INTEGRATING MISSION PLANNER TO THE FLIGHT CONTROL SYSTEM OF A ROBOTIC AIRSHIP

José Reginaldo Hughes Carvalho*, Alexandra Moutinho†, José Raul Azinheira†

*Institute of Computing, Federal University of Amazonas
Manaus-AM, Brazil

†IDMEC, LAETA, Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal

Emails: reginaldo@icomp.ufam.edu.br, alexandra.moutinho@tecnico.ulisboa.pt, jose.raul.azinheira@tecnico.ulisboa.pt

Abstract— This work proposes a mapping from flight primitives to the UAV low level controller that incorporates the various flight actions of a robotics airship. As most VTOL (vertical take-off and landing) aircraft, the airship can hover, move vertically and sustain a cruiser flight. The goal is to integrate flight plan related tasks into the low-level software, thus the flight plan can be executed without the assistance of a robotics middleware (e.g. ROS). The main motivation of this integration is to encapsulate mission critical activities in such a way that the middleware focuses on planing, mission supervision and high-level activities like fault detection, and mission re-planning. To validate the approach the authors prepared a surveillance mission in a simulation tool specifically designed for airships. This work is in the context of the development of a robotics airship for scientific application and environmental monitoring of the Amazon rain forest.

Keywords— Aerial Unmanned Vehicles, Aerial Navigation, Lateral Control.

Resumo— Este trabalho propõe um mapeamento de primitivas de voo para o controlador de baixo nível de um VANT que incorpore as diversas ações de voo de um dirigível robótico. Como a maioria das aeronaves VTOL (do inglês, vertical take-off and landing), o dirigível pode pairar, mover-se verticalmente e sustentar um voo de cruzeiro. O objetivo é o de integrar as tarefas relacionadas com o plano de voo ao software de baixo nível, de forma que possa ser executado sem a assistência de um middleware robótico (e.g. ROS). A principal motivação desta integração é encapsular as atividades de missão crítica de tal forma que o middleware concentre-se no planejamento, supervisão da missão e atividades de alto nível, como a deteção de falhas, e o re-planejamento da missão. Para validar a abordagem, os autores prepararam uma missão de vigilância em uma ferramenta de simulação projetada especificamente para dirigíveis. Este trabalho está no contexto do desenvolvimento de um dirigível robótico para aplicação científica e monitoramento ambiental da floresta amazônica.

Palavras-chave— Veículos Aéreos Não-tripulados, Navegação Aérea, Controle Lateral.

1 Introduction

This work addresses the design of a mission planner and its integration with the low level control of DRONI, the project code-name of a 13m long autonomous airship powered by four vectored DC brushless electrical motors. This new and under-development blimp is an evolution of the AURORA airship (Elfes et al., 2001).

Experiments with robotic airships for environmental monitoring have been presented in the last years. In (Azinheira et al., 2001; Elfes et al., 2002; Gerke et al., 2012) one can find results of airships aimed to overfly the rain forest. Interesting applications include scientific research, environmental monitoring, inspection, surveillance, risk mitigation of natural disasters, among many others. However, actual experiments with medium to long endurance airships in the skies of the Amazon are still to come. One of the unsolved issues is how to safely perform a complex set of tasks at the forest operation conditions.

Although considerable improvements of Unmanned Aerial Vehicles (UAVs) have been made in the military arena, the cost of a fine-tuned autonomous platform is prohibitive for most civil applications, despite that both scenarios share most challenges and issues.

Researchers typically use a robotics specific middleware (e.g. ROS - Robot Operating System at www.ros.org) to establish an abstraction layer between high level tasks and the vehicle specifics command and control. Such middlewares are present in many applications, and their reliability has been proved in the field. However, there is no formal verification of the reliability limits of a robotics middleware (from the critical mission perspective) (Pordel and Hellström, 2013). A possible step to define a clearer boundary between high and low-level critical actions is to encapsulate the flight plan together with low-level control. By doing so, the robotics middleware deals with high level supervision activities, like flight mode transitions, data acquisition from sensors, fault detection, among many others.

