Emmanuel Hygounenc
Il-Kyun Jung
Philippe Souères
Simon Lacroix

LAAS/CNRS, 7, av. du Colonel Roche
31077 Toulouse Cedex 4, France
Emmanuel.Hygounenc@laas.fr

The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping

Abstract

In this paper we provide a progress report of the LAAS-CNRS project of autonomous blimp robot development, in the context of field robotics. Hardware developments aimed at designing a generic and versatile experimental platform are first presented. On this base, the flight control and terrain mapping issues, which constitute the main thrust of the research work, are presented in two parts. The first part, devoted to the automatic control study, is based on a rigorous modeling of the airship dynamics. Considering the decoupling of the lateral and longitudinal dynamics, several flight phases are identified for which appropriate control strategies are proposed. The description focuses on the lateral steady navigation. In the second part of the paper, we present work on terrain mapping with low-altitude stereovision. A simultaneous localization and map building approach based on an extended Kalman filter is depicted, with details on the identification of the various errors involved in the process. Experimental results show that positioning in the three-dimensional space with a centimeter accuracy can be achieved, thus allowing the possibility to build high-resolution digital elevation maps.

KEY WORDS—airship control, UAV, lighter than air modeling, vision-based SLAM

1. Introduction

Our long-term objectives in field robotics is to tackle the various issues raised by the deployment of heterogeneous autonomous systems, in the context of exploration, surveillance and intervention missions. Within such contexts, aerial robots will undoubtedly play a growing role in the near future, not only during the mission preparation phase (in which drones can already gather environment information, for instance), but also on-line, during the mission execution. Aerial robots can then provide the rovers with telecommunications support, as well as with up-to-date information on the environment. They can also localize the rovers as they evolve within this environment, achieve by themselves some of the mission goals, and even be implied in tight cooperation schemes with ground rovers.

In this context, we initiated the development of an autonomous blimp project. The ever on-going developments in a wide spectrum of technologies, ranging from actuator, sensors and computing devices to energy and materials will ensure lighter than air machines a promising future. There is undoubtedly a resurgence of interest in this domain, as shown by the recent industrial developments on heavy loads transportation projects—such as the ATG Skycats (for up-to-date information on this project, see http://www.airship.com), not to mention various other prospective transportation projects—and on stratospheric telecommunication platforms. As for small-size unmanned radio-controlled models, the size of which is of the order of a few tens of cubic meters, their domain of operation is currently essentially restrained to advertising or aerial photography. However, their properties make them a very suitable support to develop heterogeneous air/ground robotics systems: they are easy to operate, they can safely fly at very low altitudes (down to a few meters), and especially their dynamics is comparable with the ground rovers dynamics, as they can hover a long time over a particular area, while being able to fly at a few tens of kilometers per hour, still consuming little energy. Their main and sole enemy is the wind; see Elfes et al. (1998) for a detailed and convincing review of the pros and cons of small-size airships with regards to helicopters and planes. Let us also note that some specific applications of unmanned blimps are more and more seriously considered throughout the world, from planetary exploration to military applications, as shown for instance by numerous contributions in the AIAA Lighter-Than-Air Conferences (e.g., AIAA

The International Journal of Robotics Research
Vol. 23, No. 4–5, April–May 2004, pp. 473-511,
DOI: 10.1177/0278364904042200

2001) and European Airship Conventions (e.g., Airship Association 2002).

The first mention of the development of unmanned autonomous blimps can be found in the literature of the late 1980s, but it is only recently that various projects have reached effective achievements. One of the most advanced is Aurora, a project held at the Information Technology Institute of Campinas, Brazil, mainly devoted to flight control (de Paiva et al. 1999; Azinheira et al. 2001), but within which other issues are also considered (Elfes et al. 1999; Elfes, Bergermann, and Bueno 2001). Other projects are also under development at the University of Virginia (Turner 2000), at the University of Stuttgart (Kungl, Schlenker, and Krplin 2001; Wimmer and Well 2001), and the University of Wales (Barnes, Summers, and Shaw 2000). More recently, a project of autonomous blimp navigation has started at the CEMIF Laboratory of the University of Evry, France. Their first results concern the system modeling and the trajectory planning problem (Bestaoui and Hima 2001). One of the interesting characteristics of such projects is that they mix various innovative technological developments and fundamental research.

Besides long-term developments related to the coordination and cooperation of heterogeneous air/ground robots, our research work on autonomous blimps is currently twofold: we concentrate on the navigation problem on the base of automatic control, and on environment modeling issues using low-altitude imagery.

In this paper we present our current achievements in these two areas. We begin with the presentation of Karma, the 15 m^3 airship within which our work will eventually be integrated. Karma's hardware and software architectures are briefly presented, and the paper is then split in two parts. Part A presents the developments related to flight control. The description of the complete model is first presented, and the control strategy based on the decoupling of the longitudinal and lateral dynamics is then explained. The description focuses on the steady lateral dynamics which constitutes the central navigation phase. Part B is devoted to the terrain mapping issue. The various algorithms that allow terrain mapping on the basis of non-registered images are sketched, and the application of a simultaneous localization and map building (SLAM) approach is presented in detail. Results are presented, which show the capacity of our approach to allow the building of very high-resolution digital elevation maps of several thousands of square meters, integrating hundreds of images taken at an altitude of a few tens of meters.

2. The Blimp Karma

2.1. An Airspeed Airship AS-500

We acquired at the end of 2001 an "AS-500" radio-controlled airship from the English company Airspeed Airship. (The company's homepage is http://www.airship.demon.co.uk/airspeed.html; the platform used in the Aurora project is an AS-800, a similar bigger model.) Criteria for this choice were the size of the blimp (which we wanted to be rather small, for the ease of deployment and storing), its available payload and its possible operation modes. The main characteristics of the airship as delivered by Airspeed Airship are the following.

- It is 8 m long, with a 1.90 m maximum diameter, giving a volume of about 15.0 m^3, and a fitness ratio of 4.25.
- The hull is made of welded mylar, and equipped with four control rudders in a "X shape" configuration. A ballonnet fed with air captured at the rear of the propellers maintains a constant hull pressure, and a radio-controlled security valve, located on the top of the hull, allows the release of helium in case of emergency.
- The available payload of the AS-500 is around 3.5 kg.

Specific modifications. In collaboration with Airspeed Airships, we specified the following modifications for our purpose.

- Electric motors. To have a finer controllability, we preferred to opt for electric motors. These do not weigh more than fuel engines nominally proposed for the AS-500, but are less powerful, thus reducing the maximum reachable speed and the possibility to fly in wind gusts. However, the main drawback of this choice is the payload loss due to the required batteries.
- Stern thruster. The rudder control surfaces require a certain speed to allow changes in both the altitude and orientation of the blimp. In order to have the possibility to control the yaw angle while hovering, we chose to add a stern thruster.

The various control parameters of the blimp are sketched in Figure 2. Note that, after a few flight tests, it appeared that the blimp yaw angle could be controlled thanks to the rudders even at very low speeds; the stern thruster appeared to be rather useless (see Section 2.4).

2.2. Hardware Architecture

To transform the blimp from a radio-controlled machine to a robot, we equipped it with a set of proprioceptive and exteroceptive sensors, and with computing and communications capabilities.

Stereovision. One of the advantage of blimps is that they can carry a wide base stereo bench, thus having the possibility to directly gather three-dimensional (3D) data on the overflown ground. We adapted two high-resolution digital B&W cameras on a rigid 2.4 m carbon profile that traverses the gondola.

Blimp state observation. In order to allow automatic flight control, we added the following sensors: a differential GPS

Fig. 1. Karma during its maiden flight in November 2001 (left, no on-board instrument were installed at that time), and during test flights in summer 2002 (right, the cameras and the Ethernet antenna are visible).

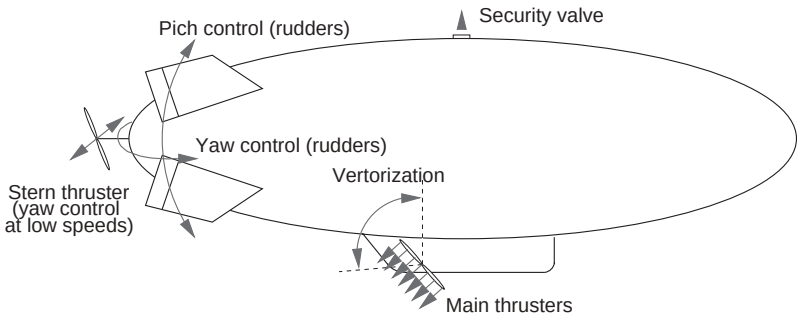

Fig. 2. The various control parameters of Karma.

receiver, a flux-gate compass, a two-axis inclinometer that provides the blimp pitch and roll angles, two solid-state gyros that provide the pitch and yaw rates, and a wind sensor (sonar transducer technology), which measures the speed and orientation of the relative wind in the lateral plane.

