV. The AFMS modules are interconnected with the same module editor as DWAV. Using the AFMS library together with DWAV, many tasks such as steering a vehicle to a specific target, performing search missions, mapping terrain elevation can be performed. Thanks to the extreme modularity of these libraries, code can easily be split to multiple computers, for instance doing simple color detection onboard and only transmitting interesting images to the ground where further validation and position estimation is performed.

4 Sensor Fusion and Control Algorithms

Attitude of a vehicle is a difficult yet particularly important part of autonomous flight, the problem lying in the fact that attitude is not easily and directly measurable.

The choice was made to use inertial attitude determination. One reason is that recent MEMS sensors become small and cheap enough for their application into 60cm span fixed wing vehicles. Another reason is that inertial MEMS sensors do not depend on external conditions as much as for instance infrared horizon detectors and are not as heavy as multi-antenna GPS solutions.

Besides inertial data, a *ublox* Smart Antenna Module (SAM) GPS receiver is used for guidance, navigation and control.

4.1 Control

Control of the vehicle is split into an inner-loop, an outer-loop and a navigation loop. The navigation loop provides a filtered commanded heading and an altitude to fly, based either on waypoints or on advanced flight management system commands in the case of a search pattern.

$$\Psi_{command} = \tan^{-1} \frac{wp_y - y}{wp_x - x} \tag{1}$$

The outer-loop converts this heading into a turn-rate command by subtracting the actual flight direction Ψ, saturating this $\Delta\Psi$ to about 20 degrees and multiplying with a gain.

The inner-loop finally converts the commanded turn rate and commanded vertical velocity into rudder and elevator commands.

$$\delta_r = K_{P_r} \cdot (r_{filt} - r_{comamnd}) \tag{2}$$

$$\delta_e = K_{P_{air}} \cdot (v_{air} - v_{air_{comamnd}} + K_{ff} \cdot r_{command}) \tag{3}$$

5 Vision

Vision code for position estimation of circles is based on edge detection, color detection, blob detection and hough transforms. These algorithms are written as modules in the formerly described workspace for autonomous vision DWAV (Figure 11).

After detection of features in the image, conversion is made to earth axes using a pinhole camera model, the estimated pitch and roll from the Kalman filter of the flight computer, the GPS coordinates and some triangulation routines[8]. Optic flow combined with ground speed from GPS (also referred as motion stereo) can significantly improve the range estimation at small distances since GPS altitude has shown to have larger offsets and since range is important in the localization routine.

The position estimates are fed into a new Kalman filter to estimate the static position of the target from the imagery, optic flow, GPS and attitude data.

6 Conclusion

Robust attitude determination is obtained combining inertial MEMS sensors and GPS. When correctly calibrated against temperature the gyroscopes and accelerometers provide very constant and high bandwidth attitude information, ideal for highly manoeuvrable small aircraft such as MAV. Moreover, to solve the short range video transmission problem, steps are taken to enable video processing onboard the micro aerial vehicle. After the initial simple video processing, only interesting parts need to be transmitted over a long range low bandwidth link. Finally adaptive elements are included in the controller to accommodate for changes in the vehicle dynamics, sensors and adapt gains to different weather conditions.

An autopilot board was developed to enable these technologies. The board has large computational capabilities and contains gyroscopes, accelerometers, GPS, USB for webcams and storage devices and has ethernet access. The total weight is a mere 185 grams with 0.5 hour of battery power included for a board of 12 by 4 centimeter, enabling a new range of MAV applications.

References

[1] Drouin, A. and Louis, “Paparazzi,” http://www.nongnu.org/paparazzi/, 2005.

[2] Rasmussen, J., *Information Processing and Human-Machine Interaction*, North-Holland, New York, 1986.

[3] Vicente, K. J. and Rasmussen, J., "Ecological Interface Design: Theoretical Foundations," *IEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 4, July/August 1992, pp. 589–606.

[4] Amelink, M., van Paassen, M., and Mulder, M., "Actor-Agent Collaborations: Abstractions for Interfacing Operators and UAVs," *Proceedings of Cognitive Systems with Interactive Sensors (COGIS'06)*, SEE, Paris, France, March 2006.

[5] FLTK, "Fast Light Toolkit," http://www.fltk.org/, 2005.

[6] OpenCV, "Open Computer Vision Library," 2003.

[7] Ettinger, S. M., Nechyba, M. C., Ifju, P. G., and Waszak, M., "Vision-Guided Flight Stability and Control for Micro Air Vehicles," *Proc. IEEE Int. Conference on Intelligent Robots and Systems*, Oct 2002.

[8] Wagter, C. D., Proctor, A., and Johnson, E., "Vision-Only Flight Control," *22nd Digital Avionics Systems Conference*, 2003.