This paper discusses this idea, based on an environmental surveillance case study, and combining recent results of natural landmark detection and tracking over the Amazon forest. A method to match real-time video with previously computed feature-maps is used in (Pinage et al., 2013) to track a sequence of pre-defined images.

The main contribution of this paper is a scheme to integrate the mission execution module with the low-level controller. The paper discusses the high-level tasks break-down, and how to map them into flight primitives. The approach is evaluated in a 600 hours wind-tunnel based airship computer model (Gomes and Ramos, 1998; de Paiva et al., 2006). The results show a considerable degree of autonomy in sequencing different flight mode primitives required to execute a mission to visit and inspect a set of landmarks.

The remainder of this paper is organized as follows: Section 2 presents the problem context. The fundamental techniques are explained in Sections 3 and 4. In Section 5, these techniques are combined to implement a surveillance mission based on detecting and visiting landmarks. Section 6 presents the simulation results and Section 7 closes the paper with some final remarks.

2 The Context: Rain Forest Surveillance

2.1 Operation Scenario: The Amazon Forest

This work is part of an initiative to apply UAVs as a tool for sustainable conservation of the Amazon. Although satellites provide information to build statistic models (e.g. www.obt.inpe.br/eng/), the large latency (time from acquisition and analysis) and granularity (area per measured unit) prevent counter-actions to mitigate damage to the ecosystem in its 5.5 million km^2 of forest (India, the 7th largest country, has approximately 3.3 km^2).

Data acquired from sensors overflying the forest have finer granularity and lower latency. However, the rain forest is deep, dense, with high temperature and humidity rates, being a challenging operation environment. Radio link coverage is strongly dumped under the canopy (Assis, 2012), and an external antenna has to surpass the 20-30 meters average height of the canopy. There are no road connection, and finding a firm terrain might be difficult during the flooded season. Ground personnel has to cope with the discomfort of the weather, the diversity of fauna (e.g. insects) and endemic diseases (e.g. yellow fever and malaria).

2.2 Robotics Airship and Surveillance Missions

Figure 1 presents the different flight phases of an UAV, from take-off to landing, and the correlation to its autonomous capability to perform mission tasks (Elfes et al., 2001). Usually, a robotics middleware bridges the interface among the different phases, creating an interesting level of abstraction between decision-driven and reactive-driven aspects. Given the operation risks of an UAV in the rain forest, and given that, despite its broad use, there is no formal proof of the reliability limits of a robotics middleware, this paper proposes the

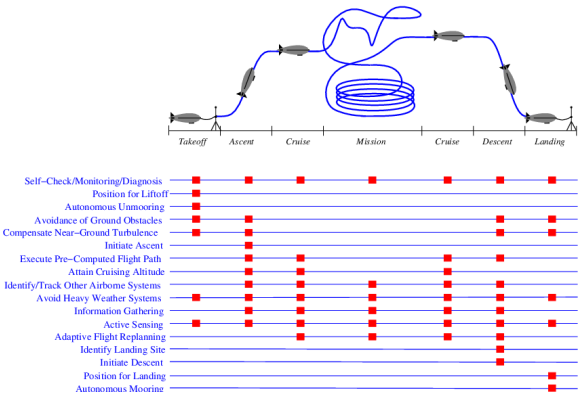

Figure 1: Flight phases for a substantially autonomous UAV.

integration of the flight plan tasks and the low-level controller. Figure 2 summarizes the main phases of the proposed integration, composed of:

1. Mission planner, to decompose the mission requirements into a set of flight primitives;
2. Mission mapper, which uses a set of directly related maneuvers to translate the flight primitives into vehicle input references;
3. Vehicle control system, to ensure the input references are followed by regulating the respective tracking error.

3 MISSION PLANNER

The mission planner is a supervised off-line activity that selects locations (defined as image landmarks) to be visited. Due to the extension of the forest, the resulting imagery database demands a searching space reduction. In this work, a multiband wavelet operator, used to decrease the image resolution and to smooth the canopy pattern (highlighting anything else), is combined with the SIFT algorithm, used to detect and match landmark key-points. The procedure used during the off-line detection is also applied during the on-the-fly matching (Pinage et al., 2013).