CPU. Since our objective is to achieve autonomous missions that include environment perception and mapping, we opted for a Pentium III motherboard, endowed with all the necessary communication ports and a PC 104 slot, on which we added four additional RS232 ports and a PCMCIA interface to host a light 11 Mbit s^{-1} Ethernet modem card.

Actuator control. The control surfaces and motor servos of the blimp are usual PWM controlled modelist devices. To ensure a precise and battery charge independent control, the thrusters are servoed on their speed thanks to a microcontroller. For safety reasons, and to enable a mixed manual/automatic control (e.g., automatic lateral control, while the pilot maintains the altitude), we conceived and developed a "switch" electronic module, that allows us to select between CPU or RC control for each actuator.

The whole hardware architecture is sketched in Figure 3. The total equipment weight is just less than 2.0 kg, and requires 40 W of power.

Energy. Energy is a critical issue for any flying device, mainly for safety considerations. In our case, the few available payloads strongly constrain the battery choices; we opted for lithium/ion batteries for the on-board instruments, as they provide a good power/weight ratio. However, the maximum instantaneous power such batteries can deliver is not sufficient for the thrusters, which are therefore fed by NiMh batteries. Finally, the radio-control receiver and the various servos are independently powered.

Each battery is managed by a electronic module, which allows both the "intelligent" charge of the battery and the dispatching of status information to the CPU via a multiplexed serial link. The charging modules are linked to a single connector, on which a power source is plugged while the blimp is on the ground (which allows booting and debugging without any power loss). This very flexible structure also allows the future use of an alternate or backup power source on flight, such as a Stirling engine of fuel cells.

2.3. Functional Architecture

To allow the integration of the various functionalities, while keeping development and control flexibility, all the flight control and terrain mapping algorithms are nested within the functional level of the LAAS three-layer architecture (Alami et al. 1998). This level is a network of "modules". A module is an active software entity that can encapsulate any kind of algorithm; it is a server that manages all the communications with the others modules, runs the algorithms when required, and paces them using its own threads or processes. The data flow between the various modules is not fixed; it is dynamically controlled by the executive layer of the architecture, depending on the plans provided by the decisional level (Alami et al. 1998).

Figure 4 presents the set of modules that are to be integrated on board Karma. Three modules manage the blimp motions (namely State monitor, Flight control, and Motion planner), while the others achieve absolute positioning and terrain mapping (namely Stereovision, Motion estimation, SLAM and Mapping). Details on the algorithms embedded in these modules are given in the rest of the paper. Note that the position provided by the SLAM module is communicated to the State monitor, to estimate the state with a minimal variance. Also, the future development of an Exploration planner will exploit the results of the Mapping module to determine goals for the Motion planner.

2.4. Current Status

Several test flights have been performed on an airfield near Toulouse during summer 2002, using a preliminary hardware integration of the cameras, compass/inclinometers and GPS, and all the actuators provided by Airspeed Airships. The objectives of these tests were to obtain piloting skills, to evaluate the blimp endurance and ability to cope with wind, and mainly to gather sensor data in order to evaluate the blimp performances and to begin algorithm testing (Extensions 1,2).

Remotely piloting Karma appears to be extremely easy, as long as the mean wind speed does not exceed 10 km h^{-1}. The rudders are very effective, even at low speed; the turning radius is about 15 m at 10 km h^{-1}. We therefore decided to remove the stern thruster, which appeared not really necessary, and whose control was rather difficult. Payload appears to be a critical point; the flight autonomy with the on-board batteries is no longer than 12 min, and there is no payload left for additional batteries or equipment. A second version of Karma is therefore currently being developed; the new envelope will be 3 m^3 bigger, allowing more payload; the gondola is totally redesigned and the original motors are replaced by brushless motors, allowing more thrust power. As the altitude estimate provided by the GPS receiver is not precise enough to safely servo the takeoff and landing phases, we will add a downward oriented sonar telemeter,[1] and we are investigating for 3D wind sensors.

Part A: Flight Control

This part of the paper presents the different flight phases and gives an account of the associated control strategies from

1. Fortunately, the main drawback of sonars in robotics, i.e., their wide perception cone which makes their data interpretation so tedious, will turn into an advantage in our case, as there will be no need to mechanically stabilize it along the vertical.

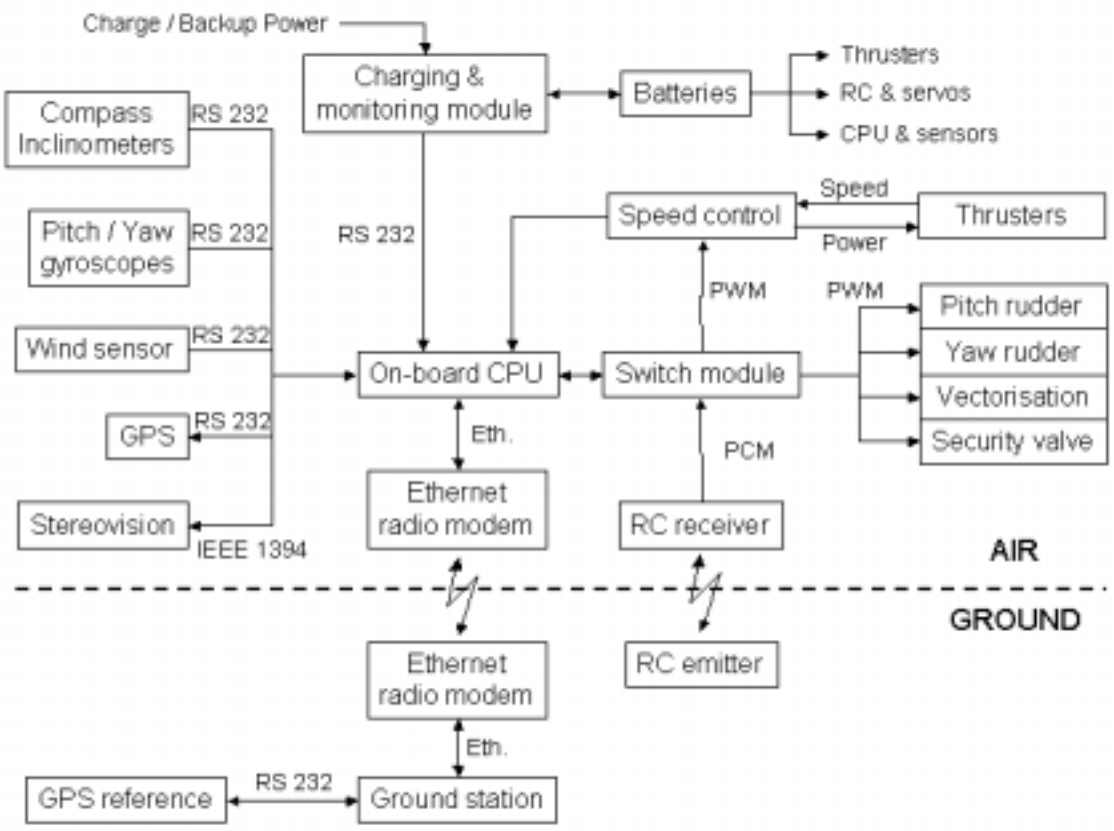

Fig. 3. The hardware architecture of Karma, showing the various formats of information exchanges between the instruments. Note that the GPS reference corrections are transmitted via the radio Ethernet link.

takeoff to landing. The description mainly focuses on the lateral control phase which constitutes the essential part of the steady-state navigation. The flight control architecture is shown in Figure 5. It involves the following three functional modules.

- The motion planner, which contains the trajectory planner and the control planner. For each flight phase, the trajectory planner provides a reference path (expected position of the airship) or a reference trajectory (position, velocity, and acceleration). A corresponding closed-loop control law is then provided by the control planner to regulate the airship's motion along the reference solution. The reference trajectory and the control law are sent to the flight controller.

- The flight controller, which computes the control input to be sent to the airship actuators on the basis of the estimated state variables provided by the state monitoring module. These values are also sent to a virtual airship (see Figure 6) from which a predicted state is determined and sent to the state monitor function.

- The state monitor, which allows us to determine the current estimated state of the system from the predicted state and the measure of the sensory output. This analysis is also used to detect abrupt variations of data that may result from perturbations such as wind gusts, thermals, or model variations (fault detection).

