

Project AURORA: Towards an Autonomous Robotic Airship

Samuel S. Bueno¹, José R. Azinheira², Josué Jr. G. Ramos¹, Ely C. de Paiva¹,
Patrick Rives³, Alberto Elfes⁴, José R. H. Carvalho¹, Geraldo F. Silveira¹

¹*CenPRA/LRVC, Campinas, Brazil, {First_name.Last_name}@cenpra.gov.br*

²*IST/IDMEC, Lisbon, Portugal, jraz@dem.ist.utl.pt*

³*INRIA Sophia-Antipolis, France, Patrick.Rives@sophia.inria.fr*

⁴*FAW, Ulm, Germany, elfes@faw.uni-ulm.de*

Abstract

Robotic UAVs have an enormous potential as observation and data-gathering platforms for a wide variety of applications. Robotic airships, in particular, are of great interest for environmental monitoring and inspection applications. This paper provides an overview of Project AURORA (Autonomous Unmanned Remote Monitoring Robotic Airship), a research effort that focuses on the development of the technologies required for substantially autonomous robotic airships. The authors treat airborne and ground hardware and software infrastructures; airship dynamic modeling and simulation; control and guidance methods; visual servoing strategies; robotic air-ground cooperation; dynamic target recognition; and hybrid airship robotic software architecture. Successful airship autonomous flight achieved through a set of pre-defined way points and simulation results of ongoing work are also presented.

1 Introduction

Besides their use for military surveillance, Unmanned Aerial Vehicles (UAVs) have a wide spectrum of potential civilian applications as observation and data acquisition platforms. They can be utilized in several fields related to biodiversity, ecological and climate research and monitoring (henceforth denoted simply by environmental monitoring). Inspection oriented applications covers different areas such as mineral and archaeological prospecting, agricultural and livestock studies, crop yield 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 requires maneuverable low altitude, low speed

airborne data gathering platforms. The vehicle should also be able to hover above an area, to present extended airborne capabilities for long duration studies, to 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 better suited than balloons, airplanes and helicopters [12], 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, the authors are currently developing Project AURORA [12], which focuses on the establishment of the technologies required for substantially autonomous operation of unmanned robotic airships for environmental monitoring and aerial inspection missions.

This article reports the current status of Project AURORA, and it is organized as follows. Section 2 briefly describes the project and presents the LTA platform used in its first phase. Section 3 describes the airborne and ground hardware and software infrastructure developed for the robotic airship. Airship dynamic modeling and simulation environments are treated in Section 4. Control and guidance strategies are the subject of Section 5, which presents successful airship autonomous flight achieved through a set of pre-defined waypoints. The remaining sections of the article address on-going work. In Section 6, visual servoing strategies for hover and line following are treated, including simulation results. Section 7 describes initial studies focusing robotic air-ground cooperation. First results concerning dynamic target recognition are the subject of Section 8. A prototype of an hybrid airship robotic software architecture is described in Section 9. Finally, Section 10 concludes this article and afterwards the references are given.

2 AURORA project

AURORA's objective is the development of the underlying technologies for substantially autonomous airship operation. This includes sensing and processing infrastructures, control and guidance capabilities, the ability to perform mission, navigation, and sensor deployment planning and execution, failure diagnosis, 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 multi-phase 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, AURORA I, a robotic prototype has been built. It is a proof-of-concept system aiming the development and experimental validation of the underlying technologies, and the realization of low demanding pilot test applications.

The LTA platform of AURORA I is the AS800 by Airspeed Airships, a non-rigid 10.5m long, 3.0m diameter, and 34m³ of volume airship, whose payload capacity is approximately 10kg and its maximum airspeed is around 50km/h (Fig. 1).



Figure 1: AURORA I Airship AS800.

The airship actuators are its deflection surfaces and two main propellers disposed on each side of the gondola. The four deflection surfaces at the stern, arranged in an 'x' shape with allowable deflections in the range $[-25^\circ, +25^\circ]$, generate the equivalent rudder and elevator commands of the classical '+' tail. The two propellers, which are driven by two-stroke engines, can be vectored within the interval $[-30^\circ, +120^\circ]$. A small stern thruster is also available, perpendicular to the airship longitudinal axis, to introduce one extra horizontal actuation for hovering tasks.

3 Hardware and basic software infrastructure

This section briefly describes the basic software and hardware infrastructure of the airship, which is composed by three main components: the onboard station, the ground station and the communication system [26, 23]. While in the sequence the specific commercial products being used are mentioned, we stress that we are replacing some of them by updated products with improved performance.

3.1 Basic software

As the underlying real time operating system, RT-Linux is presently used, which is an open source reliable and robust solution. The Linux kernel was striped down, so that the onboard version fits into a flash disk. Application software is most written in C++, following object oriented approach and the UML methodology.

3.2 Communication system

Communication between the onboard and ground stations is based on two radio links. The first one transmits analog video imagery from the airship to ground. The second one, which is composed of a pair of spread spectrum radio modems, transmits digital sensor telemetry and command data between both stations.

3.3 Onboard station

The onboard station includes processors, sensors, actuators, and part of the communication system. The CPU is a Pentium in a PC/104 standard, with serial, parallel and Ethernet interfaces, and a flash disc. Sensors are connected to the PC/104 computer via serial ports or a CAN bus. All actuators are connected to the PC/104 via microcontrollers, 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 aboard).

