TOWARDS VISION GUIDED NAVIGATION OF AUTONOMOUS AERIAL ROBOTS

Geraldo F. Silveira*,†, José R. H. Carvalho†, Marconi K. Madrid*
Samuel S. Bueno†, Patrick Rives‡

†*Robotics and Computer Vision Laboratory/ITI*
PO Box 6162, 13081-970, Campinas-SP, Brazil
Fist_name.Last_name@iti.gov.br

**UNICAMP – State University of Campinas*
PO Box 6101, 13081-970, Campinas-SP, Brazil
silveira@dsce.fee.unicamp.br

‡*INRIA Sophia-Antipolis*
2004 Route des Lucioles, 06565 Valbonne Cedex, France
Patrick.Rives@sophia.inria.fr

Abstract— Visual servoing has rapidly emerged as a new and exciting field of research. Its main characteristic is the direct feedback of visual signals into the control loop. The aim of this work is to develop a vision-based navigation framework for autonomous aerial robots from the task specification level to the control level. A line-oriented navigation scheme with simulation results are presented to validate our proposed methodology.

Key Words— Mobile robot, visual servoing, vision-based control, aerial unmanned vehicle.

1 Introduction

Recent advances on computer vision and image processing capabilities have allowed the effective inclusion of visual data as feedback signals in the control loop. The advantages of such feedback is clear. From the point of view of robotic applications, this spans and improves the variety of tasks currently able to be performed. Concerning vision aspects, it enables to control the camera motion to promote better recognition, localization and inspection of the environment.

Visual servoing is a maturing approach in which the motion of the sensor or of the objects in the scene are controlled, as opposed to be only exploited. The methodology proposed in (Espiau et al., 1992) is considered as a basis to our vision-based navigation framework. This method is based on an Interaction Matrix, whose purpose is to relate the changes in the robot's working space and its own motion by using geometric primitives as reference images. The velocities of the image feature parameters are used *directly* to compute control signals for driving the robot actuators to perform pre-defined tasks. Hence, the framework does not require any camera calibration procedure or explicit 3D scene reconstruction, which are susceptible to errors and demand high processing capabilities.

Research on the utilization of aerial unmanned vehicles (AUV) appeared in the last decade due to the huge list of potential applications. A new and special attention has been given to the use of AUVs in environmental applications, biodiversity and climate research. For these kind of tasks, airships out-rank helicopters and airplanes as the ideal platform, mainly because their aerostatic lift makes them less intrusive, noiseless, capable of hovering and able to fly during longer periods. In project AURORA (Bueno et al., 1998), a robotic airship is being developed for such kind of applications, see Fig. 1.

Figure 1. Aerial Unmanned Robot.

Recently, vision-based methodologies designed to navigate robotic vehicles have been targets of researchers. Refer to (Silveira et al., 2001*a*), (Silveira et al., 2001*c*), (Ma et al., 1999), (Rives and Borrelly, 1997) for examples of visual servoing applications.

In this work, as the basis for the development of control and navigation strategies, a 6DOF model, Simulink-based control system development environment is used for design and validation purposes. To visualize and to represent the world where the vehicle is flying around, it is used a visualizator based on Mesa, which is a freeware implementation of the openGL Graphics Library. The systems communicate with each other through TCP/IP sockets, increasing flexibility.

This paper is organized as follows: Section 2 describes the specification level of navigation tasks. In Section 3 the visual servoing framework is presented, which consists of visual and control aspects. In Section 4 the simulated results are shown demonstrating the feasibility of the system. A case study addressed for two different cases of interaction matrix is also made in this section. Finally, in Section 5 some remarks and future work are stated to conclude the paper.

2 Task specification

The main objective of this paper is to develop a pure vision-based navigation methodology for autonomous aerial vehicles which is both useful and feasible by using off-the-shelf components.

Before a task can be executed by a robotic vehicle, several appealing problems from the task planning level to the control laws synthesis must be efficiently solved. To complete the vision guided navigation task by a robot, it must be capable to identify and calculate or to be given the following requirements:

- (i) The type of virtual linkage of the task execution and the degrees of freedom constrained by the linkage;
- (ii) The type of the visual signals allowing to accomplish the task;
- (iii) The symbolic form of the associated interaction matrix, which is needed by the image-based visual servoing;
- (iv) The embedded actuators to achieve the task.

which one will be further discussed.