3.1 Off-line Planner: Landmark Identification

Figures 3 and 4 present two segments of up to 1.000 images acquired by a manned aircraft part of

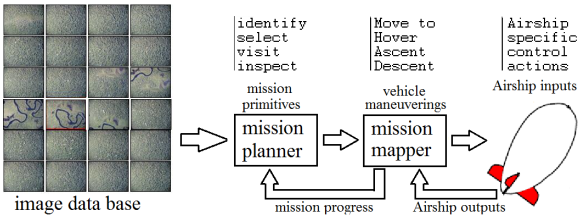

Figure 2: Direct mapping overview.

the Brazilian GEOMA project (www.lncc.org.br). The first sequence is a typical landmark for further assessment due to the evidences of human made deforestation. The second sequence is also of interest, as curvy river branches and small lakes usually have high concentration of biodiversity.

The segments in Figures 3 and 4 are the output of the Landmark Candidate Identification algorithm (Pinage et al., 2013) that reduced hundreds of images to a few segments. The resulting segments are subjected to a semantic classification of vegetation areas, water and anything else (Cavalcanti et al., 2015). The Landmark Candidate Identification steps follows:

1. Wavelet transform: smooth canopy pattern and reduce image scale (the LL Sub-band);
2. Apply SIFT algorithm to the image sequence, extracting a set of key-points for each image;
3. Whenever the number of key-points increases abruptly, label the frame as `landmark_candidate_start`;
4. When the number of key-points goes down to the steady state value, label the frame as `landmark_candidate_end`;
5. Classify the images into vegetation (grey), water (black) and neither (white).

Figure 5 presents the number of key-points detected in segments of figures 3 and 4, and the corresponding peak frame. Figure 6 presents typical outputs of Steps 1 and 5 (left), and examples of images with and without landmarks (right).

Finally, a human specialist decides what landmarks will be visited, including hover time and altitude. Although some automatic detection may be developed, the conservationists taking part of this project are more comfortable to supervise the selection. For them, it is already a relevant benefit to work over a reduced database.

The mission planner sends to the next block (mission mapper) a data structure E_r, with the fields (one list entry per landmark): 1.Landmark geodesic coordinates and cruise altitude: L_N, L_E, L_h; 2.Landmark feature description: `Landmark`; 3.Airship sensors information: `Sensor`.

4 Mission Mapper

The mission mapper processes E_r based on the vehicle maneuvering primitives, and outputs the references to the vehicle control system, closing the end-to-end integration. The call format is:

$$[N, E, u, \psi, h, t] = \\ = MMapper(L_N, L_E, L_h, Landmark, Sensors)$$

with the following mandatory methods:

$$\begin{cases} (N, E) & \leftarrow & Geo2Cart(L_N, L_E) \\ (h, t) & \leftarrow & Sense(landmark, sensors) \\ (u, \psi) & \leftarrow & Move(N_i, E_i, N_f, E_f, h) \end{cases}$$

where $Geo2Cart(\cdot)$, basically a vehicle router, converts the geodesic coordinates (L_N, L_E) to the relative North-East distance from the take-off position $(0, 0)$. $Sense(\cdot)$, a supervised method, has as output the hover altitude and time, which depends on the landmark and sensor used for inspection (e.g. camera model). $Move(\cdot)$ computes cruise reference velocity u and heading angle ψ. If (N_i, E_i) and (N_f, E_f) are too close, the airship will align with the wind and hover ($u = 0$). Variations in h mean vertical displacements. The mission mapper output is uploaded to the onboard system as a matrix.