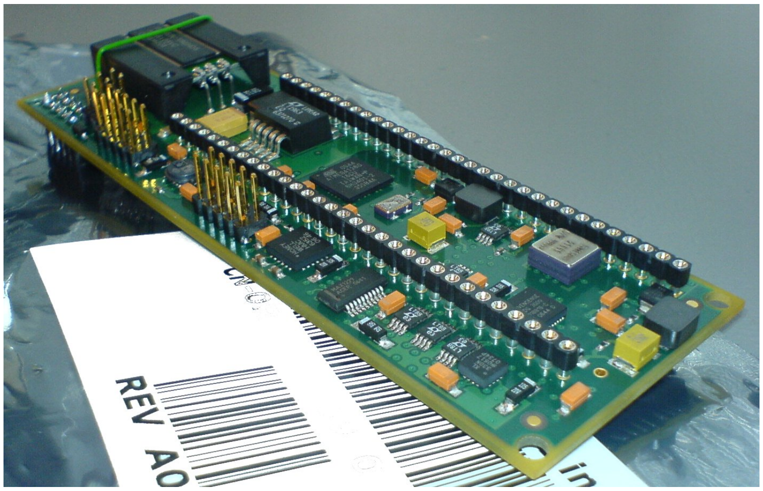


Figure 3: The MAVPilot3 board.

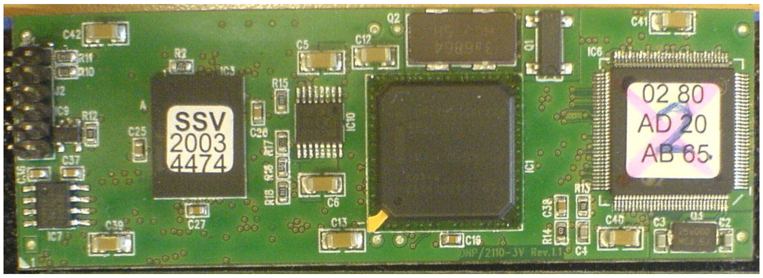


Figure 4: The processorboard with 180MHz 32bit AT91C9200 ARM processor from ATMEL that will be piggybacked to the MAVPilot3 board

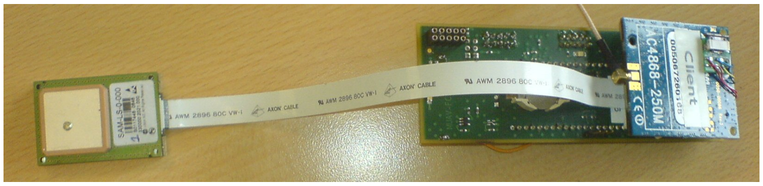
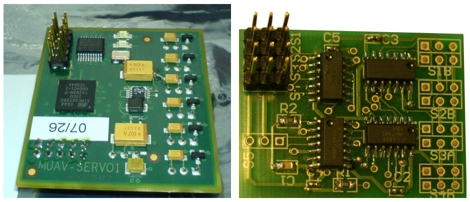


Figure 5: The long range modem and GPS module connected to the MAVPilot3



(a) Servocenter (b) Pilot-Take-Over-Switch

Figure 6: Servocenter and pilot-take-over switch are separate boards connected to the MAVPilor3 board.

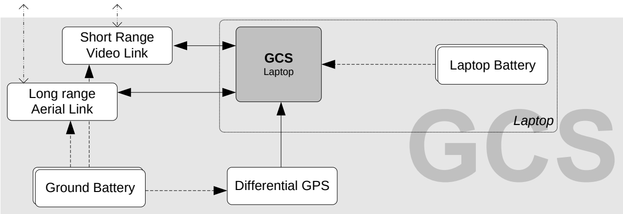


Figure 7: Functional overview of the ground control station. Depending on the configuration multiple laptops can be used to do onground image processing or postprocessing when the short-range (300m) analog video transmission is available.