Let us represent the camera in a perspective projection model. This model permits to define a local differential mapping between the configuration and the sensor output spaces ($s : SE_3 \rightarrow \mathcal{R}^n$). Now consider a set of 2D features resulting from the projection of the 3D geometric primitives in the scene onto the image frame. Denoting $s = (s_1, s_2 \dots s_n)^T$ as the vector of parameters that characterizes these 2D features, s will be used as our vector of useful visual feedback signals.

We will explain the navigation framework through an example of navigation task for didactic reasons. The authors are currently working on the generalization of such framework, expanding to any kind of sensor and robotic platform.

2.1 Navigation from lines

As an example of a navigation task, we have chosen to perform a line-oriented navigation by a robotic airship, depicted in Fig. 1, as our validating testbed. The task specification can be stated as: given a set of lines, e.g., the bands of a road, the robotic airship has to follow it with a established mission-dependent altitude and longitudinal speed.

The problem may also be formulated in the image plane as regulating the image so that the central line is intended to be vertically centered with both lateral lines lying simetrically with a certain inclination. The number of lines used here (three) is the minimum number to perform such a task avoiding the redundancy that appears due to the projection of the actual target in the image plane.

In the sequel, it is developed the visual servo control used to complete the task by using lines as our visual feedback signal.

3 Visual servo control

Visual servo control is established when data from images are used in a visual-feedback control loop (Hutchinson et al., 1996). In some cases, the pixel information is processed to rebuild the tridimensional configuration of the robot and these variables are used as control input. In other methodologies the image feature parameters are used directly in the control algorithm computation. This last possibility is explored in our approach.

3.1 Vision aspects

The methodology is based on the modeling of *scene features* as a set of tridimensional geometrical primitives (points, lines, vertices, etc.). An *image feature* is a set of primitives in the image plane which corresponds to the projection of the scene feature. A configuration of the image feature is an element P_I.

Using the camera model and without loss of generality, the focal length is assumed to be equal to 1 so that any point with coordinates $\bar{x} = (x, y, z)^T$ is projected on the image plane as a point with coordinates $\bar{X} = (X, Y, 1)^T$ with

$$\bar{X} = \frac{1}{z}\,\bar{x} \quad . \tag{1}$$

The relevant variables in the case of path following over straight lines are the parameters that describe completely those image features configuration P_I. The canonical equation of a line in the image plane is $g(\bar{X}, \bar{P}) = AX + BY + C = 0$. However, this representation is not minimal (3 parameters (A, B, C) and $dim(P_I) = 2$). Therefore, it is better to define lines using polar representation (θ, ρ), which is minimal:

$$g(\bar{X}, \bar{P}) = X\cos\theta + Y\sin\theta - \rho = 0 \tag{2}$$

where the image feature parameters θ and ρ are defined by $\theta = \arctan(\frac{B}{A})$ and $\rho = -\frac{C}{\sqrt{A^2+B^2}}$.

As we have three lines, our vector of image feature parameters is $s = (\theta_1, \rho_1, \theta_2, \rho_2, \theta_3, \rho_3)^T$, which represents the parameters of the central and lateral lines in the image plane. Thus, we have solved the type of visual signals to use. The purpose of the vision task is to extract and compute these feature parameters at each frame during robot motion, and deliver them to the control system.

Image segmentation and tracking of actual reference lines in outdoor environments are not usually straightforward. The authors are also currently working on this subject in different areas: an active perception architecture for image detection and classification, see for instance (Elfes et al., 2000) and the study of suitable image processing techniques, including real-time and robustness aspects, see Fig. 2. (Silveira et al., 2001*b*) , (Carvalho et al., 2000).

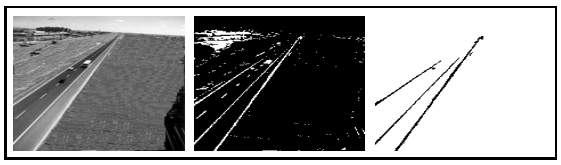


Figure 2. Sequence of Mathematical Morphology-based filters used to find the three lines from a real image of a highway.