5 Integration of Mission Tasks and Flight Control

The integration of mission mapper with the low-level control system is based on the gain-scheduling approach (Moutinho, 2007). Consider the dynamics linearization around a trim condition, and the decoupled airship longitudinal and lateral dynamics (for the no-wind case) given by:

$$\dot{x}_{lon} = A_{lon} x_{lon} + B_{lon} u_{lon} \quad (1)$$

$$\dot{x}_{lat} = A_{lat} x_{lat} + B_{lat} u_{lat} \quad (2)$$

where all variables correspond to the variation relative to the trim condition.

Longitudinal dynamics: $x_{lon} = [u, w, q, \theta, h]^\mathsf{T}$ includes airship longitudinal and vertical velocities, pitch rate, pitch angle, and airship altitude. Inputs $u_{lon} = [\delta_e, \delta_0, \delta_2, \mu_0]^\mathsf{T}$ are elevator deflection, forward thrust, differential front-rear thrust, and vectoring angle.

Lateral dynamics: $x_{lat} = [\beta, p, r, \phi, \psi]^\mathsf{T}$ comprehends sideslip angle, roll and yaw rates, and roll and yaw angles. Inputs $u_{lat} = [\delta_a, \delta_r, \delta_1, \mu_2]^\mathsf{T}$ includes aileron and rudder deflections, differential left-right thrust power, differential front-rear motor vectoring angle.

Matrices A_{lon}, B_{lon}, A_{lat} and B_{lat} depend on airspeed and altitude trim conditions, computed at steps of 0.1m/s. The resulting matrices are stored in a lookup table indexed by mission demanded airspeed and altitude.

Figure 7 shows the control system block diagram. The gain-scheduling as in (Moutinho, 2007) indexes the operation conditions stored in the lookup table, and adapts the LQR gain matrix used to compute the control signals to be sent to the actuators. The block labeled "mission mapper" is responsible for the transition from the mission planner output E_r to the references x_r. Every planned task at a given landmark is represented line-wise in the x_r state reference matrix.

Figure 3: A frame sequence of what seems to be man made clearances. Source: GEOMA project.

Figure 4: Typical locations with concentration of biodiversity. Source: GEOMA project.

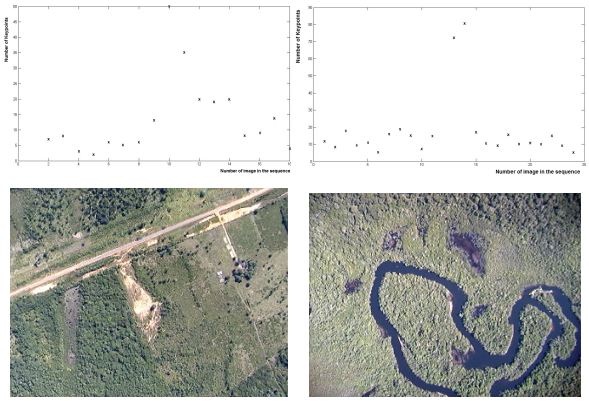

Figure 5: Key-points progress and landmark candidate of sequence in Figures 3 (left) and 4 (right).

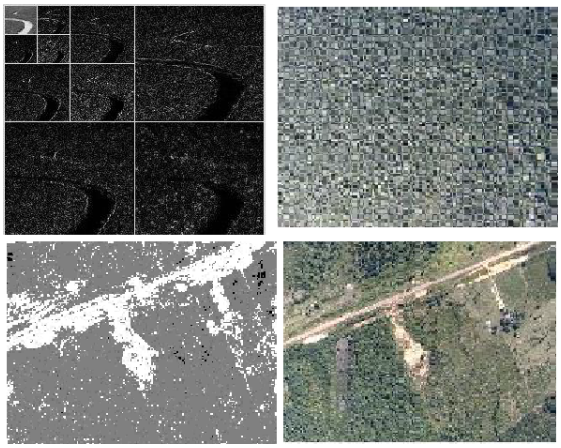

Figure 6: From top-left, clockwise: wavelets sub-bands; a typical canopy pattern with no feature in the LL sub-band; a typical landmark candidate, landmark and the three classes found by Naive Bayes classifier.

