AirGuardian - UAV Hardware and Software System for Small Size UAVs

Regular Paper

Dániel Stojcsics¹,* and András Molnár²

1 Óbuda University, Doctoral School of Applied Informatics, Budapest, Hungary
2 Óbuda University, John von Neumann Faculty of Informatics, Budapest, Hungary

* Corresponding author E-mail: stojcsics.daniel@nik.uni-obuda.hu

Received 11 Jun 2012; Accepted 27 Aug 2012

DOI: 10.5772/52759

© 2012 Stojcsics and Molnár; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract AirGuardian is a complex of an Unmanned Aerial Vehicle and ground station UAV hardware and software systems which has been developed at Obuda University. The hardware and software of the autopilot (AERObot), the antenna tracker station and the ground control station were simultaneously created resulting in the optimal cooperation of the modules.

The aim of the research on AERObot - the special autopilot - was to create a generic autopilot which is capable of controlling different designs, weights and structures of airframes without any complex mathematical model recalculation.

Keywords Unmanned Aerial Vehicle, Navigation, Ground Control Station, Virtual Cockpit, Flying Wing

1. Introduction

AirGuardian was created to meet these criteria:
Antenna tracker unit

• Compact unit without any other wired module
• Automatic self-calibration before flight

• Autonomous tracking of the UAV
• Ground Control software connectivity

Ground control software
• Multi-user server-client architecture with different capabilities (navigator, observer mission planner and pilot)
• Full mission record and replay features
• Support of custom-calibrated maps

AERObot (onboard hardware and software)
• Compact, small size and low weight electronics, without wired off-board sensors
• Module-based design with slot based architecture to achieve task specific application
• Universal communication interface allowing the usage of both wireless (different RF modems) and wired (FTDI USB) connectability
• Autopilot can store at least 200 waypoints, the parameters of planes and control functions in a non-volatile memory
• Fully autonomous operation (telemetry issues should not affect the flight)
1.1 Antenna tracker unit and ground control station

The ground electronics of the system are in the antenna tracker unit. The main task of the unit is involves RF communication with the UAV using directed high gain antennas and the wired communication with the ground control PC.

The tracker unit (fig. 1) also has an onboard GPS so it is not only able to track the UAV from a single position but so that it also possible to use it as a mobile ground station. The tracker unit has two degrees of freedom, with limitless rotation on the yaw angle [1].

![Diagram of Antenna Tracker](image)

**Figure 1.** Antenna tracker

1.2 Ground control software

The ground control software is based on client-server architecture. The server is connected to the antenna tracker, which grants the telemetry and camera data. It can handle serial ports or TCP/IP packets. Multiple clients can connect to the server from one or many PCs. The server also relays the telemetry to the Wi-Fi, allowing AirGuardian Mobile (smartphone version) to access mission data (fig. 2).

The users are basically using the client software, communicating with the server through TCP/IP using wired or wireless networks. The users have authentication in the clients, which grants them the proper permissions (pilot, navigator, observer and admin roles). They can see the data related to their task only. For example, the pilot can control the UAV remotely using the front camera but they cannot modify autonomous mode waypoints or altitude as the navigator can. The client interface is customizable for each (new) role.

![Diagram of Ground Control Software](image)

**Figure 2.** AirGuardian ground control software layout

The software is highly scalable. It can be used as a multiple workstation with one dedicated server and multiple role-based clients or else in a single, low-cost notebook with an admin role. The AirGuardian can also record and replay missions in real time. Multiple 2D and 3D maps can be imported, enabling multilayer maps and 3D satellite image-based mission planning.

1.3 Onboard electronics

The onboard electronics have two modules for a general UAV application: the AERObot main board and the Servo board. The main board has all the units, with the exception of the IMU (inertial measurement unit), which are needed for autonomous flights. This means that only
the IMU, the GPS antenna and the engine battery (for power consumption) is connected to the main board electronically. The other connectors are the static and dynamic pressure tubes from the Pitot-tube, which grants the airspeed and altitude parameters. Expander modules (e.g., the Servo panel) can be connected to the main board through two row headers. Additional electric sensors can be attached to the board (an ultrasonic altimeter, a revolution sensor, a temperature sensor, etc.), but these are not essential parts for a general UAV application.