Sensor package comprises those used for control and navigation, as well as specific mission sensors selected according to aerial data-gathering needs. The control and guidance sensor suite includes:

- A GPS receiver in a PC/104 compatible board (Trimble Navigation SvecSix) with differential correction from a second receiver located at the ground station.
- An Inertial Measurement Unit (Crossbow DMU-AHRS), which provides the roll, pitch, and yaw (heading) attitude, angular rates and body axes linear accelerations, and compass information;

- A wind sensor (the Air Data Measurement Unit) built by the IDMEC/IST [6]. It measures the airspeed, the aerodynamic incidence angles and the barometric altitude;

and propeller speed sensors, as well as sensors used for vehicle state and diagnosis purposes (e.g. control surface and vectoring position sensors, engine temperature, and fuel and battery levels). Cameras mounted on the airship's gondola, that provide aerial images to the operator on the ground, will also allow visual control and navigation based on artificial/natural features of the terrain.

3.4 Ground station

The ground station is composed of a portable computer with serial ports to which the radio modem and the differential GPS receiver are connected.

The human-machine interface (HMI) is a part of the ground station and provides the visualization and interaction mechanism between the operator and the airship onboard system. In AURORA I, commands and mission paths sent to the onboard station are specified by the operator as a flight plan composed of a set of waypoints and altitude profiles, along with the flight primitives (e.g., take-off, cruise, hover, land). During airship flight, telemetry data received from the airship onboard station, comprising control and navigation sensors as well as mission sensors information, is stored and displayed in real time to the operator, which also receives imagery generated aboard. Part of the HMI is illustrated in Fig. 2, showing inertial data and GPS-based trajectory plotted over a map. The HMI is able to perform flight playback from recorded data for post-flight analysis and mission evaluation purposes.

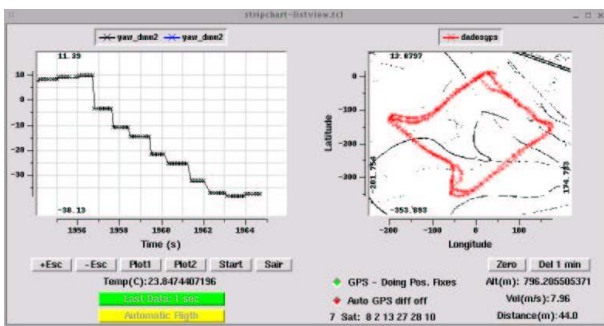


Figure 2: Real-time flight data visualization.

The HMI is being evolved to a web-based structure. It includes the animation of the airship in a virtual flight over a real-world topographical representation of the terrain (Fig. 3); and resources to make teleme-



Figure 3: VRML/Java airship simulator.

try data on-line available worldwide through the Internet. Future work will address interfacing the HMI to a geographical information system (GIS).

4 Airship dynamics and simulation

As the basis for the development of the control and guidance strategies, a 6-DOF physical model of the airship was developed including the non-linear flight dynamics of the system [20, 3, 1], based on previous extensive wind tunnel experiments [19].

The dynamic model describes the motion as referenced to a system of orthogonal body axes fixed in the airship, with the origin at the Center of Volume (CV), assumed to coincide with the gross Center of Buoyancy of the airship (CB).

The orientation of this body-fixed frame (X, Y, Z) with respect to an Earth-fixed frame (N, E, D) is obtained through the Euler angles (ϕ, θ, ψ) corresponding to the roll, pitch and yaw angles, respectively. The airship dynamics is then expressed as:

$$\mathbf{M} \dot{\mathbf{x}}_A = \mathbf{F}_d(\mathbf{x}_A) + \mathbf{F}_a(\mathbf{x}_A) + \mathbf{F}_p + \mathbf{F}_g \quad (1)$$

where \mathbf{M} is the 6×6 mass matrix; $\mathbf{x}_A = [u, v, w, p, q, r]^T$ is the vector of airship linear and angular velocities, and \mathbf{F}_d , \mathbf{F}_a , \mathbf{F}_p , \mathbf{F}_g , indicates the vector of forces and moments due to kinematics, aerodynamics, propulsion, and gravity/buoyancy, respectively.

Identification procedures are currently being conducted to refine this model from flight data [22].

Using the 6-DOF non-linear model, two simulation environments were built: one SIMULINK-based simulator allowing the design and validation of flight control and guidance strategies [7], and a

VRML/Java simulator [24], to allow the visualization of the airship flight, as well as the sensor data, in a standard airplane-type cockpit (Fig. 3).

5 Control system

The airship control and guidance system is designed as two decoupled main modules, one for the longitudinal and other for the lateral mode, each of them commanding the corresponding actuators based on a given mission profile. The longitudinal control is still under development so that, in what follows, the airship velocity is under manual control. The longitudinal and lateral control modules are presented in the sequel.

5.1 Lateral path tracking control

One of the most important mission problems is the flight path tracking through a set of pre-defined waypoints in latitude/longitude. Allowable paths are defined as a sequence of straight lines or circle arcs between the defined waypoints [2]. The tracking error for the case when the flight path is composed of straight lines is obtained as follows [4]. For a constant speed, the path tracking problem is linearized according to the model:

$$\dot{\delta} = V \sin(\varepsilon) \approx V_0 \varepsilon, \quad \dot{\varepsilon} = r \quad (2)$$

where V_0 is the reference ground speed considered for design purposes, r is the yaw rate, δ is the distance error (orthogonal) to the desired path, and ε is the angular error (angle between the ground velocity vector and the trajectory line). In order to accommodate both the distance and angular errors in a single equation, a look-ahead error, δ_a , considering a prediction horizon Δt , may be estimated some time ahead of the actual position:

$$\delta_a = \delta + \dot{\delta}\Delta t \approx \delta + V_0 \varepsilon \Delta t \quad (3)$$

The lateral control system is composed of a path tracking guidance controller (as outer loop) and a heading controller (as inner loop), yielding the rudder deflection (ζ) as output (Fig. 4).