3. Flight Control Strategy

3.1. Complete Model

Three frames are introduced to model the airship's dynamics (see Figure 7): R_0 is a global frame fixed to the Earth, while the body-fixed frame $R(N, X, Y, Z)$ and the aeronautic frame $R_a(N, X_a, Y_a, Z_a)$ are two local frames attached to the airship whose origin is at the nose of the hull N. The point N has been chosen as the origin of these local frames for the following reasons: its position is precisely determined, does not depend on parameter variations, and allows us to model easily the airship rotations with respect to its center of gravity

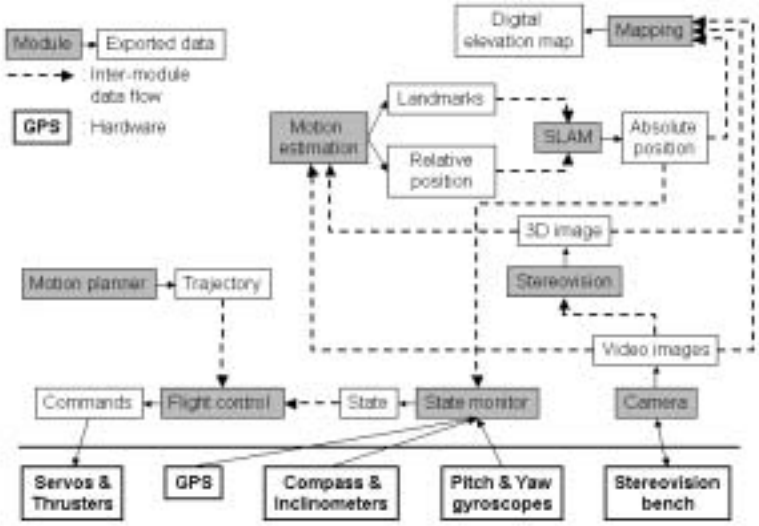

Fig. 4. The functional architecture of Karma.

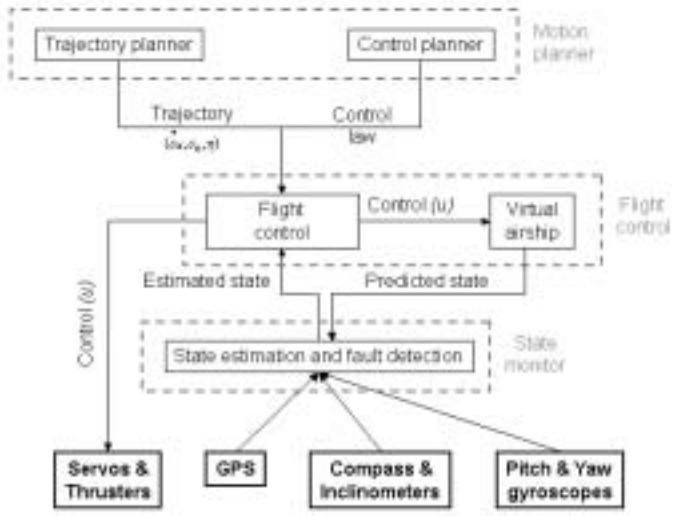

Fig. 5. Detailed architecture of the control modules.

G. Furthermore, aerodynamic torques are usually expressed at this point and more generally, external torques acting on the airship can easily be expressed with respect to this point. The X_a-axis of R_a is directed along the airship's aerodynamic velocity $V_a = V - V_w$, where V and V_w represent respectively the airship's velocity and the wind's velocity with respect to R_0 (note that in case of no wind $V_a = V$).

α is the angle of incidence within the XZ plane, while β is the skid angle within in the XY plane. To describe the airship's orientation with respect to R_0, the three orientation angles are the yaw ψ, the pitch θ, and the roll ϕ. The current configuration is then deduced from three elementary rotations (see Hygounenc, Souères, and Lacroix 2001). The following notations are used: $\eta = [\eta_1, \eta_2]^{\mathrm{T}}$ represents the configuration

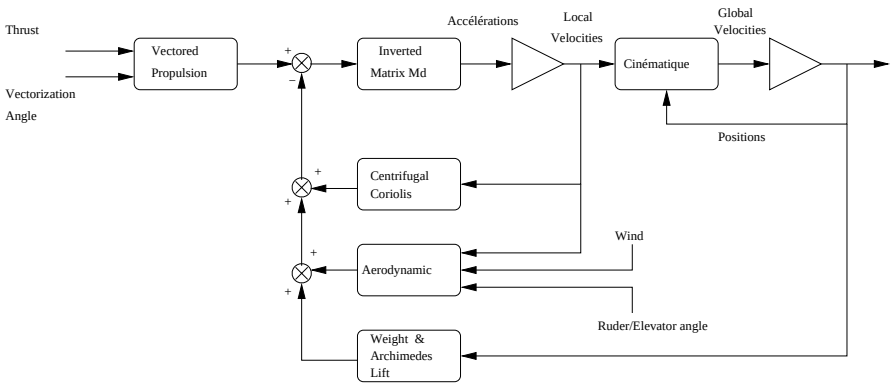

Fig. 6. Virtual airship block design.

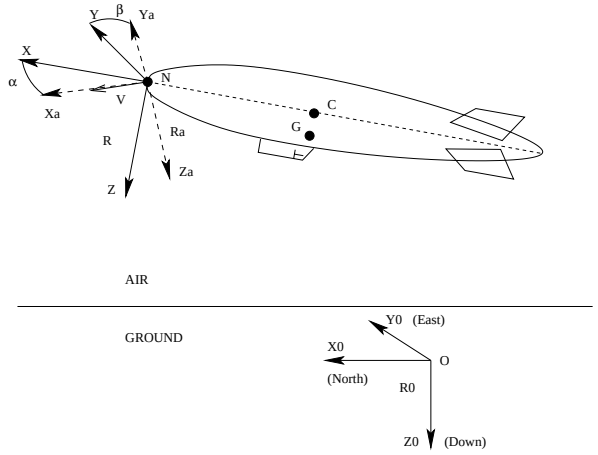

Fig. 7. The different frames.

of R with respect to R_0, where $\eta_1 = [X_0, Y_0, Z_0]^{\mathrm{T}}$ and $\eta_2 = [\phi, \theta, \psi]^{\mathrm{T}}$. $\nu = [\nu_1, \nu_2]^{\mathrm{T}}$ describes the velocity screw of the blimp expressed in the local frame R, where $\nu_1 = [u, v, w]^{\mathrm{T}}$ and $\nu_2 = [p, q, r]^{\mathrm{T}}$.

The dynamic modeling is deduced from the Newton laws of mechanics (Bamberger 1981), while the aerodynamic model (Bonnet 2003) is derived from the Kirshoff laws (Munk 1922) completed with the theory of Bryson (1954). The modeling is based on the following hypotheses.

- The equivalent density of the airship being close to the density of air, the time-varying phenomenon of added fluid induces a variation of inertia and mass that cannot be neglected. (This variation is proportional to the volume of air displaced by the hull.)

- In order to apply the mechanical theory of a rigid body, aeroelastic phenomena are neglected; the hull is considered as a solid.

- The mass of the airship and its volume are considered as constant. This strong hypothesis neglects the variation of mass induced by the inflation of air ballonets, inside the hull, which are consecutive to a variation of temperature or pressure. To reduce this error, the nominal mass is determined for an average temperature of 15°C and a 50% inflation of air ballonets.

- The aerodynamic effects which are due to gravity (modeled by the Froude number) can be decoupled from the dynamics.

- The phenomenon of internal added fluid, which is caused by the motion of molecules of helium inside the hull, is also neglected.

- The center of buoyancy is supposed to be the hull's center of volume.

- The volume of ballonets is supposed to be insufficient

to modify significantly the position of the center of buoyancy.

- As the Mach number is low for an airship, the fluid's viscosity, which depends on the temperature, can be considered constant. As a consequence the Prandtl number (which is dependent on the coupling dynamics/thermics) is neglected, and the density of air is not locally modified by the airship's motion (low Mach $\Rightarrow$ $\rho_\infty = \rho$).

Following the analysis presented in Hygounenc, Souères, and Lacroix (2001) and Lewis, Lipscombe, and Thomasson (1984), the model can be described by two equations. The first characterizes the system dynamics with respect to R, while the second represents the kinematic link between the frames R and R_0:

$$M_d\dot{\nu} + C(\nu)\nu + T_a(\nu_A) + g(\eta) = T_p \tag{1}$$

$$\dot{\eta} = J(\eta)\nu. \tag{2}$$

Here:

- The 6×6 symmetric matrix M_d includes the parameters of mass, inertia with respect to N and the coupling terms (i.e., terms issued from the coupling of translation and rotation).
- $\dot{\nu} = [\dot{\nu}_1, \dot{\nu}_2]$ is the time derivative of the airship's velocity expressed in frame R.
- $T_d = C(\nu)\nu$ is the torque of centrifugal and Coriolis terms, $C(\nu)$ being a skew-symmetric matrix.
- $T_a(\nu_A) = A\dot{\nu}_A + D_1(\nu_2)\nu_A + T_{sta}(\nu_a^2)$ is the torque of aerodynamic forces and moments, where
  - A is the 6×6 symmetric matrix of added masses, inertia at N, and coupling terms of the fluid;
  - $\nu_A = [\nu_a, \nu_2]^{\mathrm{T}}$, where $\nu_a = \nu_1 - \nu_w$ is the vector of aerodynamic translational velocity, with $\nu_w = J_1(\eta_2)^{-1}\vec{V}_w$ the velocity of wind with respect to R_0 expressed in R (note that ν_w is not constant);
  - $T_{da} = D_1(\nu_2)\nu_A$ is the torque of added centrifugal, Coriolis and damping terms of the fluid, where $D_1(\nu_2)$ is a matrix which is only dependent on the rotational velocity rotation ν_2;
  - $T_{sta}(\nu_a^2)$ is the torque of stationary forces and moments at N, which is proportional to the square of the aerodynamic velocity (Thwaites 1960). This torque contains the forces and moments produced by the control surface. Each pair of diagonally opposed control surfaces is simultaneously actuated. The resulting force increases linearly with the deflection angle. They give rise to a rotational moment within the longitudinal or the lateral plane.
- $g(\eta)$ is the torque of weight and buoyancy.
- T_p is the torque of the vectored thrust which is produced by two synchronized propellers fixed on a rotational axis. The norm and the direction μ of this thrust are adjustable within the longitudinal plane. This actuator is used as a upwards lift during takeoff and as an horizontal thruster during the lateral steady flight.
- $J(\eta)$ is the transition matrix from R to R_0.