This can be activated with an RC transmitter switch on a free channel or a with a telemetry command. For flight safety purposes, the board can handle two different RC receivers in diversity, which is important in manual control mode during the setup procedure. When the signal of a receiver is weak, the board automatically switches to the other receiver. When both lose signal, the board switches to autonomous stabilized flight.

The boards (main and servo) can control a general purpose UAV. Both the airplane and the payload can also be controlled (i.e., camera turn on/off, record, follow target, etc.). It is possible to attach further boards [2] to the main or servo board, such as the sensor board (analyse air composition, radiation, etc.), or else to further main boards (allowing redundant UAV control) (fig 5.).

AERObot saves all the parameters to a non-volatile memory, which means that after turning on the UAV and the GPS locks position, the system is then ready to use. The waypoints and mission directives can be set before the mission and can also be modified during flight.

**Figure 3. AERObot main board**

Parameters of the AERObot 5.0 main board (fig. 3):

- Redundant power supply (with redundant 2-4S LiPo batteries)
- Voltage sensing on batteries
- GPS sensor with 10Hz update rate
- Barometric altimeter with automatic calibration
- Barometric airspeed meter with Pitot-tube
- MicroSD memory card connector for data logging
- Power consumption sensor up to 100A
- 6+2 channel PWM output for servo drive
- Serial port for radio modem (telemetry and setup)
- FTDI USB port (telemetry and setup)
- Serial port for IMU (Both 3.3 and 5V with TTL or RS232 signal levels)
- CAN bus
- PWM input (ultrasonic altimeters or the revolution sensor)
- Extender boards (e.g., the servo board or the sensor board)
- External GPS port
- 3D waypoints (position, altitude, airspeed and actions)
- Autonomous waypoint navigation
- Full in-flight mission control (waypoints, parameters, controllers, etc.)
- 2 SPI channels
- 1 I2C channel

The servo board was made as an interface to connect AERObot to a common radio control plane easily. The main function of the board is to relay the control signals created by the main board to the actuators of the plane. Furthermore, the sensor board can switch the controls between manual, heterogeneous and autonomous at circuit level without modifying the connections between the boards and the actuators.

**Figure 4. AERObot servo board**

Parameters of the AERObot Servo board (fig. 4):

- Dual receiver remote control
- Autonomous, heterogeneous (8 preset autonomous-manual mixed control), manual mode
- 6+2 channel PWM output for the servo drive
- 1A onboard BEC for the low power actuators
- High current BEC connector for external actuator power
- Optional xBeel, xStream and xTend radio modem connector with FTDI USB support
- Modem and RC receiver RSSI sense
- OSD (On Screen Display and character mode SPI controller)
- Revolution meter input
- 4 switch relay output
- 4 ADC input.
2. AERObot waypoint navigation

There are many navigation approaches. Most of them use waypoints [3], defined by a geographic position (latitude and longitude), an altitude (above ground level or above sea level) and a waypoint radius (usually 25-75m is used to achieve precise path tracking). A waypoint is reached, when the distance between the UAV and the waypoint is under the predefined radius value. For precision navigation, the autopilot needs very accurate position and bearing information.

2.1 Bearing estimation

The base sensor for navigation is the GPS. This provides the position, the speed over ground (SOG), the bearing and many other parameters used by the navigation algorithms. The maximum position refresh rate of the best GPS modules is about 5-10Hz, which is not enough for precision flight control.

The other problem is that the refresh rate is not reliable because the NMEA sentences and the checksums are often incorrect. It is necessary to estimate the missing and intermediate positions for the continuous bearing control (100Hz - update frequency of the most commercial RC actuators). This estimation can be computed using the yaw rate and the GPS bearing from the last valid NMEA sentence. The Z-axis angular rate (provided by the IMU) should be converted to the heading turn rate using the UAV bank angle [4]. With this procedure, the actual bearing can be estimated between two valid GPS sentences (1). The heading angle provided by the IMU is often unreliable because of the magnetic sensor used by the internal sensor fusion algorithms. This sensor is very sensitive to strong electromagnetic fields, especially when the UAV has electric propulsion.

\[ \theta_e = \theta + \psi \]  

where:

\( \theta \): previous bearing (measured by GPS), \( \theta_e \): estimated bearing,  
\( \psi \): heading turn rate.

