Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

AUTONOMOUS FLIGHT EXPERIMENT WITH A ROBOTIC UNMANNED AIRSHIP

Josué Jr. G. Ramos+, Ely Carneiro de Paiva+, José Raul Azinheira*, Samuel Siqueira Bueno+, Silvio Mano Maeta+, Luiz Gustavo Bizarro Mirisola+, Marcel Bergerman+, Bruno Guedes Faria+

+ Information Technology Institute
Caixa Postal 6162 – 13083-970 – Campinas – Brazil
ssbueno@ia.cti.br

* Instituto Superior Técnico – IDMEC / IST
Av. Rovisco Pais 1049-001 Lisboa - Portugal
jraz@dem.ist.utl.pt

Abstract

Project AURORA aims at the development of an unmanned airship capable of autonomous flight over user-defined locations for aerial inspection and imagery acquisition. In this article the authors report a successful autonomous flight achieved through a set of pre-defined points, one of the first of its kind in the literature. The guidance control strategy is based on a path tracking error generation methodology that takes into account both the distance and the angular errors of the airship with respect to the desired trajectory. The control strategy uses a PI controller for the tail surfaces' deflection.

1. Introduction

A great interest in the utilization of unmanned aerial vehicles appeared in the last decade, due to their potential application in varied tasks such as surveillance, advertising, monitoring, inspection, exploration, and research roles [7]. A new and special attention has been given to the use of unmanned aerial vehicles in environmental applications, such as bio-diversity, ecological, climatological, and agricultural research and monitoring, among others.

In this context the authors are currently developing Project AURORA – Autonomous Unmanned Remote Monitoring Robotic Airship – which focuses on the development of the sensing, control, navigation, and inference technologies required for semi-autonomous operation of unmanned robotic airships for aerial inspection [7], [11], [12], [13], [14]. AURORA is conceived as a multi-phase project, the current one being AURORA I, which aims at establishing the underlying technologies and at performing low demanding applications.

The lighter-than-air platform of AURORA I is the AS800 by Airspeed Airships, a non-rigid, 9 m long, 2.25 m in diameter, 24 m^3 airship (Figure 1). The airship control actuators are its deflection surfaces and two-stroke internal combustion engines. The four deflection surfaces at the stern, arranged in an 'X' shape, generate the equivalent rudder and elevator commands of the classical '+' tail, with allowable deflections situated in the range -25 to +25 deg. The two engines on the sides of the gondola can be vectorized from 0 to +90 degrees up.

Figure 1: AURORA I airship AS800.

This article reports a successful airship autonomous flight achieved through a set of pre-defined points, one of the first of its kind in the literature. This flight was achieved with a PI control-based guidance strategy for the trajectory path following. The horizontal trajectory is controlled automatically by the onboard system, while altitude is controlled manually by the ground pilot.

The article is organized as follows: firstly, the authors present a path tracking error generation methodology combining distance and angular errors with respect to a given reference trajectory. Afterwards, the PI controller for the tracking regulation problem is presented. The authors then detail the experimental airship, including its onboard and ground hardware and software, and finally present the main result, an actual experimental flight.

2. Airship Dynamic Model

The AS800 aerodynamic model was adapted from the wind tunnel database acquired to model Westinghouse's YEZ-2A airship [9]. The adaptation was possible due to the same length/diameter ratio (4:1) of both airships [8].

The airship dynamic model can be described as:

$$M\, dx_A/dt = F_d(x_A) + F_a(x_A) + P + G \qquad (1)$$

where M is the 6x6 mass matrix and includes both the airship's actual inertia as well as the virtual inertia elements associated with the dynamics of buoyant vehicles; x_A is the vector of state variables, with the linear and angular airship velocities; F_d is the 6x1 dynamics vector containing the Coriolis and centrifugal terms; F_a is the 6x1 aerodynamics forces and moments vector; P is the 6x1 vector of propulsion forces and moments; and G is the 6x1 gravity vector, which is a function of the difference between the weight and buoyancy forces. For further details the reader is referred to [8].

Using this 6 DOF nonlinear model, a SIMULINK-based control system development environment was built to allow the design and validation of control and mission strategies [5].

3. Path Tracking Problem

One important mission problem is the flight path following of the vehicle through a set of pre-defined points in latitude/longitude.

The problem of path tracking for autonomous vehicles appears as a good application for very different control strategies [1], [4], [10], [15].

In this section we present the trajectory path following problem and describe the details of the straight line path following and the trajectory error generator, introduced in [2].

