System Architecture of a Robotics Airship

José Reginaldo H. Carvalho[1(✉)], Miguel Á. C. Rueda[2], José R. Azinheira[3], Alexandra Moutinho[3], Luiz G. B. Mirisola[4], Ely C. de Paiva[5], Lucas A. C. O. Nogueira[2], Gustavo A. Fonseca[2], Josué Jr. G. Ramos[2], Mauro F. Koyama[2], Samuel S. Bueno[2], and Christian Amaral[6]

[1] Instituto de Computação, UFAM, Av. Rodrigo Otávio, 6200, Manaus, AM 69077-000, Brazil
reginaldo@icomp.ufam.edu.br
[2] CTI Renato Archer, Rod. D.Pedro I (SP-65) km.143.6, Campinas, SP 13069-901, Brazil
[3] IDMEC/LAETA/IST, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisbon, Portugal
[4] IEC/ITA, Pç Marechal Eduardo Gomes, 50, São José dos Campos, SP 12228-900, Brazil
[5] FEM/UNICAMP, Rua Mendeleyev 200, Campinas, SP 13083-970, Brazil
[6] Omega Aerosystems, R. Domingos Cordeiro, 389, Campo Largo, PR 83601-120, Brazil

Abstract. This paper aims to present the conception, design, and realization of a robotics airship. Differently, from the classic bi-propulsion, this vehicle has four vectored thrusters. Tail surfaces complete the directional actuation. The project has the code-name DRONI, the acronym for "Dirigível Robótico de Concepção Inovadora" (Robotic Airship with an Innovative Design in free translation). The initial motivation to start the project was the demand for an aerial platform for environmental monitoring of flooded areas in the Amazon region. This operation scenario imposes severe restrictions on the practical usage of quadrotors and fixed wings. High maneuverability, medium to long-endurance, flexible flight modes (hover, vertical takeoff and landing) are the most essential requirements for an effective aerial platform to overfly the vast Amazon canopy. Medium to long-endurance surveillance, cargo, and telecommunication relay are other common application of airships. The paper presents the current vehicle's architecture. The main subsystems shown are the robotic embedded infrastructure, the ground station and the communication system. The 6 kg payload enables the use of several types of cameras and sensors. Moreover, the paper presents the dynamic modeling with a control based on Incremental Nonlinear Dynamics Inverse. The inaugural flight is also shown, and the results are promising, towards a multipurpose aerial platform, that can be applied beyond environmental monitoring.

Keywords: Unmanned aerial vehicle · System architecture · Environmental monitoring

1 Introduction

Unmanned aerial vehicles (UAVs), which have been used in a wide range of applications, can be classified by the way the lift force for sustained flight is generated: (i) by fixed wings (various aircraft configurations), (ii) by rotating wings (helicopters, multi-rotors), (iii) by aerostatic principle (airships).

L. Pereira et al. (Eds.): IDEAS 2018, SIST 198, pp. 13–22, 2021.
https://doi.org/10.1007/978-3-030-55374-6_2

In the case of airships, the aerostatic lift comes from a lighter than air gas conditioned in an envelope, whereas its maneuverability is typically ensured by two controllable propellers mounted on a gondola under the envelope and aerodynamic actuation surfaces mounted on the tail. Interesting characteristics of these aircraft are greater intrinsic stability, greater flight autonomy, from low altitudes up to the stratosphere, at low speeds, their ability to hover, and to perform vertical takeoff and landing. On the other hand, some difficulties are less maneuverability due to the slower dynamics, under actuation, mainly in the lateral direction to the movement of the aircraft and more pronounced at low speeds, and the abrupt and strongly non-linear transition between low speed and cruising flights. The development of control systems for airships is, therefore, a complicated task. In this scenario, from the 1990s, interest in robotic airships increased, for example: in Brazil - the AURORA project [1], Germany [2], France [3], Korea [4], USA [5], Portugal [6], China [7]. With the evolution of electric propulsion, a small prototype equipped with four vectoring propellers, but dispensing the tail surfaces, was presented in [8]. Applications include long to medium term surveillance, telecommunication relay, cargo ships, and, as in the present case, environmental monitoring.