The PI guidance controller input is the look-ahead path tracking error δ_a given in Eq. (3). The controller output, added to the trajectory heading angle ψ_{traj} , yields the reference signal ψ_{ref} for the heading controller.

The PD heading controller objectives are the tracking of a yaw angle reference input, and the attenuation of the disturbance (wind and gust), taking

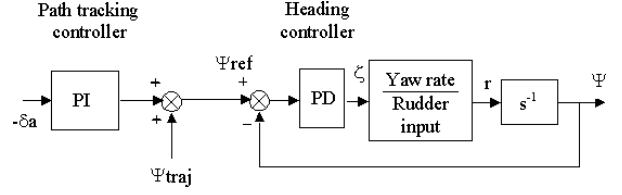


Figure 4: Guidance controller block diagram.

into account the lightly damped poles of the lateral dynamics.

Considering Eq. (3) and the PI algorithm, it can be shown that the outer loop path tracking guidance controller is equivalent to a PIV control strategy [23], as given by the control law:

$$\begin{aligned} u(t) &= (K_p + K_i \Delta t) \delta(t) + \\ &+ K_i \int \delta(t) dt + K_p \Delta t (V_0 \varepsilon) \\ &= K_{PT} \delta(t) + K_{IT} \int \delta(t) dt + K_{VT} \dot{\delta} \end{aligned} \quad (4)$$

where $\dot{\delta}$ is the velocity component orthogonal to the trajectory.

The guidance PI gains (as well as the corresponding PIV ones) can be calculated using any classical LTI design technique, by considering the linear model composed of the linearized lateral dynamics of the airship plus the linearized tracking error model in Eq. (2). The design of the heading controller gains may follow an H_2/H_∞ PID design technique, as presented in [8]. An additional roll controller, using the aileron input, may also be included [4] in order to reduce undesired rolling oscillations.

5.2 Longitudinal altitude control

The longitudinal control algorithm, at present, is composed of a single altitude controller given by [7, 23]:

$$\eta = K_{ph}(h - h_{ref}) - w + K_\theta \theta_{deg} + K_q q \quad (5)$$

where h and h_{ref} are the current and reference altitude, w is the vertical airship velocity, θ_{deg} is the pitch angle in degrees, and q is the pitch rate.

An advanced longitudinal control system, based on a feedback/feedforward structure with integrated propeller vectoring and speed control, is currently under development. It will enable the airship to automatically perform take-off, cruise, hover and landing operations for the whole range of operational airspeeds.

5.3 Validation in experimental flights

The lateral and longitudinal controls presented above were implemented in the airship onboard computer.

The airship velocities and heading, necessary in the control algorithms, were directly obtained respectively from data provided by the differential GPS (at 1Hz sampling frequency), wind sensor and the compass (both at 10Hz). No sensor fusion scheme (which is under development) is explored. Control calculations and actuation were performed at 10Hz [23].

During the experiments, the take-off and landing procedures were handled by the pilot, on ground. After take-off the pilot brought the airship to a cruise flight state and switched from manual to automatic flying mode. The results for one of the flight experiments under automatic operation are shown in Figs. 5 and 6, which correspond to 2 complete turns along a square path with sides of 200m.

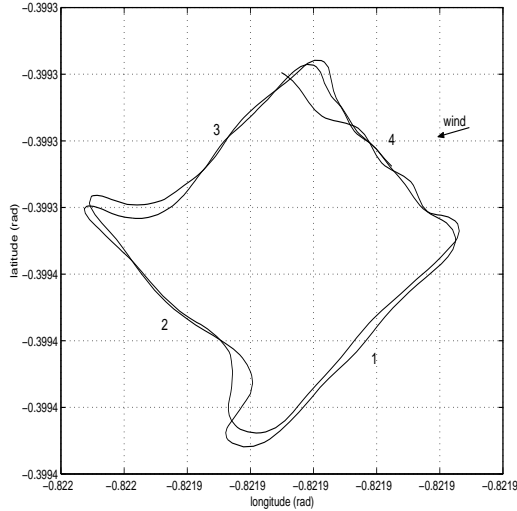


Figure 5: AURORA I under automatic guidance control following a set of four predefined waypoints.

In Fig. 5, one can clearly observe the adherence to the mission path, as well as overshoot for turns indicated by “1-2” and “2-3”, due to the disturbing wind coming from the tail. Figure 6 shows the corresponding yaw angle (ψ), pitch angle (θ), and barometric altitude (h) evolution during task execution. The transition points are represented by vertical dotted lines in the Fig. 6. One can also see 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.

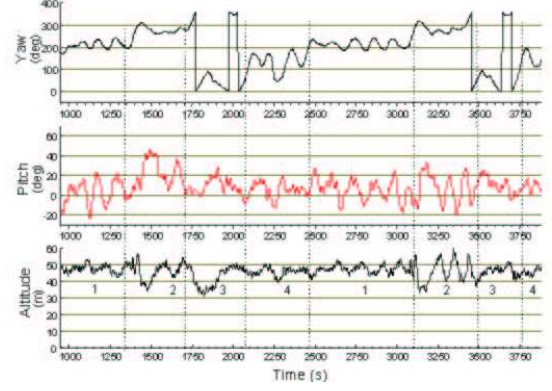


Figure 6: Corresponding yaw angle (ψ), pitch angle (θ), and barometric altitude (h).

