Project AURORA: Infrastructure and Flight Control Experiments for a Robotic Airship

Ely Carneiro de Paiva
Research Center Renato Archer
DRVC/CenPRA
Rod. D. Pedro I, km 143,6
13082-120 Campinas, Brazil

José Raul Azinheira
Instituto Superior Técnico, IDMEC/IST
Av. Rovisco Pais
1049-001 Lisboa, Portugal
e-mail: jraz@dem.ist.utl.pt

Josué G. Ramos, Jr.
Research Center Renato Archer
DRVC/CenPRA
Rod. D. Pedro I, km 143,6
13082-120 Campinas, Brazil

Alexandra Moutinho
Instituto Superior Técnico, IDMEC/IST
Av. Rovisco Pais
1049-001 Lisboa, Portugal

Samuel Siqueira Bueno
Research Center Renato Archer
DRVC/CenPRA
Rod. D. Pedro I, km 143,6
13082-120 Campinas, Brazil
e-mail: samuel.bueno@cenpra.gov.br

Received 5 September 2005; accepted 11 December 2005

Project AURORA aims at the development of unmanned robotic airships capable of autonomous flight over user-defined locations for aerial inspection and environmental monitoring missions. In this article, the authors report a successful control and navigation

Journal of Field Robotics 23(3/4), 201–222 (2006) © 2006 Wiley Periodicals, Inc.
Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.1002/rob.20111

scheme for a robotic airship flight path following. First, the AURORA airship, software environment, onboard system, and ground station infrastructures are described. Then, two main approaches for the automatic control and navigation system of the airship are presented. The first one shows the design of dedicated controllers based on the linearized dynamics of the vehicle. Following this methodology, experimental results for the airship flight path following through a set of predefined points in latitude/longitude, along with automatic altitude control are presented. A second approach considers the design of a single global nonlinear control scheme, covering all of the aerodynamic operational range in a sole formulation. Nonlinear control solutions under investigation for the AURORA airship are briefly described, along with some preliminary simulation results.

1. INTRODUCTION

Beside their use as military surveillance platforms, unmanned aerial vehicles (UAVs) have a wide spectrum of potential civilian applications as observation and data acquisition platforms. They can be used in environmental monitoring applications related to biodiversity, and ecological and climate research and monitoring. Inspection-oriented applications cover different areas, such as mineral and archaeological prospecting, agricultural, and livestock studies, crop field prediction, land use surveys in rural and urban regions, and also inspection of man-made structures, such as pipelines, power transmission lines, dams, and roads. UAV gathered data can also be used in a complementary way concerning information obtained by satellites, balloons, manned aircraft, or on ground.

Most of the applications cited above have profiles that require maneuverable low altitude low speed airborne data gathering platforms. The vehicle should also be able to hover above an observation target, present extended airborne capabilities for long duration studies, take-off and land vertically without the need of runway infrastructures, have a large payload to weight ratio, among other requisites. For this scenario, lighter-than-air (LTA) vehicles are often better suited than balloons, airplanes, and helicopters (Elfes, Bueno, Bergerman, & Ramos, 1998), mainly because: They derive the largest part of their lift from aerostatic, rather than aerodynamic forces; they are safer and, in case of failure, present a graceful degradation; they are intrinsically of higher stability than other platforms.

In this context, Project AURORA—Autonomous Unmanned Remote mOnitoring Robotic Airship—was proposed (Elfes *et al.*, 1998). AURORA focuses on establishing the technologies required for the autonomous operation of unmanned robotic airships in environmental monitoring and aerial inspection missions. This includes sensing and processing infrastructures, control and guidance capabilities, the ability to perform mission, navigation, and sensor deployment planning and execution, failure diagnosis and recovery, and adaptive replanning of mission tasks based on real-time evaluation of sensor information and constraints on the airborne system and its surroundings.

AURORA is conceived as a multiphase project, involving a sequence of prototypes capable of successively higher mission duration and ranges, with increasing levels of autonomy, evolving from mainly teleoperated to substantially autonomous systems. For the first and current phase of the project, AURORA I, a robotic prototype has been built; it is a proof-of-concept system aiming at the development and experimental validation of the underlying technologies, and the realization of low demanding pilot test applications.

Other important research related to outdoor autonomous airships in the world at this moment are the Lotte Project (Wimmer *et al.*, 2002) in Germany, the French projects at LAAS/CNRS (Hygounenc & Souères, 2002; Hygounenc, Jung, Souères, & Lacroix, 2004) and LSC-Université d'Evry (Beji & Abichou, 2005). In the USA, there is a partnership between the projects of STWing-SEAS (2002) of University of Pennsylvania and the EnviroBLIMP at Carnegie Mellon University. Recently, the DIVA Project (DIVA) started in Portugal, sharing a partnership with the AURORA Project.

Aiming the autonomous airship goal, aerial platform positioning and path tracking should be assured by a control and navigation system. Such a system needs to cope with the highly nonlinear, flight-dependent, and underactuated airship dynamics, ranging from the hovering flight (HF) to the aerodynamic (AF) or cruise flight. Hovering flight is defined here as a flight at low airspeed condition, and includes the take-off and landing maneuvers. In addition, the continuous but abrupt transition between the HF and AF dynamics, and the different use of actuators necessary within each region, make it a very difficult issue to be dealt with by the control scheme.

Basically, four main approaches can be considered for the automatic control and navigation system of an airship. The first one relies on the linear control theory to design individual compensators to satisfy closed-loop specifications, based on linearized models of the airship dynamics. One important result of the linearization approach is the separation of two independent (decoupled) motions: The motion in the vertical plane, named longitudinal, and the motion in the horizontal plane, named lateral. Following this approach, Section 4 presents the control solutions for the AURORA airship based on the linearized models for the lateral and longitudinal motions, considering the aerodynamic region (AF) only. Experimental results for the airship flight path following through a set of predefined points in latitude/longitude, along with automatic altitude control are presented.

Also based on a linearized airship model, Wimmer *et al.* (2002) introduced a robust controller design method to compensate for the lack of knowledge about the Lotte airship dynamic behavior and model parameters. The decoupled longitudinal and lateral control systems consist both of an inner H_∞-controller for the dynamics and an outer SISO proportional (P)- or proportional plus integral (PI)-controller for the remaining states. Experimental results are shown therein for the pitch and velocity control. We remark that, as far as the authors are aware, both experimental results (from Lotte and AURORA Projects) on automatic control for outdoor airships are the only ones reported at the literature until this moment. For the lateral control problem, an alternate H_∞ approach for the airship heading control is proposed by Tan and Nagabhushan (1997), and in de Paiva, Carvalho, Ferreira, & Azinheira (2001), a H_2/H_∞ approach for the design of a lateral proportional plus derivative (PD)-PI controller for the AURORA airship is proposed. Other works in the AURORA Project following this methodology of linear-based controllers can be found in literature (de Paiva, Bueno, & Bergerman, 1999; de Paiva, Bueno, Gomes, Ramos, & Bergerman, 1999; Azinheira, de Paiva, Ramos, & Bueno, 2000).

The second main control approach is based on a hierarchical scheme, where a set of linearized controllers is coordinated by a supervisory layer. The nonlinear equations of motion of the air vehicle are linearized about selected operating points over the flight envelope, and an adaptative or a gain scheduled strategy may be used to adjust the controller gains. Khoury and Gillet (1999) follow this linearized hierarchical scheme.

The third main approach consists of searching for a single global control scheme covering all of the aerodynamic range, such that the different flight regions, from HF to AF, are considered inside a sole formulation. For security reasons, simplicity, and flexibility, a global nonlinear control is more interesting than a linearized and hierarchical one. At present, three nonlinear control solutions are under investigation for the AURORA airship, namely dynamic inversion, backstepping, and sliding modes control, and some preliminary results for the design and simulation using the dynamic inversion and backstepping techniques are shown in Section 5. A backstepping technique has been proposed by the LAAS/CNRS autonomous blimp project (Hygounenc & Souères, 2002; Hygounenc *et al.*, 2004). The global control strategy studied is obtained by switching between four subcontrollers, one for each of the flight phases considered. Each controller is however still based on linearized models of the airship, what leads once again to the separate control of the longitudinal and lateral motions. Still considering the nonlinear control techniques, Beji, Abichou, and Bestaoui (2002) introduced a backstepping tracking feedback control for ascent and descent flight maneuvers, where the objective is to stabilize the airship engine around trimmed flight trajectories.

The fourth main approach for the airship control system is the visual servoing control where the airship attitude and position are controlled based on images provided by an onboard vision system. In the AURORA Project, visual servoing formulations is comprised of a hovering solution (Azinheira *et al.*, 2002) and a strategy for line-following tasks, the latter with objects composed by two (Silveira *et al.*, 2002) and three lines (Silveira, Azinheira, Rives, & Bueno, 2003). Other works in the field of visual servoing of small indoor blimps, where the difficulties due to wind influence and effects of the virtual masses are greatly reduced, focus on the aircraft stabilization problem (Zhang and Ostrowski, 1999), and the use of dynamic inversion techniques to control the position and pose of an indoor blimp (Kawai, Kitagawa, Izoe, & Fujita, 2003).

After this introductory section, the remaining parts of this paper are organized as follows. Section 2 describes the experimental platform and the hardware and software infrastructures developed for the AURORA robotic airship. Section 3 presents the airship dynamic model and the control system strategies investigated. Section 4 presents the control solutions developed based on the airship linearized dynamics, along with the experimental results obtained for trajectory tracking and altitude control. In Section 5, the nonlinear-based control strategies under development are presented along with some preliminary results on simulation for the dynamic inversion and the backstepping techniques. Finally, Section 6 stresses the concluding remarks.

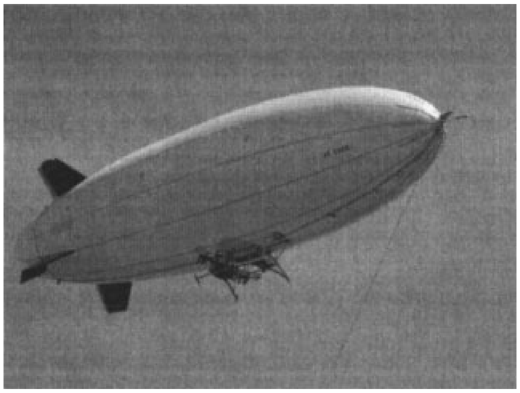


Figure 1. AURORA I LTA robotic prototype.

2. AURORA EXPERIMENTAL PLATFORM AND OVERALL INFRASTRUCTURE

In this section, we briefly describe the AURORA airship platform. Then, we present the software environment conceived to support the autonomous robotic airship from development to operation, as well as the hardware and software infrastructures of the onboard system, the ground station and the communication system between them. Additional information can be found in the following references: (Ramos *et al.*, 1999a; 1999b; Ramos, 2002; Bueno *et al.*, 2002).

2.1. Robotic Airship

For this first phase of the Project AURORA, the LTA robotic prototype has been built as an evolution of the Airspeed Airships' AS800. It is a nonrigid airship with 10.5 m in length, 3.0 m in diameter, and 34 m^3 of volume, whose payload capacity is approximately 10 kg and maximum speed is around 50 km/h (Figure 1). As main control actuators, the airship has: (i) Four deflection surfaces at the tail; (ii) the thrust provided by two propellers driven by two-stroke engines, and; (iii) the vectoring of this propulsion group. Additional actuators, which are not presently used, are the differential action of the two propellers and a stern horizontal electric thruster, useful in hover operation.