3.2 Control aspects

Given that the vision aspects are solved in the camera frame $\mathcal{F}_c$, one has to derive the relationship between the 2D motion in the image frame to the frame attached to the robot $\mathcal{F}_r$. The *Interaction Matrix* L^T is a suitable mathematical relationship between the motion of the parameters vector $\dot{s}$ of certain geometric primitives and the motion T_c of $\mathcal{F}_c$. T_c has translational and rotational velocities and it is given as $T_c = [v_x, v_y, v_z, w_x, w_y, w_z]^T$.

$$\dot{s} = L^T \, T_c \tag{3}$$

The derivation of the Interaction Matrix for our specific case of navigation task is described briefly in the sequel. It is related to straight lines as the geometric primitives h in the image plane. For other types of primitives, the reader may refer to (Espiau et al., 1992).

Derivation of L^T

In order to achieve an expression for the lines, we have to represent them in the SE_3. A straight line can be represented as the intersection of two planes, having configuration $\bar{p}$:

$$h(\bar{x}, \bar{p}) = \begin{cases} a_1 x + b_1 y + c_1 z + d_1 = 0 \\ a_2 x + b_2 y + c_2 z + d_2 = 0 \end{cases} \tag{4}$$

The differentiation of Eq. (2) gives

$$\frac{\partial g}{\partial \bar{P}}(\bar{X}, \bar{P})\dot{\bar{P}} = -\frac{\partial g}{\partial \bar{X}}(\bar{X}, \bar{P})\dot{\bar{X}} \tag{5}$$

The well know optical flow equation is

$$\dot{\bar{X}} = L_{of}^T(\bar{X}, z) \; T_c \tag{6}$$

where L_{of}^T is given by

$$L_{of}^T = \begin{bmatrix} \frac{-1}{z} & 0 & \frac{X}{z} & XY & -(1+X^2) & Y \\ 0 & \frac{-1}{z} & \frac{Y}{z} & 1+Y^2 & -XY & -X \end{bmatrix} \tag{7}$$

Isolating z from Eq. (4) and using Eq. (1), with $i = 1$ if $d_1 \neq 0$ or $i = 2$ if $d_2 \neq 0$, the substitution in L_{of}^T achieves $L_{of}^{'T}$ and (6) can be rewritten. Finally, using this result in (5) we obtain:

$$\frac{\partial g}{\partial \bar{P}}(\bar{X}, \bar{P})\dot{\bar{P}} = -\frac{\partial g}{\partial \bar{X}}(\bar{X}, \bar{P}) \; L_{of}^{'T}(\bar{X}, z) \; T_c \tag{8}$$

which corresponds to the general equation for deriving the interaction matrix. For more complex geometric primitives, simply find $L_{of}^{'T}$ that best represent them and solve (8).

Solving the first term of (8) to suit our case,

$$\frac{\partial g}{\partial \bar{P}}(\bar{X}, \bar{P})\dot{\bar{P}} = -X \sin\theta \; \dot{\theta} + Y \cos\theta \; \dot{\theta} - \dot{\rho} \tag{9}$$

From (2) we take X expressed as a function of Y, when $\cos\theta \neq 0$ (or Y as a function of X in the other case). Then, after some manipulations, it is possible to rewrite (8) where the second term may be separated between those terms which depend on the value of Y and those which are not,

$$-\frac{\dot{\theta}}{\cos\theta} Y + (\dot{\rho} + \rho \tan\theta \; \dot{\theta}) = Y K_1 T_c + K_2 T_c \tag{10}$$

and comparing term-by-term we derive

$$\begin{matrix} \dot{\theta} = -K_1 \cos\theta \; T_c \\ \dot{\rho} = (K_2 + \rho \sin\theta \; K_1) \; T_c \end{matrix} \Rightarrow \begin{bmatrix} \dot{\theta} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} L_\theta^T \\ L_\rho^T \end{bmatrix} T_c \tag{11}$$

After some calculations, we have L_θ^T and L_ρ^T in the final form:

$$\begin{matrix} L_\theta^T = [& \lambda_\theta \cos\theta & \lambda_\theta \sin\theta & -\lambda_\theta \rho \\ & -\rho\cos\theta & -\rho\sin\theta & -1 \;] \\ L_\rho^T = [& \lambda_\rho \cos\theta & \lambda_\rho \sin\theta & -\lambda_\rho \rho \\ & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \;] \end{matrix} \tag{12}$$

with $\lambda_\theta = (a_i \sin\theta - b_i \cos\theta)/d_i$ and $\lambda_\rho = (a_i \rho \cos\theta + b_i \rho \sin\theta + c_i)/d_i$.