In the nominal case, when the external wind is weak or null, the following approximation holds: $\nu_a = \nu_1$, that is $\nu_A = \nu$. Under this condition, the dynamical model can be written

$$M\dot{\nu} + C(\nu)(\nu) + T_a(\nu) + g(\eta) = T_p$$

where $M = M_d + A$ is the matrix of inertia due to both the mass of the airship and the added mass of air at N, expressed in R.

3.2. Decoupled Model for Control

Although the system described by eqs. (1) and (2) models rather precisely the complete blimp's dynamics, it is hardly tractable for control. As the dynamics of the state parameters involved in longitudinal and lateral motions turn out to be weakly dependent, the 12 state variables can be split into two subsystems in the following way:

- $\eta_{long} = [X_0, Z_0, \theta]^{\mathrm{T}}$ and $\nu_{long} = [u, w, q]^{\mathrm{T}}$ to describe the dynamics within the longitudinal plane;
- $\eta_{lat} = [Y_0, \phi, \psi]^{\mathrm{T}}$ and $\nu_{lat} = [v, p, r]^{\mathrm{T}}$ to describe the dynamics within the lateral plane.

REMARKS.

- Although the state variable X_0 and u are common to both planes, they do not appear explicitly in the lateral dynamic as, according to the proposed control strategy, they are supposed to be stabilized to a steady value during the lateral flight.
- The rolling dynamics does not appear in the proposed control submodels. The reason for this choice is that the corresponding mode is structurally stable and not controllable. Indeed, as explained in the description above, the pair of opposed control surfaces being simultaneously actuated cannot induce a rolling torque. Such a torque can neither be produced by the vectored thrust, which always provides the same power to both propeller. The simulation results presented in the last section (which have been done by considering the complete model) show that an effective rolling motion occurs during lateral navigation but that the corresponding dynamics is stable, sufficiently fast, and quite well damped.

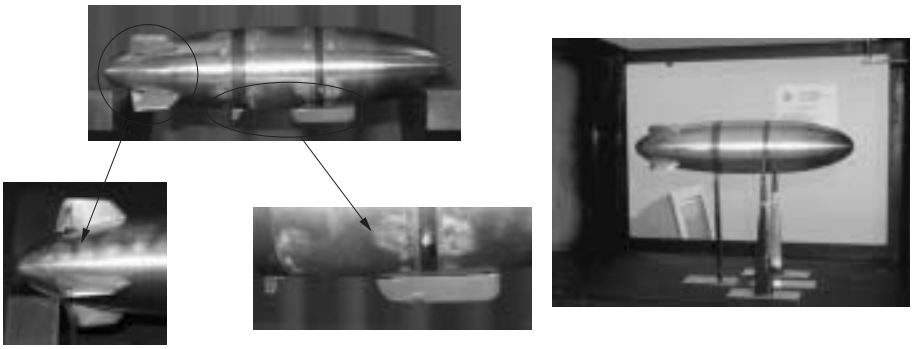

Fig. 8. Scale model of Karma in a wind tunnel.

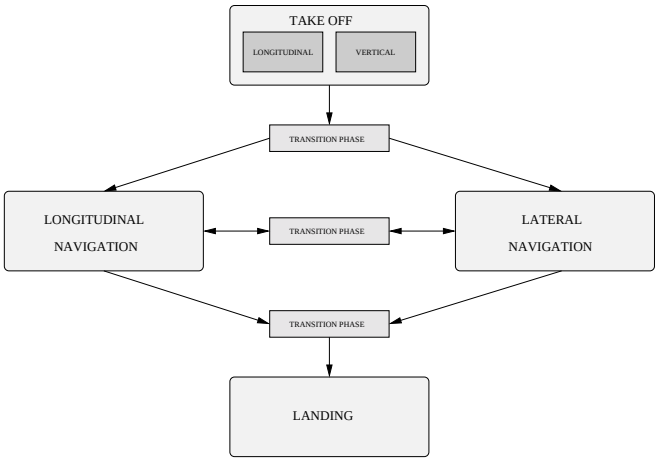

Fig. 9. The different flight phases.

The proposed navigation strategy consist of a separate longitudinal and lateral control on the basis of the submodel described above. During each phase, a part of the state variables needs to be stabilized to a nominal value while the remaining part is controlled. This requires us to consider transition phases between each nominal phase.

The different flight phases and the transition between them are defined according to the scheme of Figure 9. Four flight phases are to be considered, takeoff, longitudinal navigation, lateral navigation and landing.

3.3. Global Control Strategy

In this section we present a short description of the different control phases. The steady lateral control which constitutes the main part of the navigation process is then detailed in Section 4.

3.3.1. Takeoff

Two control strategies are proposed to perform the takeoff phase. The first aims at controlling the airship longitudinally in the same way as a plane, while the second involves first a vertical motion. In both cases, as the lateral motion is not controlled during this phase, the airship must be directed so as to face the wind.

Longitudinal takeoff. In this approach the vectored thruster is initially directed horizontally and the motors are supplied with maximum power. Initially maintained at a non-zero altitude, the action of elevator becomes rapidly efficient to produce the pitching torque which is necessary to takeoff. A simultaneous control of speed and altitude is then performed to control the rising motion.

Vertical takeoff. Another way to perform the takeoff phase is to apply the following three-step procedure. First, the thrust is maintained almost vertically until the blimp has reached a reference altitude and longitudinal position. An adaptive nonlinear backstepping-based controller is used to this end. Secondly, a trajectory tracking involving a similar controller is used to drive longitudinally the airship up to a threshold velocity. Finally, the velocity and the altitude are simultaneously controlled to perform the rising motion.

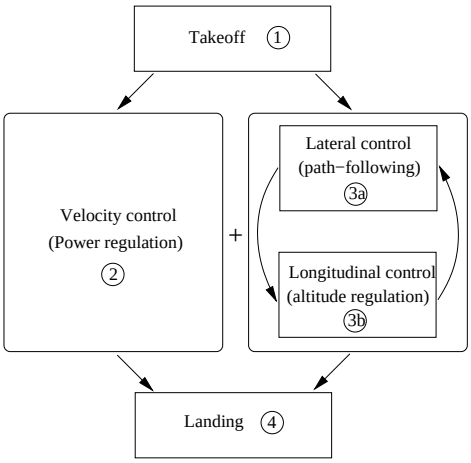

Fig. 10. The flight control diagram.

3.3.2. Longitudinal and Lateral Navigation

According to the previous description, the last step of the takeoff phase allows us to drive the airship to a reference altitude and velocity. Stabilizing the longitudinal dynamics to a steady value constitutes a necessary condition for starting the lateral control. Once this equilibrium has been reached, the navigation is driven by two parallel regulations. A first control loop is used to regulate the velocity (block 2 in Figure 10) with respect to the Earth, in order to make the control matrix associated with the mobile surfaces stationary. Another loop is used to control either the lateral motion by means of a path-following procedure (block 3a in Figure 10), or the altitude on the base of a backstepping regulation of the elevators angle (Hygounenc and Souères 2002). While the velocity regulation is maintained during the whole steady navigation, the transition between lateral and longitudinal control is determined by the motion planner. A necessary condition for the control switch to be possible is that the corresponding subsystem be stabilized.

3.3.3. Landing

As for the takeoff, the blimp needs to face the wind during the landing procedure. The reason for this constraint is that the landing procedure aims at regulating the longitudinal dynamics only. Using the longitudinal controller 3b, the altitude is reduced down to a security value, while the airship is driven to a target position. The last step is performed either by stopping the engine or even reversing the thrust direction for a while to stop the velocity.

This paper focuses on the steady lateral navigation. The description of the controllers that allow us to perform this flight is detailed in the next section.

4. Steady Lateral Navigation

In order to elaborate a complete autonomous navigation strategy, the first objective is to guarantee the stability of the steady-state motion within the lateral plane, and the robustness of the controller with respect to wind perturbations. The closed-loop velocity regulation and the path-following process, which are to be performed in parallel during this phase, are detailed in the next section. Simulation results are given in the last part.