2.2. AURORA Software Infrastructure

Robotic UAVs require an overall software environment with custom tools to support their development, operation, and postflight stages, considering the specific needs of the different actors involved: Developers, operational staff, and client for specific applications (Ramos *et al.*, 1999b). The development stage is characterized by the design of all autonomous control strategies (in a large sense), their validation in simulation and further implementation in the real vehicle. The operation stage is associated with the autonomous execution of a specified mission, involving the interaction between the operator and the vehicle during flight, including tools for mission planning, execution and monitoring, as well as a human-machine interface. Finally, in the postflight stage, flight data should be scrutinized in a systematic way, allowing for the analysis of the aircraft flight, the mission execution, and the application data acquired as well. During either the operation or the postflight stage, access to specific gathered information and its dissemination are also important issues.

To support the AURORA's development, operation, and postflight stages, an integrated software environment (Figure 2) with custom tools was created using the Internet as its base, where TCP/IP protocol allows for easy software integration. Nevertheless, during the operation stage at the flight site, we remark that a specific communication system, with appropriated bandwidth and reliability, is used.

The set of tools available, corresponding to those commonly used by the different actors, are gathered in Table I and briefly detailed in the sequel. They are instantiated over the onboard, ground station (explained later) and other computers integrated to the software environment.

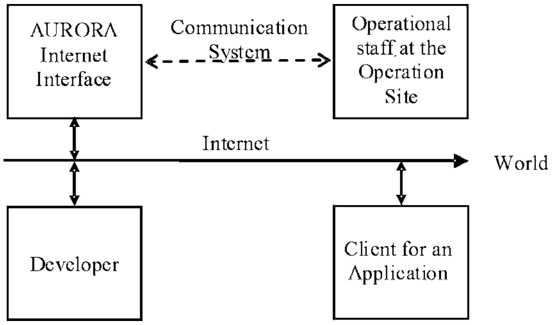

AURORA Internet Interface
Communication System
Operational staff at the Operation Site
Internet
World
Developer
Client for an Application

Figure 2. AURORA software environment architecture and the different actors involved.

2.2.1. Flight Data Storage, Visualization, and Play Back Tool

AURORA's environment provides a tool that allows the user to select the flight data storage file that will be used and the activation of a set of viewers (see next section). During the postflight stage, this tool also allows the user to browse through the flight data files for play back in different speeds.

2.2.2. Viewers: Sensory and Flight Data, Airship, and Surrounding World 3D Representation

Plotting utilities (Figure 3) are used for showing relevant information about the vehicle state. It includes a longitude versus latitude window depicting the trajectory over a real map of the region and other user selected variables plotted against time.

An avionics panel, in a pilot intuitive view (left part of Figure 4), presents the airship states as aircraft instruments such as turning bank, artificial horizon, heading, and so on.

With a three-dimensional (3D) Virtual Viewer, synthesized images of the airship and the surrounding region it operates are provided (one example among several representations is shown on the right part of Figure 4), giving an animation of the vehicle's position and attitude against geographic data. This tool is useful: (i) During the development stage of control and guidance strategies; (ii) to perform a previous mission study, by virtually navigating the airship over the region of interest; (iii) in some cases of flight images lacking during mission execution; or (iv) for postflight animation in playback studies stage.

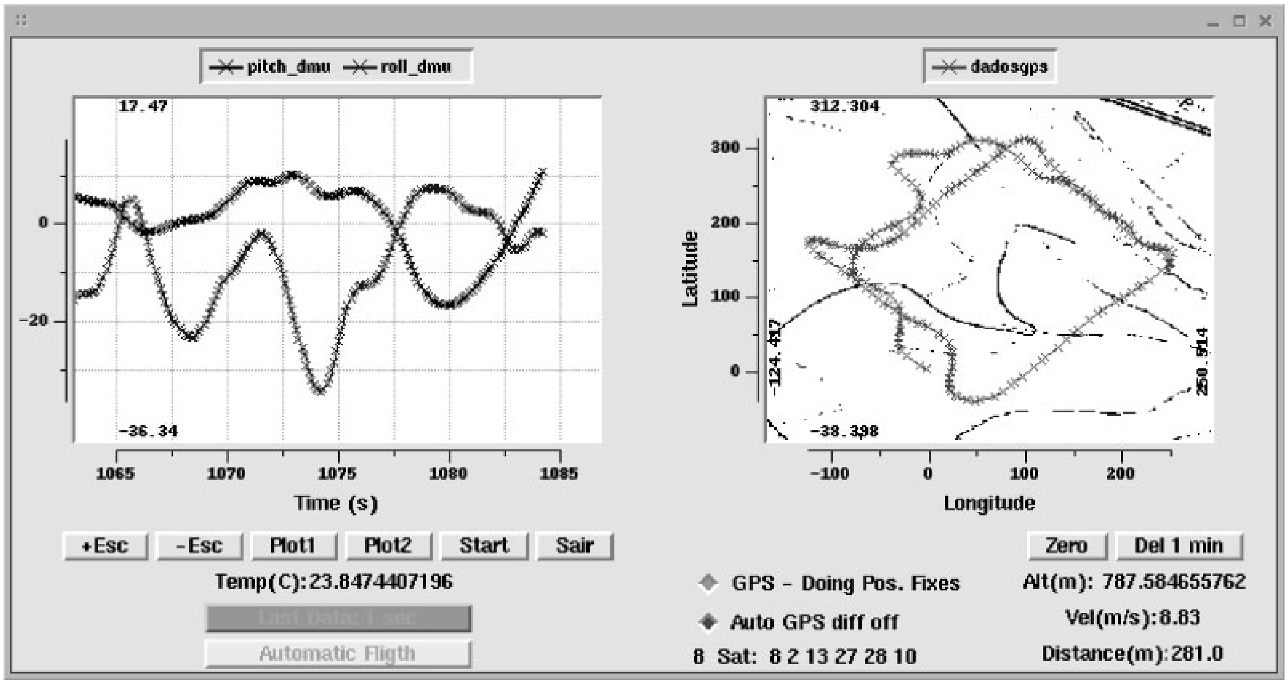
2.2.3. Mission Programming and Mission Control Interface, and Setup of Flight Control System

On AURORA, the mission programming and mission control tool allows the user to: (i) Define the mission over a real map of the flight region; (ii) upload this mission to the aircraft; and (iii) follow its evolution with the possibility of interfering on its execution. The interface of this tool (left part of Figure 5) is composed by a set of mission-related command buttons and, most important, one graphics panel showing the map of the flight region, enabling the user to employ the mouse to easily define, erase, or edit mission way points.

Another component of this interface is a status panel (right part of Figure 5) showing relevant information about the aircraft status, such as the vehicle speed, the distance from the base, the status of the

Table I. AURORA stages and software tools according to their significances, ranging from extremely important (√√√) to useless (blank).

	Stage		
Software tool	Development	Operation	Postflight
Flight data storage and visualization	√√√	√√√	√√√
Flight and data play black	√√√		√√√
Airship and surrounding world 3D virtual viewer	√√√	√√	√√√
Sensory and flight data visualization	√√√	√√√	√√√
Mission programming and mission control interface	√√√	√√√	√√√
On flight setup of control system parameters and mode	√√√	√√	
Airship simulator	√√√	√√	√
CAD for control and navigation strategies	√√√	√	√√√



Temp(C):23.8474407196

GPS - Doing Pos. Fixes

Auto GPS diff off

8 Sat: 8 2 13 27 28 10

Alt(m): 787.584655762

Vel(m/s):8.83

Distance(m):281.0

Figure 3. Plotting utilities showing selected variables and longitude × latitude plot.

global positioning system (GPS) receiver, the status of the flight controller (automatic or manual mode), time elapsed since take off, the engines speed, and so on.

Additionally, the setup of the flight control system part of this interface provides a set of windows for "on flight" setup of control and navigation system tuning parameters and operation mode transition—choosing between manual and automatic, the control algorithms to be activated, etc.

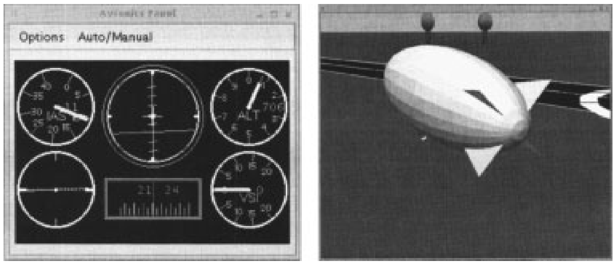


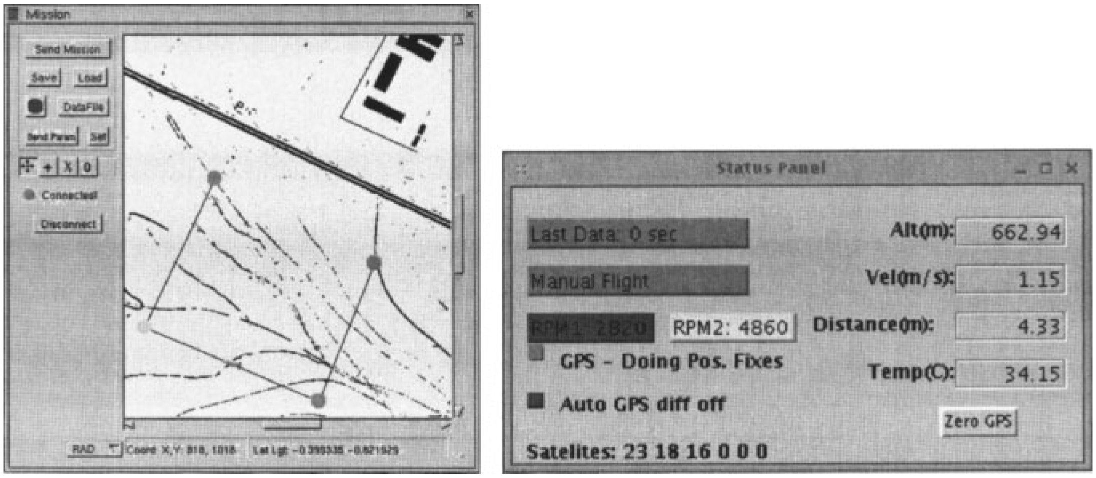
Figure 4. Screen shots for the different viewer tools: Avionics panel (left), virtual viewer (right).

2.2.4. Airship Simulator and CAD for Control and Navigation Strategies

Simulation resources are necessary for: (i) Computer aided design (CAD)-based development and evaluation of control, navigation, and intelligent autonomous strategies; (ii) testing their implementation on the onboard system software prior to real use; and (iii) pilot training. For these purposes, and based on the airship dynamic model detailed next in Section 3, two simulator versions were created: A first one running in MATLAB (de Paiva, Bueno, Gomez *et al.*, 1999), and another one in Java/VRML (Ramos *et al.*, 1999b), which is depicted in Figure 6. They make use of some of the viewers previously described.

2.3. Onboard System and Ground Station

As stressed before, the tools previously described are instantiated over the onboard system and ground station, whose hardware, software, and communication systems are presented in the following.