This article addresses the development of an unmanned robotic airship, with an innovative arrangement of four independent vectored electric propellers. This new configuration shows better stabilization and maneuverability at low speeds (hovering and vertical takeoff and landing), whereas maintains the efficiency of the aerodynamic control surfaces at the tail in the averages to high speeds (cruise flight). The airship is named NOAMAY - a word in the language spoken by the Yanomami Indians in the Amazon in Brazil, which means to take care of, to protect. Indeed, the purpose of the robotic airship is the realization of pilot-scale applications of environmental observation in the Amazon [9]. NOAMAY comprises the airship itself, its robotic infrastructure (onboard system, control station and communication system), as well as control and autonomous guidance system.

Figure 1 shows the first airship prototype. It has 11 m of length, 43 m^3 of volume and 6 kg as payload. The envelope has two layers: the inner one is composed of polyethylene polymer film, while the outer one is composed of high tensile rip-stop nylon synthetic fabric. At the bottom of the envelope, an aluminum frame receives the front and back propellers structure and, at its center, a rack that houses the embedded system. A distinct vectoring mechanism of $\pm 225°$ couples each of the four propellers. The aerodynamic surfaces, as actuators, can be arranged in "+" configuration (as in the figure) or in "x"

Fig. 1. NOAMAY airship during its inaugural flight.

pattern. The power system consists of 4 packs of LiPo batteries, which provide 22.2 V, 40 Ah, 888 Wh.

In this ongoing research context, this article focuses on the development of the robotics infrastructure associated with the aircraft and on the modeling and design of a nonlinear control system. Thus, after this introductory section, Sect. 2 summarizes the dynamic modeling of the aircraft and presents a control strategy based on inverse nonlinear incremental dynamics. Then, Sect. 3 describes the robotic infrastructure developed for the aircraft. Section 4 presents sensory data acquired during the airship's inaugural flight, in a remotely controlled way, demonstrating the capabilities of the airship and its robotic infrastructure. Finally, Sect. 5 concludes the article.

2 Robotics Infrastructure

Typically unmanned aerial vehicle holds at least three, well defined, subsystems i) the vehicle itself, avionics and embedded electronics within; ii) a ground station with mission planning capabilities iii) a reliable communication system to establish a secure link between the other sub-systems.

NOAMAY's robotics infrastructure, which is based on AURORA's project airship [10], is composed by:

- The onboard system, including autopilot, application processor, sensors and application-specific hardware for commuting between automatic operation mode (autopilot, autonomous flight mode), and manual (radio-controlled mode);
- A ground station, installed on a notebook.
- A communication system via radio modem.

NOAMAY solution consists of the same three subsystems, concerning to its functionalities, but with updated components (AURORA infrastructure is ten years old). Figure 2 shows the new infrastructure, also detailed in the sequence.

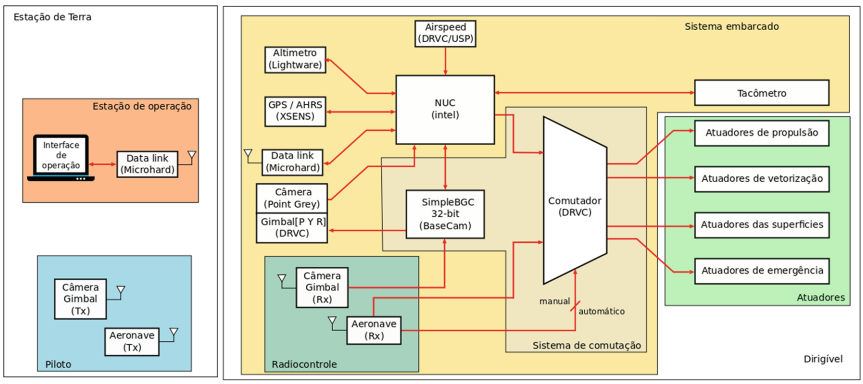

Fig. 2. Airship general architecture.