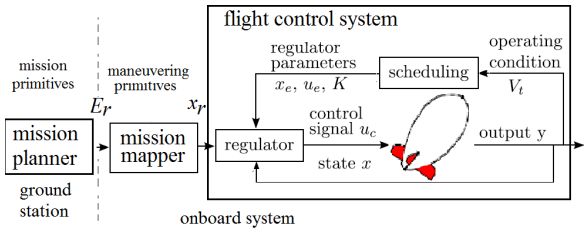

Figure 7: Framework block diagram.

6 Following Maneuvering Primitives

In this section, the gain-scheduling controller successfully controls the airship during the sequence of maneuvers. The closed-loop low-level algorithm is implemented using a MATLAB®/Simulink®-based simulator of the airship (Gomes and Ramos, 1998; de Paiva et al., 2006; Moutinho, 2007). This nonlinear model is based on parameters extracted for an airship airframe in 600 hrs wind-tunnel tests. It includes wind/turbulence disturbances, and propellers and battery models, being the most important computed aided tool of this project. It is the most precise airship simulator to the best of the authors knowledge.

The case-study mission consists of overflying two specific locations in the limits a natural reserve, and returning to the same launching position (see figure 8). Maneuvering primitives are: take-off and hover for 120s; move to location 1 at same altitude; hover for 120s; descend 20m and hover again for 120s; move to location 2 maintaining altitude; hover for 120s; ascend 20m; hover again for 120s; move back to initial position at same altitude; hover for 120s and land. Landmark 1 is at coordinates (-1296, 1537)m, and landmark 2 at (5925, 7881)m relative to the (North, East) frame, with origin at the take-off location.

The mission maneuvering matrix (M) for the three way-points including take-off and landing is:

$$\mathrm{M} = \begin{bmatrix} 0 & 0 & -5 & 1 & 0 \\ 0 & 0 & -120 & 8 & 0 \\ -1200 & 1423 & -120 & 4 & 0 \\ -1296 & 1537 & -120 & 0 & 120 \\ -1296 & 1537 & -120 & 0.5 & 0 \\ -1296 & 1537 & -100 & 8 & 0 \\ 5875 & 7837 & -100 & 4 & 0 \\ 5925 & 7881 & -100 & 0 & 120 \\ 5925 & 7881 & -100 & 1 & 0 \\ 5925 & 7881 & -120 & 8 & 0 \\ 100 & 133 & -120 & 4 & 0 \\ 0 & 0 & -120 & 0.5 & 0 \\ 0 & 0 & -5 & 0 & 120 \end{bmatrix}$$

where each $(N_i, E_i, h_i, V_t, t_{\text{hover}})$ line are position references, given by the i-landmark coordinates,

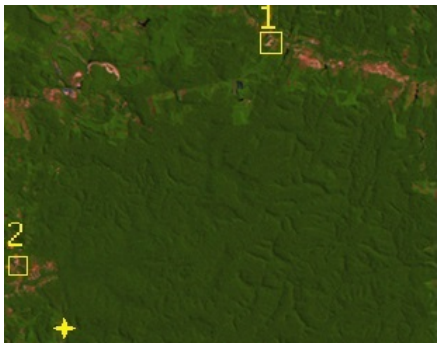

Figure 8: Google Earth image of the reserve with landmarks 1, 2 and launching (+) locations.

the speed reference (air or ascend/descend) V_t to move to the *i+1*-landmark, and the hover duration in case $V_t = 0$. As the airship has a substantial virtual mass, an intermediate point is added to reduce the airspeed to half before the target location. This is necessary to force the reduction of the approaching speed to a value that makes hover feasible. A speed profile could also be considered. However, the intermediate point is easier to implement, demands lower memory overhead and it works well.

The simulation was run with a north wind of 4m/s. Figure 9 shows the 2D view of the executed trajectory (the zoomed windows show the influence of the wind). Note the perfect alignment with the wind when hovering. Figure 10 presents the 3D view of the trajectory. The gain-scheduling controller ensured the execution of the sequence of tasks from take-off to landing.

