# Developing a Low-Cost Autonomous Indoor Blimp

P. González[1], W. Burgard[2], R. Sanz[1] and J.L. Fernández[1]

*Abstract*— **This paper describes the design of an autonomous blimp-based robot and its navigation system. The robot was based on a commercial kit and its dimensions were suitable for use in indoor environments. Our main goal was to develop a simple and safe model for evaluating different autonomously controlled navigation techniques. Due to the special requirements of this application, two specific electronics boards to control the blimp and to communicate with the PC ground station were designed and two different altitude controllers and also a controller to maintain distance from obstacles were implemented. Finally, comparative results on both altitude controllers are presented.**

*Index Terms*— **Blimp, autonomous robot, robot navigation, altitude controller, control avoidance controller, fuzzy logic control.**

## I. INTRODUCTION

Non-rigid airships, also known as blimps, are basically unmanned aerial vehicles (UAVs) that use gas (usually helium) balloons. In contrast to a rigid airship, a blimp has no internal structure to maintain the shape of its hull envelope. Rather, its shape is maintained by a higher pressure of the gas. The only rigid components are the driving elements, the fins and the gondola attached to the envelope.

Unmanned blimp robots can be used in both indoor and outdoor environments. The buoyancy force provides an energy-free form of lift, offering a non-traditional approach to long-duration missions for which conventional aircrafts are not well-suited. Miniaturization of sensors and actuators and the development of long-duration batteries have also opened up opportunities for further progress in the development of these small-scale autonomous vehicles.

The first rigid airships, which were constructed in the early 20th century, consisted of a balloon with a metal frame covered by fabric and filled with a gas (helium or hydrogen). These airships were mainly used in wars for military aerial exploration and transportation. Nowadays, however, they are mainly used for advertising and aerial filming. Nevertheless, they have great potential in terms of applications such as search and rescue missions, traffic monitoring, urban planning, inspection of power lines and pipelines, mineral and archaeological site prospection, law enforcement and telecommunication relay systems [1]. Blimps are well-suited for these applications because their ability to remain stationary for long periods of time in the air enables data to be gathered. Blimps can also be used for research purposes in a variety of applications including ecological, biodiversity and climate research and monitoring in different environments [2].

Our primary interest was the development of a low-cost blimp designed to operate autonomously in indoor environments where different control strategies and navigation paradigms are tested and evaluated. The design of a blimp imposes certain restrictions, primarily because of its limited payload capability, given that a blimp relies on its neutral buoyancy to stay afloat. A key challenge was to build an electronic board that was sufficiently light to be carried on board the blimp. Electronic components were selected to fit our main navigation requirements including limited autonomous navigation capabilities.

This paper is structured as follows. Section 2 discusses related works. The commercial blimp selected and other main components are described in detail in Section 3. Section 4 describes the design and implementation of the navigation system, which basically consists of a fuzzy logic obstacle avoidance controller and an altitude controller. Two different control techniques were experimented with for the altitude controller: a simple proportional–integral–derivative (PID) linear controller and a fuzzy logic controller. The experimental results for these systems are also compared in this section. Finally, Section 5 summarizes our conclusions.

## II. RELATED WORKS

Several researchers have recently developed autonomous robotic systems based on blimps and studied appropriate control paradigms. Much of this research is devoted to large-scale systems, with payloads of kilograms and therefore capable of including a number of sensors (such as cameras) and remaining airborne for long periods of time. Elfes and colleagues [1] presented arguments that favor blimps over airplanes and helicopters as ideal platforms for standard aerial exploration missions. Kantor and colleagues [2] discussed the use of solar energy as a renewable source of power for airships using an outdoor blimp. Hygounenc and colleagues [3] focused on flight control and terrain mapping issues in cooperation between ground and aerial robots.

This work was partially funded by Spanish Ministerio de Ciencia y Tecnología (DPI2005-06210) and the Xunta de Galicia (PGIDIT06PXIC 303194PN).

[1] Dept. of Systems Engineering and Automation, University of Vigo, Spain.

[2] Dept. of Computer Science, University of Freiburg, Germany.

Other authors described the use of blimps in indoor environments. Motoyama and colleagues [4] designed an autonomously controlled indoor blimp and an action-value function for motion planning based on a potential field method, evaluating its effectiveness in a simulated environment [5]. Geoffrey and colleagues [6] used a commercial indoor blimp, concluding that the vertical motor was severely underpowered in tele-operated control. These researchers used a commercial wireless board to send the sensor measurements to the ground computer and a separate board to control the motors. A PC running Linux was used to process sensor data and send control signals to the blimp and a wireless communication unit (WCU) was used for sensor data communication and blimp-side servo control. Hydrogen and helium were used to increase the payload capacity.

We used the same type of blimp as Geoffrey and colleagues [6], but with a different vertical motor, as a more powerful motor and a big propeller were necessary to control altitude so as to have full altitude control in each indoor environment in which the vehicle was tested. We also developed a lighter specific board to control the motors and sensors and thus had more payload capacity for additional components. We only used helium, because hydrogen is flammable and so is very dangerous in indoor environments.