6 Visual servoing

In this section, the conceptual developments and simulation results regarding the visual servoing of the airship are presented. Techniques were applied to control the vehicle to ensure the capture of the desired task-dependent image frame from the camera aboard.

The mapping between the motion of the image feature parameters \dot{s} and the motion of the camera frame $T_c = [V, \Omega]^T$ is modeled as:

$$\dot{s} = L^T T_c + \frac{\partial s}{\partial t} \quad (6)$$

where L^T is called the interaction matrix and $\frac{\partial s}{\partial t}$ represents possible target motion. The methodology to derive the interaction matrix for generic geometric primitives presented in [16] is used in the tasks described in the sequel. More details on the design of robust sensor-based tasks can be found in [28].

6.1 Hovering tasks

For the problem of automatic hovering of the airship over a reference target, both the target and the corresponding visual signals were selected to leave the heading unobserved, such that the yaw angle is indetermined what allows to cope with a previously unknown wind direction.

The selected target is composed of a circle (radius $r_c = 2\text{m}$) on the ground and a ball floating just over the center of the circle at a height $h_b = 4\text{m}$. In the image plane, the circle appears as an ellipse with center (x_c, y_c) and the centroid of the ball has coordinates (x_b, y_b) . The vector of visual signals for the hovering case is then chosen as:

$$s_{hov} = [x_c, y_c, x_b, y_b, \zeta]^T, \text{ with } \zeta = \frac{1}{\sqrt{\mu^{2,0} + \mu^{0,2}}} \quad (7)$$

where $\mu^{2,0}$ and $\mu^{0,2}$ are the *reduced moments* of the ellipse.

From such a vector of visual signals, an analytical form of the interaction matrix may be derived [27]. By using the model of L_{hov} at the desired configuration, whose corresponding desired vector of visual signals is $s_{hov}^* = [0, y_c^*, 0, y_b^*, \frac{1}{2r_c}]^T$, the interaction matrix is given by

$$L_{hov}^T|_{s=s^*} = \begin{bmatrix} \frac{-1}{d^*} & 0 & 0 & 0 & 1 & y_c^* \\ 0 & \frac{-1}{d^*} & y_c^{*2} & 1 - y_c^{*2} & 0 & 0 \\ \frac{-1}{(d^* - h_b)} & 0 & 0 & 0 & 1 & y_b^* \\ 0 & \frac{-1}{(d^* - h_b)} & y_b^{*2} & 1 - y_b^{*2} & 0 & 0 \\ 0 & 0 & \frac{1}{2r_c} & \frac{3y_c^* d^*}{4r_c} & 0 & 0 \end{bmatrix} \quad (8)$$

where d^* is the desired “vertical” distance from the optical center to the center of the circle.

Concerning control, through the linearization of the non-linear model given in Eq. (1), a LTI model that tackles both the dynamics of the airship and the vision process can be obtained. After its discretization and by applying a steady-state LQR approach, an output feedback control scheme is used,

$$\mathbf{u}_{hov}(k) = -K_{hov}(\infty)y(k), \quad (9)$$

with output vector $y = [s, \dot{s}]^T$.

The results with the airship initially located at the position $(N_0, E_0, D_0) = (-15, 0, -25)$, and the center of the target located at $(0, 20, 0)$ depicted by the inner circle, are presented in Fig. 7. The objective is to reach and to keep the camera frame on the dashed circle, with an appropriate translation to the image plane (s_{hov}^*). The errors in the image plane are depicted in Fig. 8. For further information and different simulation cases, the reader may refer to [5].

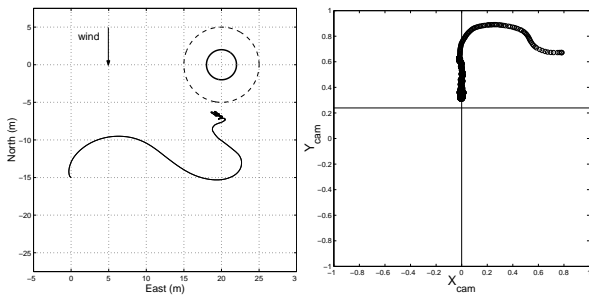


Figure 7: Left: evolution of the airship CV position (located 1.25m behind the camera frame) in Earth-fixed frame. Right: target center path in image plane (case: wind=2m/s, gust=1m/s).

6.2 Line following tasks

Other example of sensor-based task of substantial interest for aerial vehicles is the capability to follow

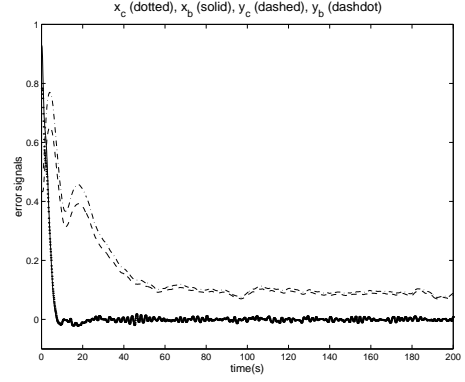


Figure 8: Error signals in the image plane (case: wind=2m/s, gust=1m/s).

linear features lying on the ground, such as pipelines, power lines, rivers, and roads. The visual servoing framework developed in [16] allows to tackle such tasks with the same approach made for the hovering case.