3.1. Controller Design Objective

Path tracking is a typical regulation problem, where one looks for a command input able to reduce the path tracking error for a given mission path.

Allowable mission paths are defined as a sequence of straight lines and circle arcs between the given way-points. The heading change at each way-point (between consecutive segments) may vary in the ±180° range and the distance between the actual airship position and the mission path is to be minimized in all cases.

The longitudinal motion is maintained at a constant altitude and airspeed and acceptably decoupled from the lateral motion, which is a common assumption in aerial vehicle control.

3.2. Linearization for Constant Speeds

The variables used in the path tracking control are depicted in Figure 2, where δ is the distance error to the desired path, ε is the angular error, V is the ground speed, and ψ_{traj} is the heading angle of the trajectory with respect to the north-east reference frame.

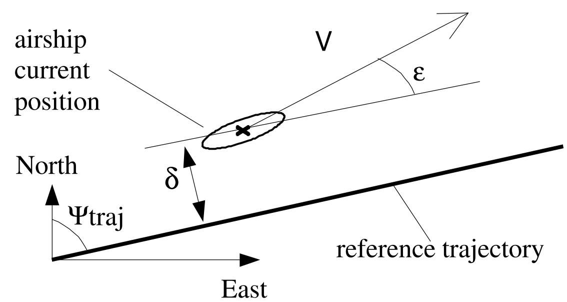

Figure 2. Path tracking signals.

For constant speed and small angular error ε, the following simplified path tracking linearized model results:

$$d\delta/dt = V \sin(\varepsilon) \approx V_o\, \varepsilon \qquad (2)$$
$$d\varepsilon/dt = R$$

where Vo is the reference ground speed considered for design purposes and R is the yaw rate.

In order to accommodate both the distance and angular errors in a single equation, a look-ahead error, δ_a, may be estimated some time ahead of the actual position:

$$\delta_a \approx \delta + V_o . \Delta t . \varepsilon \qquad (3)$$

where Δt is the prediction horizon. This strategy has already been successfully used for the guidance of both unmanned aircraft [10] and ground mobile robots [4].

4. Guidance Control Strategy

In this section we present a proportional-integral path tracking control method based on our previous work [2]. This guidance methodology is part of AURORA's overall airship control architecture, described in [5]. The control structure consists of a heading control inner loop and a path-tracking outer loop. It is shown in Figure 3, where ψ is the airship heading (yaw) angle. The control signal is the rudder deflection (ζ).

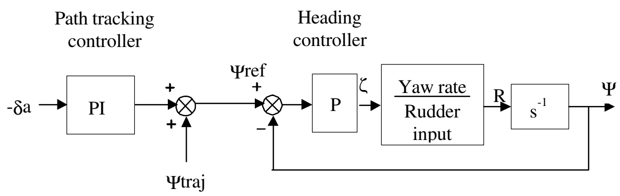

Figure 3. PI control block diagram.

The heading controller is composed solely by a proportional controller. The path-tracking controller is a proportional plus integral (PI) controller whose output, added to the trajectory heading angle ψ_{traj}, yields the reference signal ψ_{ref} for the heading controller, described in [5], [6]. The PI controller input is the opposite of the look-ahead path tracking error δ_a given in equation (3). The idea is that the PI controller uses the tracking error (δ_a) to correct the reference signal for the heading controller, with the necessary correction forcing the tracking error to decrease.

The PI controller uses an anti-wind up strategy to avoid saturation of the integral term.

5. Experimental Platform

AURORA I experimental platform is composed by the AS800 airship shown in Figure 1, and an onboard system, a ground station, and a communication system.

5.1 Hardware Infrastructure

The onboard infrastructure is composed by the following main components (Figure 4): CPU, sensors, actuators, and part of the communication system. Most of the components are mounted inside the airship's gondola. Figure 5 presents a pictorial representation of the airship with its actuators and onboard sensors, described in the sequel.

Figure 4: Onboard CPU and sensors.

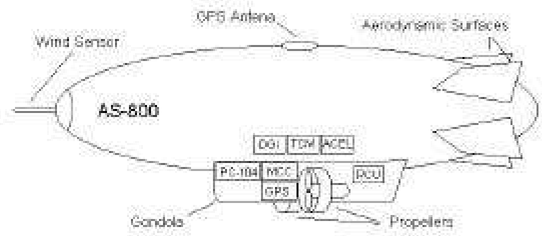

Figure 5: Onboard hardware components.