From this point, we already have the type of signals and how those signals change for a motion of the camera, which is described by the symbolic form of the interaction matrix L^T.

Control law

The pose (position and orientation) of the camera is an element $\bar{r}$ of $\mathcal{R}^3 \times SO_3$ which is a six dimensional differential manifold. A classical approach in robotics is to consider the process of achieving a task, such as tracking or positioning, as a problem of regulation to zero of a certain function. In other words, the control objective is considered as perfectly achieved during $[0,T]$ if $e(\bar{r},t) = 0,\ \forall t \in [0,T]$.

The application of the task function approach to vision-based control is straightforward: the task is now described as desired values for a certain set of elementary visual signals. Our main visual task function vector e is then:

$$e(\bar{r},t) = L^{T^+}(s(\bar{r},t) - s_d) \qquad (13)$$

where $s(\bar{r},t)$ is the value of the visual features currently observed by the camera, which only depends on the relative pose between the camera and the scene; s_d is the desired value for s and L^{T^+} is the pseudoinverse of L^T.

We can design a very simple *velocity control scheme* using visual feedback, which leads to a task function to be regulated according to

$$T_c = -\lambda e, \quad \text{with} \quad \lambda > 0 \qquad (14)$$

for an exponential convergence.

The three lines-oriented navigation task, defined in section 2.1, is required to satisfy a *Prismatic-type linkage* by analogy to the theory of mechanisms. Such a linkage is characterized by 5 constrained degrees of freedom (DOF), therefore $C = dim(SE_3) - 5 = 1$ degree remains free. This DOF comes from the translation along the lines. In our application, we will control the DOF constrained by the linkage using visual servoing techniques and for the remaining DOF we will give a reference constant velocity ν for simplicity. Other trajectory tracking technique may also be applied. Thus, our final control law is given below:

$$T_r = -{}^rM_c \left(\lambda e + \alpha(\mathbb{I}_6 - W^+W)\ {}^c\nu\right) \qquad (15)$$

where:

- α is a scalar which ponderates the secondary objective, which was set to 1;
- $(\mathbb{I}_6 - W^+W)$ is an orthogonal projection operator on the null space of W, which is the same as L^T, to promote the coupling between the two objectives;
- ${}^rM_c$ is a matrix of frame transformation $\mathcal{F}_c \rightarrow \mathcal{F}_r$.

From this point, we already have the type of the virtual linkage, the degrees of freedom constrained by such linkage and the final form of the control law as well.

4 Results

The setup used to obtain the simulation results presented here is based on the airship 6DOF model depicted in Fig. 1. The pair simulator/visualizator communicate to each other through TCP/IP sockets. The simulation hardware was a workstation Sun ULTRA 10 (UltraSPARC IIi-330MHz) with 128Mb of RAM.

4.1 Case study of L^T

As previously shown, the associated interaction matrix L^T depends both on parameters measured in the image, $\bar{P}$, and the 3D configuration of the used primitives, $\bar{p}$, which is *a priori* unknown. In this case, it is necessary to choose a model $\hat{L}$ of L. Some possibilities are given below:

- $\hat{L} = L(\hat{\bar{p}}, \bar{P})$, where $\bar{p}$ is concurrently estimated;
- $\hat{L} = L(\bar{p}_d, \bar{P})$. By using this model, the parameters $\bar{p}$ is set to be the value at the desired position $s(\bar{r},t) = s_d$ and the others are time-varying.
- $\hat{L} = L(\bar{p}_d, \bar{P}_d)$. In this case both values of the parameters are set at the final camera configuration with respect to the environment. Therefore, they are constants.

For the proper selection of such model, the designer has to take into consideration aspects such as computing time versus required performance and a possibly crossing of an isolated singularity, which leads to a loss of rank. Figure 3 shows the rate of convergence using the two last cases for a quick comparison of how the entire airship behavior is affected by incorporating more information.