To control the 3D airship position and attitude with regard to the ground frame, the image of the three parallel straight lines (e.g., the two sides of a road and its central line) are sufficient. The problem is formulated in the image plane as the regulation of the image so that the central line is intended to be vertically centered with both lateral lines lying symmetrically with a certain inclination. Using polar representation for the lines, the vector of image feature parameters is

$$s_{line} = [\theta_1, \rho_1, \theta_2, \rho_2, \theta_3, \rho_3]^T, \quad (10)$$

whose analytical form of the associated interaction matrix computed at the desired configuration s_{line}^* is given by:

$$L_{line}^T|_{s=s^*} = \begin{bmatrix} -\frac{\cos^2 \theta_1^*}{h^*} & -\frac{\cos \theta_1^* \sin \theta_1^*}{h^*} & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 \sin \theta_1^* & -\cos \theta_1^* & 0 & 0 \\ -\frac{\cos^2 \theta_2^*}{h^*} & -\frac{\cos \theta_2^* \sin \theta_2^*}{h^*} & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 \sin \theta_2^* & -\cos \theta_2^* & 0 & 0 \\ -\frac{\cos^2 \theta_3^*}{h^*} & -\frac{\cos \theta_3^* \sin \theta_3^*}{h^*} & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 \sin \theta_3^* & -\cos \theta_3^* & 0 & 0 \end{bmatrix} \quad (11)$$

where h^* is the desired height.

For the vision-based lateral control, only one line is necessary and such case is presented in [30].

The null space of the interaction matrix given in Eq. (11) reveals the degree of freedom not constrained by the visual servoing (motions of the camera frame that leaves the projection of the three lines unchanged). For this DOF, a secondary task may be designed and it is given a constant forward speed ν in this work.

As for the hovering case, an augmented and discretized LTI system can be derived after linearization of the non-linear system for the trajectory tracking. The controller design is performed by using a steady-state LQR formulation in the sensor space. The resulting control law has the form

$$\mathbf{u}_{line}(k) = -K_{line}(\infty)C^+y(k), \quad (12)$$

obtained through the substitution of $x(k) = C^+y(k)$ where the superscript $(\cdot)^+$ denotes the pseudoinverse.

The navigation of the airship in Cartesian space under the visual servo control scheme, with the airship initially located at the position $(N_0, E_0, D_0) = (-300, -40, -30)$ with Euler angles $\phi_0 = \theta_0 = \psi_0 = 0$ and $\nu = 10\text{m/s}$, is presented in Fig. 9. The objective is to track the trajectory in the image plane, with an altitude of $h^* = -15\text{m}$ properly translated to the desired vector of visual signals (s_{line}^*) . The evolution of the error in the image plane is shown in Fig. 10. Preliminary results of using a steady-state Kalman filter is presented in [31]. Further information and simulation cases are available in [29].

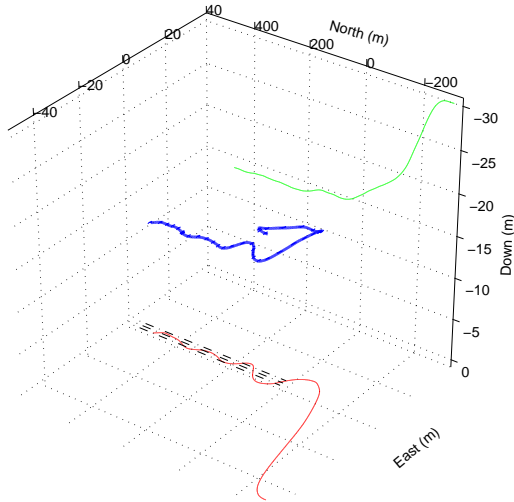


Figure 9: Airship navigation in Cartesian space (blue) with its projections onto NE (red) and ND (green) ($gust = 1\text{m/s}$).

The hovering and line following approaches are under refinements, with their practical implementation and validation on the way. Future work will address visual-based landing as well as tracking of others visual targets.

7 Cooperative air-ground robotics ensemble

Robotic collectives composed of properly selected and coordinated heterogeneous robots will greatly

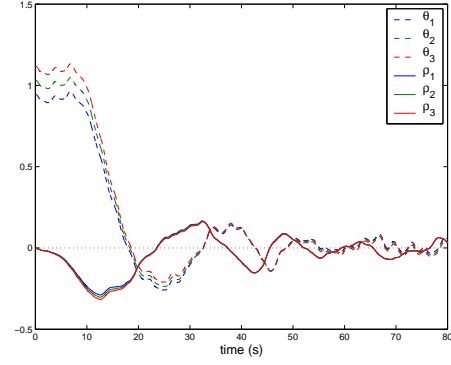


Figure 10: Evolution of the error in the image feature parameters θ_i and ρ_i , $i = 1, 2, 3$ ($gust = 1\text{m/s}$).

extend the capability of human beings to perform complex tasks. This section presents initial studies concerning air-ground robotic ensembles, which are of substantial interest for a large class of applications.

The basic idea is to use images obtained from a camera aboard an aerial vehicle to navigate a ground robot. The methodology framework being developed is composed of three levels, respectively image processing, path planning, and path following (Fig. 11). Further details can be found in [14].

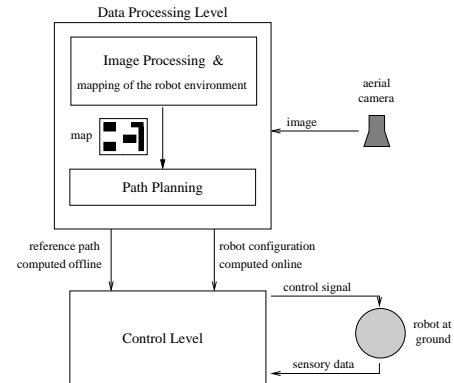


Figure 11: Three-level structure of the visual-based navigation framework.

