

Line Following and Ground Vehicle Tracking by an Autonomous Aerial Blimp

David Jerónimo¹, Ricardo Alcácer¹, F. C. Alegria², Pedro U. Lima³

Abstract – In this paper we introduce an autonomous aerial blimp testbed. The robot has onboard vision and computation capabilities, based on a digital signal processor. We also present the realistic hardware-in-the-loop simulator developed over USARSim. This development environment enabled fast prototyping and implementation of navigation primitives for the blimp, namely vision-based line following and ground vehicle tracking. Results of the indoor operation of the real blimp are presented.

Keywords: Aerial blimp; autonomous navigation; vision-based path following; vision-based vehicle tracking.

I. INTRODUCTION

Aerial blimps, together with fixed wing airplanes, helicopters and quad-copters, have been among the most popular unmanned aerial vehicles (UAVs) used in research in recent years. They have, over the other types of UAVs, the advantage of intrinsic stability and safeness, low noise, low vibration, vertical take-off and landing with hovering capabilities for a higher payload-to-weight-and-energy consumption ratio [5]. On the other hand, their dynamics is hard to identify, and non-conventional methods may have to be used to determine unknown parameters [3]. Previous work on blimps has used single camera vision to fly around targets in indoor environments [1], to emulate (indoors) underwater station keeping and docking operations [6], to track simple objects [4] or for outdoor environmental monitoring [5]. Stereovision was used in [2] for outdoor terrain mapping. Indoor solutions often use non-fully autonomous blimps, as the control algorithms run on ground stations that communicate through radio-frequency (RF) signals with the blimp motors and cameras. On the other hand, larger outdoor blimps have enough payload to carry on board more sensors, such as GPS, gyroscopes or wind speed meters.

In this paper, we introduce an indoor autonomous aerial blimp with onboard vision and computation capabilities, based on a digital signal processor (DSP), shown in Fig. 1. We also present the realistic hardware-in-the-loop simulator developed over the USARSim simulator. This development

environment enabled fast prototyping and implementation of navigation primitives for the blimp, namely vision-based line following and ground vehicle tracking in real outdoor scenarios. Results of the indoor operation of the real blimp are presented.

The work is part of a long-term project of the Institute for Systems and Robotics (ISR) at Instituto Superior Técnico, in the area of cooperative navigation of rescue robots [7]. This project aims at endowing a team of outdoor (ground and aerial) robots with cooperative navigation capabilities, so as to demonstrate the ability of the robots to act individually and cooperatively in search and rescue-like operation scenarios. This project intends to integrate a number of autonomous agents working in formation capable of interacting and co-operating between each other in a disaster situation such as an earthquake, where conditions are too adverse or difficult for human intervention and a rapid intervention of rescue teams is essential so as to prevent or minimize casualties. The blimp's mission is to survey the land while mobile robots on the ground move in, keeping permanent contact with the blimp and obtaining information about the ground and other matters, thus serving as an information transmission relay between the land robots and the base station.

This paper is focused on the design, development and implementation of all the blimp electronics, sensing and control systems that enable its full autonomy. The linearized blimp dynamics was identified using measurements made with the real robot flying in an indoor sports pavilion, and control algorithms were designed to follow ground lines and to track a ground vehicle.



Fig. 1. The Passarola blimp.

The authors' affiliations are:

¹ Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, davidjeronimo@gmail.com

² Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, ricardoalcacer@gmail.com

³ Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal, falegria@lx.it.pt

⁴ Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal, pal@isr.ist.utl.pt

The control algorithms were implemented in the onboard DSP and the whole system was calibrated using both

simulation tests and the real blimp. Results with real robot indoor experiments are presented.