2.1 Embedded System

The embedded system integrates off-the-shelf components and overlaps a conventional radio control solution with PWM (pulse width modulation) commands. In addition to its primary functionality (ensuring the calculation on the embedded computer of control signals for the actuators when in automatic mode of operation), an additional feature is the logging of all commands sent to the actuators when in the manual operation mode. This peculiarity is useful, especially in the development phase, for example, when executing specific maneuvers to derive the dynamic behavior of the airship and collect data for the identification of model parameters. This solution is also attractive for the possibility of "robotization" of other radio-controlled vehicles.

The embedded system consists of an Intel NUC form factor computer, and a set of sensors integrated with USB, Ethernet and serial (RS-232) interfaces. This computer operates with Linux 14.04 Ubuntu and the Robot Operating System framework (ROS) [11], Indigo version, as *middleware*. Characteristics such as weight, power consumption, and interfacing capacity were the criteria for choosing the robotic components explained in Fig. 2, and listed as follows:

- GPS/AHRS MTi-G-700 (Xsens);
- Altímetro Laser SF11/C (Lightware);
- CMOS Flea 3 USB (Point Grey) 3.2MP camera;
- SimpleBGC 32-bit Gimbal Stabilizer (Basecam electronics);
- Airspeed/wind speed sensor
- Standard tachometer (rotating velocity of the propellers);
- Battery charge sensor.

The last three are self-made and are in development. As for the wind speed sensor, it uses a low-pressure sensor SDP-1000 (Sensirion), in the −50 to 500 Pa range and communicates with the on-board computer via Xbee at 900 MHz; a self-made pitot tube, designed for final calibration in a wind tunnel, completes the hardware.

A driver as ROS node was created for each of the sensors. There is also a Basecam electronics SimpleBGC 32-bit card and inertial units that will eventually stabilize a gimbal with the camera.

The actuators of the aircraft are four brushless motors for propulsion, four servo-actuators for vectoring the propellers and four aerodynamic surfaces with servo-actuators for their deflection. Besides, there is an emergency solenoid valve to release the helium gas out of the envelope.

The aircraft has two operation modes: manual (via radio control) and automatic (via embedded computer). The self-made electronics provide proper switching between these modes. The sensor data acquisition and the command dispatcher to the actuators are fully operational via ROS nodes. Also, in the ROS nodes, the control cycles with higher rates, necessary to implement the nonlinear control approach discussed in Sect. 3, will receive priority scheduling, reducing delays of communication.

2.2 Ground Station

The ground station uses a laptop computer with Linux Ubuntu and ROS Indigo. It is coupled to the onboard computer by a remote radio link (see Sect. 2.3), using the MAVLink protocol, which is computationally light enough to run on low-performance microcontrollers. Due to this, MAVLink is widely adopted by the UAV R&D community, especially the small ones [12]. Concerning the Ground Station core software, it was decided to use the program QGroundControl [13]. It was the first to implement the MAVLink protocol, built in synergy with the protocol development team. In addition, one can run it on Windows, Mac, and Linux. Then, it was only necessary to develop software elements to perform the communication between the ROS system in the vehicle and the MAVLink protocol. The user interface, shown in Fig. 3, enables:

- The mission schedule in terms of waypoints (latitude and longitude coordinates and altitude profile) and flight attributes (vertical takeoff or landing, hovering or cruising).
- The selection of control and navigation algorithms and the definition of their adjustment parameters.
- The reception and display, during the flight, of telemetry data (sensory information, aircraft status, power supply systems, etc.) as well as images from the camera.

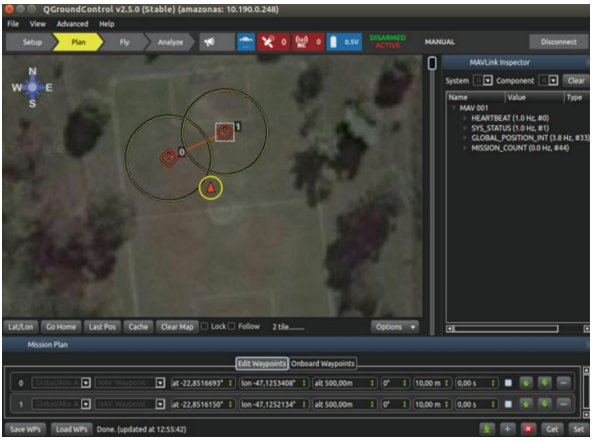

Fig. 3. Ground station main software window.