The upper level: (i) receives an aerial image as input; (ii) identifies on it the ground robot current location, its target position, and obstacles in the environment; and (iii) builds a map which comprises the available free space. The middle level uses this information to find a path for robot navigation. Finally, the lower level takes as inputs the path planned and the kinematic and dynamic descriptions of the robot, and guides it along the computed trajectory. An example with the Nomad XR4000 navigating in an outdoor environment is shown in Fig. 12, where the white line represents the planned path. Extensions of such strategy are under development. Fur-



Figure 12: Path planning for a ground robot in an outdoor environment.

ther work will address mapping and self-localization issues.

8 Dynamic Target Recognition

In this section, the development of autonomous target identification and classification capabilities that are essential for systematic and long-term airborne inspection and monitoring is briefly outlined. Details may be found in [13, 15]. The approach implements a cycle of hypothesis formulation, experiment planning for hypothesis validation, experiment execution, and hypothesis evaluation [9] to confirm or reject the classification of targets into given object classes (Fig. 13) [11, 10].

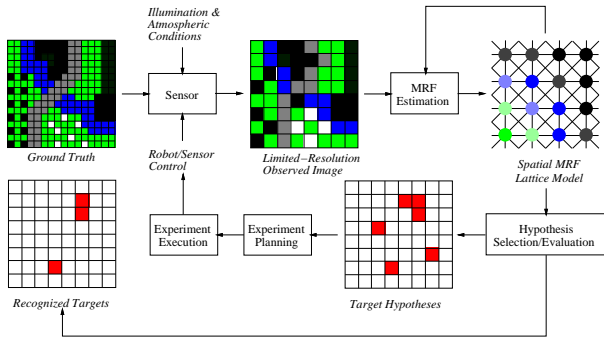


Figure 13: System architecture for dynamic target identification: a target selection, validation and classification cycle is implemented.

8.1 Approach

As a representational framework, sensor observations are encoded using stochastic lattice models [21]. For target identification and classification a classical hypothesis testing approach is used [18].

For a set of sensor observations X , the observations are to be classified into one of c classes $\omega_1, \dots, \omega_c$. This is done by assigning the observation X to class

k such that $p_k[\omega_k|X] = \max_i p_i[\omega_i|X]$. In environmental monitoring applications, it is often necessary to discriminate certain target classes against a variety of non-targets and background imagery. Because of their large number and variety, it would be unfeasible to determine the distributions for all potential non-targets before designing a classifier system. In the approach presented here, this is done using a single hypothesis test, which involves measuring the distance of the object hypothesis to the cluster mean (in feature space), normalized by the target covariance matrix, and applying a threshold to determine whether the observation is classified into one of the relevant classes.

To control the acquisition of new data in an optimal way, an approach derived from the theory of optimal design of experiments [17] is used to discriminate hypotheses based on the entropy measure. As above, consider the need to assign a given target to a certain class among a number of c classes. From the observations obtained so far, a set of prior probabilities $p_0[H_j], j = 1, \dots, c$ is available that corresponds to the hypotheses of the target belonging to the classes ω_j . An optimal experiment (new sequence of observations) will be the one that maximizes the expected mean increment of information $E[\Delta I(\mathcal{E})]$, given by:

$$\begin{aligned} \Delta I(\mathcal{E}) &= \eta(p[H]) - \eta(p_0[H]) \\ &= - \sum_{j=1}^c p_0[H_j] \ln p_0[H_j] + \sum_{j=1}^c p[H_j] \ln p[H_j] \end{aligned} \quad (13)$$

8.2 Target Identification and Tracking Using Aerial Imagery

Some initial results obtained from adaptive aerial target identification and classification using visual imagery from airborne cameras are presented next [11, 10].

Initial experiments were done using the approach outlined above to perform autonomous identification, recognition and tracking of man-made targets using aerial imagery. Target classes of relevance to specific perceptual tasks of a robot mission are described through parameterization of the expected sensor data, and the candidate areas are selected using object templates that are matched probabilistically against the existing map. The regions corresponding to potential targets of interest are labelled as object hypotheses, and are called Loci of Interest. As regions are identified as having a significant probability of being instances of the desired targets, data acquisition is planned in such a way as to confirm or deny the presence of these targets in the candi-



Figure 14: Identification and tracking of a paved road using an airborne camera. The aerial imagery is shown on the left, and the results of the target classification procedure on the right.

date sites, and validate the classification. The experiments planned by the system include sensor selection, setting of sensor parameters, attention focusing on the Loci of Interest, computation of Loci of Observation to allow the relevant regions to be observed with greater accuracy, and acquisition of multiple views of the Loci of Interest to help in the disambiguation of target hypotheses.

Fig. 14 shows results from paved road identification and tracking. Identification and segmentation of the roads in the images was done using probabilistic measures based on the spectral characteristics of the targets in the visible RGB bands. As can be observed in the results, due to atmospheric conditions and sensor limitations the approach provides a higher correct classification rate for road portions closer to the airborne camera, while misclassifying parts of the imagery that are further away from the airship. As the airborne vehicle comes closer to the new target regions, the change in the distributions of the observations leads to a correct reclassification.