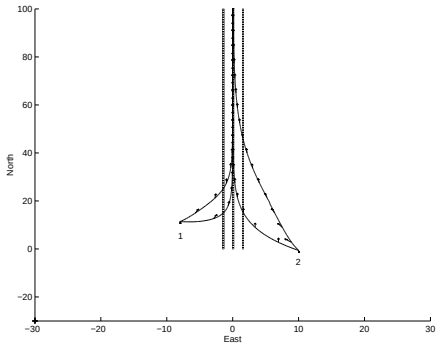


Figure 3. Comparison of the control behavior for two models of L^T.

In the first set of curves the lateral gain is set to $\lambda = 1.5$, with the initial pose of the robot at $12m$ North, $-8m$ East and yaw angle of $\psi = \pi/5$. For the second set of curves, the lateral gain is $\lambda = 0.5$ and launched at $0m$ North, $10m$ East and $\psi = -\pi/4$. Both with cruising speed set to $\nu = 8m/s$, roll and pitch angles (ϕ, θ) to 0^o, initial altitude of $-30m$ and final altitude $h_{ref} = -20m$.

$$L^T = \begin{bmatrix} -\frac{\cos^2\theta_{rht}}{h_ref} & -\frac{\cos\theta_{rht}\sin\theta_{rht}}{h_{ref}} & \frac{\rho_{rht}\cos\theta_{rht}}{h_{ref}} & -\rho_{rht}\cos\theta_{rht} & -\rho_{rht}\sin\theta_{rht} & -1 \\ \frac{\rho_{rht}\sin\theta_{rht}\cos\theta_{rht}}{h_{ref}} & \frac{\rho_{rht}\sin^2\theta_{rht}}{h_{ref}} & -\frac{\rho^2_{rht}\sin\theta_{rht}}{h_{ref}} & (1+\rho^2_{rht})\sin\theta_{rht} & -(1+\rho^2_{rht})\cos\theta_{rht} & 0 \\ -\frac{\cos^2\theta_{cnt}}{h_ref} & -\frac{\cos\theta_{cnt}\sin\theta_{cnt}}{h_{ref}} & \frac{\rho_{cnt}\cos\theta_{cnt}}{h_{ref}} & -\rho_{cnt}\cos\theta_{cnt} & -\rho_{cnt}\sin\theta_{cnt} & -1 \\ \frac{\rho_{cnt}\sin\theta_{cnt}\cos\theta_{cnt}}{h_{ref}} & \frac{\rho_{cnt}\sin^2\theta_{cnt}}{h_{ref}} & -\frac{\rho^2_{cnt}\sin\theta_{cnt}}{h_{ref}} & (1+\rho^2_{cnt})\sin\theta_{cnt} & -(1+\rho^2_{cnt})\cos\theta_{cnt} & 0 \\ -\frac{\cos^2\theta_{lft}}{h_ref} & -\frac{\cos\theta_{lft}\sin\theta_{lft}}{h_{ref}} & \frac{\rho_{lft}\cos\theta_{lft}}{h_{ref}} & -\rho_{lft}\cos\theta_{lft} & -\rho_{lft}\sin\theta_{lft} & -1 \\ \frac{\rho_{lft}\sin\theta_{lft}\cos\theta_{lft}}{h_{ref}} & \frac{\rho_{lft}\sin^2\theta_{lft}}{h_{ref}} & -\frac{\rho^2_{lft}\sin\theta_{lft}}{h_{ref}} & (1+\rho^2_{lft})\sin\theta_{lft} & -(1+\rho^2_{lft})\cos\theta_{lft} & 0 \end{bmatrix} \quad (16)$$

The final form of the interaction matrix for each case is given in the next subsections. We clearly notice that a time-varying L^T promotes a faster rate of convergence, since it accounts for the instantaneous configuration of the robot with respect to the scene.

Time-varying L^T

Taking the model $\hat{L}$ of L as $L = L(\bar{p}_d, \bar{P})$, the parameters $\bar{P}$ are measured at the frame rate and $\bar{p}$ has to be found for the desired configuration of the robot. The horizontal plane desired configuration $h_{2d} = (a_{id}, b_{id}, c_{id}, d_{id})$ that characterizes all three lines is $h_{2d} = (0, 1, 0, h_{ref})$.

After the appropriate substitution of those parameters and considering our vector of image features that represents the parameters of the central and lateral lines in the image plane as $s = (\theta_{rht}, \rho_{rht}, \theta_{cnt}, \rho_{cnt}, \theta_{lft}, \rho_{lft})^T$, the time-varying interaction matrix is given in Eq. (16).