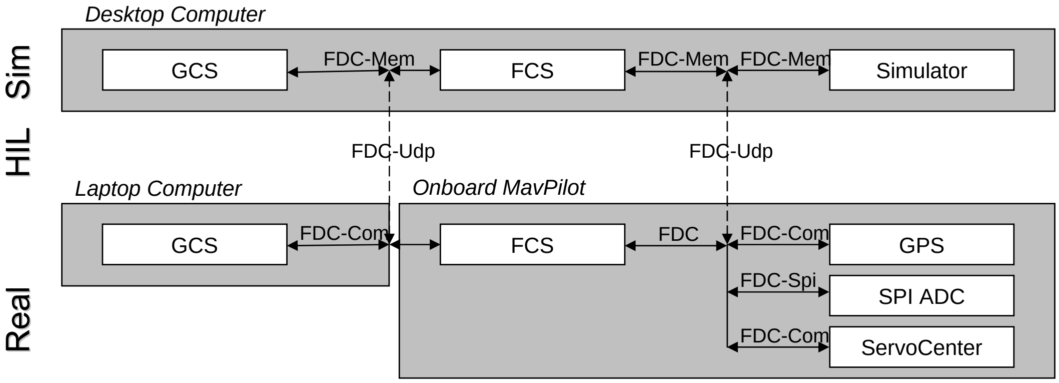
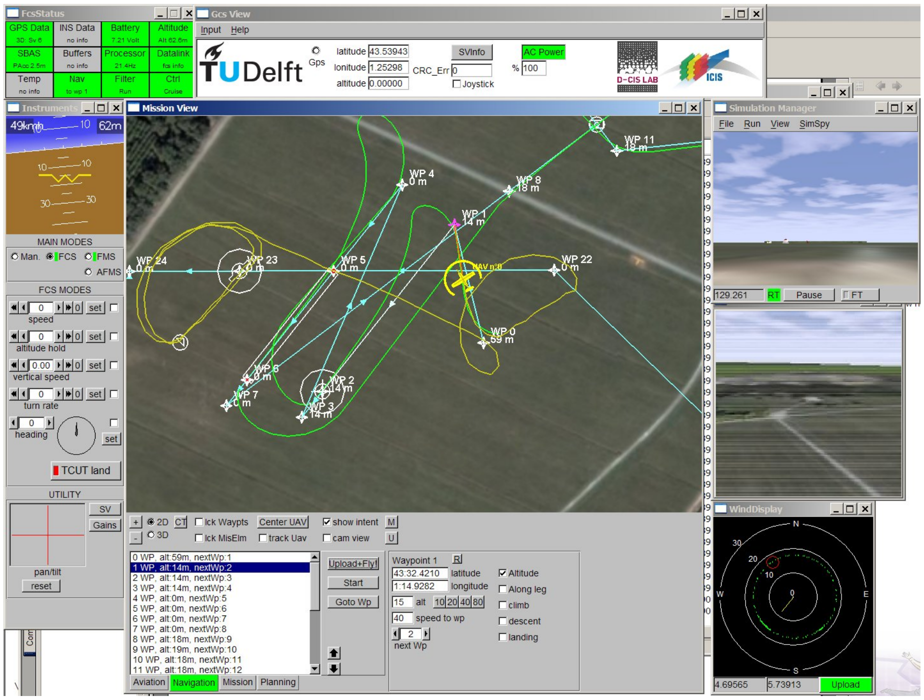


Figure 8: Flexible Data Channels allow seamless interconnections of libraries in either full simulation mode, flight mode, or any Hardware-In-the-Loop (HIL) combination without the need of recompiling.



GPS Data 3D: Sv 6 | INS Data no info | Battery 7.21 Volt | Altitude Alt 62.6m
SBAS PAcc 2.5m | Buffers no info | Processor 21.4Hz | Datalink fcs info
Temp no info | Nav to wp 1 | Filter Run | Ctrl Cruise

○ Gps
latitude 43.53943
lonitude 1.25298
altitude 0.00000
CRC_Err 0
□ Joystick
AC Power
% 100

49km/h 62m

○ Man. ● FCS ○ FMS
○ AFMS

0 speed
0 altitude hold
0.00 vertical speed
0 turn rate
0 heading

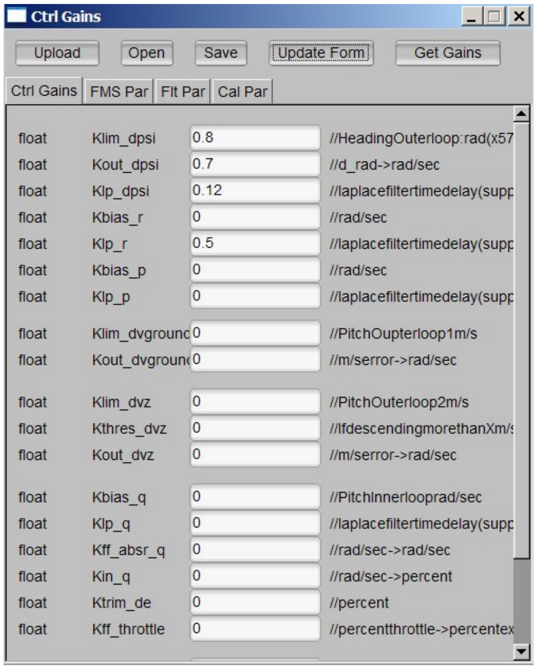
● 2D ○ 3D
□ lck Waypts □ lck MisElm
□ track Uav
☑ show intent □ cam view