4.1. Velocity Regulation

Once the reference altitude has been reached the airship's velocity is stabilized in order to start the lateral steady navigation.

4.1.1. Embedded Control Structure

The velocity regulation is obtained by means of the embedded control structure described in Figure 11. The system input is the reference value u_e of the longitudinal component of the airship's velocity, i.e., the component of the velocity along the X-axis of frame R. Indeed, once the vectored thrust angle μ is set to zero, the sole component of the velocity which is controllable is u. As the remaining two components v and w are very small with respect to u, the airship velocity can be approximated by its longitudinal component: $u \approx V$. From the difference between the current and the expected value of u the first control block is a velocity controller which determines the reference thrust T to be provided by the propellers. On this base, a converter computes the corresponding power which has to be sent to the motors. A PI controller is then used to servo the motor. Finally, the current value of the longitudinal component of the airship's velocity is deduced from the differential GPS measures and sent to the velocity controller, while the aerodynamic velocity measured by the anemometer is sent to the converter. The successive blocks are presented in more detail in the following.

4.1.2. Velocity Controller

By considering only the tangential effort from the previous model, making the hypotheses that the aeronautic angles are small and that the propulsion is horizontal ($\mu = 0$), the dynamics of the longitudinal velocity component $u \geq 0$ can be expressed by the following differential equation:

$$m_x \dot{u} = -F_T u^2 + T \tag{3}$$

where m_x is the inertia due to both the blimp's mass and the added mass of air along the X-axis, $F_T u^2$ is the normalized drag force, and T is the propeller thrust which is considered here as the input. Introducing the notation, $a = F_T/m_x$,

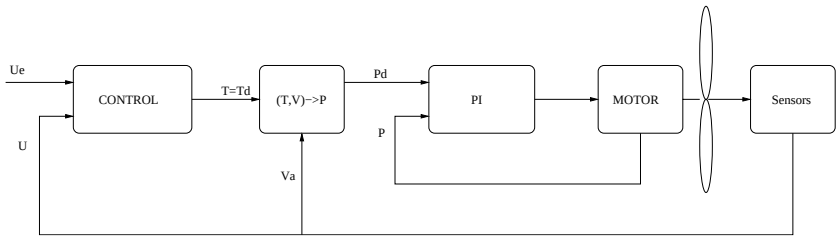

Fig. 11. The embedded velocity controller.

$b = 1/m_x$, the following simpler expression of system 3 is considered:

$$\dot{u} = -au^2 + bT.$$

Denoting by $\epsilon = u - u_e$ the velocity error, the following strictly positive storage function is introduced:

$$V = \frac{1}{2}\epsilon^2.$$

As the reference velocity u_e is constant, the time derivative of V is

$$\dot{V} = \epsilon(-au^2 + bT).$$

In order to obtain a definite negative expression of the form $\dot{V} = -cz^2$, the following choice is made for the controller (Kokotovic, Krstic, and Kanellakopoulos 1995):

$$\mathbf{T} = \mathbf{b}^{-1}(-\mathbf{c}\epsilon + \mathbf{au}^2).$$

Figure 12 represents a result of simulation of this controller. The control objective was to stabilize the system to the cruising speed of 5 m s^{-1}. Despite the simulation of a strong frontal wind gust (3 m s^{-1}), only a slight variation of the velocity occurs, showing the robustness of the controller to external perturbation.

4.1.3. Thrust Converter

The propeller is characterized by its advance number J, the thrust coefficient C_T, and the power coefficient C_P. Two equations are usually used to describe the propeller's model. The first expresses the thrust T, while the second describes the absorbed power P:

$$T = \rho D^4 C_T(J) n^2 \qquad n \geq 0 \tag{4}$$

$$P = \rho D^5 C_P(J) n^3. \tag{5}$$

In these relations $J = \frac{V_a}{nD}$, ρ is the air density, D is the diameter of the propeller, $n = \frac{\Omega}{2\pi}$ is the number of rotations per second (rps), Ω is the propeller's rotation velocity, and V_a is the advance speed at the propeller, which is equal to the airship's aerodynamics velocity. The thrust coefficient C_T and the power coefficient C_p, which depend on the advance parameter, can be written under the following form (see Fossen 1996):

$$C_T = \alpha_{1T} + \alpha_{2T} J \tag{6}$$

$$C_P = \alpha_{1P} + \alpha_{2P} J + \alpha_{3P} J^2. \tag{7}$$

Karma's propellers have been chosen on the basis of a collaboration with the Laboratory of Aerodynamics and Propulsion of the Ecole Nationale Superieure de l'Aéronautique et de l'Espace (ENSAE), Toulouse France (http://www.supaero.fr). They are made of plastic and the type is 12 × 8 "Master Airscrew". The curves represented in Figure 13 describe experimental tests of traction and absorbed power, for various values of the motor's rotation velocity, and for different values of the airship's velocity. The model equations (4) and (5) have been identified thanks to relations (6) and (7), on the basis of a polynomial approximation of these curves:

$$T = T_1 n^2 + T_2 n V_a \tag{8}$$

$$P = P_1 n^3 + P_2 n^2 V_a + P_3 n V_a^2. \tag{9}$$

As the propeller thrust cannot be directly observed, this quantity has to be converted into another one directly measurable. Two output functions can be measured to this end: the rotation velocity of the motors and the power delivered by the batteries. For practical reasons, the successful solution was to servo the motor on the basis of the power delivered by the batteries. From the desired thrust T provided by the velocity controller, the reference power P is determined by the converter as follows. The expected rotation velocity n is first computed from eq. (8) by replacing T and V_a by their current values, and P is then deduced from eq. (9). This value must then be augmented in order to compensate the power dissipated by the system which has been experimentally evaluated to 40%.

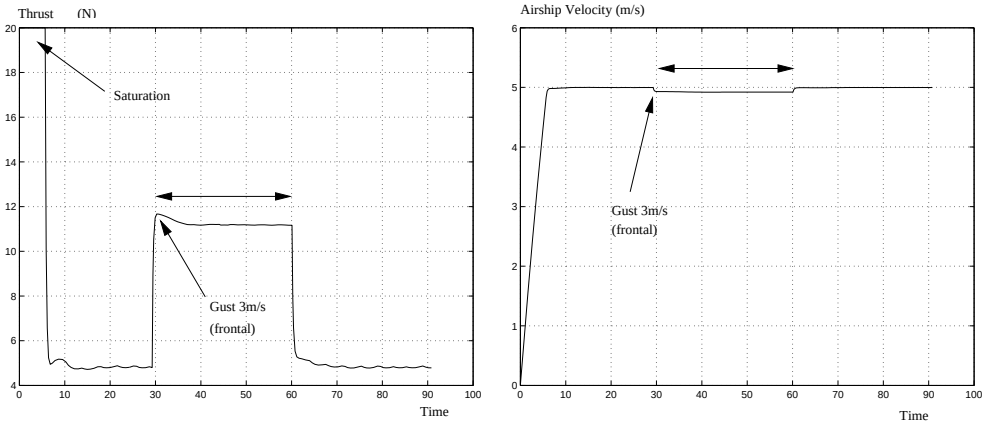

Fig. 12. Velocity stabilization with frontal gust.

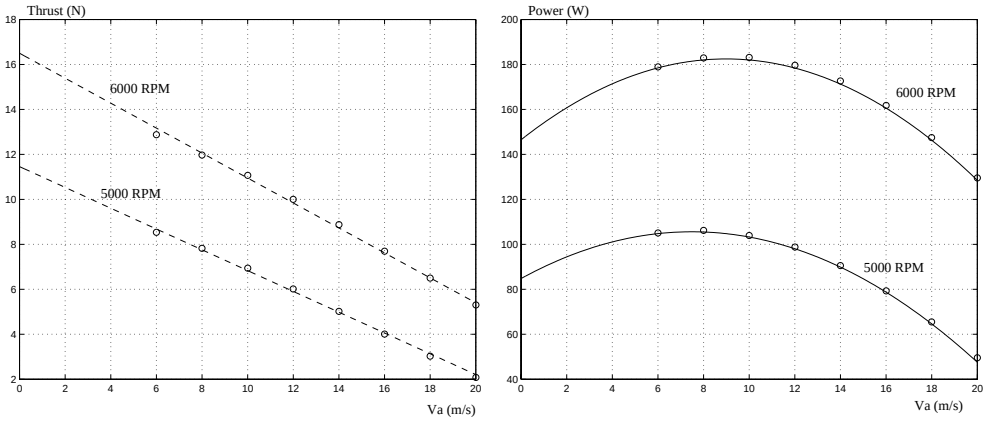

Fig. 13. Thrust T and absorbed power P versus aerodynamic velocity V_a for 5000 and 6000 rpm.