9 Airship robotic software architecture

The complexity of the autonomous capabilities for the system being developed calls for an hybrid (deliberative-reactive) robotic software architecture. Here, a first prototype of the airship Robotic Soft-

ware Architecture (RSA) is briefly described. It was developed in a distributed software environment [25] that comprises the airship model and simulation tools and mimics the onboard and ground stations. Simulation tests performed had covered a large set of situations leading to different deliberative and reactive actions [23].

The deliberative part of the airship RSA is mainly related to planning actions; its reactive part deals with internal and external contingencies. Internal contingencies come from malfunctioning of any component of the onboard, ground, or communication systems, whereas an external contingency is typically generated by disturbances such as wind or meteorological conditions, or those arising from the flying environment. Also, contingencies varies in terms of seriousness, asking for different recovery actions.

The airship RSA follows the multi-agent approach proposed in [33] and is composed by the Ground Agent (GA) and the Onboard Agent (OA), that exchange information through the communication system, as shown by Fig. 15.

Planners are written in high level programming languages. Executive levels are built by using the Task Description Language (TDL) [32], that provides a framework to support the construction and execution of deliberative and reactive modules. Behaviors

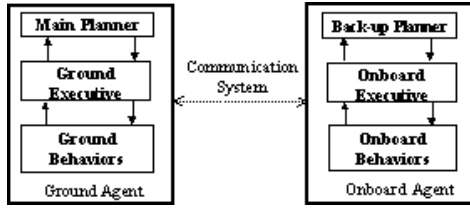


Figure 15: General view of AURORA's RSA.

are written in under RTLinux to satisfy real time constraints.

The GA and OA components are the following:

Main Planner (MP): From a user mission request, the MP deliberative actions generate a mission script composed by a set of tasks. This script is sent to the onboard station via Ground Executive. During the airship flight, the MP can be called to generate an alternative plan.

Ground Executive (GE): During the airship flight, the GE monitors the script execution as well as the environmental situation in order to detect and correct for any mismatch between the planned and the current status. In case of a significant difference, due to a serious contingency for example, the GE will trigger the MP for generating a new plan.

Ground Behaviors (GB): The GB interact with the human-machine interface, the ground station sensors and the communication system. They are mainly of support nature, but also present basic reactive actions.

Back-up Planner (BP): The BP is a simplified version of the MP, compatible with the onboard processing power and memory limitations.

Onboard Executive (OE): The OE receives a mission script from the GE and maps it into a set of behaviors that have to be triggered or suspended to accomplish the mission execution. During the flight, the OE assure the behaviors evolve properly. In case of a serious contingency, where a replanning reaction is needed, if there is a lack of communication with ground, the BP is activated; otherwise, replanning is treated at ground by the GE that triggers the MP.

Onboard Behaviors (OB): Support type OB interact and perform basic reactive actions with respect to the sensors, actuators and communication system. Control OB are directly associated to the airship control and guidance system. They constitute a set of basic movements, such as take-off, cruise flight at a certain trajectory, hover and landing.

This architecture is being implemented in the airship system and it will be gradually evolved.

10 Conclusions

This article presented the current status of Project AURORA, which focuses on the development of substantially autonomous robotic airships. The main motivation for such project is the use of this airborne platform for inspection and environmental monitoring missions.

Most relevant practical field results obtained so far consist of path tracking through way points following a given altitude profile. Visual servo control schemes, validated in a realistic simulation environment, are being incorporated to enable hovering and line following tasks. Dynamic target recognition and tracking from aerial imagery represent an important framework to extend the airship's autonomous capabilities. Initial studies were also described for a three-level vision-based air-ground robotic cooperation, as well as for an airship robotic software architecture to support deliberative and reactive actions.

The overall robotic system is constantly being evolved, and new research areas are brought to the project as it progresses.

Acknowledgments

The authors recognize the enormous contribution to Project AURORA brought by M. Bergerman, S.B.V. Gomes, S.M. Maeta, L.G.B. Mirisola, B.G. Faria and M.F.M. Campos.

This work is partially funded by the Brazilian agencies FAPESP under grant 97/13384-7 and CNPq under grants 680260/01-3 (CNPq-ProTeM-CC/INRIA) and 910094/99-3 (CNPq/ICCTI). Dr. José R. Azinheira was also supported by the Portuguese Operational Science Program (POCTI), co-financed by the European FEDER Program.

References

- [1] J. Azinheira, E. de Paiva, and S. Bueno. Influence of wind speed on airship dynamics. *AIAA Journal of Guidance, Control and Dynamics*, 2002. accepted for publication.
- [2] J. R. Azinheira, E. C. de Paiva, J. J. G. Ramos, and S. S. Bueno. Guidance of an autonomous unmanned airship. In *15th Bristol International UAV Systems Conference*, Bristol, UK, 2000.
- [3] J. R. Azinheira et al. Extended dynamic model for AURORA robotic airship. In *14th AIAA Lighter-than-air Technology Conf.*, USA, 2001.
- [4] J. R. Azinheira et al. Lateral/directional control for an autonomous unmanned airship. *Aircraft Engineering and Aerospace Technology*, 73(5):453–458, September 2001.
- [5] J. R. Azinheira et al. Visual servo control for the hovering of an outdoor robotic airship. In *IEEE Int.*