An important navigation problem is automatic control of altitude and of horizontal movement. If the blimp can be maintained at a specific altitude, it can be moved in a horizontal plane. Kadota and colleagues [7] used PID controllers to control blimp altitude and horizontal movement, arguing that blimp trajectory could be unstable in the vertical direction. We also used PID and fuzzy logic controllers to control blimp altitude and evaluated the performance of the control systems developed using the controllers in two different environments.

A second important navigation problem for blimps (and for autonomous mobile robots in general) is obstacle detection and collision avoidance. Green and colleagues [8] used an infrared sensor to detect obstacles: when the collision avoidance system detects an obstacle, the blimp turned 180 degrees to avoid collision. The collision avoidance system does not take account of blimp dynamics, however; consequently, we used an ultrasonic sensor to measure the distance to potential obstacles and implemented a fuzzy logic controller to avoid collisions.

## III. BLIMP DESIGN

The design of an autonomous blimp has certain restrictions arising from the assembled hardware. Various issues were evaluated simultaneously in terms of making appropriate choices. For a blimp system, the higher the volume of the envelope—that is, the higher the ascending force—the higher the possible payload. However, a blimp for indoor applications has to be fairly small. For our blimp system, the goal was to minimize both the size of the blimp and the weight of the necessary onboard hardware.

Due to the limitation on payload, onboard hardware could not be equipped with sensors to measure absolute vehicle position and this limited the autonomous navigation capabilities of our vehicle (as described below).

The main characteristics of the selected blimp components and the design of other elements are described immediately below.

### A. Blimp selection

Payload, which depends on balloon volume, is a critical aspect that constrains the choice of other onboard components such as the battery.

The blimp that we selected was a hobby radio-controlled (RC) blimp from Plantraco, which comes with an RC transmitter than can be connected via USB to a PC and controlled via a connection to a TCP socket. It has a 52” non-rigid hull made of a light material, achieving 200 grams of estimated payload capacity (Figure 1) [9] [10]. This size is very appropriate for indoor laboratory applications. It can fly in a corridor with people and enter rooms through standard-sized office doors. The payload capacity of this blimp was considered adequate for our purposes.

The blimp has a tri-turbofan gondola with three light DC micro-motors: a vertical motor allows altitude to be regulated and the other two motors control speed and rotation. Since the original vertical motor does not have enough power to properly control altitude, we replaced it with a Futaba 3003 servo motor and a bigger propeller blade. With all these changes the gondola components weighed only 55 grams, leaving 74 grams free for other hardware components including the battery (Table I).

TABLE I
BLIMP COMPONENT WEIGHTS

| Component | Weight (g) |
|---|---|
| Envelope (52"x37") | 60.5 |
| Gondola components | 55.0 |
| Fins and propellers | 10.5 |
| Onboard hardware | 39.0 |
| Battery | 35.0 |
| Total blimp weight | 200.0 |

Fig. 1. The commercial blimp selected for this research, with a tri-

turbofan gondola at its base. Balloon size was 52". The vertical motor controls altitude and the side motors control horizontal movements.

The choice of battery was another key aspect because of the weight issue: a very light battery that still provided enough current and autonomy was required. We selected a 350mA lithium polymer battery based on a new technology. Weighing only 35 grams, it supplies a maximum current of 3A and provides around 40 minutes of autonomy for the blimp. The nominal voltage level of 11V was too high for the electronic components, and so a power management circuit to provide the required voltages (8V and 5V) was designed. A circuit to control voltage level was also added to avoid damage to the battery when the voltage was under 9V.

### B. *Ultrasonic sensors*

Two ultrasonic sensors provided the autonomous blimp with information on the environment. An ultrasonic, lightweight SRF05 sensor (with a resolution of 1 mm and a very narrow beam) was mounted facing downwards at the bottom of the gondola to measure the distance from the blimp to other objects. Sensor measurements were integrated by means of a Kalman filter which sequentially estimated blimp altitude. The other ultrasonic sensor, located in the forward-facing part of the balloon, was used by the obstacle avoidance controller. In this case, we selected a lightweight SRF10 ultrasonic sensor with an operating range of up to 6 meters that could be connected to the microcontroller via a standard I2C bus interface.

### C. *Electronic components*

An electronic board was custom-designed for this application because no commercial board met with our requirements (Figure 2). An appropriate communication system was also designed and implemented. The main reasons motivating this approach were the following:

- A new motor speed control was necessary, firstly, because the blimp could not be controlled using the original circuit, which only worked at maximum speeds, and secondly, because we could not control the new vertical motor with the original board.
- Good wireless communication coverage with a lightweight circuit was desirable, so lighter components for communications between the vehicle and the PC ground station were selected, given that a wireless access point or a wireless router could not be used because of the limited payload capability.
- A bidirectional communication link to send and receive data from the blimp was required in order to be able to send data to and from the blimp and the PC ground station (in other words, we needed to close the control loop).

For the above reasons, two specific boards were designed: a gondola onboard card and a PC interface card.