II. BLIMP'S HARDWARE

We have designed the blimp's navigation system, by adequately integrating several off-the-shelf components:

- 1 non-rigid envelope filled with helium;
- 2 dual blade main propellers (279x120 mm);
- 1 dual blade tail propeller (180x80mm);
- 1 shaft for adjustment of the angle of the main propellers;
- 1 Analog Devices ADSP-BF561 Blackfin evaluation board;
- 1 Sony HQ1 Helmet Camera;
- 2 Graupner Speed 400 motors for the main propellers;
- 1 Multiplex Permax 480 motor for the tail propeller;
- One motor and encoder for the main propellers' shaft;
- PWM controllers for the motors;
- 1 Reedy Black Label 2, Ni-MH battery (7.2 V, 3700 mAh);
- 1 Reedy Black Label 2, Ni-MH battery (7.2 V, 3300 mAh);
- 2 Flight Power EVO 25, Li-Po batteries (11.1 V, 1500 mAh).

In Fig. 2 we show how the different electronic components are connected.

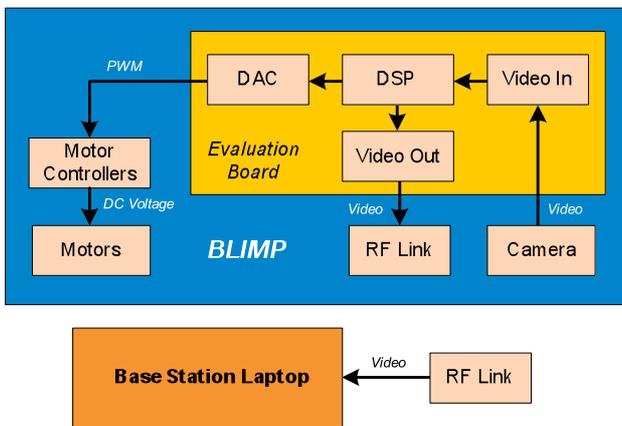


Fig. 2. Connection diagram of the hardware inside the blimp.

The DSP is used to implement the image processing algorithms, which determine the relative position of the blimp regarding the target (line or vehicle on the ground). It also implements the controller and generates the PWM signals, which are sent to the propeller motors to control their rotational speed. Although the blimp has the possibility to adjust the angle of attack of the main propellers in order to adjust the blimps height, this was not implemented in the current control loop.

All the electronics of the blimp are housed in two canopies attached to the underside of the envelope as seen in

Fig. 3. The video camera can be seen between the two canopies. In the same figure some cables can be seen coming out of the rear canopy and which go to the ground. They are used to load the software, developed in C using Analog Devices' VisualDSP++ programming environment, into the DSP, and to charge the batteries. During normal operation of the blimp they are removed. An RF link is used to transmit an image from the blimp to ground control in order to monitor the blimp's operation.



Fig. 3. Underside of the blimp showing the two canopies which contain the hardware and the main propellers.

In order to reduce the development time, an evaluation board for the DSP was used. This board has, besides the DSP, video coders and decoders and the corresponding connectors for video input and output (used for debug purposes), as well as digital-to-analogue converters for audio output. These audio outputs are used to drive the motor controllers. A proper pulse width modulation signal is created in software with the appropriate duty cycle to set the rotational speed of the motors (Fig. 4).

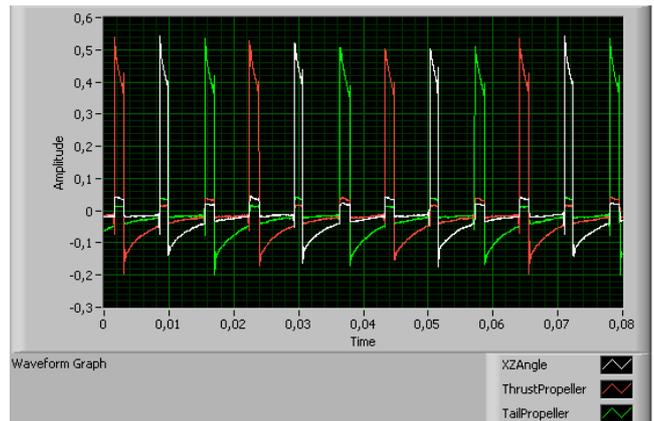


Fig. 4. Motor control signals acquired by the ADC and represented in LabVIEW used in the hardware-in-the-loop simulation setup.