4.1.4. Power Control

The propellers are driven by two brushless motors equipped with a variator. The variator input is a PWM signal which corresponds to a value ranging from 0 to 1023 for the microcontroller. Figure 14 represents the power output and the intensity output to a step, for various values of the tension. A residual noise in steady state and a strong overshoot can be observed on both curves. As this overshoot does not correspond to a rotation velocity increase it has to be filtered. To this end, a first-order filter has been used. A discretized expression of this filter has been considered to a sampling period of 50 Hz (see Figure 15):

$$\frac{bZ^{-1}}{1+aZ^{-1}}. \tag{10}$$

In order to minimize the output error the control loop includes an integral action. The discrete expression of the controller is then of the type:

$$\left(K_p + \frac{K_i}{1-Z^{-1}}\right)\delta P$$

where K_p and K_i are the proportional and integral gains, and δP is the power error. The control diagram is represented in Figure 16, which shows control experiments in which the power is successively stabilized to follow a reference input step of 50 and 100 W.

4.2. Path Following

In this section we present the path-following control loop which is to be executed in parallel to the velocity regulation described in the previous section (Hygounenc and Souères 2003). Making the hypothesis that the longitudinal component of the airship velocity is stabilized to a reference value

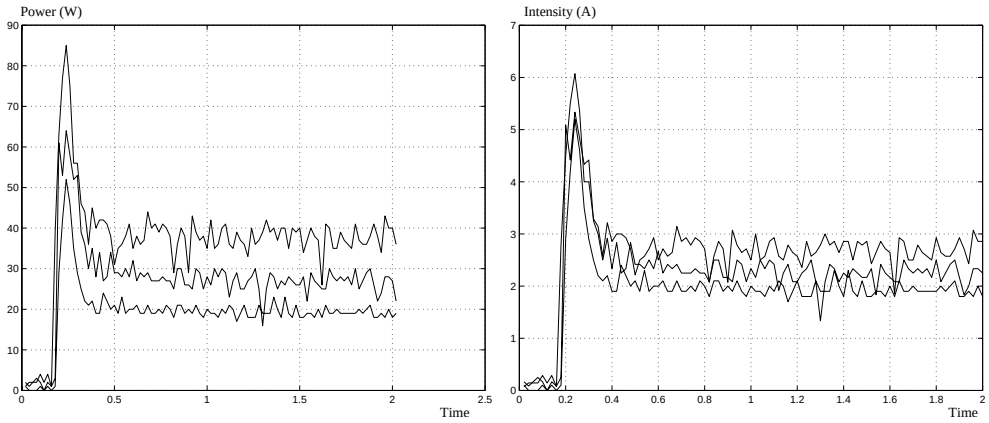

Fig. 14. Power and intensity output for various values of the tension.

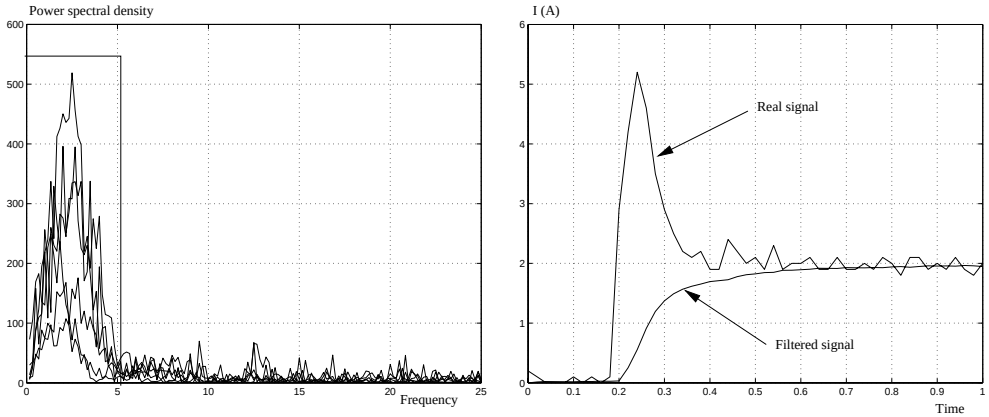

Fig. 15. Power spectral density and filtered signal.

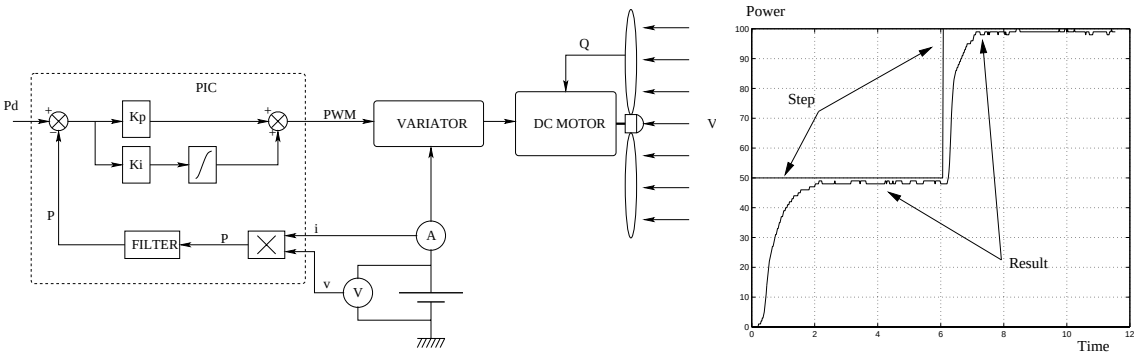

Fig. 16. Power control scheme and experiment.

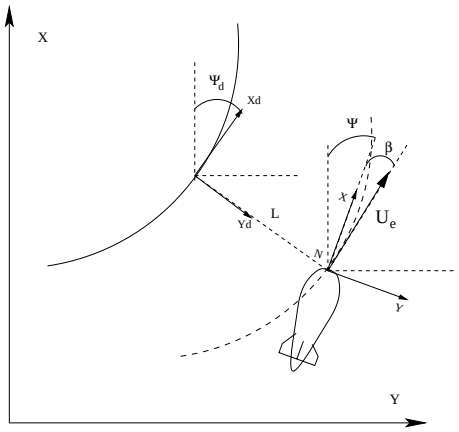

Fig. 17. Path following.

u_e, the lateral blimp's dynamics is described by the system:

$$\begin{aligned}
\dot{X}_0 &= u\cos\psi - v\sin\psi = u_e\cos(\psi+\beta)\\
\dot{Y}_0 &= u\sin\psi + v\cos\psi = u_e\sin(\psi+\beta)\\
\dot{\psi} &= r\\
\dot{\beta} &= a_{11}\beta + \tfrac{a_{12}}{u_e}r + \tfrac{b_1}{u_e}\delta_r\\
\dot{r} &= a_{21}u_e\beta + a_{22}r + b_2\delta_r
\end{aligned}$$

where the a_{ij} coefficients $i, j = 1, 2$ are constant.

As the control objective is to regulate the blimp's motion within the lateral plane while its velocity u_e is constant, a path-following strategy appears to be very appropriate. As we consider a planar motion, the problem can be formulated as the regulation of the lateral distance and the orientation error with respect to a mobile Frenet frame whose orientation is defined by projecting perpendicularly the point N on the path (see Figure 17). However, contrary to what is done for mobile robots (Canudas de Wit et al. 1993), the frame to be stabilized along the reference path is the aeronautic frame R_a instead of local frame R. The reason for this difference is that the blimp is not constraint to move tangentially to its main axis as non-holonomic robots. Indeed, due to the lateral slippage, the blimp velocity is directed along the X_a-axis which differs from the local X-axis by the skid angle β. Let L be the lateral distance between N and the path, and $\tilde{\psi}$ the angular error between the blimp velocity u_e and the mobile frame X_d-axis whose orientation is given by ψ_d. The error dynamics reduces to

$$\begin{aligned}
\dot{L} &= u_e\sin\tilde{\psi}\\
\dot{\tilde{\psi}} &= \dot{\psi} + \dot{\beta} - \dot{\psi}_d.
\end{aligned}$$

Under the hypothesis that the angular error $\tilde{\psi}$ remains small, a first-order approximation can be considered: $\sin\tilde{\psi} \approx \tilde{\psi}$. Finally, the lateral path following including the skid dynamics can be represented by the following fourth-order linear system:

$$\begin{pmatrix} \dot{L}\\ \dot{\tilde{\psi}}\\ \dot{\beta}\\ \dot{r}\end{pmatrix} = \begin{pmatrix} 0 & u_e & 0 & 0\\ 0 & 0 & a_{11} & \frac{u_e+a_{12}}{u_e}\\ 0 & 0 & a_{11} & \frac{a_{12}}{u_e}\\ 0 & 0 & a_{21}u_e & a_{22}\end{pmatrix}\begin{pmatrix} L\\ \tilde{\psi}\\ \beta\\ r\end{pmatrix}$$

$$+\begin{pmatrix} 0\\ \frac{b_1}{u_e}\\ \frac{b_1}{u_e}\\ b_2\end{pmatrix}\delta_r + \begin{pmatrix} 0\\ -1\\ 0\\ 0\end{pmatrix}\dot{\psi}_d.$$

Usually the reference path is determined by a sequence of passing through points. Using the on-board GPS, the distance L to the path and the orientation error with respect to it can be well approximated. However, the term $\dot{\psi}_d$, which is proportional to the instantaneous path curvature, appears to be less easy to determine. As the reference path curvature is supposed to be upper bounded, a practical way to solve this problem is to consider $\dot{\psi}_d$ as a piecewise constant perturbation. Defining L as the system output and introducing $C = [1\ 0\ 0\ 0]$ and $G = [0, -1, 0, 0]^{\mathrm{T}}$, the path-following error dynamics can be written under the classical form:

$$\begin{aligned}
\dot{X} &= AX + B\delta_r + G\dot{\psi}_d\\
Y &= C\ X.
\end{aligned} \tag{11}$$

From the previous model, as $\dot{L} = u_e\tilde{\psi}$ and $u_e \neq 0$, a sufficient condition to stabilize $\tilde{\psi}$ to zero is to stabilize the lateral distance L to zero. The control objective can then be specified in terms of stabilizing the perturbed system (11), while insuring a zero output error. To this end, a stabilizing state feedback with integral control is applied.