*Gondola onboard card.* This board included all the electronic components necessary to control the three motors and the onboard wireless communication link (see Figure 3). The main components are:

*Power supply unit.* It consists of a power regulator which provides two stabilized voltage levels: 8V and 5V and a battery circuit which prevents total discharge of the battery.

*Microcontroller unit (MCU).* It is based on a PIC microcontroller, which remains the best balance of cost, processing power, complexity, and power consumption. The PIC 16F873 microcontroller from Microchip appears to be a good choice. It has USART, analog ports, and I2C ports required to connect sensors and motor drivers. Some other inputs and outputs are used to communicate with the wireless transceiver.

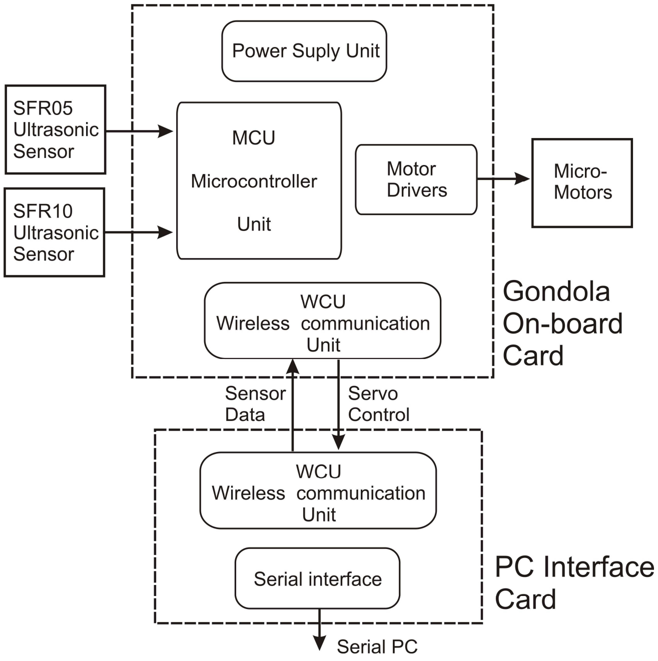


Fig. 2. Block diagrams of the gondola onboard and PC interface cards.

*Motor drivers.* They are necessary to control the speed of each motor. The drivers are based on a L293B circuit, which offers 1A per channel and can modulate the voltage and control the motor speed. The pulse-width modulation (PWM) is directly controlled by the microcontroller.

*Wireless communication unit (WCU).* It is employed to transmit data between the blimp and the PC ground station. It consists of a serial wireless modem that works in the 400 MHz band, and it is based on an ER400TRS transceiver. It also includes a buffer to protect the transceiver. Data are transferred in real time at a speed of 19,200 bps. The WCU sends sensor data to the ground PC and receives servo positions from the ground PC to control the blimp (Figure 2).

*PC interface card.* The blimp was remotely controlled using a customized computer board that also contained a WCU. We also developed a software package module with generic functions that enabled easy control of the blimp by the programmer.

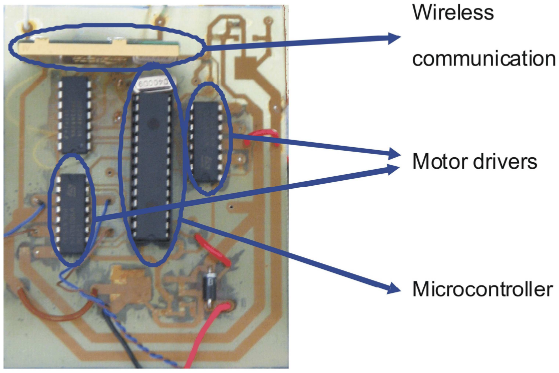


Fig. 3. Electronic components of the gondola onboard card.

The ground computer could both write and read from the WCU using these functions. The design of the function interface was such that the programmer was not required to deal with serial port characteristics and the communication protocol between the blimp and the computer. The interface had ten user-friendly functions so that the programmer could develop software to move the blimp in few minutes.

### D. *Onboard software*

The onboard software processed the sensor measurements and sent them to the PC ground station. These data were received by the PC ground station and used to compute new control signals for the gondola micromotors. The WCU in the PC interface card sent control data to the onboard WCU, which were used by the microcontroller to set the corresponding commands for the micromotors.

## IV. BLIMP NAVIGATION SOFTWARE

The navigation software provided limited autonomous operation of our blimp due to:

- The lack of odometry.
- Misalignment caused by wind gusts and temperature changes.
- The non-linear nature of propeller action.
- Blimp operation in three dimensions.

The control system for our autonomous blimp was designed basically to keep the blimp moving in a straight line if its path was clear, maintaining the desired horizontal speed and reference altitude. When a head-on obstacle was detected the control system attempted to maintain a certain distance from it. Assuming a low speed, the control problem could be decoupled in two sub-systems describing motions in both vertical and horizontal planes [11].