2.3 Communication System

The communication system consists of two setups. One is intended for manual piloting and uses two conventional 2.4 GHz radios-controls (Hitec Ltd). One of the radios is in charge of controlling the aircraft's actuators, while the other controls the camera's gimbal.

The second solution establishes the communication between the Ground Station and the onboard system in the aircraft. It is based on a radio link IPnDDL2450 (Microhard), with an arrangement of 8 dBi omnidirectional antennas mounted on the aircraft and, next to the Ground Station, a 17 dBi Yagi directional antenna guided by a pan-tilt tracker (Flir D48-E). The arrangement has a maximum nominal range of 15 km, with a bandwidth of 12 MBps in two channels, one for data and another one for command and control.

3 Airship Modeling and Control

The airship dynamic model is defined considering a local coordinate system as its reference, whose origin is at the center of volume of the envelope. The vector of linear and angular velocities is represented by $V = \left[u, v, w, p, q, r\right]'$, and the dynamics of the aircraft is expressed as:

$$M\acute{V} = F_d(V) + F_a(V) + F_p(V) + F_g(V) + F_w(V)$$

where M is the 6×6 mass matrix which includes the elements of real and virtual inertia, typical of floating air vehicle dynamics, and F_d, F_a, F_p, F_g and F_w are, respectively, inertial, aerodynamic, propulsive, gravitational/thrust and wind-influenced forces. The main actuators in the airship (Fig. 4) are four aerodynamic surfaces whose deflection function as rudder, elevator, aileron, and four vectorizable electric thrusters that compensate the excess weight of the aircraft at low speeds, besides improving maneuverability throughout the flight profile.

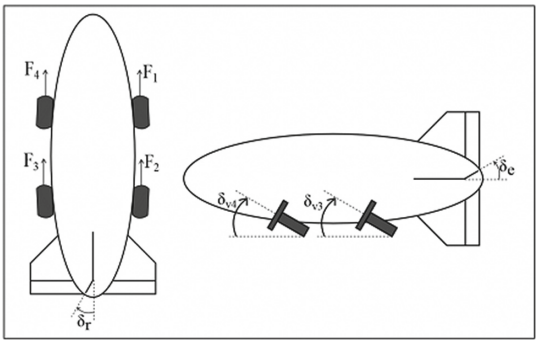

Fig. 4. Airship main actuators.

The vector of control inputs is $u = [\delta_e, \delta_a, \delta_r, \delta_1, \delta_2, \delta_3, \delta_4, \mu_1, \mu_2, \mu_3, \mu_4]'$, where δ_e, δ_a, and δ_r are the deflections of elevator, aileron and rudder, δ_i is the normalized input voltage of the *i-th* thruster and μ_i is the vectoring angle of the *i-th* propeller. To make the controller design more natural, the propeller powers were configured for joint use at forward power $\delta_f = (\delta_1 + \delta_2 + \delta_3 + \delta_4)/4$; left/right differential $\delta_{lr} = (\delta_1 + \delta_2 - \delta_3 - \delta_4)/4$, front/back differential $\delta_{fb} = (\delta_1 - \delta_2 - \delta_3 + \delta_4)/4$, and cross-differential $\delta_c = (\delta_1 - \delta_2 + \delta_3 - \delta_4)/4$. Similar configurations have been set for vectorizations.

This dynamic model of 6 degrees of freedom, evolved from [14], forms the basis of a simulator in MATLAB/Simulink [15] that allows the design and validation of airship control and navigation strategies. Among several possible approaches to control the NOAMAY airship flight, one that has been very promising is the *Incremental Non-linear Dynamic Inversion* (INDI). This automatic control technique considers that the vector of states x and its derivative $\acute{x}$ are observable and measurable at a sufficiently rapid sampling rate, when then the control law for a request for change of states is predominantly dependent on the respective non-linear input function F_u being given by:

$$u = u_0 - F_u^{\{-1\}}\left(\acute{x}_d - x_0\right)$$

in which $\acute{x}_d = K(x_d - x_0)$, where x_d is the desired state, F_u is a function of the state vector, and K is a gain matrix that stabilizes the closed-loop system. Figure 5 depicts the closed-loop block diagram of an INDI controller for any dynamic system.