III. SIMULATION SETUP

In the development stage, a setup was built which allows the closed loop control software to be tested with hardware in the loop. This setup, whose diagram can be seen in Fig. 5,

consists of two personal computers (PCs): the Development PC and the Simulation PC. The former, running VisualDSP++, is used for software coding and debugging. The latter, running USARSim [8], is used to create the virtual world in which a virtual blimp would exist. USARSim is a high-fidelity simulation of robots and environments based on the Unreal Tournament game engine. A dynamically accurate model of the Passarola blimp was created using vehicle classes from the Karma Physics Engine, which is a rigid multi-body dynamics simulator that is part of the Unreal development environment.

In order to ease as much as possible the transition from software development to full system deployment, a hardware-in-the-loop type of setup is used in which the blimp DSP is inserted in a loop together with the USARSim simulator (see Fig. 5).

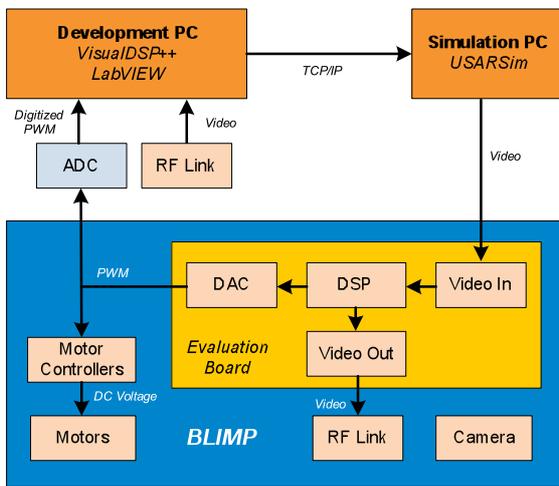


Fig. 5. Connection diagram of the simulation setup.

The image of the virtual world (Fig. 6), from the view point of the blimp's video camera is fed to the DSP and the control signals produced by the DSP (audio outputs of the DSP evaluation board) are digitized, using a National Instruments USB-9233 data acquisition board, and transferred to the development PC using LabVIEW. This data, in turn, is sent to the Simulation PC in order to control the speed of the virtual blimp propellers.

The testbed with the simulation setup is depicted in Fig. 7. The DSP signals for motor control were also connected to the real blimp motors in the laboratory to assert that they operated correctly.

IV. MODELING AND VISION-BASED CONTROL

For vehicles with 6 degrees of freedom (DoF), 6 independent coordinates are needed to determine their position and orientation. The first 3 coordinates \mathbf{v} and their temporal derivatives corresponds to the position and translational velocity of the vehicle, along the x , y and z axes, while the last 3 coordinates $\boldsymbol{\eta}$ and their temporal derivatives are used to describe its orientation and its rotational velocity.



Fig. 6. Image of the virtual world from the point of view of the blimp's camera.



Fig. 7. Picture showing the hardware-in-the-loop simulation setup.

The PASSAROLA blimp is under-actuated, i.e., it has less control inputs than DoF, so the vehicle control is limited.

A. DYNAMIC MODEL

The equations of movement of vehicles that move immersed in a fluid, can be written in a vector form [10]:

$$\begin{aligned}\dot{\boldsymbol{\eta}} &= J(\boldsymbol{\eta})\mathbf{v} \\ M\dot{\mathbf{v}} + C(\mathbf{v})\mathbf{v} + D(\mathbf{v})\mathbf{v} + \mathbf{g}(\boldsymbol{\eta}) &= \boldsymbol{\tau}\end{aligned}$$

where J is the Jacobian matrix, M is the system inertia matrix, C is the Coriolis-centripetal matrix, D is the damping matrix, \mathbf{g} is the vector of gravitational/buoyancy forces and moments and $\boldsymbol{\tau}$ is the vector of applied torques and moments. Expressed in the blimp center-of-mass centered frame (with the x axis pointing towards the movement direction, over the longitudinal axis of the blimp, the z axis pointing downwards, and the y axis completing an orthonormal coordinate system), the linearization of the kinematic and dynamic equations leads to the state space model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{6 \times 6} & \mathbf{I}_{6 \times 6} \\ -M^{-1}\mathbf{g} & -M^{-1}\mathbf{D} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{6 \times 3} \\ M^{-1}\mathbf{B} \end{bmatrix} \mathbf{u}$$