The CPU is a PC104, a small size, low energy consumption computer widely utilized in embedded and industrial applications. In our case it includes: a MOPS lcd5 board with a Pentium 133 MHz processor, 32 MB RAM, two serial ports, one parallel interface, a 10base-T Ethernet network interface, and keyboard and IDE adapters; an Emerald MM multiserial board with four additional serial ports (yielding a total of six serial ports); a V104 power supply fed by a 12 V battery; and a 42 MB flash drive.

The sensor package consists of the following components:

- GPS with differential correction: in AURORA we utilize a Trimble SveeSix GPS receiver composed by a PC104-compatible board with digital output and an antenna mounted on the top of the airship with a magnetic disk. Another GPS receiver located by the ground station sends correction data to the onboard GPS, yielding an enhanced precision;

- inclinometer and compass: roll, pitch, and yaw (heading) measurements are provided by two modules: a Precision Navigation TCM module composed by a bubble sensor and three magnetometers, and a KVH digital gyro inclinometer (DGI) based on the capacitance of plates immerse in a dielectric fluid. The latter also provides measurements of the roll and pitch rates, utilized to compensate the instantaneous acceleration measurements. A filter is utilized to fuse the data from the TCM and digital giro inclinometer, furnishing the roll and pitch values utilized in the control algorithms.

- accelerometers: a Crossbow CXL01M3 accelerometer provides measurements of the accelerations in all three axes. Since it operates in the range -1 g to +1 g, where g = 9.8 m/s^2, it saturates when reading the acceleration in the z axis. To account for this behavior we added an auxiliary

Analog Devices ADXL05JH accelerometer in the z axis to achieve a larger range of ±1.5 g.

- wind sensor: a wind sensor specially built by one of the authors measures the relative airship air speed in all three axes as well as the barometric altitude [3].

Except for the onboard GPS receiver, which connects directly to the PC104 bus, all other sensors are connected to the CPU via serial ports and a 80C517 microcontroller (MCC). The microcontroller also receives PWM signals from a radio control unit (RCU) and generates PWM signals to the actuators. It communicates with the CPU via RS232.

5.2 Software Infrastructure

Operating System

For proper and reliable operation, Project AURORA necessitates a robust, real time, and small (in terms of disk space) operating system. The OS that best attends to these requirements is Linux. Linux is known to be more robust than other OS's available for personal computers, and attends to our real time requirements via the use of real time Linux (RT-Linux). Additionally, because of its open source code philosophy, Linux allows us to strip down the kernel to the absolute minimum necessary.

RT-Linux is a kernel patch that turns Linux into a real time operating system, allowing one to define periodic tasks. RT-Linux tasks are implemented as dynamic modules of the Linux kernel. They communicate to regular Linux processes through first-in-first-out queues known as rt-fifos. The Linux user may access real time tasks through special files in the /dev directory and the usual file input/output functions (open, read, write, etc.).

The rt_com module is used to give the RT-Linux access to the serial ports of the onboard and ground station computers, that are connected to radio modems (described in Section 5.3).

Onboard System

The onboard system is responsible for reading and sending sensor data to the ground station, and for executing automatic control strategies sending commands to the actuators. Both tasks are executed every 100 ms by RT-Linux. The sampling interval was selected based on simulated flights and corroborated in actual experimental flights.

Ground System

The ground station is responsible for sending commands, including mission paths, to the onboard station and for receiving sensor data and displaying them in real time during simulated and actual flights. Additionally, the ground system records all data received from the onboard station for post-flight analysis and visualization.

For pilot training and airship manual control we utilize a conventional remote control unit (RCU), connected to one of the ground computer's serial ports through a microcontroller. An RT-Linux task is responsible for reading the RCU pulses and sending them to the onboard station via the radio-modem.

During simulated or actual flights a Tcl/Tk script reads the inertial sensor and GPS data and displays them in real time (Figure 6). Additionally, we utilize our VRML/Java airship simulator [17], [18] to visualize the airship flying, as well as the sensor data in a standard airplane-type cockpit (Figure 7). These features allow us to follow the progress of the current mission and make decisions such as send in new mission plans to the airship or abort the mission in case of unforeseen events.

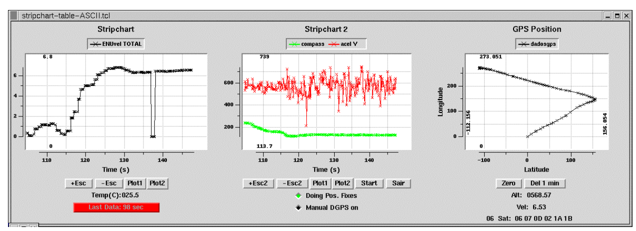

Figure 6: Real time flight data visualization.