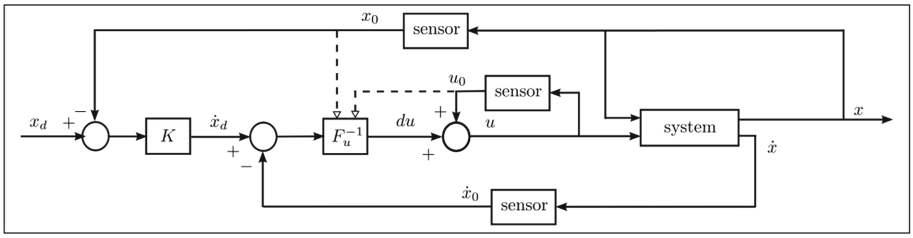

Fig. 5. Incremental Non-linear Dynamic Inversion (INDI) block diagram.

Eventually, for the particular case of a linearized model around an equilibrium condition, the input function F_u will correspond to the input matrix B of the linearized dynamics. This is the case when considering the decoupled models of the lateral and longitudinal dynamics of the NOAMAY airship, resulting from the linearization around a trimming condition, where we obtain:

$$\acute{x}_{lon} = A_{lon}x_{lon} + B_{lon}u_{lon}$$

$$\acute{x}_{lat} = A_{lat}x_{lat} + B_{lat}u_{lat}$$

and the matrices $A_{lon}, B_{lon}, A_{lat}, B_{lat}$ mainly depend on the trimming conditions (airspeed and altitude).

Longitudinal Dynamics: The state vector $x_{lon} = \left[u, w, q, \theta, h\right]$' comprises, respectively, the longitudinal and vertical velocities, rate of attack angle, attack angle, and altitude. The inputs $u_{lon} = \left[\delta_e, \delta_f, \delta_{fb}, \mu_f\right]$' are the elevator deflection, forward power, forward differential power, and synchronized vectorization (same angles).

Lateral Dynamics: The state vector $x_{lon} = \left[\beta, p, r, \varphi, \psi\right]$' comprises the lateral slip angle, roll, and yaw rates, and roll and yaw angles respectively. The inputs $u_{lat} =$

$[\delta_a, \delta_r, \delta_{lr}, \delta_c, \mu_{lr}]$' include aileron, and rudder deflections, left/right differential power, cross-differential power and front/back differential vectoring, respectively.

A detailed description of INDI control for the lateral movement of the NOAMAY airship is given in [16], where are stressed the advantages of the approach, including its simplicity of design and tuning, and its efficiency. It should be emphasized that the INDI approach dispenses the knowledge aerodynamic parameters that are state-dependent, as these parameters usually are challenging to measure or to estimate.

Aiming the automatic control of both lateral and longitudinal airship dynamics, in order to cover all the phases of a complete mission (vertical takeoff, hovering, cruise flight, hovering and vertical landing) the INDI methodology is being extended, as well as a second control technique based on gain scheduling, similar to the one developed in [1]; this latter technique is also adopted in the incorporation of a planning level for airship flight [17]. Additional approaches being considered for the NOAMAY control use the LQR gain scheduling for the longitudinal mode, combined with the sliding modes for lateral movement [18], as well as a multivariate approach to Backstepping combined with Sliding Modes.

This challenging research about an automatic control scheme for a complete airship mission is being conducted in simulation, based on the 6 DOF dynamic model described earlier. Afterward, the conceived techniques will be implemented on the airship onboard system for experimental validation on real flight situations, subject to wind disturbances, mass variations, and so on.

4 Experimental Results

To validate the airship along with its robotic infrastructure, a set of flight tests were performed. The airship was flown under remote control, whereas sensory data was acquired by the onboard system and transmitted to the ground control station. A video of the ground station, showing the airship movement and other data received from the embedded system, is available at https://youtu.be/Ld8Hsmeak2U.