where x_1 is the 6×6 position and translational velocity vector and x_2 the 6×6 orientation and rotational velocity vector. \mathbf{B} is the actuators position matrix, in the blimp's frame. Observing carefully the achieved model, it is clear that it can be divided in two different systems, entirely decoupled. Consequently, we are in the presence of two independent systems, one that describes the blimp's motion on the vertical plane with $\mathbf{u} = [X \ Z]^T$ (X and Z force

components) and the other that models its rotational motion over the z axis with $u = F_{MT}$ (heading moments).

The model parameters were identified by adequate experimental tests carried out on the aerial blimp, using the transfer function version of the linear state equations above.

Briefly, the accomplished experiments were made separately for the two decoupled subsystems. With the robot at rest, it was applied (at $t=0$ s) a step at the appropriate input of each subsystem, corresponding to a sudden change on the PWM value of the actuators, equivalent to a 2 N force. The sequence of blimp positions/orientations was measured for each test and the final motion was plotted for visual inspection and comparison with Simulink blimp model simulations.

The system identification was carried out using the Matlab *Ident* toolbox, from the input/output set obtained for each subsystem and using the ARX parametric model to find the best match for each subsystem.

B. CONTROL LOOP

A feedback linearization control law was successful tested in simulation, using the identified blimp dynamic model, and provoking mismatches between the parameters of the actual model and of the model used by the controller. However, the blimp onboard DSP has not enough power to implement the full controller for the real blimp. Therefore, we only used the inertia matrix in the control law

$$u = K_{3 \times 6} M (K_{P_{6 \times 6}} e + K_{D_{6 \times 6}} \dot{e})$$

and introduced the gain matrices K , K_P and K_D to scale the error e between the desired and actual blimp positions (or orientations), measured on the image, and its derivative.

Due to the decoupling between the x - z vertical plane and rotational motion models, two separate controllers were designed, one for the translation along the x axis (the altitude was not controlled and simply kept by the balance between the blimp impulsion and weight) and another for the rotation over the z axis (*yaw*). The former keeps a desired blimp speed when following a line or tracks the ground vehicle speed, while the latter keeps the blimp aligned with a desired direction (the ground line tangent or the ground vehicle heading). The last two subsections focus on the image processing for each of the cases.

C. GROUND LINE FOLLOWING

To determine ground lines to be followed, a Sobel edge detection algorithm [9] was applied to the image acquired by the video camera (Fig. 6).

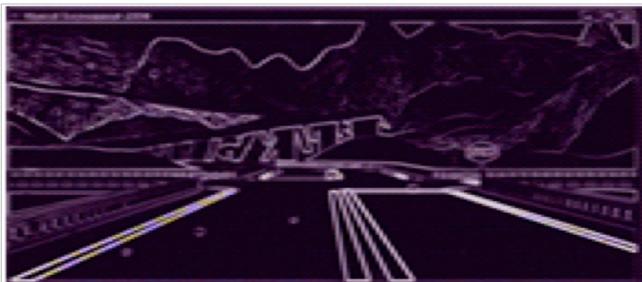


Fig. 8. Result of the edge detection algorithm. The gray level is proportional to the luminosity gradient.

The gradient image obtained, depicted in Fig. 8, was transformed into a black and white image (Fig. 9) by the use of a threshold.

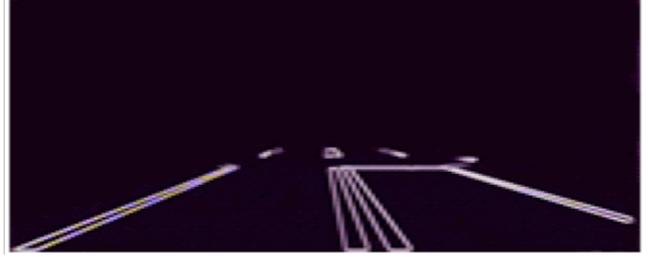


Fig. 9. Black and white image of the edges.