Small sized (around 1 m wingspan) and low weight (under 2 kg) UAVs are quite agile and so turn rates can reach high levels. For these airframes, bearing estimation is essential (fig 6).

2.2 Position estimation

During navigation, both the bearing and the position have to be estimated with a navigation formula (2), which is well known in nautical terms [5]. The aim is not for a complete inertial navigation system with minimal error but rather the computation of intermediate positions and bearings in order to achieve better performance. As such, some things can be ignored [6, 7, 8].

\[ \text{Lat}_2 = \text{asin}(\sin(\text{Lat}_1) \cdot \cos\left(\frac{d}{R}\right) + \cos(\text{Lat}_1) \cdot \sin\left(\frac{d}{R}\right) \cdot \cos(\theta_e)) \]

\[ \text{Lon}_2 = \text{Lon}_1 + \text{atan}2(\sin(\theta_e) \cdot \sin\left(\frac{d}{R}\right) \cdot \cos(\text{Lat}_1) \cdot \cos\left(\frac{d}{R}\right) - \sin(\text{Lat}_1) \cdot \sin(\text{Lat}_2)) \]  

(2)

where:

\( \text{Lat}_1 \): previous estimated latitude of the UAV,

\( \text{Lon}_1 \): previous estimated longitude of the UAV,

\( \theta_e \): estimated bearing (in radians, clockwise from north),

\( d \): distance travelled,

\( R \): The Earth's radius.

Usually, there are 200ms between two valid GPS sentence bursts, but the distribution of the measured positions is not uniform. Even if one or two is corrupted, the refresh time is always under 1000ms. Because of this, the direction and velocity of the wind can be ignored. During the estimation, the last known SOG can be used no matter how the bearing of the UAV changed (compared to the wind). Obviously, there will be some errors, but usually it is low (under 1m) and will not cause any serious false position estimation.

The estimation (Fig. 7) can be calculated from the previous known SOG, estimated position and bearing (when the known position and bearing are unavailable).
2.3 Vector field navigation

The vector field navigation calculates a desired bearing in every position, no matter what the actual bearing of the UAV is. This can be easily represented as a vector field. The desired bearing depends on the position of the UAV, the source and destination waypoints and the cross track error (3).

\[ \delta = \kappa \sqrt{D_{CT} + K_c \cdot (\varphi_T - \varphi_d)} \cdot \text{sign}(D_{CT}) \]

\[ \gamma = \min(1, D_T) \]

\[ \varphi_d = \varphi_T + \delta \cdot \gamma \]

(3) (3)

where:

- \( D_{CT} \): Cross track error,
- \( \varphi_d \): Desired bearing,
- \( \varphi_T \): Bearing from the UAV to the destination waypoint,
- \( K_c \): Path reaching smoothness parameter,
- \( K_c \): Cross track error sensitivity parameter,
- \( \gamma \): Waypoint bearing sensitivity parameter.

The cross track error sensitivity can be set with \( K_c \) parameter (fig. 8) and the path reaching smoothness (fig. 9) with \( K_d \). It is a common problem that a UAV will miss the waypoint if the waypoint radius is low and there is a strong wind or else the distance between the source and the destination points is short. In this case, the UAV should turn back and the flight path will contain unnecessary loops. The first possibility for avoiding this problem is the dynamic radius which must suit local airflow characteristics.

The alternative - and better - solution makes use of \( \gamma \). If the UAV is close to the destination point (under 1000m), the parameter \( \gamma \) guarantees arrival at the waypoint, ignoring \( \delta \) depending upon the UAV – waypoint distance (fig. 10).
The first advantage of this form of navigation is the accessible graphical setup of the internal parameters $K$ ($-10.0$) and $K_c$ ($-0.5$). The other is that it has only one output parameter ($\varphi_d$) and so nearly any controller type (such as third-order nonlinear, PID, fuzzy, etc.) can fit this navigation because only one value - the difference between the desired and actual bearing - should be minimized. Furthermore, and with a little modification (4), it is capable of loiter navigation over a desired position (fig. 11). This function is quite useful when the UAV is observing a destination position with a stabilized electro-optical or thermal imaging camera.

$$\varphi_d = \varphi_T + \frac{\pi}{2} + \min \left( D_{CT} \cdot K_{e_{\varphi} \frac{\pi}{2}} \cdot \text{sgn}(D_{CT}) \right) \quad (4)$$

where:

$D_{CT}$: Cross track error (distance from the radius),

$\varphi_d$: Desired bearing,

$\varphi_T$: Bearing from the UAV to the destination waypoint,

$K_e$: Cross track error sensitivity parameter.