Last Data: 0 sec

Manual Flight

RPM1: 2820 RPM2: 4860

GPS - Doing Pos. Fixes

Auto GPS diff off

Satelites: 23 18 16 0 0 0

Alt(m): 662.94

Vel(m/s): 1.15

Distance(m): 4.33

Temp(C): 34.15

Figure 5. Mission programming tool (left) and aircraft status panel (right).

2.3.1. Onboard System

The onboard system includes a central processing unit (CPU), sensors, and actuators. It is responsible for sensor data acquisition, control and navigation calculations, and actuator command. By activating algorithms and intelligent strategies for autonomous operation, it assures the execution of mission flight profiles uploaded from the ground and, most important, it takes into account all the features required for a safe flight.

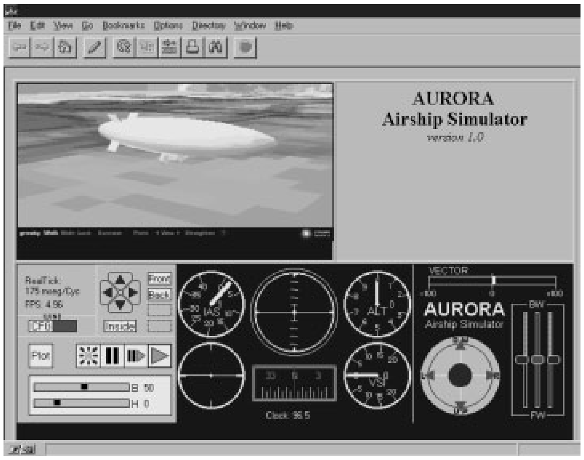


Figure 6. Screen shot of the airship simulator running in Java/VRML.

The onboard operating system (OS) is based on Linux, due to its robustness and open source code philosophy. A real time linux (RT-Linux) distribution, a kernel patch that turns Linux into a real time OS, was striped down to the absolute minimum necessary, resulting in a small embedded solution that fits on a flash disk.

The onboard CPU is a personal computer (PC)/104 standard, with serial, parallel, and Ethernet interfaces, and flash disk. Sensors are connected to the PC/104 via serial ports or a CAN bus. All of the airship actuators are connected to the PC/104 through a microcontroller, which also assures the transition between manual control mode (actuation signals coming from the pilot at ground, through a standard radio control unit) and automatic control mode (signals computed onboard).

A sensor package is comprised of those used for control and navigation, as well as vehicle state and diagnosis sensors (including control surface and vectoring position sensors, engine temperature, and fuel, and battery level).

Main control and navigation sensors currently used on the airship are the following:

- GPS with differential correction: We utilize a Trimble Navigation GPS receiver composed of a PC104-compatible board. Another GPS receiver, located at the ground station, sends correction data to the onboard GPS.

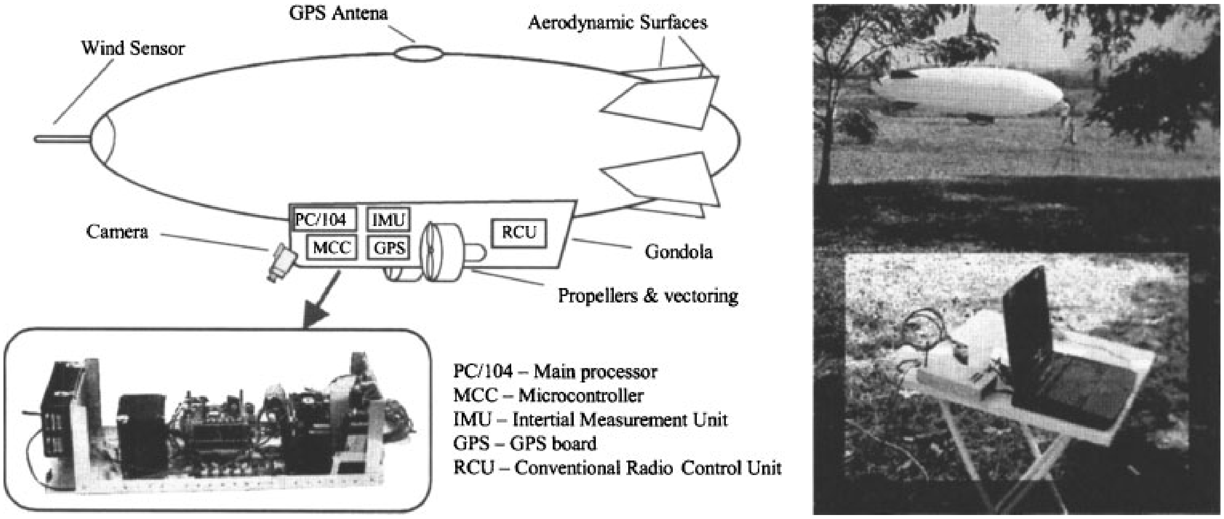


Figure 7. Airship onboard components (left) and the ground station (right).

- Inertial Measurement Unit: We use a Crossbow Dynamic Measurement Unit which provides the roll, pitch, and yaw (heading) attitude, the angular rates and body axes linear acceleration, serving as an inclinometer and compass as well.
- Wind sensor: We use the air data measurement unity, a wind sensor built by the IDMEC/IST in Portugal. It measures the relative airship air speed in all three axes, the aerodynamic incidence angles, as well as the barometric altitude.
- Vision System: A Sony digital camera mounted on the airship's gondola, with a firewire IEEE 1394 interface for Linux, provides aerial images to segmentation algorithms that runs on the PC/104, as well as for the operator on the ground.

The left part in Figure 7 presents a pictorial representation of the airship with its actuators and onboard system.

The communication between the onboard system and the ground station is assured by two links. The first one is comprised of a pair of spread spectrum FreeWave radio modems and transmits command data from ground to the airship and a telemetry package from onboard sensors and actuator commands on the opposite direction. The second link transmits video imagery from the airship to the ground, using one of two alternatives: A short-range wireless Ethernet solution or a long-range analogical link.

2.3.2. Ground Station

The ground station provides the interface between the airship and the "on site operator." It consists of a portable computer, shown on the right part in Figure 7, and a distributed processing infrastructure supported by a Linux OS, where critical time tasks are executed under RT-Linux. The human machine interface and functional facilities at the ground station are build from the tools previously described in Section 2.2.

3. AIRSHIP DYNAMICS AND CONTROL SYSTEM STRATEGIES

3.1. Airship Nonlinear Dynamics

As stated before, two airship simulators were built aiming the development of the control and navigation strategies. The simulators rely on a six degrees of freedom physical model of the airship, including the fully nonlinear flight dynamics of the vehicle. We

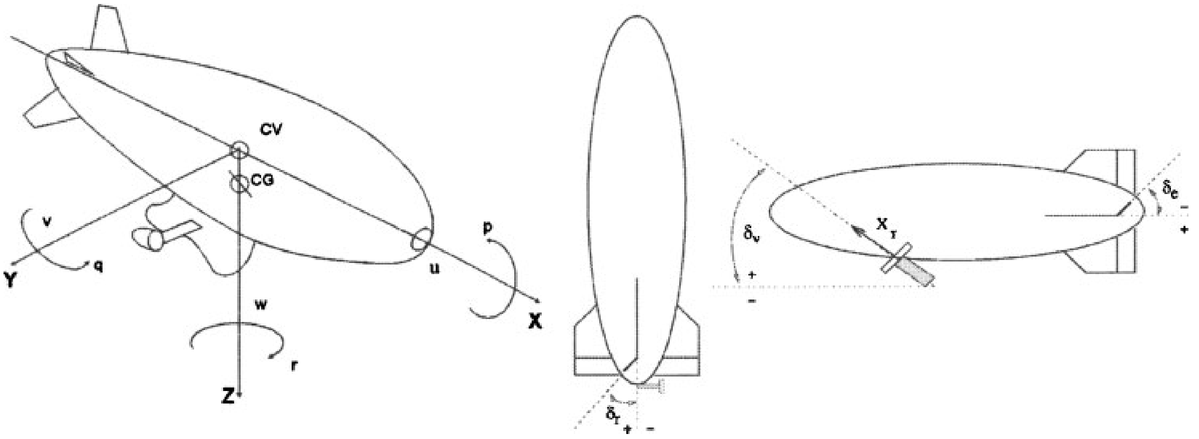


Figure 8. Airship reference frame (left) and airship main actuators (right).

briefly review this model here, while a more detailed presentation can be found in Gomes & Ramos (1998).

In order to develop an accurate mathematical model of the airship flight dynamics, the following aspects were taken into account: (i) The model considers the airship virtual masses and inertias due to the large volume of air displaced by the airship; (ii) the airship motion is referenced to a system of orthogonal axes fixed to the vehicle, whose origin is the center of volume, assumed to coincide with the gross Center of Buoyancy (see Figure 8); and (iii) the airship is assumed to be a rigid body, so that aeroelastic effects are ignored.

As seen above, the dynamic model is defined in the airship frame. The orientation of this body-fixed frame (X,Y,Z) with respect to an Earth-fixed frame (X_E,Y_E,Z_E) is obtained through the Euler angles (ϕ,θ,ψ), corresponding to the roll, pitch, and yaw angles, respectively. The airship linear and angular velocities are given by (u,v,w) and (p,q,r). The angular velocity components (p,q,r) may also be considered as approximations of the roll, pitch, and yaw rates, respectively.

Taking into account the above assumptions, the airship dynamics may be expressed as

$$\mathbf{M}\dot{\mathbf{V}} = \mathbf{F_d}(\mathbf{V}) + \mathbf{F_a}(\mathbf{V}) + \mathbf{F_p} + \mathbf{F_g}, \qquad (1)$$

where $\mathbf{M}$ is the 6×6 mass matrix that includes both the actual inertia of the airship and the virtual inertia elements associated with the dynamics of buoyant vehicles; $\mathbf{V}=[u,v,w,p,q,r]^T$ is the 6×1 vector of airship velocities; $\mathbf{F_d}$ is the 6×1 dynamics vector containing the Coriolis and centrifugal force terms, and also the wind-induced forces (Azinheira, de Paiva, & Bueno, 2001a); $\mathbf{F_a}$ is the 6×1 vector of aerodynamic forces and moments; $\mathbf{F_p}$ is the 6×1 vector of propulsion forces and moments; and $\mathbf{F_g}$ is the 6×1 gravity forces and moments, which are a function of the difference between the weight and buoyancy forces.

The aerodynamic model developed is based on the seminal work presented in Gomes (1990) and takes advantage of information from a wind tunnel database built to model the Westinghouse YEZ-2A airship. The aerodynamic coefficients available in the database are a function of the aerodynamic incidence angles and the deflections of the tail surfaces. The adaptation to the AS800 platform was possible due to the similar length/diameter ratio of both airships. However, the aerodynamic incidence angles considered in this original database were limited to 30°. In order to widen the coverage of the modeling to the whole range of airspeeds and incidence angles of a usual flight, including hover operation, an extended model was developed using a curve fitting and extrapolation procedure as shown in Azinheira, de Paiva, Ramos, Bueno, & Bergerman (2001b). Another improvement was the derivation of a wind-induced force and torque term on the airship dynamics equation (Azinheira *et al.*, 2001a) that was not present on the original model developed by Gomes (1990).