Figure 7: Airship visualization during a flight.

All data received by the ground station is recorded for future data analysis and dissemination, and for flight playback. A Tcl/Tk script with the usual VCR buttons (play, stop, ff, rew) allows us to replay the airship flight for mission analysis purposes.

5.3 Communication Infrastructure

The communication system is composed by two radio links. The first one operates in analog mode to transmit video imagery from the airship to the ground station. The second one operates in digital mode to transmit sensor and command data between the ground and onboard stations. The video system is composed by a color CCD camera, a transmission antenna mounted onboard, and a reception flat plate antenna mounted on ground. The data transmission system is composed by a pair of spread spectrum radio modems operating at 115.2 kbps in the 902-928 MHz frequency range. The data can reach up to 30 km with direct line of sight. They are connected to the serial ports of the ground and onboard computers, providing a connection that is undistinguishable from an actual serial cable. An error detection scheme utilizing CRC and packet retransmission ensures data integrity.

6. Experimental Validation

For experimental validation purposes, the PI guidance control method was implemented in C in the onboard computer.

On March 4th, 2000, the airship was flown in a military field in Campinas, Brazil. In this experiment, the airship was manually flown for a few minutes before automatic control was switched on. The mission path was defined as a square with vertices distant 150 m from each other. Wind speed was on the range of 0 to 10 km/h, blowing approximately in the southwest direction.

The airship velocities and heading, necessary in the control algorithm, were obtained respectively from differential GPS data and the compass. Airship path following was controlled automatically by the onboard system, while altitude was controlled manually by the ground pilot.

The look-ahead distance used for the controller design was chosen with a reference speed of 10 m/s and prediction horizon of 2.5 s:

$$\delta_a = \delta + 25\varepsilon \tag{4}$$

The controller's proportional and integral gains were obtained by trial and error.

Figure 8 presents the result obtained. The dotted line represents the airship motion under manual control from take-off until automatic control was switched on. The continuous line represents the airship motion under PI path tracking control. Finally, the dashed line shows the motion of the airship under manual control until landing. In this figure, one can clearly see the adherence to the mission path, as well as an overshoot when the airship turns from southwest to northwest due to the wind disturbance.

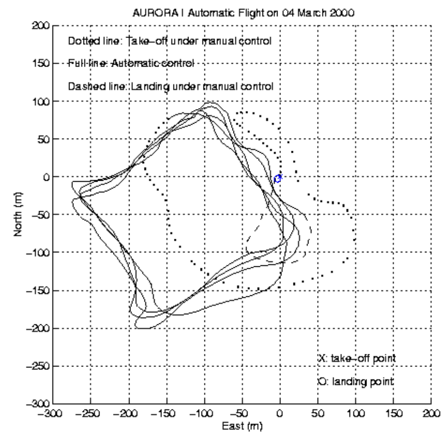

Figure 8: AURORA I under automatic PI control following a set of four points arranged in a square.

Figure 9 presents one of the airship loops around the square, where the dots represent the airship position and the lines represent its heading. Note that the control method composed by the tracking and heading controllers automatically adjusts the airship heading to compensate for wind disturbances; for example, in the lower left part of the square loop, the airship navigates "sideways", while in the upper left it navigates mostly facing towards the trajectory.

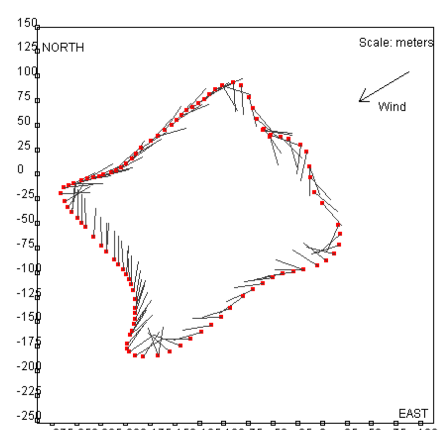

Figure 9: AS800 position and heading along a loop.

7. Conclusions

This paper presents the first results obtained for the trajectory path following problem of AURORA I airship, where the objective is to make the vehicle follow a set of pre-defined points. A path tracking error generation methodology combining distance and angular errors regarding a given reference trajectory is presented, and a control strategy for the guidance problem is proposed. Experimental flight results validate the control methods presented. The authors are currently working on the implementation, in the onboard computer, of automatic control of the airship altitude and attitude, integrating the inertial sensor package available.