The drag rudder approach used by AEROBot is based on a conventional model glider brake system - the “butterfly” (fig. 12) - which does not need any further mechanical control surfaces or actuators, using only the four control surfaces on the main wing. The inner two control surfaces (the flaps) are open down around 45-80°, while theouters (the ailerons) are open up by 5-45°, with a little elevator compensation on the tail. Based on the butterfly brake, a new drag rudder was developed (fig. 13, 14, 15). On the wing, there are four control surfaces - the inners are the elevators, the outers are the ailerons - which are all set to +3-4 mm up, as is usual on flying wings.

**Figure 11.** Loiter navigation

3. Flying wing extension

Flying wings has no vertical or horizontal control surfaces - only elevons, which are a mix of elevator and rudder. This means that flying wings generally have no control along the vertical axis. However, some approaches exist to achieve vertical control. These are the so-called “drag rudders”, which are usually represented by asymmetric operated airplane dive brakes, such as Scheppe-Hirth or the trailing edge brakes used on Horten 229 and Northrop YB-35.

**Figure 12.** Butterfly brake on a model glider

**Figure 13.** Ailerons on a flying wing from behind (left roll, straight flight, right roll)

**Figure 14.** Elevators on a flying wing from behind (nose up, straight flight, nose down)

**Figure 15.** Drag rudder on a flying wing from behind (left turn, straight flight, right turn)

On turns, the elevator moves up by 100%, the elevator down by 50% of the rudder command on the desired side (5).

$$D_L \begin{cases} 1, & \text{if } \varphi_{\text{Elev}} < 0 \\ 0, & \text{otherwise} \end{cases}$$

$$D_R \begin{cases} 1, & \text{if } \varphi_{\text{Elev}} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\varphi_{\text{Ele}}; \varphi_{\text{Elev}} \rightarrow \frac{\varphi_{\text{Rudd}}}{2} \cdot D_L$$

$$\varphi_{\text{Elev}} = \varphi_{\text{Ele}} - \varphi_{\text{Elev}} \rightarrow \frac{\varphi_{\text{Rudd}}}{2} \cdot D_L$$

$$\varphi_{\text{Ail}}; \varphi_{\text{Ail}} \rightarrow \frac{\varphi_{\text{Rudd}}}{2} \cdot D_L$$

where:

$\varphi_{\text{Ele}}$: Left side elevator control surface deflection

$\varphi_{\text{Elev}}$: Right side elevator control surface deflection

$\varphi_{\text{Ail}}$: Left side aileron control surface deflection

$\varphi_{\text{Ail}}$: Right side aileron control surface deflection

$\varphi_{\text{Rudd}}$: Aileron deflection command

$\varphi_{\text{EleCMD}}$: Elevator deflection command

$\varphi_{\text{RuddCMD}}$: Rudder deflection command

4. Experiences of the AirGuardian system

The setup procedure for a new plane is always begun with a manual test flight so as to examine the flight abilities and preset the control surfaces. Next, the control surface idle positions for the autopilot must be adjusted. The control loops must be set up independently first and then fine tuned during flight.
Fig. 16 represents an altitude plot of the first test flight of Chelidon UAV. It contains a manual take-off and multiple heterogeneous/autonomous flights. The red parts are manual control. In the early stage of the flight, the independent tuning of the control loops was made through the AirGuardian ground control software. During this period, the UAV was flown in heterogeneous mode 6 times. After the independent tuning, there were 4 autonomous flights when the combined control loop fine tunings were made. The first three flights were short, because the controllers were not well-tuned. The last autonomous part was a successful fine tuned flight, with many laps on a three waypoint route (fig. 17). The whole procedure took 30 minutes.