As main control actuators (Figure 8), the AS800 airship has four deflection surfaces, and a pair of propellers driven by two engines. The four deflection surfaces at the tail are arranged in an "×" shape, but they generate the equivalent rudder and elevator commands of the classical "+" tail, with allowable deflections situated in the range from −30° to +30°. The main propellers generate the necessary forces to control the airship motion. Their vectoring (ranging from −30° to +120° up) is used for vertical load compensation and to control the longitudinal motion at low airspeed.

3.2. Linearized Dynamics

The linearization of the airship nonlinear dynamic model given in Eq. (1) makes it easy to design dedicated linear-based controllers. The linearization of the dynamic equations is performed for trimmed conditions around equilibrium, which is commonly a horizontal straight flight, without wind incidence.

In such conditions, the equations are written for a perturbation vector $\mathbf{x}$ of the states around the equilibrium value, perturbed by the control vector $\mathbf{u}$ and the disturbance $\mathbf{w}$, resulting in the dynamic equation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w}. \tag{2}$$

The stochastic disturbance input ($\mathbf{w}$) corresponds mainly to the atmospheric turbulence, generated in the simulation by a Dryden model with three white noise inputs (McLean, 1990).

One important result of the linearization approach is the separation of two independent (decoupled) airship motions: The motion in the vertical plane, named longitudinal, and the motion in the horizontal plane, named lateral. The assumption of decoupled movement is a usual practice in aerial vehicle control, yielding a simplified model with quite acceptable performance. The linearized models, i.e., the dynamic matrix $\mathbf{A}$ and the input matrices $\mathbf{B}$ and $\mathbf{E}$, depend on the trim point chosen for the linearization, and in particular of the airspeed V_{to} and altitude h_0 chosen.

3.2.1. Lateral Dynamics

The dynamics of the airship in the horizontal plane (lateral dynamics) is approximated by the fourth-order linear state space system:

$$\dot{\mathbf{x}}_{\mathbf{h}} = \mathbf{A}_{\mathbf{h}}\mathbf{x}_{\mathbf{h}} + \mathbf{B}_{\mathbf{h}}\mathbf{u}_{\mathbf{h}} + \mathbf{E}_{\mathbf{h}}\mathbf{w}_{\mathbf{h}}, \tag{3}$$

where the state $\mathbf{x}_{\mathbf{h}}=[v,p,r,\phi]^T$ includes the lateral velocity component (v), roll rate (p), yaw rate (r), and roll angle (ϕ). The control input ($\mathbf{u}_{\mathbf{h}}$) is composed of the rudder surface deflection, the aileron surface deflection, and the differential thrust of the main engines (δ_r, δ_a, T_D).

The lateral control task is to coordinate the turning maneuvers, mainly through the rudder surface deflection, making the airship follow a given flight path reference in the horizontal plane. The airship lateral control is relatively easy to implement using simple control algorithms, and successful experimental results were already obtained as shown in the sequel.

3.2.2. Longitudinal Dynamics

For the longitudinal case, in the vertical plane, the state vector considered to evaluate the dynamic characteristics is $\mathbf{x}_{\mathbf{v}}=[u,w,q,\theta,h]^T$ corresponding to the longitudinal velocity (u), the vertical velocity (w), the pitch rate (q), the pitch angle (θ), and the altitude (h). The control vector is given by $\mathbf{u}_{\mathbf{v}} =[\delta_e, X_T, \delta_v]^T$ corresponding to elevator deflection, thrust force, and vectoring angle. The result is the following fifth-order linearized dynamics:

$$\dot{\mathbf{x}}_{\mathbf{v}} = \mathbf{A}_{\mathbf{v}}\mathbf{x}_{\mathbf{v}} + \mathbf{B}_{\mathbf{v}}\mathbf{u}_{\mathbf{v}} + \mathbf{E}_{\mathbf{v}}\mathbf{w}_{\mathbf{v}}. \tag{4}$$

The longitudinal control task is to execute the takeoff, cruise, hover, and landing maneuvers using the elevator deflection and the engines propulsion/vectoring.

3.3. Control System Strategies

The airship dynamics is highly nonlinear flight-dependent, and underactuated, with very different behavior as the airspeed varies from the hovering, or low speed flight (HF) to the cruise or aerodynamic (AF). At low airspeeds, the airship behaves like a balloon, with two badly damped oscillation modes (pitch and roll pendulum modes), which are greatly damped as the airspeed increases. In addition, the abrupt and continuous transition between the HF and AF in the dynamics implies a different use of actuators for each situation. For AF, the most important actuators are the propellers thrust and the aerodynamic elevator/rudder control surfaces, whereas for HF, the effective actuators are the propellers total thrust and vectoring (differential propulsion and a stern thrust may also be used when available).

Moreover, in the low airspeed region, the system may be considered underactuated as:

- The tail surfaces have reduced authority, leaving the airship to be controlled by the force inputs only;
- the main propellers provide four coupled force components (longitudinal and vertical forces, pitching, and rolling torques) with only three inputs (total thrust, vectoring angle, and differential thrust); and
- the stern thruster only provides a yawing torque and its lateral force is negligible, so that no actuator is really available to oppose the aerodynamic side forces.

It is important to note that at low airspeeds, the extra weighting mass of the airship needs to be compensated for with the vertical forces coming from the propellers vectoring, whereas in the aerodynamic flight the extra forces come from the lift resulting from the airship angle of attack.

As stated previously, four main approaches can be considered for the design of the airship control and navigation system, and the main results presented herein are related to the first approach, corresponding to the design of dedicated linear-based controllers. In the next section, control solutions based on the linearized airship models are presented for the lateral and longitudinal motions, both for the aerodynamic flight (AF) only, and experimental results are shown together.

The authors also started the development of controllers in the third approach using a global nonlinear scheme. In this way, the different aerodynamic operational ranges can be considered in a sole formulation. At present, three nonlinear control solutions are under investigation, namely dynamic inversion, backstepping, and sliding modes control. The preliminary results for the design and simulation using the dynamic inversion and the backstepping techniques are shown in Section 5.

4. LINEAR-BASED CONTROL AND EXPERIMENTAL RESULTS

4.1. Lateral Control Approach

One of the most important navigation problems is the flight path following of the vehicle through a set of predefined points in latitude/longitude, characterizing what is called the lateral guidance. In the following, we present the trajectory path following problem, and describe the details of the path following and the trajectory error generator, introduced in Azinheira *et al.* (2000). The guidance and heading airship control system is also presented in the sequel. The approach was successfully validated in experimental flights (Ramos, 2002; Carvalho *et al.*, 2001; Ramos *et al.*, 2001a, 2001b).

4.1.1. 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 navigation paths are defined as a sequence of straight lines 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 path is to be minimized in all cases. We remark that other possible navigation paths, such as arcs of circle for example, fit in the same methodology.

The longitudinal control task in path tracking, as shown in Section 4.2, is to maintain the velocity and to follow an altitude profile.

4.1.2. Linearization Model for Constant Speeds

The variables used in the path tracking control are depicted in Figure 9 (top), where δ is the distance error to the desired path, ε is the angular error, V_g is the ground speed, and ψ_{traj} is the heading angle of the trajectory with respect to the north-east reference frame.

For a constant speed and a small angular error ε, the path tracking problem may be linearized and results in the following model:

$$\dot{\delta} = V_g \sin(\varepsilon) \approx V_{go}\varepsilon,$$

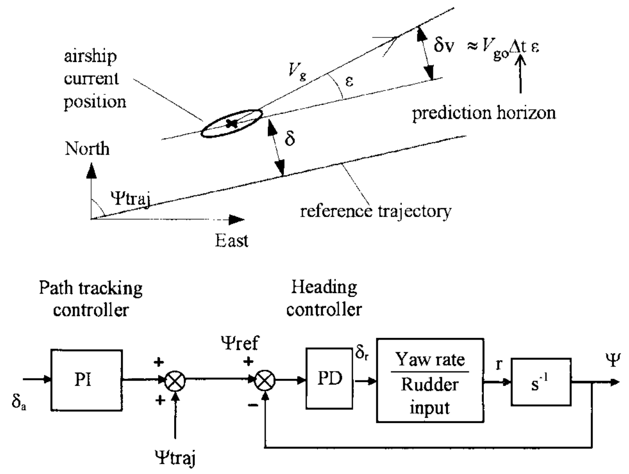


Figure 9. Path tracking signals (top) and guidance controller block diagram (down).

$$\dot{\varepsilon} = r, \tag{5}$$

where V_{go} is the reference ground speed considered for design purposes and r is the yaw rate.

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

$$\delta_a = \delta + V_{go}\Delta t \varepsilon, \tag{6}$$

where Δt is the prediction horizon. This strategy has already been successfully used for the guidance of both unmanned aircraft (Lourtie, Azinheira, Rente, & Felicio, 1995), and ground mobile robots (Botto, Azinheira, & Costa, 1999).

4.1.3. Guidance Control System

The guidance control system is composed of a path tracking controller (outer loop) and a heading controller (inner loop) (de Paiva *et al.*, 2001; Azinheira *et al.*, 2000). A block diagram of the controller is shown in Figure 9 (down), where ψ is the airship heading (yaw) angle, and δ_r is the rudder deflection.

The path-tracking controller is a PI controller whose output, added to the trajectory heading angle ψ_{traj}, yields the reference signal ψ_{ref} for the heading controller. The PI controller input is the look-ahead path tracking error δ_a given in Eq. (6). 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 heading controller is a PD controller, whose objectives are the tracking of a heading reference input, the attenuation of the disturbance (constant wind and turbulence), and the reduction of the amount of rolling oscillation during yaw maneuvering caused by the slightly damped poles of the lateral dynamics.

In these first experimental flights, the PI and PD controller gains were obtained by trial and error, although alternative options were analyzed, through various classical design techniques. A H_2/H_∞ approach for the design of the heading PD gains was by instance presented in de Paiva *et al.* (2001).

4.2. Longitudinal Control Approach

The longitudinal control is responsible for the airship altitude following and speed regulation, and is much more complex and difficult than the lateral control case, because the airship dynamics is nonlinearly dependent on the airspeed, with a complex transition between the HF and AF zones, and using different actuators in the different zones, as stated before.

In a first attempt, we considered a simplified design approach for the airship in the aerodynamic flight only (airspeed above 6 m/s). The engines throttle controls the airship forward speed, and the elevator surface deflection controls the altitude, resulting in a simple PD controller for altitude and a P controller for speed (de Paiva, Bueno, & Bergerman, 1999a; de Paiva, Bueno, Gomez *et al.*, 1999). As the HF zone is not considered, the propulsion vectoring does not appear as a control actuator input in such strategy.

The altitude controller is a state feedback control law with a PD action, where the input is the airship altitude error, and the output is the elevator surface deflection δ_e. The control law for the tracking of an altitude reference $h^* = h^*(t)$ is given by

$$\delta_e = K_{\text{ph}}(h - h^*) + K_{dh}\dot{h} + K_t\theta + K_q q, \tag{7}$$

where θ is the airship pitch angle, q is the pitch rate, and K are the state feedback gains. As the airship altitude rate can be approximated by $\dot{h} = -w + u\theta$, where u and w are the forward and vertical velocities, respectively, and assuming $u \approx u_o = 10$ m/s as a typical velocity for design purposes, then the altitude control law presented before can be rewritten as

$$\delta_e = K_{\text{ph}}(h - h^*) - K_{dh}w + K_\theta \theta + K_q q. \tag{8}$$

Likewise, the speed controller may be easily set up as a single P controller, where the input is the airship velocity error and the output corresponds to the engines throttle. However, in the first experimental flights presented in the sequel, the airship speed was controlled manually by a human operator.