Although some blimp control approaches are based on the vehicle mathematical model (see [11] and [12], for example), we preferred not to implement an analytical controller because the blimp has complex dynamics due to its nonlinear characteristics and environmental influences (air gusts, temperature, altitude references, etc). We implemented both PID and fuzzy logic controllers. PID parameters can be experimentally adjusted on the basis of well-known methods and fuzzy logic controllers can be easily tuned on the basis of trial and error.

Our control system was composed of two different controllers that would ensure safe autonomous navigation in indoor environments: an altitude controller and a collision avoidance controller. Both controllers were implemented in the PC ground station using the distance measurements sent by the WCU, with computed control signals sent back to the onboard microcontroller.

The reference blimp altitude was specified in the program user interface, with the altitude control algorithm endeavoring to automatically maintain the blimp at this altitude. Controlling vertical motion reduced blimp movements by one degree of freedom. Two different altitude controllers were implemented and tested.

The collision avoidance controller enabled control of the horizontal movements of the blimp so as to avoid frontal collisions. Only a fuzzy controller was implemented for this purpose because the results for PID controllers were poor. When the front sensor did not detect an obstacle in its path, the controller navigated the blimp along a straight line; when an obstacle was detected, the controller kept the blimp at a certain distance from the obstacle.

### A. *PID altitude controller*

The current altitude of the blimp was measured by the SRF05 ultrasonic sensor and then sent to the PC. Sensor errors were corrected with a Kalman filter, thus obtaining the estimated altitude. The filter parameters depended on sensor characteristics, the dynamic model of the blimp and the previous measurements, but in our case they were tuned experimentally. We did not make use of the Kalman filter variance because of real time restrictions in controlling motor action.

The first altitude controller implemented was a PID-type controller. Control actions were calculated as:

$$u_{PID} = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{1}$$

where $K_p$, $K_i$ and $K_d$ were parameters experimentally calculated using the Zieger-Nichols method and $u_{PID}$ was the command signal to the vertical propeller, responsible for up-and-down movements of the aerial vehicle.

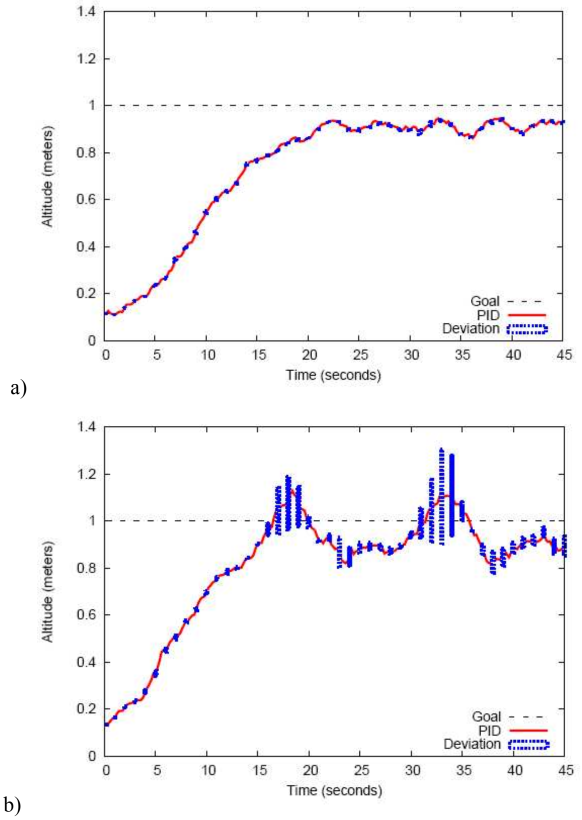


Fig 4. Behavior of the PID altitude controller when the altitude reference is set to 1 meter. a) Experiments in environment 1. b) Experiments in environment 2.

Blimp altitude in Figure 4 is depicted in red, while average values are represented in blue. The figure shows the blimp PID altitude controller behavior for two different scenarios. The first was a corridor in Building 79 of Freiburg University (Figure 5), while the second was a computer laboratory in the same university (Figure 6). The computer laboratory had a door that, when opened, could alter the draught inside. Eight similar experiments were performed in both environments, all commencing with the blimp lying on the ground and then moving to its target position, one meter above the ground.

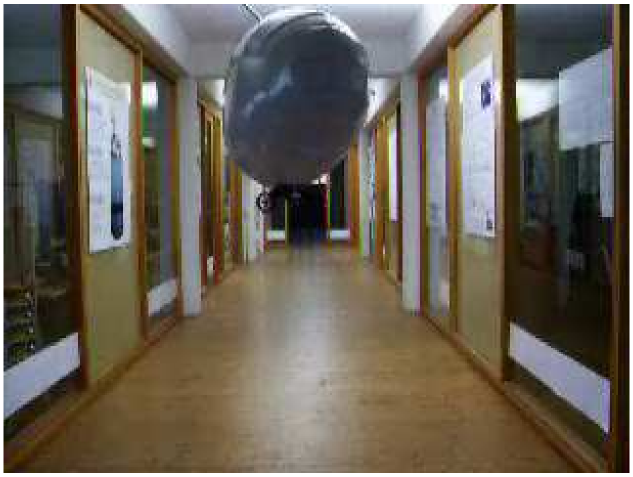


Fig 5. Experimental environment 1: a corridor in Building 79, Freiburg University.