**Figure 16. Altitude diagram - Chelidon first test flight (red: manual, white: heterogeneous, green: autonomous flight)**

This short time is remarkable since Chelidon was a brand new custom built prototype airplane, and so no preliminary flight data was available. The experiment confirmed that AERObot is robust and that the setup procedure is quick and effective.

| UAV         | Propulsion                                         | All up weight | Wingspan |
|-------------|----------------------------------------------------|---------------|-----------|
| Tiger 60    | 80 cc m two-stroke internal combustion             | 4.5 kg        | 1.5m      |
| Sky walker  | Electric 500W brushless motor with 8Ah 11.1V LiPo | 3.5 kg        | 1.6m      |
| BHE – 03    | Electric 1300W brushless motor with 10Ah 22.2V LiPo| 16 kg         | 3.5m      |
| Fun Cub     | Electric 500W brushless motor with 3Ah 11.1V LiPo  | 1.5 kg        | 1.4m      |
| Chelidon    | Electric 1300W brushless with 6:3 reduction unit and 11Ah 22.2V LiPo | 7 kg         | 3.6m     |
| Flying Wing | Electric 200W brushless motor with 3Ah 11.1V LiPo  | 1.4 kg        | 1.6m      |

**Table 1. UAVs applied with AERObot and tested with AirGuardian system**

5. Conclusions

Various experiments proved that AirGuardian met the criteria targeted at the beginning of the research. The onboard electronics and redundant systems were working reliably. The switch between manual, heterogeneous and autonomous is less faulty and transient.

The antenna tracker is capable of tracking the UAV in real time. After the GPS lock, it automatically turns to the UAV and starts following it. In case of telemetry loss, the antenna turret can be controlled manually for the plane with a remote.

The UAVs used to test AERObot and AirGuardian are in table 1. The examination displayed the abilities of the system using multiple, different sized and shaped UAVs. Comprehensive ground control software was developed which can effectively serve the UAV and the ground personnel. It can be used as a single or as a multiple workstation or even as a mobile application (fig. 18).

![Figure 17. Chelidon flight route in the AirGuardian Navigator view (partial)](image-url)
As a result, the various parts of the system can be used as standalone units - e.g., an extension module to another existing system or as a part of a whole new development.

6. Acknowledgments

The authors gratefully acknowledge the grant provided by project TÁMOP-4.2.2/B-10/1-2010-0020 and the support of the scientific training, workshops and established talent management system at Óbuda University.

7. References

[1] D. Stojcsics and L. Somlyai: “Improvement methods of short range and low bandwidth communication for small range UAVs”, 8th IEEE International Symposium on Intelligent Systems and Informatics, ISBN: 978-1-4244-7394-6, Subotica, Serbia, September 10-11, 2010, pp. 93-97.

[2] D. Stojcsics: “Heterogeneous control of small size unmanned aerial vehicles”, Proceedings of 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, ISBN: 978-963-7154-96-6, Budapest, Hungary, 2009, pp. 745-752.

[3] R. Beard, et al.: “Autonomous Vehicle Technologies for Small Fixed-Wing UAVs”, Journal Of Aerospace Computing Information and Communication (2005) vol. 2, no. 1, pp. 92-108, ISSN: 19403151.

[4] S. Leven, J. Zufferey and D. Floreano: „A minimalist control strategy for small UAVs“, in Proc. IROS, 2009, pp. 2873-2878.

[5] F. Ucan, D. T. Altilar: “Navigation and Guidance Planning for Air Vehicles”, 20th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2008, vol. 2, pp. 534-538.

[6] Q. Li, Z. Fang and H. Li: “The Application of Integrated GPS and Dead Reckoning Positioning in Automotive Intelligent Navigation System”, Journal of Global Positioning Systems (2004) vol. 3, no. 1-2, pp. 183-190.

[7] P. Davidson, J. Hautamäki and J. Collin: “Using Low-Cost MEMS 3D Accelerometers and One Gyro to Assist GPS Based Car Navigation System”, Proceedings of 15th Saint Petersburg International Conference on Integrated Navigation Systems, May 2008.

[8] L. Zhao, W. Y. Ochieng, M. A. Quddus and R. B. Noland: “An Extended Kalman Filter algorithm for Integrating GPS and low-cost Dead reckoning system data for vehicle performance and emissions monitoring”, Journal of Navigation, 2003, volume 56, issue 2, pp. 257-275.