4.3. Experimental Results

4.3.1. Results on Lateral Control

For experimental flight purposes, the guidance/heading PID control method, presented in Section 2.2 was implemented in C language in the airship onboard computer (Ramos, 2002; Ramos *et al.*, 2001b). The airship velocity components and heading, necessary in the control algorithm, were obtained, respectively, from the differential GPS data (at 1 Hz sampling frequency) and the compass data (at 10 Hz) obtained from the inertial measurement unit. Control calculations and actuator commands were performed at 10 Hz.

On March 4, 2000, the airship flew in Campinas, São Paulo, Brazil. In this first flight test, only the lateral control was tested. The airship path following was controlled automatically by the onboard system, while altitude and speed were controlled manually by the pilot at ground.

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

The look-ahead distance from Eq. (6) used for the controller design was chosen with a reference speed of 10 m/s and time of 2.5 s:

$$\delta_a = \delta + 25\varepsilon. \tag{9}$$

The PI path tracking controller used an antiwind up strategy to avoid saturation of the integral term. The heading controller implemented was a simple PD controller for the rudder deflection.

Figure 10 presents the results obtained. The dotted line represents the airship motion under manual control from takeoff until automatic control was switched on. The continuous line represents the airship motion under the path tracking automatic control. Finally, the dashed line shows the motion of the airship under manual control until landing. 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 disturbing wind.

In order to emphasize the influence of the wind input and demonstrate the control behavior, Figure 10 (right) 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 of 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 aligned with the followed segment.

4.3.2. Results on Longitudinal Control

Figure 11 shows the results of a second experimental flight (Bueno *et al.*, 2002; Ramos, 2002) with simultaneous lateral and altitude control, the last one using control law from Eq. (8). The airship velocities and heading, necessary in the control algorithms, were directly obtained, respectively, from the differential GPS and the compass. The altitude signal was measured using a barometric altimeter.

The mission path was defined as a square with sides of 200 m, and the reference altitude was set to $h^* = 50$ m. During the experiments, the take-off and landing procedures were assured by the pilot, on ground. After takeoff, the pilot brought the airship to a cruise flight state and commuted from manual to automatic flying mode.

The results for one of the flight experiments, under automatic operation are shown in Figure 11; they correspond to two complete turns in the trajectory path. One can clearly see the adherence to the mission path, as well as an overshoot for the turns indicated as "1–2" and "2–3," due to the disturbing winds coming from the tail. The right part of Figure 11 also shows the corresponding yaw angle (ψ), pitch angle (θ), and barometric altitude (h). The tran-

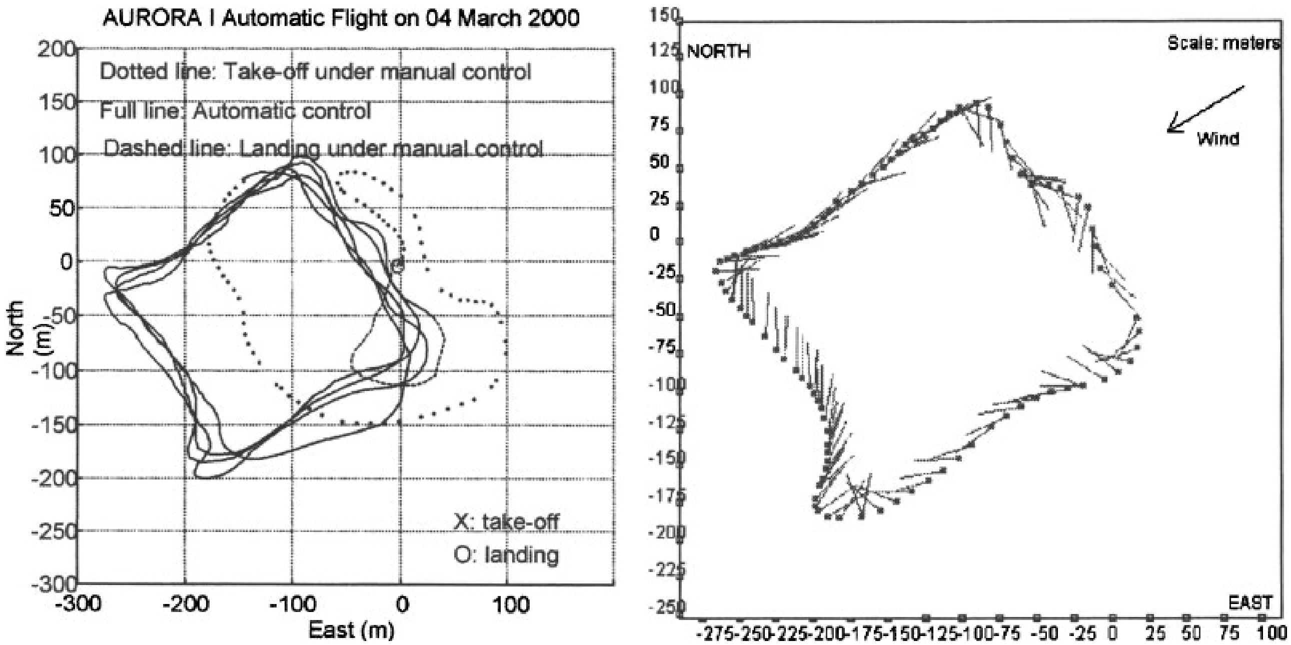


Figure 10. AURORA I under automatic lateral control following a set of four points arranged in a square (left). On the right, airship position and heading along a loop are shown.

sition points can be seen as dotted lines between the numbers marked in the altitude plot. It can be seen that most altitude drops occur during the turning maneuver, due to the longitudinal-lateral coupling that results from saturation in the aerodynamic surface actuators, with the greatest oscillations occurring in transitions 1–2 and 2–3 as explained by the wind disturbance coming from the back. The abrupt changes in the yaw signal are due to the sensor limiting angles, in the range of 0°–360°.

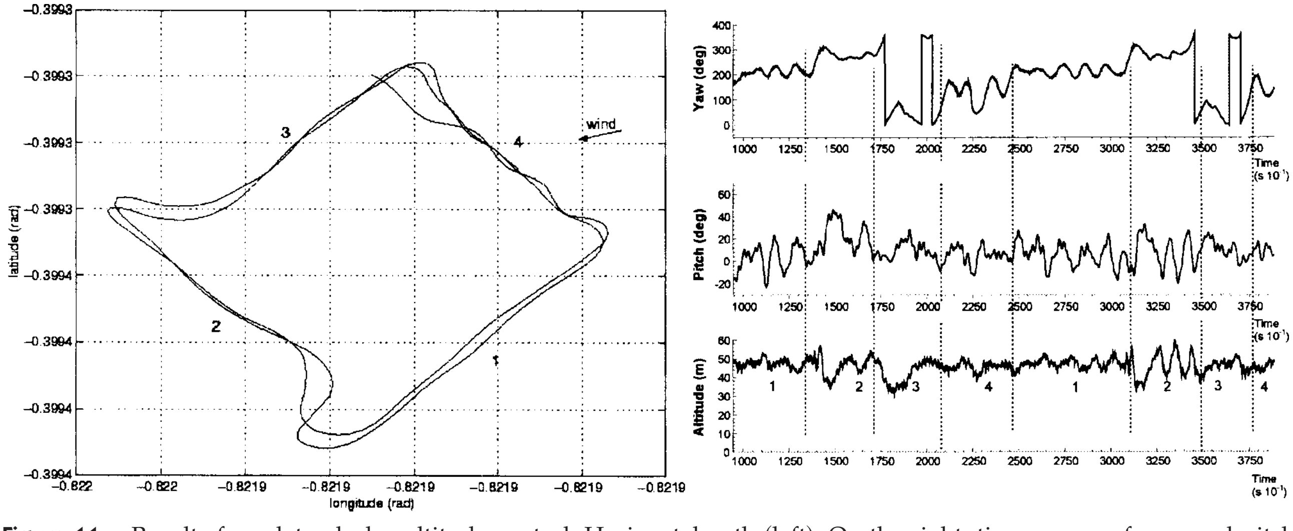


Figure 11. Results from lateral plus altitude control. Horizontal path (left). On the right, time curves of yaw and pitch angles and barometric altitude are shown.

5. NONLINEAR-BASED CONTROL UNDER DEVELOPMENT

As stated previously, a global nonlinear-based controller may introduce a more general approach for the full airship flight envelope, from HF to AF. The main challenge here is the nonlinear and abrupt change behavior of the airship dynamics between these two zones, where a global approach may lead to better performance results. Moreover, as it is very difficult to precisely determine this transition zone, it is interesting that the controller covering the full flight envelope presents strong robustness properties.

With this in mind, three nonlinear control design approaches were recently developed, namely, dynamic inversion, backstepping, and sliding modes control. Preliminary results for the design and simulation using dynamic inversion and backstepping are shown here.

5.1. Dynamic Inversion

Dynamic inversion is a methodology to design closed-loop control laws for nonlinear systems (Isidori, 1989; Slotine & Li, 1991), searching for a global controller from a nonlinear model of the plant. Its application in flight control (Enns, Bugajski, Hendrick, & Stein, 1994) is justified since it can explicitly address the nonlinearities in the aircraft dynamics and provides a control law that is valid over the entire flight envelope.

By using dynamic inversion, the set of existing deficient or undesirable dynamics are cancelled out and replaced by a designer selected set of favored dynamics. This is accomplished by careful algebraic selection of a feedback function, which is why dynamic inversion methodology is also called feedback linearization. In this preliminary investigation of dynamic inversion the airship control problem is still divided into the longitudinal and the lateral motions. Yet, a nondecoupled approach is possible and is the next step in the control system development.

5.1.1. Position Tracking Using Dynamic Inversion

The airship nonlinear system, comprised of dynamic and cinematic equations, suggests a cascaded systems appearance. Considering an affine in control input dynamics and the separation of the state vector $\mathbf{x}$ in its velocity $\mathbf{V}=[u,v,w,p,q,r]^T$ and position $\mathbf{P}=[N,E,D,\phi,\theta,\psi]^T$ components, we have

$$\dot{\mathbf{V}} = \mathbf{f}(\mathbf{V},\mathbf{P}) + \mathbf{G}\mathbf{u},$$

$$\dot{\mathbf{P}} = \mathbf{h}(\mathbf{V},\mathbf{P}), \qquad (10)$$

where $\mathbf{f}$ and $\mathbf{h}$ are nonlinear functions, and $\mathbf{y}$ contains the output variables to be controlled by the input vector $\mathbf{u}$.