The next step was to use the Hough Transform [9] to detect the straight lines in the image (Fig. 10). The four more predominant straight lines were selected.



Fig. 10. Hough transform space. The horizontal coordinate is the straight line angle and the vertical one is the straight line distance to the origin.

In order to determine the target direction, a novel procedure was adopted. This procedure consists in intersecting the four straight lines determined previously with a half-circle centred at the image centre and with a diameter of half the image's height (red half-circle in Fig. 11). This results in at most 4 points over the circle (there may be straight lines that do not intersect the circle at all).

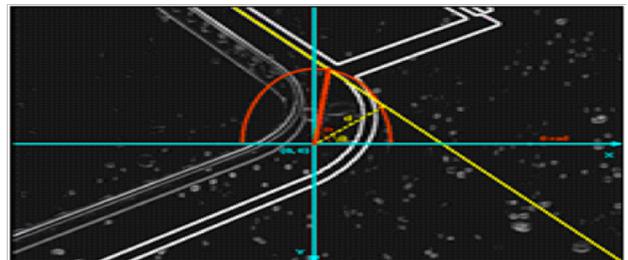


Fig. 11. Result of the image processing algorithm. The target direction is represented by the red line segment.

In the initial stages of the algorithm, when the autonomous navigation system is started, the closest intersection point to the image vertical axis is selected. In the following steps, the selected intersection point is the one closest to the previous intersection point.

The desired blimp speed along the x -axis is kept constant, but multiplied by a reduction factor that depends on the trajectory characteristics, e.g., the speed reduction is larger on curved lines.

D. GROUND VEHICLE TRACKING

In this case, the two control systems must track dynamic references.

The x -axis reference is the x coordinate of the tracked vehicle centre of mass in the blimp image frame. The goal of the control system is to reduce this coordinate to zero, by actuating on the speed of the blimp blade main propellers.

The y coordinate of the tracked vehicle centre of mass in the blimp image frame, and the angle between the image frame y -axis and the vector E that links the image frame origin to the vehicle centre of mass (see Fig. 12), play a role in the rotation controller, which acts on the blimp blade tail propeller to reduce the angle to $\pi/2$.

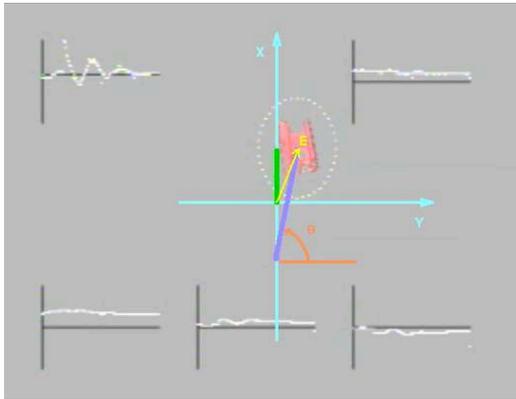


Fig. 12. Real blimp following a line on the ground. The image includes plots of the temporal evolution of different quantities such as the duty cycle of the motor control PWM signals.

V. INDOOR TESTS

After the image processing and control algorithms were completed and tested in the virtual environment of the simulation setup the blimp was assembled and tested in real conditions, in an indoor sports pavilion, under changing light conditions but no wind. Fig. 13 shows the blimp successfully following a white line in the pavement of a sports arena where other white lines were present.

In Fig. 14 shows the image sent by the blimp to the ground station using the RF link. This image includes plots of the line being followed (blue line) as well as of the temporal evolution of different quantities such as the duty cycle of the motor control PWM signals.



Fig. 13. Real blimp following a line on the ground.