Constant L^T

When $L = L(\bar{p}_d, \bar{P}_d)$, the value of the desired configuration for both parameters is used. Translating to our case of navigation task, we have a vector of image features at the final configuration given by $s_d = (\theta_{rhtd}, \rho_{rhtd}, \theta_{cntd}, \rho_{cntd}, \theta_{lftd}, \rho_{lftd})^T$, which represents the parameters of the central and lateral lines in the image plane, equals to

$$h_{rht}\begin{cases} y + h_{ref} = 0 \\ x + \frac{l}{2} = 0 \end{cases} \Rightarrow \begin{cases} \theta_{rhtd} = \arctan(-\frac{l}{2h_{ref}}) \\ \rho_{rhtd} = 0 \end{cases}$$

$$h_{cnt}\begin{cases} y + h_{ref} = 0 \\ x = 0 \end{cases} \Rightarrow \begin{cases} \theta_{cntd} = 0 \\ \rho_{cntd} = 0 \end{cases}$$

$$h_{lft}\begin{cases} y + h_{ref} = 0 \\ x + \frac{l}{2} = 0 \end{cases} \Rightarrow \begin{cases} \theta_{lftd} = \arctan(\frac{l}{2\ h_{ref}}) \\ \rho_{lftd} = 0 \end{cases}$$

In such case, the interaction matrix is said to be constant and it has the final form, suited to our case, through the substitution $s = s_d$ in the Eq. (16).

4.2 Vehicle behavior during task execution

We launched our navigation task to observe its behavior under the best case of the Interaction Matrix and at the same conditions as the first curve of the Fig. 3.

It is shown in Fig. 4, from the airship's cockpit view, both the initial and final frame acquired by an on-board camera during the navigation. For the initial configuration, we have our vector of visual feedback signal equals to $s = (17.09, -0.68, 18.25, -0.69, 19.29, -0.70)^T$ and the final configuration as given in section 4.1, achieving the task. A comprehensive illustration of the complete task is depicted in Fig. 5. In this case it is shown, by an external fixed camera, the beginning and the end of the complete task execution.

The evolution of the Euler angles and how the desired altitude was reached is depicted in Fig. 6. The control signals sent to the simulated robot is depicted in Fig. 7. In this same figure, it is presented the exponential convergence of the norm$_2$ of the output error $||e = s(\bar{r}, t) - s_d||_2$.

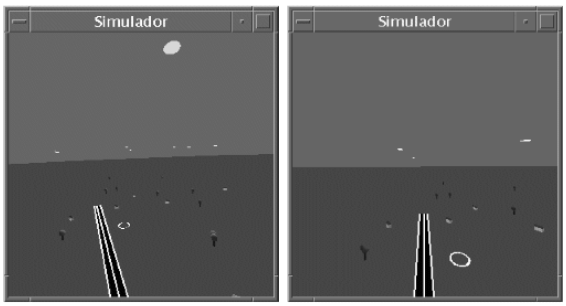
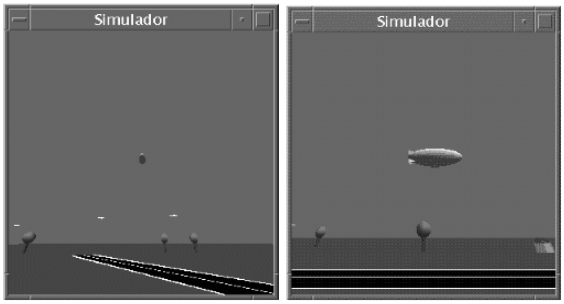


Figure 4. airship's cockpit view from the initial and final position, achieved using the methodology.

5 Conclusion

Aerial robots, mainly airships, experienced an increasing interest over this past decade thanks to their enormous untapped potential in low-speed, low-altitude exploration, monitoring, biodiversity and climate research as well as telecommunication relay systems.

After a briefly recall on visual servo control techniques, a new vision guided navigation framework for aerial robots, from the task specification up to the control level, is proposed. The methodology is validated through the results shown in the preceding section.