Introducing the additional state variable

$$x_0(t) = \int_0^t Y(\tau)\mathrm{d}\tau.$$

A fifth-order augmented linear system is defined as follows:

$$\begin{pmatrix} \dot{x}_0\\ \dot{X}\end{pmatrix} = \begin{pmatrix} 0 & C\\ 0_{4*1} & A\end{pmatrix}\begin{pmatrix} x_0\\ X\end{pmatrix} + \begin{pmatrix} 0\\ B\end{pmatrix}\delta_r + \begin{pmatrix} 0\\ G\end{pmatrix}\dot{\psi}_d.$$

Let us denote by $\tilde{X} = (x_0, X)^{\mathrm{T}}$ the augmented state, and by $\tilde{A}$, $\tilde{B}$, $\tilde{G}$, $\tilde{E}$ the augmented matrices. Using a stabilizing controller $u = -\tilde{K}\tilde{X} = -k_0x_0 - KX$, the closed-loop system can be stabilized while insuring a zero output error. Indeed, as $\dot{\tilde{X}}(t) = (\tilde{A} - \tilde{B}\ \tilde{K})\tilde{X} + \tilde{G}\dot{\psi}_d + \tilde{E}e$ converges asymptotically to zero, $\dot{x}_0(t) = L$ converges to zero as well. The controller is presented in the block diagram 18.

REMARK 1. The path-following controller allows us to stabilize asymptotically the aeronautic frame X_a-axis along the

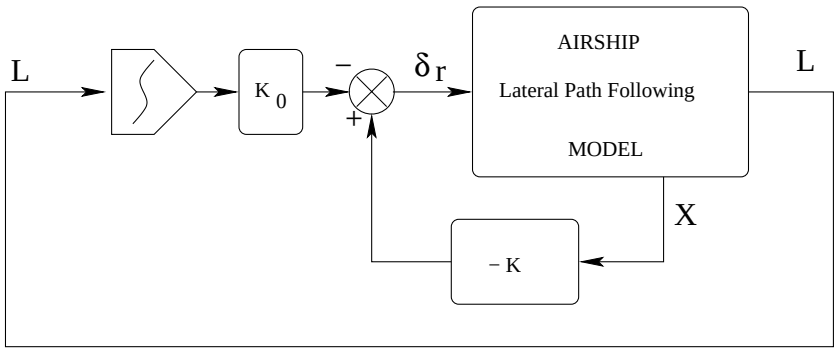

Fig. 18. State feedback with integral control.

reference path. The body-fixed frame R, which is tangent to the blimp's main axis, will then make a skid angle β with the reference frame. If the reference path is a circle, the skid angle β is then stabilized to a constant value. This illustrates the necessary condition for the blimp's dynamics to include a constant lateral slippage to follow a circle. If the reference path becomes rectilinear, β converges to zero and the blimp's orientation changes progressively until being tangent to the path.

4.3. Simulations

4.3.1. Introduction

A flight simulator involving the complete mathematical model of the airship has been developed by using the Matlab/Simulink software. It allows us to test the different controllers by taking into account the whole dynamic and aerodynamic forces and torques which have been identified. Before starting the simulation, the following configuration parameters are to be set:

- the equilibrium parameters (relative position of the center of gravity);
- the payload (lighter than air, heavier or in aerostatic equilibrium);
- the limits on the actuator's dynamics.

The environment can also be configured:

- density of air (which directly modifies the aerodynamic effects and the buoyancy);
- the external wind parameters: velocity and orientation angles (these parameters are then expressed with respect to the body-fixed frame).

Two simulations of the steady lateral control of Karma are presented. As explained in the previous sections, the lateral control involves the simultaneous regulation of the velocity and the path-following process. The modulus of the reference velocity u_e with respect to the Earth frame has been set to 5 m s^{-1}.

Table 1. Airship Configuration

Equilibrium	Null pitch angle at rest
Payload	Heavier than air +0.4 kg
Maximal thrust	20 N
Maximal mobile parts angle	|45|°

Table 2. Environment Configuration

	Simulation A	Simulation B
Air density ρ	1.225 kg m^3	1.225 kg m^3
External wind	Null	Front, horizontal 2 m s^{-1}

The path-planning procedure is based on the definition of a sequence of reference geometric points $Pt_i(X_i, Y_i)$ within the lateral plane. The reference trajectory is then defined as a broken line by considering successive pairs of points. The lateral error L is deduced from the measure of the current position of the airship by means of the differential GPS or by using the stereovision. The angular error $\tilde{\psi}$ is deduced from the absolute orientation angle given by the compass and the skid angle given by the anemometer. Note that the trajectory can be progressively constructed by defining additional pairs of points during the motion. Although this planning process aims at being directed by the motion planner it has been performed manually for the simulations.

For both simulations the takeoff phase has also been performed manually. Once a security altitude is reached the

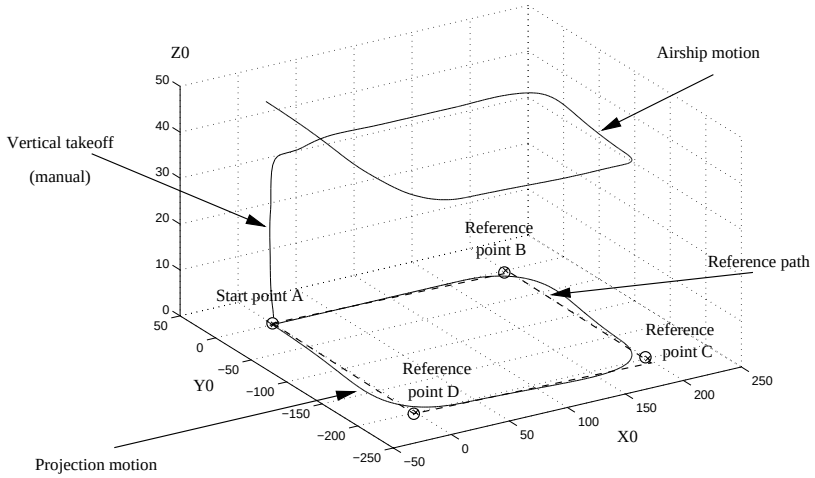

Fig. 19. 3D trajectory without wind.

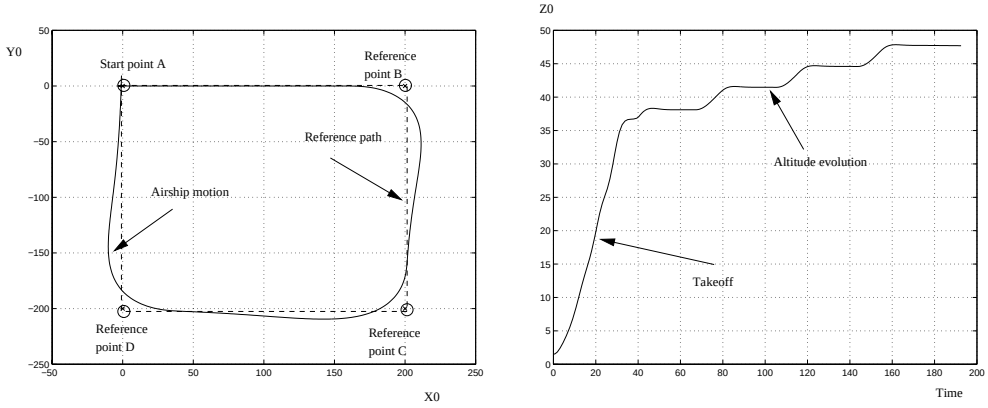

Fig. 20. Lateral motion and altitude variation.

vectored thrust angle is turned to zero (horizontal propulsion). The velocity controller is activated until the airship has reached the cruising speed. At this stage the first pair of points is defined and the autonomous lateral control is initiated. The control objective was to follow four successive line segments to execute a square trajectory. The first simulation is performed for the nominal system, that is without wind perturbations, while the second is performed with a constant horizontal wind to test the control robustness.