Consider again the problem of the path tracking of the airship through a set of predefined points. In this tracking problem, the output variables to be controlled are obviously the position and orientation of the airship. This new formulation of output feedback dynamic inversion, besides representing explicitly the velocity $\mathbf{V}$ and position $\mathbf{P}$, takes the output $\mathbf{y}$ as the position vector. The dynamic system may then be summarized by

$$\dot{\mathbf{V}} = \mathbf{f}(\mathbf{V},\mathbf{P}) + \mathbf{G}\mathbf{u},$$

$$\dot{\mathbf{P}} = \mathbf{h}(\mathbf{V},\mathbf{P}),$$

$$\mathbf{y} = \mathbf{P}. \qquad (11)$$

In order to deduce the input necessary to track a desired output, $\mathbf{y}$ is derived twice, yielding

$$\ddot{\mathbf{y}} = \mathbf{H}_v[\mathbf{f}(\mathbf{V},\mathbf{P}) + \mathbf{G}\mathbf{u}] + \mathbf{H}_p\mathbf{h}(\mathbf{V},\mathbf{P}), \qquad (12)$$

where $\mathbf{H}_v$ and $\mathbf{H}_p$ are, respectively, the partial derivative matrices of $\mathbf{h}(\mathbf{V},\mathbf{P})$ relatively to $\mathbf{V}$ and $\mathbf{P}$.

The inversion of the output dynamic equation thus gives

$$\mathbf{u} = (\mathbf{H}_v\mathbf{G})^{-1}[\ddot{\mathbf{y}} - \mathbf{H}_v\mathbf{f}(\mathbf{V},\mathbf{P}) - \mathbf{H}_p\mathbf{h}(\mathbf{V},\mathbf{P})], \qquad (13)$$

which states the necessary input value $\mathbf{u}$ for a desired output acceleration $\ddot{\mathbf{y}}$, as long as the matrix $\mathbf{H}_v\mathbf{G}$ may be inverted, i.e., as long as $\phi-\theta\neq\pi/2$, a reasonable assumption in the case of the stable airship platform.

When a reference position $\mathbf{P}_r$ is to be tracked ($\mathbf{y}=\mathbf{P}-\mathbf{P}_r$), the desired output dynamics must include the second derivative of the reference:

$$\ddot{\mathbf{y}} = \ddot{\mathbf{P}} - \ddot{\mathbf{P}}_r. \tag{14}$$

In order to define the control law, a desired (or model) output acceleration $\ddot{\mathbf{y}}=\ddot{\mathbf{y}}_m$ must be specified, and it must be in agreement with the position tracking objective. The definition of this model dynamics corresponds to an eigenvalue assignment, which may be arbitrary as long as the real limitations of the system are verified.

As an example, the model may be expressed as a LTI second-order tracking model based on a state feedback:

$$\ddot{\mathbf{y}}_m = \mathbf{L}(\mathbf{V} - \mathbf{V}_r) + \mathbf{M}(\mathbf{P} - \mathbf{P}_r). \tag{15}$$

5.1.2. Lateral Control Case

Let the control of the airship be in the horizontal plane only, assuming that the longitudinal behavior is regulated (by a proportional LQ regulator by instance). Neglect the rolling motion (Khoury & Gillet, 1999), which is a stable pendulum motion of the airship around its longitudinal axis and for which no dedicated actuator is available. Assume also the airship is to track a straight line and the position is given relatively to this line that, for simplicity, will be taken as aligned with the North axis.

The control action may then be obtained by

$$\delta_r = (\mathbf{H}_v\mathbf{G})^{-1}[\mathbf{L}(\mathbf{V} - \mathbf{V}_r) + \mathbf{M}(\mathbf{P} - \mathbf{P}_r) - \mathbf{H}_v\mathbf{f}(\mathbf{V},\mathbf{P}) - \mathbf{H}_p\mathbf{h}(\mathbf{V},\mathbf{P})], \tag{16}$$

where $\mathbf{V}=[v,r]^T$, $\mathbf{P}=[E,\psi]^T$, and the control input δ_r corresponds to the rudder deflection.

Figure 12 presents simulation results of the AURORA airship lateral control using this control law. The NE trajectory, lateral position error δ, and orientation ψ may be seen in this figure. This simulation example considers an aerodynamic flight subjected to three phases:

1. Initial alignment on a straight line segment, with no wind incidence. The airship starts to deviate from the reference trajectory at $(N_i,E_i)=(0,-10)$ m and with an orientation $\psi_i=-10°$.
2. Reference trajectory following. The airship has to track a two-segment trajectory in the shape of a 50° elbow, corresponding to the crossing of a route way-point, again with no wind incidence.
3. Robustness to disturbances. At $t=60$ s, wind starts blowing from the northwest at 3 m/s.

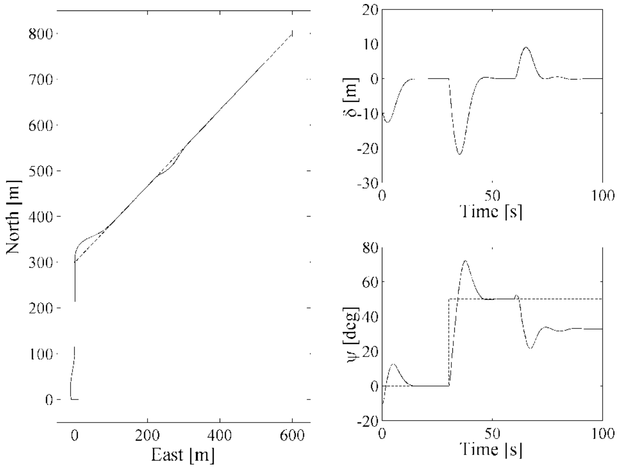


Figure 12. Lateral control: NE trajectory, east error δ, and yaw angle ψ (dashed—reference, solid—output).

The dynamic inversion control law allows good lateral control, with the airship always being able to track the reference path, after the initial deviation and orientation are corrected.

5.1.3. Longitudinal Control Case

Let the control of the airship be in the vertical plane only, assuming that the horizontal behavior is controlled (Moutinho & Azinheira, 2004). Assume further that the airship is supposed to track a straight line and the position is given relatively to this line, that for simplicity will here be taken as aligned with the North axis.

The control action may then be obtained by

$$\begin{bmatrix} \delta_e \\ X_T \\ \delta_v \end{bmatrix} = (\mathbf{H}_v\mathbf{G})^{-1}[\mathbf{L}(\mathbf{V} - \mathbf{V}_r) + \mathbf{M}(\mathbf{P} - \mathbf{P}_r) - \mathbf{H}_v\mathbf{f}(\mathbf{V},\mathbf{P}) - \mathbf{H}_p\mathbf{h}(\mathbf{V},\mathbf{P})], \tag{17}$$

where $\mathbf{V}=[u,w,q]^T$, $\mathbf{P}=[N,D,\theta]^T$, and the control in-

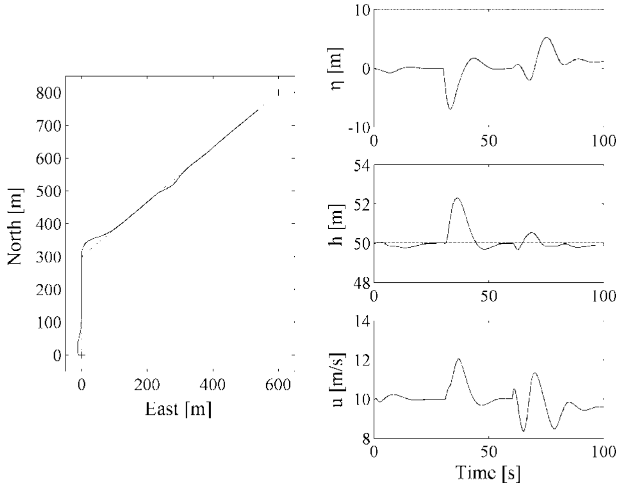


Figure 13. Longitudinal control: *NE* trajectory, north error η, altitude h, and longitudinal ground speed u (dashed—reference, solid—output).

put $\mathbf{U}=[\delta_e, X_T, \delta_v]^T$ corresponds, respectively, to the elevator deflection, the engines mean thrust, and vectoring angle.

The simulation results in Figure 13 show the dynamic inversion longitudinal control of the AURORA airship using this control law. The mission is the same as the one presented for the lateral control, where the longitudinal objective is to keep the groundspeed $u=10$ m/s and the altitude $h=-D=50$ m. Observing the results depicted in Figure 13, we again conclude the dynamic inversion control law leads to a good position tracking.

A fundamental assumption in the dynamic inversion methodology is that the plant dynamics are perfectly modeled and may be cancelled exactly. In practice, this assumption is obviously not realistic, and the robustness of the closed-loop dynamics must be secured, in order to suppress any undesired behavior due to plant uncertainties. For this reason, the stability of the dynamic inversion control system of the AURORA airship was analyzed applying Lyapunov's stability tools by Moutinho and Azinheira (2005). In the same work, the closed-loop robustness to wind and turbulence disturbances, as well as to uncertainties in the model parameters, was verified. After some simulation tests, where either different models were used for the dynamic inversion controller synthesis or wind disturbance was present, good results were still obtained. Moreover, the more significant model parameters were pointed out, for which a more careful identification is necessary, namely the aerodynamic coefficients $C_{M_{\delta_e}}$, C_{Y_β}, C_{d_0}, c_{mq}, and c_{nr}.

A fully coupled version of the dynamic inversion control, without the separation of the longitudinal and lateral motions is currently under development. A more detailed description and results of the application of dynamic inversion to the control of the AURORA airship may be found in Moutinho & Azinheira (2004, 2005).

5.2. Backstepping

Another nonlinear control approach under development for the AURORA robotic airship is the backstepping technique (Khalil, 2002; Krstíc, Kanellakopoulos, & Kokotovic, 1995), which is a Lyapunov-based nonlinear control design approach that presents important robustness properties against unmatched uncertainties, like unmodeled dynamics. By formulating a scalar positive function of the system states and then choosing a control law to make this function decrease, the stability of the resulting nonlinear control system is assured, in the Lyapunov sense.

Backstepping has recently been considered in different flight control applications. Frazzoli, Dahleh, and Feron (2000) and Mahouny, Hamel, and Dzul (1999) have applied it to helicopters control, while Beji and Abichou (2005) and Hygounenc and Souères (2002) have simulated it with airships.

In our case, and in a first attempt, a backstepping nonlinear control law was developed for the hovering flight of the AURORA robotic airship (Azinheira, Moutinho, & de Paiva). The controller is based on a nonlinear dynamic model valid for the hover flight over a ground target. Prior to the application of the backstepping formulation, a new and synthetic modeling of the airship dynamics was introduced, using a modified quaternion formulation of the kinematics equations.