0 WP, alt:59m, nextWp:1
1 WP, alt:14m, nextWp:2
2 WP, alt:14m, nextWp:3
3 WP, alt:14m, nextWp:4
4 WP, alt:0m, nextWp:5
5 WP, alt:0m, nextWp:6
6 WP, alt:0m, nextWp:7
7 WP, alt:0m, nextWp:8
8 WP, alt:18m, nextWp:9
9 WP, alt:19m, nextWp:10
10 WP, alt:18m, nextWp:11
11 WP, alt:18m, nextWp:12

Waypoint 1
43:32.4210 latitude
1:14.9282 longitude
15 alt 10 20 40 80
40 speed to wp
2 next Wp
☑ Altitude
□ Along leg
□ climb
□ descent
□ landing

129.261 RT
4.69565 5.73913

Figure 9: Ground Control Station containing the Instruments window showing flight data and FCS control modes, the Mission view with the map which is the heart of the control interface, FCS status window, the simulation manager, wind display and simulated camera view and the GCS window showing status information about the differential GPS connected to the ground control station.



float	Klim_dpsi	0.8	//HeadingOuterloop:rad(x57
float	Kout_dpsi	0.7	//d_rad->rad/sec
float	Klp_dpsi	0.12	//laplacefiltertimedelay(supp
float	Kbias_r	0	//rad/sec
float	Klp_r	0.5	//laplacefiltertimedelay(supp
float	Kbias_p	0	//rad/sec
float	Klp_p	0	//laplacefiltertimedelay(supp
float	Klim_dvgrounc	0	//PitchOupterloop1m/s
float	Kout_dvgroun(	0	//m/serror->rad/sec
float	Klim_dvz	0	//PitchOuterloop2m/s
float	Kthres_dvz	0	//IfdescendingmorethanXm/
float	Kout_dvz	0	//m/serror->rad/sec
float	Kbias_q	0	//PitchInnerlooprad/sec
float	Klp_q	0	//laplacefiltertimedelay(supp
float	Kff_absr_q	0	//rad/sec->rad/sec
float	Kin_q	0	//rad/sec->percent
float	Ktrim_de	0	//percent
float	Kff_throttle	0	//percentthrottle->percentex

Figure 10: Gain changer window allows adjustments to flight control gains to be uploaded in flight.

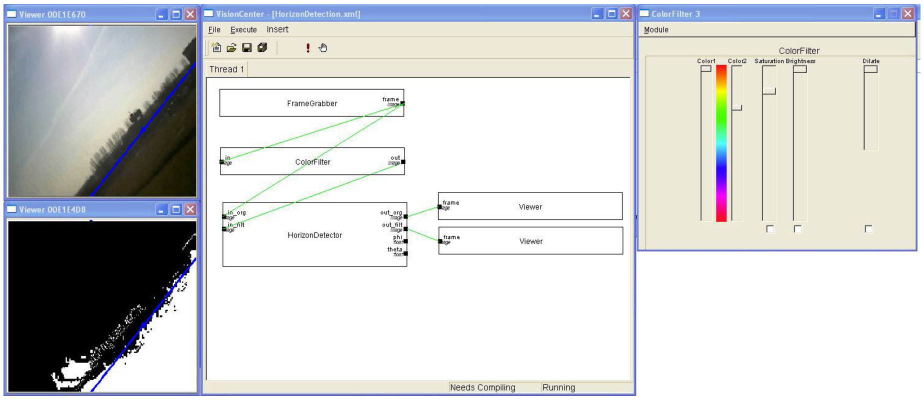


Figure 11: Vision Center

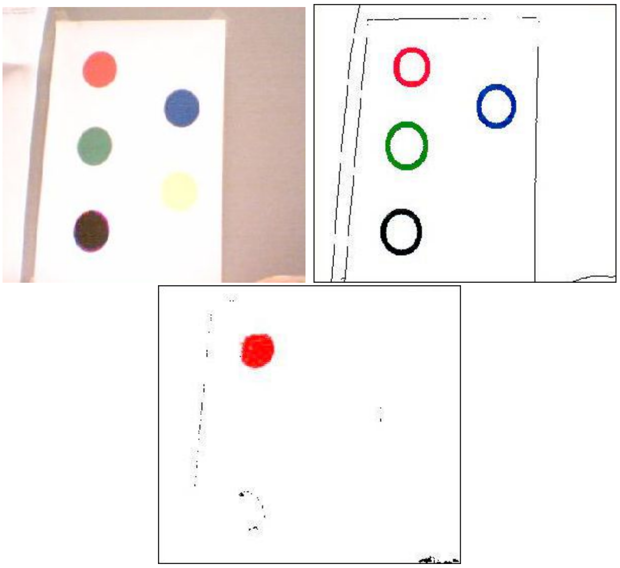


Figure 12: Red Circle Detection Using a combination of blobs, color and hough circle detections.