Figure 6 presents a screenshot of the user interface at the ground operation station. One can also see the trajectory performed by the aircraft in one of the flights, determined by the Xsens Mti-G-700 sensor. To analyze the behavior of the sampling period of each acquired sensory data, with respect to their nominal rates, for the total acquisition period, we computed the increments $\Delta t_k = t_k - t_{\{k-1\}}$ where t_k and $t_{\{k-1\}}$ are the time intervals between two consecutive acquisitions, and also the metrics $\Delta t_{\{MED\}} = mean(\Delta t_k)$ and $\Delta t_{\{MAX\}} = max(\Delta t_k)/\Delta t_{\{MED\}}$.

As an example, consider the Xsens Mti-G-700 sensor (GPS, inertial, altimetry and fused GPS + inertial information) whose data were acquired by the "ROS node" software at a nominal rate of 100 Hz - the highest sampling rate used in the robotic infrastructure. The Fig. 6 also shows the behavior of the $\Delta t_k/\Delta t_{\{MED\}}$ ratio, revealing a uniformity in acquisition at the specified 100 Hz, without loss of information, since the $\Delta t_k/\Delta t_{\{MED\}}$ ratio reaches at most a value of 1.7, staying below 1.2 most of the time. For the other sensors, whose sampling rates are much lower, the dispersion of the $\Delta t_k/\Delta t_{\{MED\}}$ around the unit was even lower.

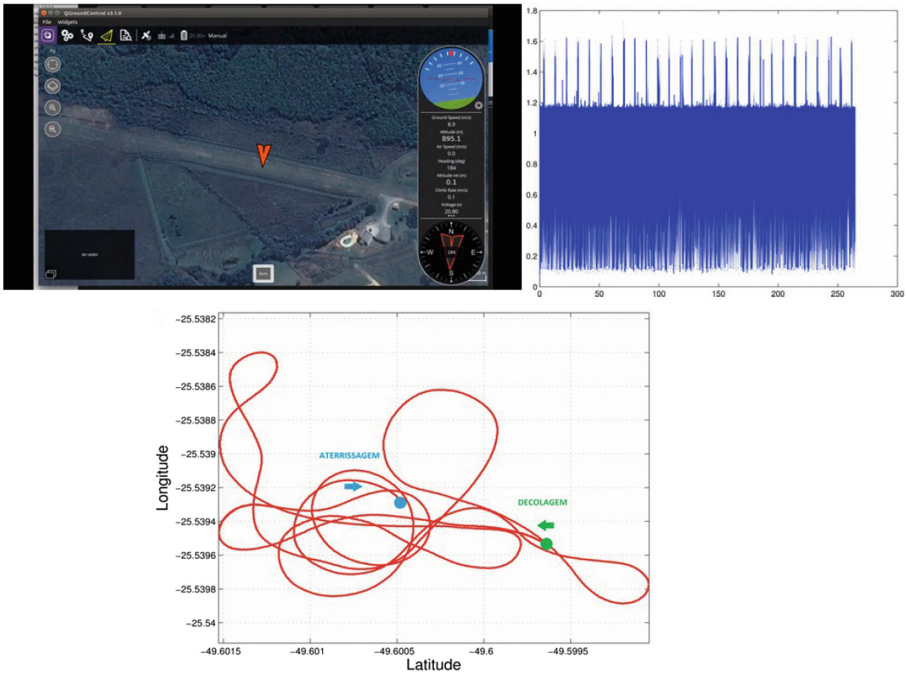

Fig. 6. Top-left: Screenshot of the Ground Station, with the airship current position marked over a map. Attitude, speed, compass and other variables are displayed on a panel on the right side of the map. Top-right: dispersion of the $\Delta t_k / \Delta t_{\{MED\}}$ ratio as measured by the Xsens sensor. Bottom: trajectory (red) of the airship, with the takeoff (green) and landing (blue) dots.

5 Conclusions

Unmanned airships are interesting platforms; both for its potential applications and for the research challenges they pose in control and other areas of robotics. This paper described the robotic infrastructure for a lighter than air aircraft, equipped with four vectoring electric propellers and aerodynamic surfaces, and presented a control architecture based on Nonlinear Incremental Inverse Dynamics.

The experimental flight was encouraging. In addition to proving the validity of the project as a whole, the vehicle presented maneuverability beyond expectations.