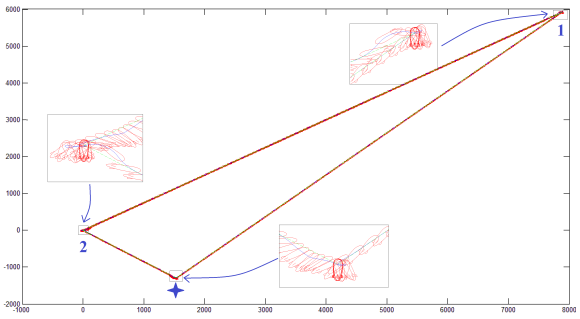

Figure 9: The 2D view (North-East) executed trajectory. Small windows present the airship alignment with the wind during hover/ascent/descent.

Figure 11 shows the altitude, groundspeed and trajectory errors along the simulation time. Note the peaks due to the abrupt change of reference when switching tasks (e.g. hover, move). Those peaks are well dumped by the controller.

The simulations confirmed that the low-level controller successfully monitored and executed each mission primitive along the task sequence. Furthermore, this approach promotes the dynamic

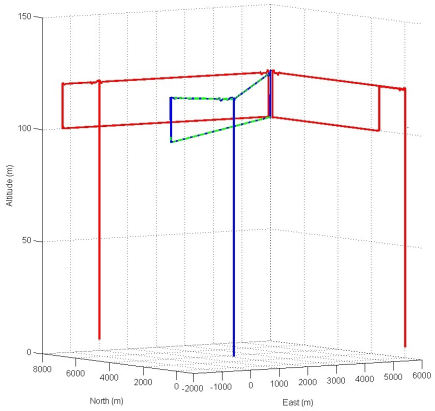

Figure 10: The 3D view. Blue: executed trajectory. Red: Its projections in xz and yz planes.

re-planing of the sequence of primitives by inserting or deleting rows in the reference matrix. The robotics middleware keeps playing its part in the supervision of the mission execution, and facilitating the inclusion of support tasks, like fault detection, safety procedures, and re-routing due to weather incidents, among others (see Figure 1).

7 Conclusion

This paper proposes a scheme to integrate the mission tasks directly into the low-level controller of an autonomous airship performing surveillance missions. The operation scenario is the Amazon rain forest. An end-to-end surveillance example is used to validate the proposal, consisting of overflying and hovering landmarks.

The objective is to implement a low-level control system capable to execute the flight plan without the assistance of a companion software (e.g. robotic middleware). The tasks primitives are intrinsically mapped into the control reference variables. The resulting closed-loop system is flexible in terms of maneuvering capability (when compared with fixed-wing aircrafts), and, in principle, it is also applicable to any VTOL UAV.

The simulated airship is able to complete the mission following the sequence of locations and performing different actions, like hover, and vertical displacements, under the coordination of its low-level controller. Simulations are sufficient to validate the approach, as they capture all aspects related to the mission planning and mapping to the low-level gain scheduling control system, which is the scope of this paper.

It is important to remark that a robotic middleware is an important component of the DRONI's platform. The purpose of our approach is to reduce its influence on the real-time aspects, shifting to the low-level control the role to monitor and correct the execution of the planned sequence of the flight. The robotic platform will still be