The experiments have shown diverse results using the same controller in these two environments. An overall good performance can be observed during the experiments carried out in environment 1. However, the same PID controller in environment 2 has shown an oscillating behavior with an important deviation from the mean. Although it can reach the altitude reference, there is a significant error as it is shown in Figure 4b.

Note that the controller response in the second environment could be improved by tuning PID parameters. In fact, these parameters had to be recalculated to take account of changes in environmental conditions while the blimp was navigating.

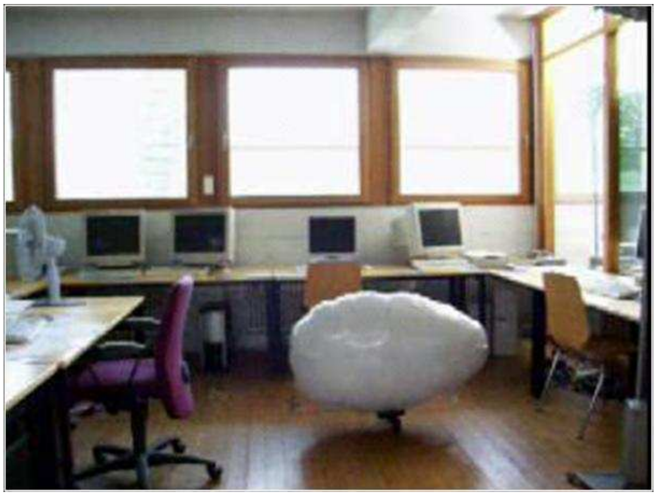


Fig 6. Experimental environment 2: a computer laboratory in Freiburg University.

### B. *Fuzzy altitude controller*

A non-linear altitude controller based on fuzzy logic was also implemented. Fuzzy logic uses fuzzy sets to model designer knowledge about the system to control, with knowledge representation modeled using fuzzy rules. This kind of controller has several advantages because it does not need to recalculate parameters when environmental conditions changes.

A fuzzy controller is composed of a knowledge base and an inference engine. The knowledge base contains rules and linguistic variable descriptions, while the inference engine generates a control action as a function of state variable values in a given time instant.

The altitude fuzzy logic controller in our blimp had two inputs: altitude error and estimated current vertical speed. Altitude error was the difference between the desired altitude and current altitude (Figure 7). A change in altitude error indicated whether the aerial vehicle was approaching the reference altitude or moving away from it. The controller output was the vertical motor command. Note that the fuzzy controller structure differed from the PID controller in that the latter has only one input, which ensures better and more accurate altitude control.

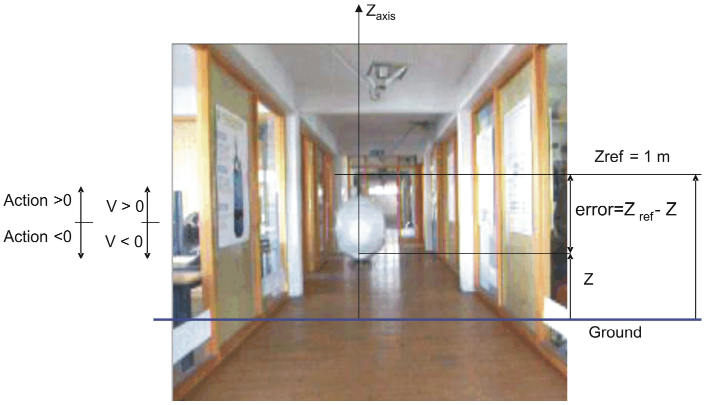


Fig. 7. Variable relations regarding altitude control of the blimp. Zref is the altitude reference; Z is the estimated height using a Kalman filter. Other variables are shown on the left.

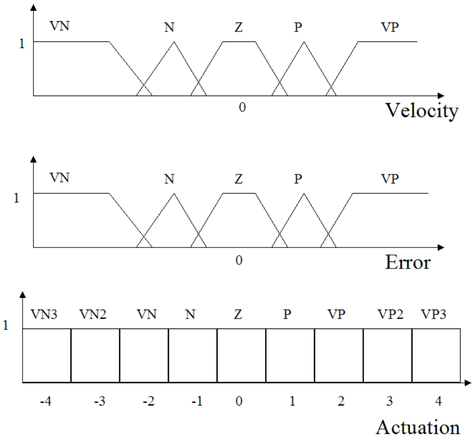


Fig 8. Fuzzy linguistic variables for the fuzzy altitude controller.

The fuzzy logic controller is characterized by a set of linguistic variables and fuzzy if-then rules. All input and output linguistic variables have a finite number of linguistic values with membership functions that are empirically defined after exhaustive simulation studies. The linguistic values representing the linguistic variable *altitude error* and *vehicle speed* are *very negative* (VN), *negative* (N), *zero* (Z), *positive* (P) and *very positive* (VP). Linguistic values describing the output linguistic variable are *very negative 3* (VN3)*, very negative 2* (VN2), *very negative* (VN), *negative* (N), *zero* (Z), *positive* (P), *very positive* (VP), *very positive 2* (VP2) and *very positive 3* (VP3). Their corresponding membership functions are shown in Figure 8.