5.2.1. Backstepping Controller Design

The complete dynamic and cinematic model is given by (with the purpose of maintaining the notation used in the original work, the variables are different from the rest of this paper)

$$\dot{\mathbf{x}} = \mathbf{K}\mathbf{x} + \mathbf{M}^{-1}(\mathbf{ESg} + \mathbf{F} + \mathbf{f}),$$

$$\dot{\boldsymbol{\eta}} = \mathbf{DC}\mathbf{x},$$

$$\dot{\mathbf{S}} = -\boldsymbol{\Omega}_3\mathbf{S},$$

$$\dot{\mathbf{D}} = \mathbf{D}\boldsymbol{\Omega}_7, \tag{18}$$

where $\mathbf{x}=[\mathbf{v}^T,\boldsymbol{\omega}^T]^T$ is the state vector formed by the linear and angular velocities vectors, $\boldsymbol{\eta}=[\mathbf{p}^T,\mathbf{q}^T]^T$ is the airship position error described by its Cartesian coordinates in an earth frame and by a quaternion description for its angular attitude, **M** is the symmetric inertia matrix, **S** is the unitary transformation matrix from earth to local frame, **g** is the gravity acceleration vector, **F** represents the aerodynamic forces, $\mathbf{f} = [\mathbf{F}_u^T,\mathbf{T}_u^T]^T$ is the force and torque input vector generated by the airship actuators, $\boldsymbol{\Omega}_3=\boldsymbol{\omega}\times$ and $\mathbf{K}=-\mathbf{M}^{-1}\boldsymbol{\Omega}_6\mathbf{M}$ (please see Azinheira, Moutinho, & de Paiva for the remaining variables description).

The desired stabilization corresponds to a control system objective where both the velocity **x** and the position $\boldsymbol{\eta}$ are regulated to zero.

Consider two intermediate output variables $\mathbf{y}_1$ and $\mathbf{y}_2$:

$$\mathbf{y}_1 = a\boldsymbol{\eta} + b\mathbf{T}\mathbf{x},$$

$$\mathbf{y}_2 = \boldsymbol{\Delta}\mathbf{x}, \tag{19}$$

with two positive scalars a, b and a diagonal positive matrix $\boldsymbol{\Delta}$, and **T** is defined as $\mathbf{T}=\mathbf{DC}$. Let us define the Lyapunov function candidate:

$$W = \frac{1}{2}\mathbf{y}_1^T\mathbf{y}_1 + \frac{1}{2}\mathbf{y}_2^T\mathbf{y}_2. \tag{20}$$

If we choose the control input in Eq. (18) as

$$\mathbf{f} = -\mathbf{M}\mathbf{T}^{+}(\rho^2\boldsymbol{\Lambda}\boldsymbol{\eta} + \boldsymbol{\Lambda}_1^2\mathbf{T}\mathbf{x}) - \mathbf{M}(\mathbf{C}^T\boldsymbol{\Omega}_7\mathbf{C} + \mathbf{K})\mathbf{x} - \mathbf{ESg} - \mathbf{F}, \tag{21}$$

where $\rho^2=a/b$, $\mathbf{T}^{+}$ is the pseudoinverse of **T**, and $\boldsymbol{\Lambda}_1$ is defined as $\boldsymbol{\Lambda}_1^2=\rho^2\mathbf{I}_7+(1+1/b^2)\boldsymbol{\Lambda}$, then it can be shown that the Lyapunov function derivative is negative definite and is given by Azinheira, Moutinho, & de Paiva:

$$\dot{W} = \mathbf{y}_1^T\dot{\mathbf{y}}_1 + \mathbf{y}_2^T\dot{\mathbf{y}}_2 = -\left(a\boldsymbol{\eta} + \left(b+\frac{1}{b}\right)\mathbf{T}\mathbf{x}\right)^T\boldsymbol{\Lambda}\left(a\boldsymbol{\eta} + \left(b+\frac{1}{b}\right)\mathbf{T}\mathbf{x}\right) - \frac{a}{b}(\mathbf{T}\mathbf{x})^T(\mathbf{T}\mathbf{x}), \tag{22}$$

resulting in a closed-loop system with global asymptotic stability. Although the Lyapunov function (20) is not directly a function of the state variables, it is easily verified that through the change of variables (19) the derivative of W only vanishes at the origin for $\mathbf{x}=0$ and $\boldsymbol{\eta}=0$.

5.2.2. Simulation Results

In order to cope with the limitations due to the reduced actuation in hovering flight, saturations are also introduced in the control design, and the global asymptotic stability of the system under saturation is demonstrated. In addition, to deal with the airship lateral underactuation considering the wind disturbances, a reference path from the current state to the desired state (target) is defined to minimize the needs of lateral forces. Extensive simulations were conducted to evaluate the approach, considering a hovering stabilization of the airship under low and high wind conditions, including turbulences (Azinheira, Moutinho, & de Paiva).

We shall describe here some of the simulation results obtained using the fully nonlinear platform developed in the AURORA project (Bueno *et al.*, 2002; Ramos *et al.*, 2001a, 2001b) to simulate the motion of the AURORA airship prototype weighting 33 kg and with a volume of 30 m^3 (Figure 1).

The case described here corresponds to the hovering stabilization of the airship at ($N_r=0$, $E_r=0$) and constant altitude h_r. The airship starts from an initial horizontal error ($N_i=-25$ m, $E_i=5$ m), where N and E stand for the north and east relative position. For illustrative purposes, the attitude of the airship is also set out of equilibrium, with 10° in each one of the Euler angles. We shall examine the airship behavior when in the presence of a constant wind of 3 m/s blowing from North, plus a turbulent gust of

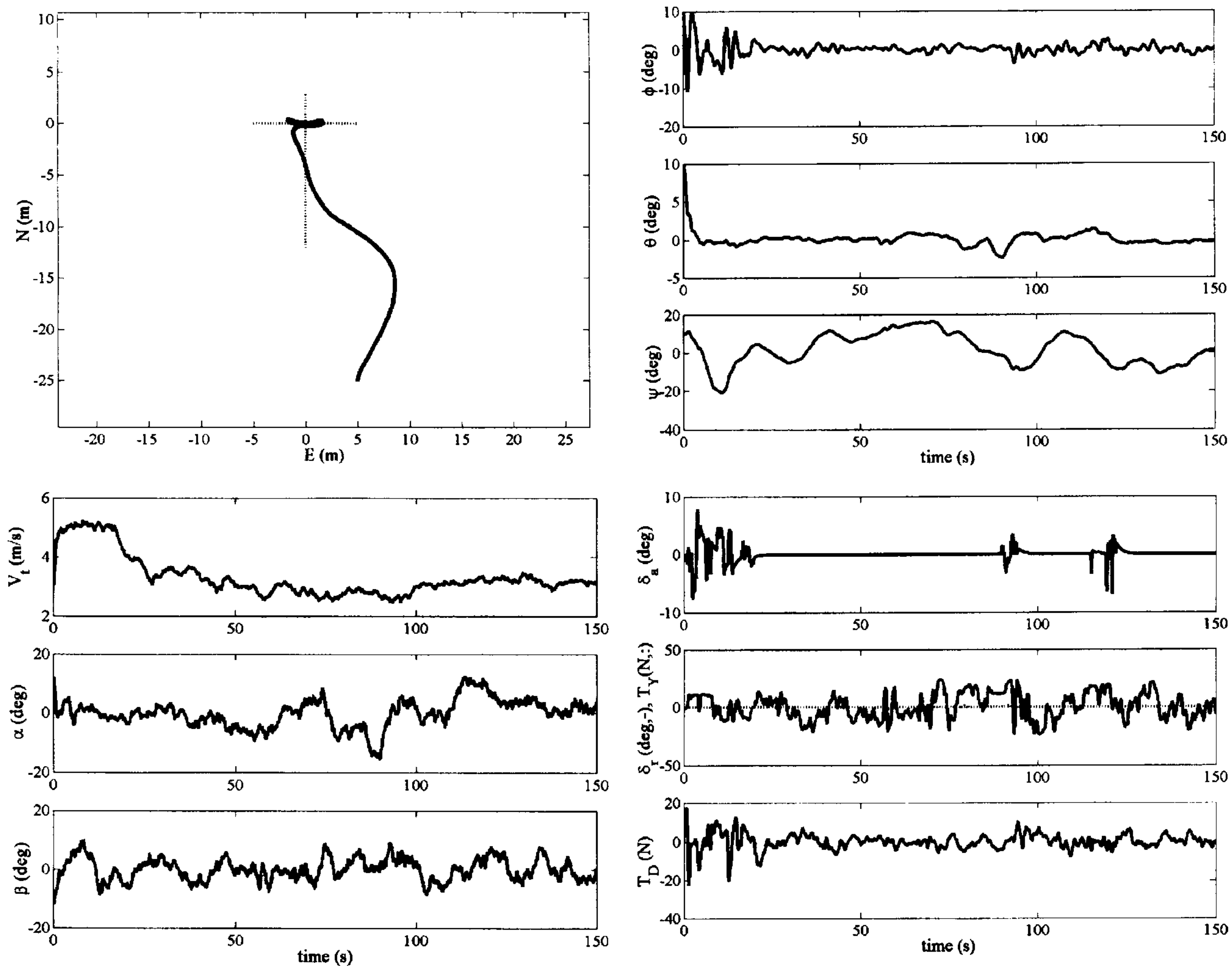


Figure 14. Simulation results of backstepping stabilization of the airship in face of wind disturbances.

3 m/s.[1] This is an intermediate condition with a relatively high turbulence, with wind heading changes reaching 40°, but one may consider that the airship should stabilize a hover in these conditions, for ground or target observation.

The results of this simulation are presented in Figure 14. The horizontal northeast path is represented on the left upper corner. We can observe that the reference shaping leads to a priority in the alignment of the airship against the wind. When the airship reaches the target point, the wind input results in a mostly lateral oscillation around 2 m wide. The influence of the turbulent disturbance is particularly visible for the aerodynamic variables (airspeed V_t, angle of attack α, and sideslip angle β) depicted below the *N-E* path. With airspeed oscillations above 1 m/s, the aerodynamic angles go up to almost 15°. The longitudinal control is still quite good, suffering very little with from the disturbance.

The lateral control is more difficult, which is simply explained by the high aerodynamic lateral forces due to the wind inputs, being the airship clearly underactuated in this axis. The overall stabilization objective is still achieved, and namely the roll ϕ and pitch θ angles are very well regulated as seen on the right of the *N-E* path (the attitude is here

[1] In the AURORA AS800B simulation platform, the continuous atmospheric turbulence is modeled with a Dryden model.

Table II. rms values for hovering in 3 m/s wind and turbulence simulation (last 100 s).

Mean wind speed	N	E	h	ϕ	θ	ψ	V_t
3 m/s	20 cm	114 cm	5.8 cm	0.9°	0.7°	8.5°	0.23 m/s

presented in terms of the Euler angles, which are more meaningful and intuitive than the quaternions).

In terms of control input (see the lower right graphic in Figure 14), the rudder deflection δ_r and the differential thrust T_D are the more active inputs, necessary to oppose the lateral forces and rolling torques.

A quantitative measurement of the control performance may be obtained from the root-mean-square (rms) measurement of the last 100 s by instance (approximately equivalent to the stabilized region), leading to the results in Table II. The results are in perfect agreement with the previous comments, and they confirm that the hovering stabilization is well achieved, despite the high turbulent disturbance. Namely, the roll ϕ and pitch θ angles are quite stable.

Extensions of this backstepping approach for the whole airship flight envelope are currently under development.

6. CONCLUSIONS

In this paper, the authors present the current status of control and navigation strategies currently under development in the AURORA Autonomous Robotic Airship Project, showing experimental results and indicating subsequent evolutions. This work, remarked as one of the first controlled flights of an outdoor airship reported in the literature, forms the basis for the development of autonomous capabilities for a robotic airship.

Initially, the AURORA I airship is presented, with the overall software environment conceived to provide support from development to operation, as well as the hardware and software infrastructures of the onboard system, the ground station and the communication system between them.