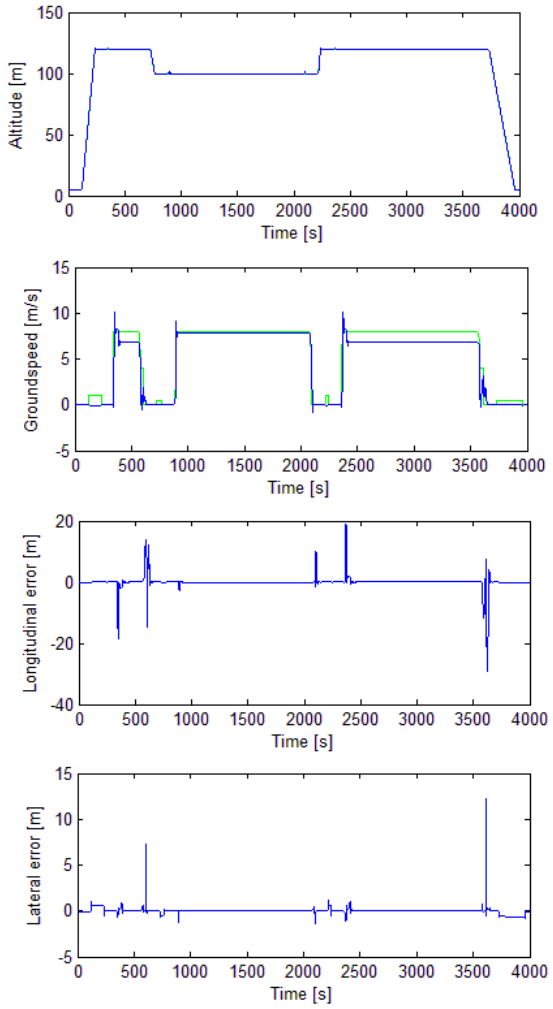

Figure 11: The simulated mission. Take-off, move to point 1, hover, descend, move to point 2, hover, ascend, move to take-off location.

there to monitor and take decisions on other important parts of the system, such as to incorporate support tasks and mission execution supervision.

This work is part of the development of an autonomous airship for environmental monitoring, surveillance and scientific research missions.

Acknowledgments

This work was supported by Brazilian National Council for Scientic and Technological Development (CNPq), process 402112/2013-0, and the Foundation for Research Support of the State of Amazonas, (FAPEAM), processes 087/2014 and 114/2014, Brazil; Fundação para a Ciência e a Tecnologia (FCT), through IDMEC, under LAETA UID/EMS/50022/2013, Portugal.

References

Assis, M. S. (2012). Radio wave propagation in the amazon jungle: A tutorial, *Revista de Tecnologia de Informação e Comunicação* **2**(1): 37–4.

Azinheira, J. et al. (2001). Lateral/directional control for an autonomous, unmanned airship, *Aircraft Engineering and Aerospace Technology* **73**(5): 453–459.

Cavalcanti, L. C., Gatto, B., Carvalho, J. R. H. and dos Santos, E. M. (2015). A comparison on supervised machine learning classification techniques for semantic segmentation of aerial images of rain forest regions, *(VISAPP) 10th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*.

de Paiva, E. et al. (2006). Project aurora: Infrastructure and flight control experiments for a robotic airship, *Journal of Field Robotics* **23**(3-4): 201–222.

Elfes, A., Carvalho, J. R. H., Bergerman, M. and Bueno, S. S. (2001). Toward a perception and sensor fusion architecture for a robotic airship, Vol. 4571, pp. 65–75.

Elfes, A. et al. (2002). *Sensor Based Intelligent Robots*, Vol. 2238 of *Lecture Notes in Computer Science*, Springer, chapter Modelling, Control and Perception for an Autonomous Robotic Airship, pp. 216–244.

Gerke, M. et al. (2012). *Future Security*, Vol. 318 of *Communications in Computer and Information Science*, Springer, chapter Lighter-Than-Air UAVs for Surveillance and Environmental Monitoring, pp. 480–483.

Gomes, S. and Ramos, J. (1998). Airship dynamic modeling for autonomous operation, *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, Vol. 4, pp. 3462–3467 vol.4.

Moutinho, A. (2007). *Modeling and Nonlinear Control for Airship Autonomous Flight*, PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisbon.

Pinage, F., Hughes Carvalho, J., Viana Freitas, E. and Pinheiro de Queiroz Neto, J. (2013). Feature transform technique for combining landmark detection and tracking of visual information of large rain forest areas, *Robotics Symposium and Competition (LARS/LARC), 2013 Latin American*, pp. 30–37.

Pordel, M. and Hellström, T. (2013). Robotics architecture frameworks, available tools and further requirements, *Technical Report 13.02*, Umeøa University, Department of Computing Science.