Fuzzy rules describe the controller behavior in terms of relationships between input and control variables. A rule is usually of the type:

*If $x_1$ is $A_1$ and $x_2$ is $A_2$ then y is B,*

where $x_i$ and $y$ are, respectively, input and control linguistic variables, and $A_i$ and $B$ are linguistic terms.

The altitude controller can be described with a small set of rules, for example:

*If altitude error is negative and vertical speed is very positive then motor command is positive.*

Figure 9 depicts, in tabular form, the fuzzy rule set used to generate the motor command. These rules endeavor to maintain the blimp at a specific height and represent human expertise on how to control the system. Note that the rule descriptions were developed on the basis of multiple experiments.

Figure 10 depicts one of the situations described in the table of Figure 9. In this situation, the blimp is descending and the altitude error is significant. In the position shown in the figure, the input variables have the linguistic labels VN for velocity and VP for altitude error, respectively. So, the command signal value must be VP3, as shown in the fuzzy rule table depicted in Figure 9.

Note that one set of rules describes when motor action must be null. These rules have a dead band to avoid continuous motor action.

Experimental results for the fuzzy altitude controller are depicted in Figures 11a and 11b. The blimp showed only slight oscillations in both environments and deviations from the mean, shown in blue, were very small. A significant number of tests carried out in different circumstances led to similar good results in both environments.

Error

| Velocity | VN | N | Z | P | VP |
|---|---|---|---|---|---|
| VN | Z | P | VP | VP2 | VP3 |
| N | N | Z | P | VP | VP2 |
| Z | VN | N | Z | P | VP |
| P | VN2 | VN | N | Z | P |
| VP | VN3 | VN2 | VN | N | Z |

⇨ Action

Fig 9. Fuzzy rules for altitude control.

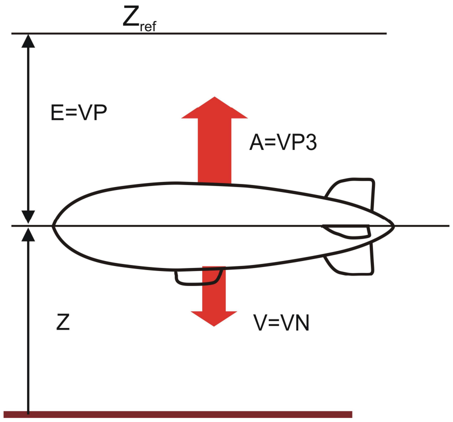
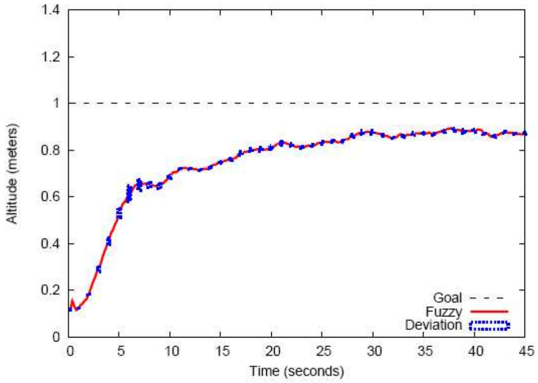
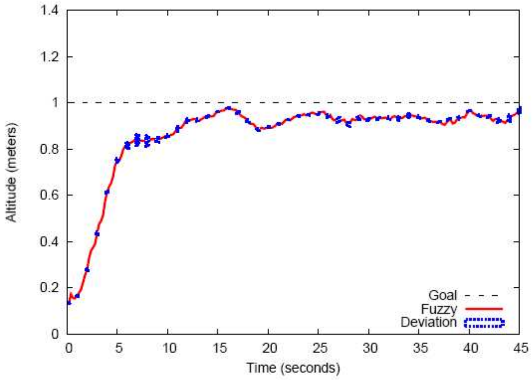


Fig. 10. Behavior of the fuzzy altitude controller when the blimp descends.



a)



b)

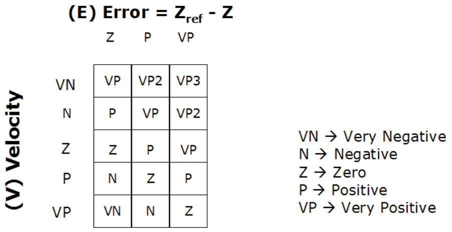
Fig. 11. Behavior of the fuzzy altitude controller when the altitude reference is set to 1 meter. a) Experiments in environment 1. b) Experiments in environment 2.

### *C. Collision avoidance controller*

The collision avoidance system should cause the vehicle to stop reliably when the frontal distance sensor detects an obstacle in the vicinity. In such circumstances, the horizontal navigation speed is changed by the collision avoidance controller. Only the frontal sensor no longer detects an obstacle close to the blimp, horizontal blimp speed is reset to a certain value. Different approaches can be used to implement the collision avoidance controller.