- Conf. on Robotics and Automation*, pages 2787–2792, Washington, D.C., USA, May 2002.
- [6] J. R. Azinheira, H. V. Oliveira, and B. F. Rocha. Sistema de medies aerodinmicas para um dirigvel do projeto AURORA. Technical report, IDMEC/IST, Lisbon, Portugal, 2000.
 - [7] E. C. de Paiva, S. S. Bueno, S. B. V. Gomes, J. J. G. Ramos, and M. Bergerman. A control system development environment for AURORA's semi-autonomous robotic airship. In *IEEE Int. Conf. on Robotics and Automation*, pages 2328–2335, Detroit, USA, May 1999.
 - [8] E. C. de Paiva, J. R. H. Carvalho, P. A. V. Ferreira, and J. R. Azinheira. An H_2/H_∞ PID heading controller for AURORA-I semi-autonomous robotic airship. In *14th Lighter-than-air Systems Technology Conf.*, Ohio, USA, 2001.
 - [9] A. Elfes. Robot navigation: Integrating perception, environment constraints and task execution within a probabilistic framework. In L. Dorst, M. van Lambalgen, and F. Voorbraak, editors, *Reasoning With Uncertainty in Robotics*, volume 1093 of *Lecture Notes in Artificial Intelligence*, Berlin, Germany, 1996. Springer-Verlag.
 - [10] A. Elfes, M. Bergerman, and J. R. H. Carvalho. Dynamic target identification by an aerial robotic vehicle. In G. Barattoff and H. Neumann, editors, *Dynamic Perception*. AKA, Berlin, September 2000.
 - [11] A. Elfes, M. Bergerman, and J. R. H. Carvalho. Towards dynamic target identification using optimal design of experiments. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000.
 - [12] A. Elfes, S. S. Bueno, M. Bergerman, and J. G. Ramos. A semi-autonomous robotic airship for environment monitoring missions. In *IEEE International Conference on Robotics and Automation*, pages 3449–3455, Louvain, Belgium, May 1998.
 - [13] A. Elfes, J. R. H. Carvalho, M. Bergerman, and S. S. Bueno. Towards a perception and sensor fusion architecture for a robotic airship. In G. T. McKee and P. S. Schenker, editors, *Sensor Fusion and Decentralized Control in Robotic Systems IV, Proceedings of SPIE*, volume 4571, pages 65–74, Oct. 2001.
 - [14] A. Elfes et al. Air-ground robotic ensembles for cooperative applications: Concepts and preliminary results. In *International Conference on Field and Service Robotics*, Pittsburgh, USA, August 1999.
 - [15] A. Elfes et al. *Perception and control for an autonomous robotic airship*. Modelling of Sensor-Based Intelligent Robot Systems. Springer-Verlag, New York, USA, Nov. 2001. H. Bunke, H. I. Christensen, G. Hager, R. Klein, Eds.
 - [16] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Rob. and Automation*, 8(3):313–326, 1992.
 - [17] V. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.
 - [18] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition, 1990.
 - [19] S. B. V. Gomes. *An investigation of the flight dynamics of airships with applications to the YEZ-2A*. PhD thesis, Cranfield Institute of Technology – College of Aeronautics, Inglaterra, Outubro 1990.
 - [20] S. B. V. Gomes and J. J. G. Ramos. Airship dynamic modeling for autonomous operation. In *IEEE Int. Conf. on Robotics and Automation*, pages 3462–3467, Louvain, Belgium, May 1998.
 - [21] T. Kämpke and A. Elfes. Markov sensing and superresolution images. In *Proceedings of the 10th INFORMS Applied Probability Conference (AP99)*, Ulm, Germany, July 1999.
 - [22] E. D. Paiva, J. Azinheira, J. Ramos, B. Faria, and S.S.Bueno. Identification methodology for the dynamics of AURORA project airship. In *4th International Airship Convention and Exhibition*, Cambridge, U.K., July 2002.
 - [23] J. J. G. Ramos. *Contribuição ao Desenvolvimento de Dirigíveis Robóticos*. PhD thesis, UFSC – Federal University of Santa Catarina, Brazil, 2002.
 - [24] J. J. G. Ramos et al. Development of a VRML/JAVA unmanned airship simulating environment. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1354–1359, Kyongju, Korea, Oct. 1999.
 - [25] J. J. G. Ramos et al. A software environment for an unmanned autonomous airship. In *Int. Conf. on Advanced Intelligent Mechatronics*, pages 1008–1013, Atlanta, GA, USA, Sept. 1999.
 - [26] J. J. G. Ramos et al. Project AURORA: A status report. In *3rd International Airship Convention and Exhibition*, Friedrichshafen, Germany, July 2000. Article B7.
 - [27] P. Rives and H. Michel. Visual servoing based on ellipse features. In *SPIE*, Boston, USA, Sept. 1993.
 - [28] P. Rives, R. Pissard-Gibollet, and L. Pelletier. Sensor-based tasks: From the specification to the control aspects. In *6th Int. Symp. on Robotics and Manufacturing*, France, May 1996.
 - [29] G. F. Silveira. Controle servo visual de veículos robóticos aéreos. Master's thesis, UNICAMP – State University of Campinas, Brazil, April 2002.
 - [30] G. F. Silveira et al. Lateral control of an aerial unmanned robot using visual servoing techniques. In *IEEE 2nd Workshop on Rob. Mot. and Control*, pages 263–268, Poland, Oct. 2001.
 - [31] G. F. Silveira et al. Optimal visual servoed guidance of outdoor autonomous robotic airships. In *American Control Conf.*, pages 779–784, Anchorage, USA, 2002.
 - [32] R. Simmons and D. Apfelbaum. A task description language for robot control. In *Conf. on Intelligent Robotics and Systems*, Canada, Oct. 1998.
 - [33] R. Simmons et al. First results in the coordination of heterogeneous robots for large-scale assembly. In *Int. Symposium on Experimental Robotics*, Honolulu, Hawaii, December 2000.