The work continuity includes aircraft optimization, with a general review of the project aiming at weight reduction, implementation, and experimental validation of control strategies, complementary to the existing ones. Moreover, there will have additional tests, to exercise the vehicle in flight conditions closer to the field operation, including multiple transitions between hovering and aerodynamic flight.

Acknowledgments. This work is sponsored by DRONI (CNPQ 402112/13-0), INCT (CNPQ 465755/14-3, FAPESP 2014/50851-0), Brasil; FCT (LAETA UID/EMS/50022/2013), Portugal.

References

1. Elfes, A., Bueno, S.S., Bergerman, M., Ramos, J.G.: A semi-autonomous robotic airship for environmental monitoring missions. In: Proceedings of 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998 (1998)
2. Wimmer, D.-A., Bildstein, M., Well, K.: Research airship 'lotte' make: development and operation controllers for autonomous flight phases. In: IEEE International Conference on Intelligent Robots and Systems, pp. 55–68, Lausanne, Switzerland, October 2002 (2002)
3. Hygounenc, E., Soueres, P., Lacroix, S., Jung, I.-K.: The autonomous blimp project at LAAS/CNRS: achievements in flight control and terrain mapping. Int. J. Robot. Res. **23**(4), 473–511 (2004)
4. Lee, S.-J., Kim, S.-P., Kim, T.-S., Kim, H.-K., Lee, H.-C.: Development of autonomous flight control system for 50 m unmanned airship. In: Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, December 2004 (2004)
5. Elfes, A., Montgomery, J.F., Hall, J.L., Joshi, S.S., Payne, J., Bergh, C.F.: Autonomous flight control for a titan exploration Aerobot. In: 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS 2005, Munich, Germany (2005)
6. Moutinho, A., Mirisola, L., Azinheira, J., Dias, J.: Project diva: guidance and vision surveillance techniques for an autonomous airship. In: Robotics Research Trends, pp. 77–120. Nova Science Publishers (2008)
7. Zheng, Z., Huo, W., Wu, Z.: Autonomous airship path following control: theory and experiments. Control Eng. Pract. **21**(6), 769–788 (2013)
8. Earon, E., Rabbath, C., Apkarian, J.: Design and control of a novel hybrid vehicle concept. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, South Carolina, USA, August 2007 (2007)
9. Carvalho, J., Bueno, S., Modesto, J.: Sistemas aéreos não-tripulados para o monitoramento e gestão de risco do bioma amazônico. Revista Computacão Brasil, March 2014 (2014)
10. OSRF: ROS - Robot Operating System (2009). www.ros.org. Accessed 29 Jan 2018
11. Meier, L., Tanskanen, P., Heng, L., Lee, G.H., Fraundorfer, F., Pollefeys, M.: The Pixhawk open-source computer vision framework for MAVs (2011). www.pixhawk.org. Accessed 29 Jan 2018
12. Meier, L., Tanskanen, P., Heng, L.: Qgroundcontrol: ground control station for small air-land-water autonomous unmanned systems (2010). www.qgroundcontrol.org. Accessed 29 Jan 2018
13. Azinheira, J.R., de Paiva, E.C., Bueno, S.S.: Influence of wind speed on airship dynamics. J. Guid. Control Dyn. **25**(6), 1116–1124 (2002)
14. Arias, R.: Modelagem de um dirigível robótico com propulsão elétrica de quatro motores. Master's thesis, Universidade Estadual de Campinas, SP, Brasil (2014)
15. Azinheira, J., Moutinho, A., Carvalho, J.: Lateral control of airship with uncertain dynamics using incremental nonlinear dynamics inversion. In: 11th IFAC Symposium on Robot Control SYROCO, Salvador - BA, Brazil, August 2015 (2015)
16. Carvalho, J., Moutinho, A., Azinheira, J.R.: Integrating mission planner to the flight control system of a robotic airship, XXI Congresso Brasileiro de Automática - CBA (2016)
17. Vieira, H., Azinheira, J.R., Moutinho, A., de Paiva, E.C.: Controladores não lineares para um dirigível robótico de propulsão quádrupla. XXI Congresso Brasileiro de Automática – CBA (2016)