We again chose a fuzzy logic approach, based on a PID controller which demonstrated oscillating behaviors in almost all the experiments carried out in both environments. The goal of this controller was to keep the blimp at a safe distance from frontal obstacles. Controller inputs were frontal distance and estimated speed and controller output was a speed index for the horizontal motors.

Fuzzy rules for the collision avoidance controller are shown in Figure 12. Figure 13 depicts a possible situation in which the blimp is very close to an obstacle detected by the frontal sensor. The blimp is heading straight for a vertical wall; speed is V=VN and the distance error value is E=VP (near the wall). In this situation, which is critical for the blimp as it is likely to collide with the wall, the fuzzy collision avoidance controller transmits the maximum control command (A=VP3) to the motors, thereby transmitting the maximum opposite power to ensure that the blimp avoids the collision. However, if this speed is used for a long time the blimp develops inertia and it becomes impossible to stop it in the desired reference. When the speed is reduced so as to be comprised within the linguistic variable V=N, the blimp does not need maximum power and control action is reduced to A=VP2. These few rule examples illustrate how the controller works.



| (V) Velocity \ (E) Error = $Z_{ref}$ - Z | Z | P | VP |
|---|---|---|---|
| VN | VP | VP2 | VP3 |
| N | P | VP | VP2 |
| Z | Z | P | VP |
| P | N | Z | P |
| VP | VN | N | Z |

VN → Very Negative
N → Negative
Z → Zero
P → Positive
VP → Very Positive

Fig. 12. Fuzzy rules for the collision avoidance controller.

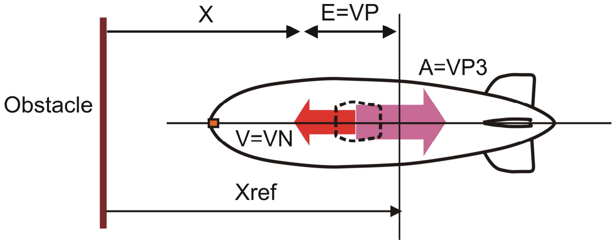


Fig. 13. Behavior of the fuzzy collision avoidance controller when an obstacle is detected.

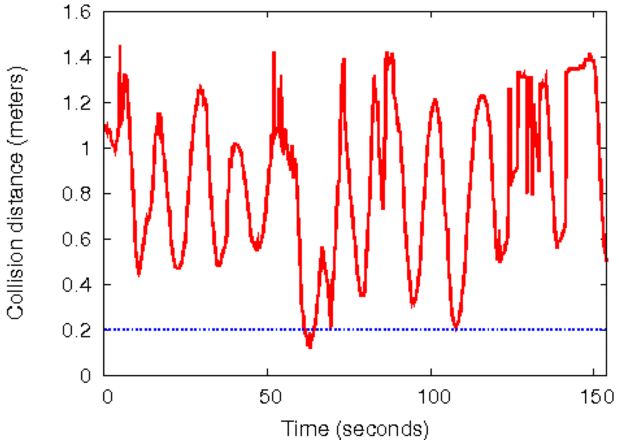


Fig. 14. Behavior of the blimp in a corridor. The blue line is the safety distance from the floor.

Figure 14 shows the blimp distance to a vertical wall while it is navigating in environment 1. Initially, it maintained a constant horizontal speed. When it approached a wall at the end of this corridor, the obstacle avoidance controller reduced the speed accordingly. In the experiment, the vehicle maintained a security distance of about 0.2 meters away from the obstacle without crashing at any time.

## V. CONCLUSIONS

In this paper we have described the construction of a small blimp based on a commercial kit, with the main goal of designing and implementing different control navigation techniques. It was necessary to change the vertical motor of the commercial blimp and develop onboard control hardware, including a wireless radio connection with a PC ground station. Due to the vehicle's small payload only two distance sensors were used for navigation; the small payload also did not permit inclusion of an inertia sensor or miniature wireless camera usually used for global position calculation [6].

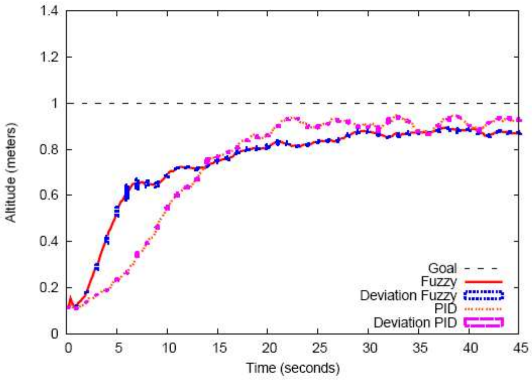
We also implemented onboard and PC programs to control the blimp with the PC ground station, with the communication software module capable of receiving onboard sensor data and sending commands to the blimp (as well as facilitating future application developments).

Two different controllers were implemented. An altitude controller maintained the blimp at a certain distance from the floor and a second controller avoided obstacles in the path of the blimp. The altitude controller was implemented using two different approaches: PID and fuzzy logic. A first experiment was designed to compare the two controllers operating in the same conditions, so as to determine which controller was better suited to controlling blimp altitude.