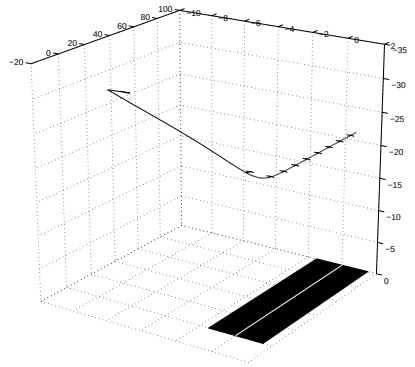


Figure 5. airship behavior from an outer fixed camera.

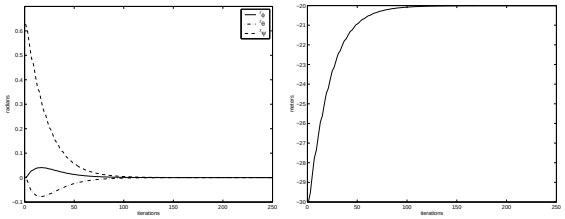


Figure 6. Euler angles and altitude behavior

In this paper, it is also presented an application of a pure vision-based control. The task here, described in terms of image parameters, was to follow a pre-defined trajectory composed by lines. The control system aligns the aerial autonomous robot with those lines, having an exponential convergence of the error, through useful visual information feedback. Another contribution of the paper is the case study for two models of the interaction matrix. It is shown the influence of adding more information to L^T to the rate of convergence.

Further steps include a generalization of such framework; a deeper study on the vision aspects, testing for robustness of the image filters and, finally, a future work that will concentrate on implementing this navigation system in the real aerial robot AS-800 produced by Airspeed Airships, which is in the context of the Project AURORA (Bueno et al., 1998).

Acknowledgments

This project earns partial grant from FAPESP under no. 97/13384-7 and 98/13562-5 and CNPq/CTPETRO under grant no. 466713/00-2.

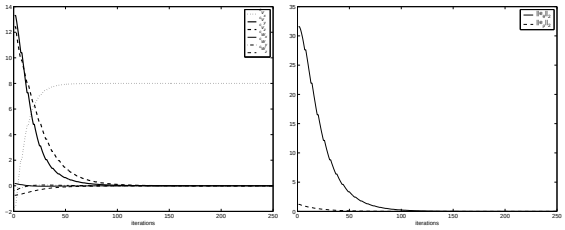


Figure 7. Control signals and the error in the image plane.

References

Bueno, S. S., Elfes, A., Bergerman, M. and Ramos, J. G. (1998). A semi-autonomous robotic airship for environment monitoring missions, *ICRA*, Belgium.

Carvalho, J. R. H., Rives, P., Santa-Bárbara, A. and Bueno, S. S. (2000). Visual servo control of a class of mobile robot, *Int'l Conference on Control Applications*, Alaska, USA.

Elfes, A., Carvalho, J. R. H. and Bergerman, M. (2000). Towards dynamic object identification using stochastic lattice models and optimal design of experiments, *ICRA*, San Francisco-CA, USA.

Espiau, B., Chaumette, F. and Rives, P. (1992). A new approach to visual servoing in robotics, *IEEE Transactions on Robotics and Automation* **8**: 313–326.

Hutchinson, S., Hager, G. D. and Coke, P. I. (1996). A tutorial on visual servo control, *IEEE Transactions on Robotics and Automation* **12**(5): 651–670.

Ma, Y., Kosecka, J. and Sastry, S. S. (1999). Vision guided navigation for a nonholonomic mobile robot, *IEEE Transactions on Robotics and Automation* **15**(3): 521–537.

Rives, P. and Borrelly, J. (1997). Visual servoing techniques applied to an underwater vehicle, *ICRA*, Albuquerque, USA.

Silveira, G. F., Carvalho, J. R. H., Madrid, M. K., Bueno, S. S. and Rives, P. (2001). Lateral control of an aerial unmanned robot using visual servoing techniques, *Proceedings of the IEEE 2nd. RoMoCo*, Poland.

Silveira, G. F., Carvalho, J. R. H., Madrid, M. K., Rives, P. and Bueno, S. S. (2001). A fast vision-based road following strategy applied to the control of aerial robots, *Proceedings of the XIV SIBGRAPI*, Florianópolis, Brazil.

Silveira, G. F., Carvalho, J. R. H., Shiroma, P. M., Rives, P. and Bueno, S. S. (2001). Visual servo control of nonholonomic mobile robots, *Proceedings of the 16th COBEM*, Uberlândia, Brazil.