8. Acknowledgments

This work is partially funded by the Brazilian funding agencies FAPESP, under grant nos. 97/13384-7, 98/13563-1, 99/04631-6, 99/04645-7, and 00/01000-4, and CNPq under grant no. 68.0140/98-0.

9. References

[1] Aguilar, L. "Variable structure Control for path following of mobile robots by linearizing techniques." *IFAC Int.Conf. IAV'98*, Madrid, 1998.

[2] Azinheira, J.R., de Paiva, E.C., Ramos, J.J.G., Bueno, S.S. "Mission Path Following for an Autonomous Unmanned Airship." *IEEE Intl. Conf. on Robotics and Automation*, San Francisco, USA, April 2000.

[3] Azinheira, J. R.; Oliveira, H. V.; Rocha, B. F. "Aerodynamic measurement system for Project AURORA airship: Calibration Report and User Manual." Mechanical Engineering Institute, Superior Technical Institute, Lisbon, Portugal, 1999. (In Portuguese.)

[4] Botto, A., J.R. Azinheira and J. Sá da Costa. "A comparison between robust and predictive autonomous guidance controllers." *Eighteenth IASTED Int. Conf.*, Innsbruck, Austria, 1999.

[5] de Paiva, E.C., Bueno, S.S., Gomes, S.B.V., Ramos, J. J. G., Bergerman, M. "A Control System Development Environment for AURORA's Semi-Autonomous Robotic Airship." *IEEE Int. Conference on Robotics and Automation*, Detroit, MI, USA, 1999.

[6] de Paiva, E.C., Bueno, S.S., Bergerman, M. "A Robust Pitch Attitude Controller for Aurora's Semi-Autonomous Robotic Airship." *13th AIAA Lighter-Than-Air Systems Tech. Conf.*, Norfolk, VA, 1999.

[7] Elfes, A., Bueno, S.S., Bergerman, M., Ramos, J.J.G. "A semi-autonomous robotic airship for environmental monitoring missions." *IEEE Intl. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 3449-3455.

[8] Gomes, S.B.V., Ramos, J. J. G. "Airship Dynamic Modeling for Autonomous Operation." *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.

[9] Gomes, S.B.V. "An Investigation of the Flight Dynamics of Airships with Application to the YEZ-2A", *Ph.D. Thesis*, College of Aeronautics, Cranfield University, 1990.

[10] Lourtie,P., J.Azinheira, J.P. Rente and P. Felício. "ARMOR Project - Autonomous Flight Capability." *AGARD FVP95 Specialist Meeting, Design and Operation of Unmanned Air Vehicles*, Turkey, 1995.

[11] Mowforth, E. *An introduction to the airship*. Airship Association Publication no. 3, The Airship Association Ltd., 1991.

[12] Munson, K. *Jane's unmanned aerial vehicles and targets*. Jane's Information Group Limited, Surrey, 1996.

[13] NASA National Air and Space Administration. *Theseus Project*. http://www.hq.nasa.gov/office/mtpe/Theseus.html, 1997.

[14] Netherclift, O.J. *Airships today and tomorrow*. Airship Association Publication no. 4, The Airship Association Ltd., 1993.

[15] Normey-Rico, J. Gomez Ortega, E.F. Camacho. Smith-Predictor Based Generalized Predictive Controller for Mobile Robot Path-Tracking, *IFAC Int.Conf. IAV'98*, Madrid, 1998.

[16] Ramos, J.J.G. et al. "Project AURORA: a status report." *3rd International Airship Convention and Exhibition*, Friedrichshafen, Germany, July 2000.

[17] Ramos, J.J.G.; Maeta, S.S.; Bergerman, M.; Bueno, S.S.; Bruciapaglia, A.; Mirisola, L.G.B. "Development of a VRML/JAVA unmaned airship simulating environment." *International Conference on Intelligent Robots and Systems*, Kyongju, Korea, Oct. 1999, pp. 1354-1359.

[18] Ramos, J.J.G.; Bueno, S.S.; Maeta, S.M.; Paiva, E.C.; Asanuma, K.; Nascimento, L.G.; Bergerman, M.; Elfes, A.; Beiral, J.A.R. "Project AURORA: autonomous unmanned remote monitoring robotic airship." *2nd International Airship Convention and Exhibition*, Bedford, UK, June 1998, pp. 91-103.