Afterward, two main approaches for the design of the airship control and navigation system are described. The first one is based on the airship linearized models, resulting in decoupled lateral and longitudinal controllers for the lateral and longitudinal motions, both for the aerodynamic flight only. Experimental results using this methodology are presented for the case where the airship should follow a trajectory specified as a set of predefined way points in latitude/longitude, as well as an altitude profile.

It is important to remark that for isolated and particular applications, such as HF only, or cruise flight only, the linear-based controllers may give acceptable performance, as shown by the experimental results. However, if a complete mission is to be implemented under automatic control, including take-off, cruise, hovering and landing, then a global nonlinear control approach would help to cope with the highly nonlinear, flight-dependent, and underactuated dynamics of the airship, allowing for a higher level of robustness.

Consequently, the second approach presented here considers the nonlinear control design, where the final objective is to derive a single global nonlinear and robust control scheme for the whole flight envelope. Preliminary results and simulations are shown using dynamic inversion or backstepping. the dynamic inversion case is still based on decoupled motions, although covering the whole aerodynamic flight envelope; the backstepping case considers a coupled formulation, though it is initially only applied to the airship hovering flight. The nonlinear control design is the main subject of our ongoing works and should soon be ready for flight tests.

ACKNOWLEDGMENTS

The present work was sponsored by the Brazilian agencies Fapesp under Grant No. 04/13467-5, and CNPq under Grant No. 150181/2003-5, and 382739/2005-1. The work was also partly developed in the frame of project DIVA-POSI/SRI/45040/2002, approved by the Portuguese Ministry of Science FCT and POSI program, with support from FEDER European Fund. We also would like to thank the

AURORA team, namely Bruno Faria, Douglas Figueiredo, Geraldo Silva, Geraldo Silveira, Hélio Azevedo, and Sidney Cunha.

REFERENCES

Azinheira, J.R., de Paiva, E.C., Ramos, J.J.G., & Bueno, S.S. (2000). Mission path following for an autonomous unmanned airship. International Conference on Robotics and Automation, San Francisco, CA.

Azinheira, J.R., de Paiva, E.C., & Bueno, S.S. (2001a). Influence of wind speed on airship dynamics. Journal of Guidance, Control and Dynamics, 25(6), 1116–1124.

Azinheira, J.R., de Paiva, E.C., Ramos, J.G., Bueno, S.S., & Bergerman, M. (2001b). Extended dynamic model for AURORA robotic airship. 14th AIAA Lighter-Than-Air Systems Technology Conference, Akron, OH.

Azinheira, J.R., Moutinho, A., & de Paiva, E.C. Airship hover stabilization using a backstepping control approach. Journal of Guidance, Control and Dynamics (to appear).

Azinheira, J.R., Rives, P., Carvalho, J.R.H., Silveira, G., de Paiva, E.C., & Bueno, S.S. (2002). Visual servo control for the hovering of an outdoor robotic airship. Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, pp. 2787–2792.

Beji, L., Abichou, A., & Bestaoui, Y. (2002, November). Stabilization of a nonlinear underactuated autonomous airship—A combined averaging and backstepping approach. Proceedings of the 3rd International Workshop on Robot Motion and Control, Bukowy Dworek, Poland.

Beji, L., & Abichou, A. (2005). Tracking control of trim trajectories of a blimp for ascent and descent flight manoeuvres. Int. J. Control, 78(10), 706–719.

Botto, A., Azinheira, J.R., & Costa, J.S. (1999). A comparison between robust and predictive autonomous guidance controllers. Eighteenth IASTED International Conference, Innsbruck, Austria.

Bueno, S.S., Azinheira, J.R., Ramos, J.G., de Paiva, E.C., Rives, P., Elfes, A., Carvalho, J.R.H., & Silveira, G.F. (2002, October). Project AURORA—Toward an autonomous robotic airship. Workshop on Aerial Robotics, 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems—IROS 2002, Lausanne, Switzerland.

Carvalho, J.R.H., Ferreira, P.A.V., de Paiva, E.C., Azinheira, J.R., Ramos, J.G., Bueno, S.S., Maeta, S.M., Mirisola, L.G.B., Faria, B.G., Bergerman, M., & Elfes, A. (2001, July). Application of classical and robust PI control to an unmanned robotic airship. 9th International Symposium on Intelligent Robotic Systems—SIRS'2001, Toulouse, France.

de Paiva, E.C., Bueno, S.S., & Bergerman, M. (1999). A robust pitch attitude controller for Aurora's semi-autonomous robotic airship. 13th AIAA Lighter-Than-Air Systems Technology Conference, Norfolk, VA.

de Paiva, E.C., Bueno, S.S., Gomes, S.B.V., Ramos, J.J.G., & Bergerman, M. (1999). A control system development environment for AURORA's semi-autonomous robotic airship. International Conference on Robotics and Automation, Detroit, MI.

de Paiva, E.C., Carvalho, J.R.H., Ferreira, P.A.V., & Azinheira, J.R. (2001). An H_2/H_∞ PID heading controller for AURORA-I semiautonomous robotic airship. 14th AIAA Lighter-Than-Air Systems Technology Conference, Akron, OH.

DIVA Project—Autonomous airship for aerial surveillance. Retrieved November 25, 2005 from ⟨http://paloma.isr.uc.pt/diva/⟩

Elfes, A., Bueno, S.S., Bergerman, M., & Ramos, J.J.G. (1998). A semiautonomous robotic airship for environmental monitoring missions. International Conference on Robotics and Automation. Leuven, Belgium, pp. 3449–3455.

Enns, W., Bugajski, D., Hendrick, R., & Stein, G. (1994). Dynamic inversion: An evolving methodology for flight control design. Int. J. Control, 59(1), 71–91.

Frazzoli, E., Dahleh, M.A., & Feron, E. (2000, June). Trajectory tracking control design for autonomous helicopters using a backstepping algorithm. Proceedings of the American Control Conference, Chicago, IL.

Gomes, S.B.V. (1990). An investigation of the flight dynamics of airships with application to the YEZ-2A. Unpublished Ph.D. thesis, College of Aeronautics, Cranfield University, Cranfield, England.

Gomes, S.B.V., & Ramos, J.J.G. (1998). Airship dynamic modeling for autonomous operation. International Conference on Robotics and Automation, Leuven, Belgium.

Hygounenc, E., & Souères, P. (2002). Automatic airship control involving backstepping techniques. Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Hammamet, Tunisia.

Hygounenc, E., Jung, I.-K, Souères, P., & Lacroix, S. (2004). The autonomous blimp project of LAAS-CNRS: achievements in flight control and terrain mapping. Int. J. Robotics Research, 23(4–5), 473–511.

Isidori, A. (1989). Nonlinear control systems, 2nd edition. New York: Springer.

Kawai, Y., Kitagawa, S., Izoe, S., & Fujita, M. (2003, August). An unmanned planar blimp on visual feedback control: Experimental results. Proceedings of the 42th SICE Annual Conference, Fukui, Japan, pp. 680–685.

Khalil, H.K. (2002). Nonlinear systems, 3rd edition. Englewood Cliffs, NJ: Prentice–Hall.

Khoury, G.A., & Gillet, J.D. (1999). Airship technology. New York: Cambridge University Press.

Krstíc, M., Kanellakopoulos, I., & Kokotovic, P. (1995). Nonlinear and adaptive control design. New York: Wiley.

Lourtie, P., Azinheira, J.R., Rente, J.P., & Felício, P. (1995). ARMOR Project—Autonomous flight capability. AGARD FVP95 Specialist Meeting on Design and Operation of Unmanned Air Vehicles, Turkey.

Mahouny, R., Hamel, T., & Dzul, A. (1999, December). Hover control via Lyapunov control for an autonomous model helicopter. Proceedings of the 38th Conference on Decision & Control, Phoenix, AZ.

McLean, D. (1990). Automatic flight control systems. Englewood Cliffs, NJ: Prentice–Hall.

Moutinho, A., & Azinheira, J.R. (2004, July). Path control of an autonomous airship using dynamic inversion. Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal.

Moutinho, A., & Azinheira, J.R. (2005, April). Stability and robustness analysis of the AURORA airship control system using dynamic inversion. Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain.

Ramos, J.J.G., Maeta, S.M., Mirisola, L.G.B., Bergerman, M., Bueno, S.S., Sousa, P.G., & Bruciapaglia, A. (1999a). A software environment for an autonomous unmanned airship. Proceedings 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

Ramos, J.J.G., Maeta, S.M., Bergerman, M., Bueno, S.S., Mirisola, L.G.B., & Bruciapaglia, A. (1999b). Development of a VRML/Java unmanned airship simulating environment. Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3.

Ramos, J.G., de Paiva, E.C., Azinheira, J.R., Bueno, S.S., Maeta, S.M., Mirisola, L.G.B., & Bergerman, M. (2001a, May). Autonomous flight experiment with a robotic unmanned airship. IEEE International Conference on Robotics and Automation—ICRA'2001, Seoul, Korea, pp. 4152–4157.

Ramos, J.G., de Paiva, E.C., Carvalho, J.R.H., Ferreira, P.A.V., Azinheira, J.R., Bueno, S.S, Maeta, S.M., Mirisola, L.G.B., Faria, B.G., Bergerman, M., & Elfes, A. (2001b, June). Path tracking flight test of an autonomous unmanned robotic airship. 3rd International Conference on Field and Service Robotics, Otaniemi, Espoo, Finland.

Ramos, J.J.G. (2002). Contribuição ao Desenvolvimento de Veículos Aéreos: Caso de um Dirigível Autônomo Não-Tripulado. Unpublished Ph.D. thesis (in Portuguese), Universidade Federal de Santa Catarina—UFSC, Brazil.

Silveira, G., Carvalho, J.R.H., Rives, P., Azinheira, J.R., Bueno, S.S., & Madrid, M.K. (2002). Optimal visual servoed guidance of outdoor autonomous robotic airships. Proceedings of the American Control Conference, USA, pp. 779–784.

Silveira, G., Azinheira, J.R., Rives, P., & Bueno, S.S. (2003). Line following visual servoing for aerial robots combined with complementary sensors. Proceedings of the IEEE International Conference on Advanced Robots, Coimbra, Portugal, pp. 1160–1165.

Slotine, J.-J.E., & Li, W. (1991). Applied nonlinear control. Englewood Cliffs, NJ: Prentice–Hall.

STWing-SEAS of University of Pennnsylvania in partnership with the EnviroBLIMP Project of CMU (2002). Retrieved November 25, 2005, from ⟨http://www.stwing.org/blimp/⟩ and ⟨http://www.frc.ri.cmu.edu/projects/enviroblimp/⟩

Tan, S.B., and Nagabhushan, B.L. (1997). Robust heading-hold autopilot for an advanced airship. 12th AIAA Lighter-Than-Air Technology Conference, AIAA-97-1488.

Wimmer, D.-A., Bildstein, M., Well, K.H., Schlenker, M., Kingl, P., & Kröplin, B.-H. (2002). Research airship "Lotte:" Development and operation of controllers for autonomous flight phases. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland.

Zhang, H., & Ostrowski, J.P. (1999, May). Visual servoing with dynamics: Control of an unmanned blimp. Proceedings of the IEEE International Conference on Robotics and Automation, MI, pp. 618–623.