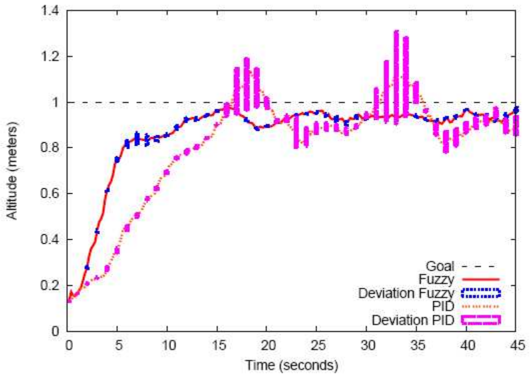
In the first scenario, both controllers were tested on an alternating basis in one environment (a corridor), an experiment that was repeated eight times. Similarly, the blimp was tested in a second environment (a computer room) with the same perturbations on the blimp for both controllers.

Comparative results show that the fuzzy logic controller produced balanced behavior in either of the two environments. The PID altitude controller performed slightly better than the fuzzy logic controller in environment 1 (see Figure 14a) but its performance in environment 2 was significantly poorer than that of the fuzzy logic controller. A better PID controller in other environments would require online adjustment of PID parameters. Using a fuzzy logic controller, on the other hand, does not affect the behavior of the blimp too much and the controller parameters do not require modification.

A second fuzzy logic controller, designed to avoid head-on collision with obstacles while the vehicle was navigating in the indoor environment, showed good experimental performance.



a)



b)

Fig. 13. Comparative results for the PID and fuzzy logic controllers. a) Experiments in environment 1. b) Experiments in environment 2.

Note that we could not measure the absolute position of our vehicle with the onboard sensor due to the low payload of the blimp. This limited blimp navigation in a desired direction without global path planning. For this reason we are developing a new design with a bigger balloon than described in this paper that will enable new vehicle control functions to be incorporated [13].

## ACKNOWLEDGMENTS

We would like to thank the people of the University of Freiburg (Germany) for their assistance during Pablo González's visit to the Autonomous Intelligent Systems Laboratory.

## REFERENCES

[1] A. Elfes, S. Siquiera, M. Bergerman, and J. Guimaraes, "A semi-autonomous robotic airship for environmental monitoring missions," in *Proceedings of the 1998 IEEE. International Conference on Robotics and Automation*, 1998.

[2] G. Kantor, D. Wettergreen, J. P. Ostrowski, and S. Singh, "Collection of environmental data from an airship platform," in *Proceedings SPIE Vol 4571, Sensor Fusion and Decentralized Control in Robotic Systems*, 2001.

[3] E. Hygounenc, I.-K. Jung, P. Soueres and S. Lacroix, "The autonomous blimp project of LAAS-CNRS: Achievements in flight control and terrain mapping," *The International Journal of Robotics Research*, vol. 23, n. 4-5, pp. 473-511, 2004.

[4] K. Motoyama, H. Kawamura, M. Yamamoto, and A. Ohuchi, "Development of autonomous blimp robot with intelligent control", in *Entertainment Computing: Technologies and Applications*, Eds. R. Nakatsu and J. Hoshino, Kluwer Academic Pub. 2003, pp. 191-198.

[5] K. Motoyama, H. Kawamura, M. Yamamoto, and A. Ohuchi, "Design of evaluation function in motion planning for autonomous balloon robot", in *Proceedings of 2003 Asia Pacific Symposium on Intelligent and Evolucionary Systems*, 2003.

[6] G. A. Hollinger, Z. A. Pezzementi, A. D. Flurie, and B. A. Maxwell, "Design and construction of an indoor robotic blimp for urban search and rescue tasks", *Swarthmore College Senior Design Thesis*, Spring 2005.

[7] H. Kadota, H. Kawamura, M. Yamamoto, T. Takaya, and A. Ohuchi, "Vision-based positioning system for indoor blimp robot", *Proceedings of the 4th International Conference on Advanced Mechatronics-Toward Evolutionary Fusion of IT and Mechatronics*, 2004.

[8] W. E. Green, K. W. Sevcik and P. Y. Oh, "A competition to identify key challenges for unmanned aerial robots in near-earth environments", in *Proceedings. 12th International Conference on Advanced Robotics ICAR '05*, July 2005.

[9] Balloon shop. http://www.heo.com.

[10] Plantraco shop. http://www.plantraco.com.

[11] S. van der Zwaan, M. Perrone, A. Bernardino, and J. Santos-Victor, "Control of an aerial blimp based on visual input", in *Proceedings of the 8th International Symposium on Intelligent Robotic Systems SIRS'00*, 2000.

[12] N. Rooz and E. N. Johnson, "Design and modeling of an airship station holding controller for low cost satellite operations," in Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, August 2005.

[13] A. Rottmann, M. Sippel, T. Zitterell. W. Burgard, L. Reindl, and C. Scholl." Towards an Experimental Autonomous Blimp Platform", in Proceedings of 3rd European Conference on Mobile Robots (ECMR '07), September 2007.