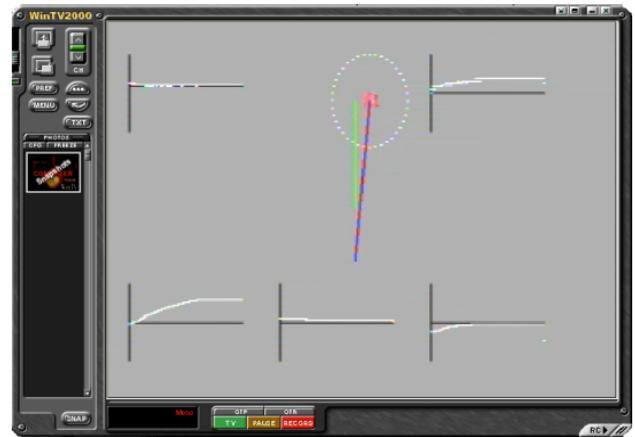


Fig. 14. Output of the DSP sent to ground control.

Fig. 15 shows the blimp tracking a tele-operated iRobot ATRV-Jr vehicle. The results of tracking figure-8 and U-shaped trajectories of the ground robot were quite satisfactory and matched quite accurately previous simulations made with the simulation setup.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we introduced an autonomous indoor blimp, with onboard computation electronics and a video camera. A hardware-in-the-loop test setup was also presented which enables accurate and fast system development and easy portability to real operation conditions.



Fig. 15. Real blimp tracking a tele-operated iRobot ATRV-Jr robot.

There are several developments that can be carried out in the future to improve the system. The electronics can be miniaturized through the development of a custom made board containing the DSP, the image coders, the DACs and the motor controllers. This would diminish the size, since a lot of unused electronics existent in the DSP evaluation board used would be eliminated. It would also lower the weight and reduce the power consumption which, in turn, would allow fewer batteries to be used.

Towards outdoor operations, we intend to endow the current blimp with more powerful motors. Regarding the navigation system, we intend to install onboard a GPS receiver, so that the blimp can follow a set of predefined waypoints.

ACKNOWLEDGMENTS

The authors acknowledge the support, by their Pluriannual fundings, of the Institute for Systems and Robotics and the Instituto de Telecomunicações at Instituto Superior Técnico.

REFERENCES

- [1] T. Fukao, K. Fujitani and T. Kanade, "An Autonomous Blimp for a Surveillance System", Proceedings of the 2003 IEEE/RSJ Intl. Conference on intelligent Robots and Systems, Las Vegas, Nevada. October 2003
- [2] E. Hygounenc, I.-K. Jung, P. Soueres, S. Lacroix, "The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping", International Journal of Robotics Research, Vol. 23, No. 4-5, 473-511, 2004
- [3] J. Ko, D. J. Klein, D. Fox, D. Haehnel, "Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp", IEEE International Conference on Robotics and Automation, pp. 742-747, 2007
- [4] H. Zhang, J. Ostrowsky, "Visual Servoing with Dynamics: Control of an Unmanned Blimp", IEEE International Conference on Robotics and Automation, Michigan, 1999
- [5] A. Elfes, S. Bueno, M. Bergerman, J. Ramos, "A Semi-Autonomous Robotic Airship for Environmental Monitoring

- Missions", IEEE International Conference on Robotics and Automation, 1998
- [6] S. Zwaan, A. Bernardino, J. Santos-Victor, "Vision Based Station Keeping and Docking for an Aerial Blimp". Proc. of the IEEE/RSJ international Conference on Intelligent Robots and Systems, 2000.
- [7] P. Lima, M. Isabel Ribeiro, Luís Custódio, José Santos-Victor, "The RESCUE Project – Cooperative Navigation for Rescue Robots", Proc. of ASER'03 - 1st International Workshop on Advances in Service Robotics, March 13-15, 2003 – Bardolino, Italy.
- [8] Jijun Wang, "USARSim – A Game-based Simulation of the NIST Reference Arenas", Carnegie Mellon Technical Report.
- [9] R. J. Schalkoff, Digital Image Processing and Computer Vision. New York: Wiley, 1989
- [10] T. Fossen, Marine Control Systems Guidance, Navigation, and the Control of Ships, Rigs and Underwater Vehicles. Trondheim: Marine Cybernetics, 2002.