4.3.2. Simulation A: Navigation Without Wind

Figure 19 describes the 3D trajectory of the airship while the projection of the motion in the lateral plane and the time variation of the altitude are described in Figure 20. The airship follows closely the reference trajectory segments and remains close to the reference points. This result is confirmed by Figure 24 in which the variation of the lateral position error L and of the orientation error $\tilde{\psi}$ are described. Three peaks correspond to the change of orientation.

In this simulation, the airship's altitude is not regulated. The altitude increases slightly (less than 10 m for 800 m). The variation of the altitude is dependent on the airship configuration parameters. In the case of a heavier-than-air airship, with a negative pitch angle at the equilibrium, the altitude would have decreased.

Figure 21 shows the evolution of the airship velocity components with respect to the local frame. As expected, the longitudinal component u is correctly stabilized to the reference value of 5 m s^{-1}. Furthermore, the longitudinal component appears to be the essential part of the velocity as predicted ($u \gg v$ and $u \gg w$). The curve of the propeller's thrust represented in Figure 23 shows the importance of the velocity regulator. Each time the orientation changes, the rudder's

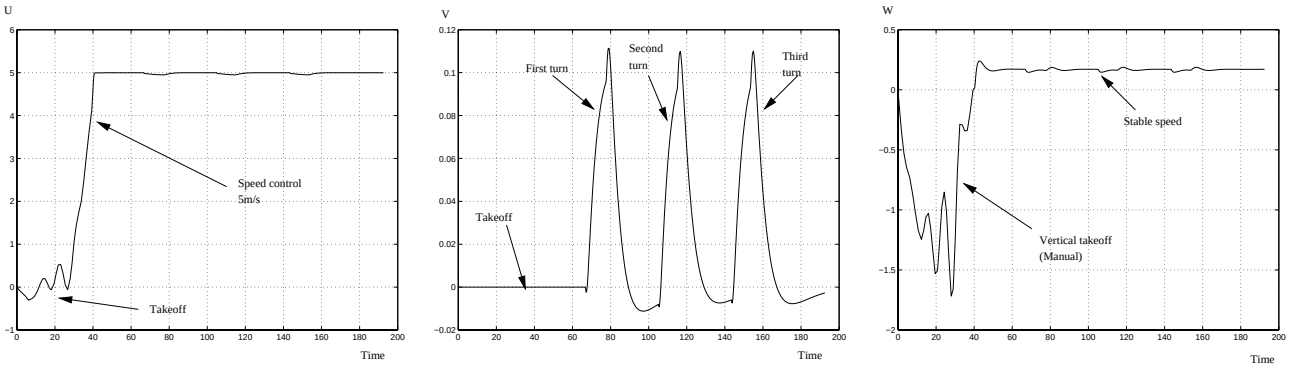

Fig. 21. Velocity components with respect to the local frame.

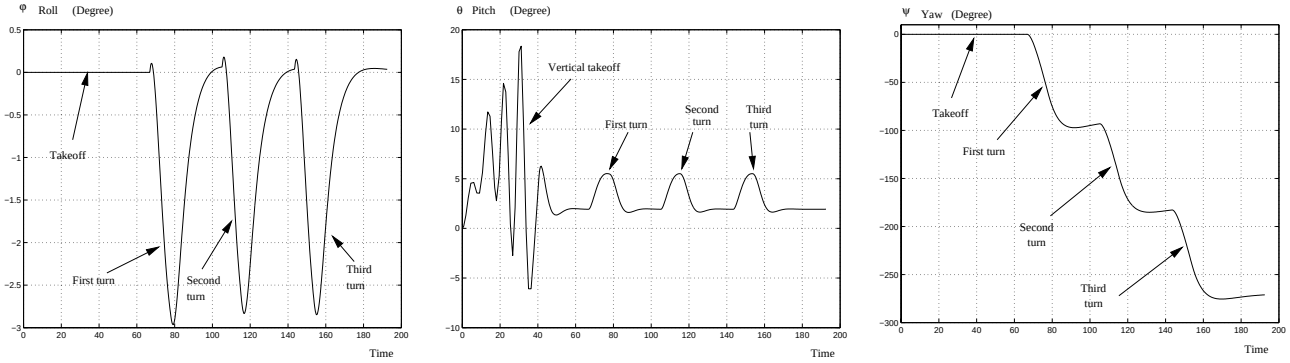

Fig. 22. Airship attitude.

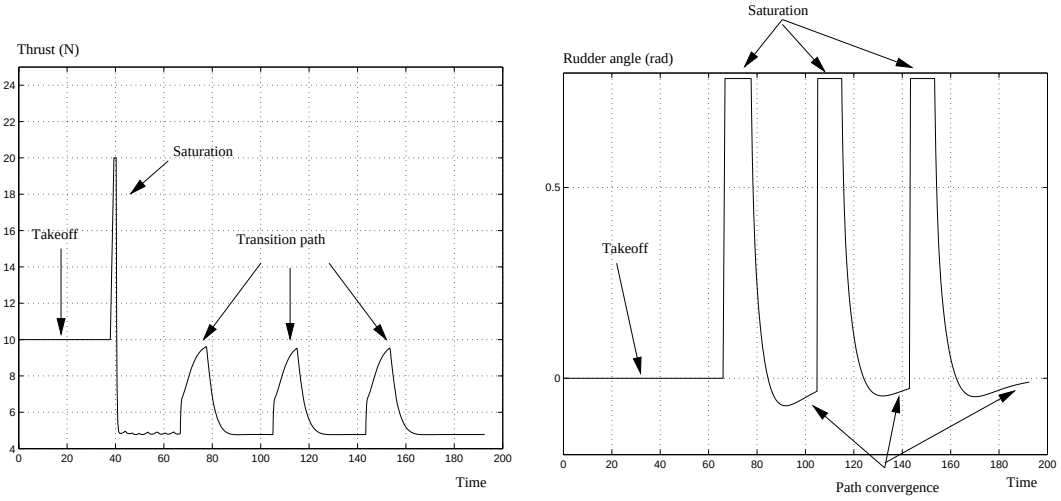

Fig. 23. Actuator activity.

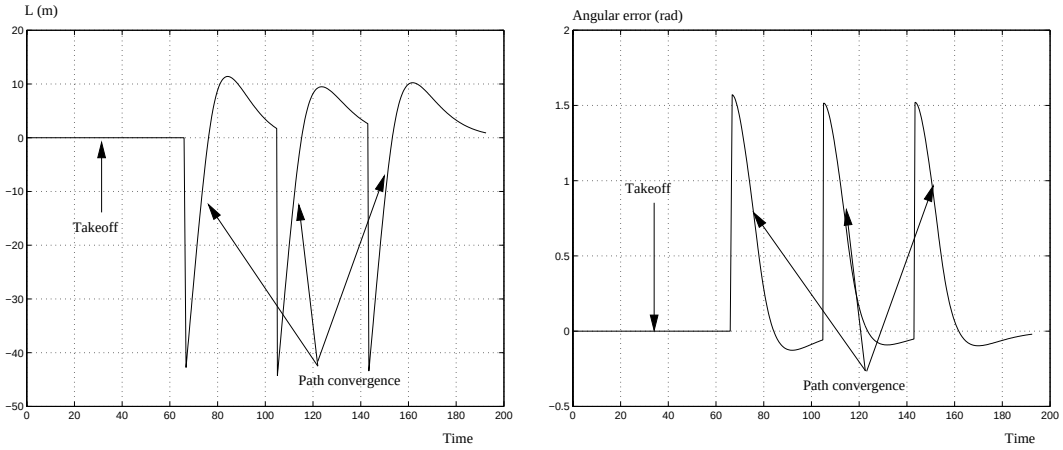

Fig. 24. Lateral and angular error evolution.

angle saturates inducing a strong increase of the tangential effort. To compensate for the phenomenon and to avoid the reduction of the velocity, the controller increases the propeller thrust. Three peaks corresponding to the saturation of the rudder-angle saturation appear on the thrust curve in Figure 23.

Figure 22 represents the evolution of the attitude angles. The variation of the roll angle ϕ justifies the choice to consider it as a well-damped stable variable. The gondola fixed under the airship's hull acts as a damped pendulum. Furthermore, the amplitude of oscillations remains very low.

The oscillations of the pitch angle θ that can be observed have been generated during the vertical takeoff phase which has been performed manually. The following three peaks result from the increase of the pitching moment which is induced by the thrust. The evolution of the yaw angle ψ is quite regular despite the three successive changes corresponding to the switch of reference segments.

This result shows the validity of the approach which consists of regulating the lateral motion while considering the roll angle as a stable well-damped state variable. On the other hand, the stabilization of the longitudinal component of the velocity allows us to regulate the altitude quite well. Following the control scheme of Figure 10 the altitude has to be adjusted from time to time by using the longitudinal controller. This switch between lateral and longitudinal control is determined by the motion planner.

4.3.3. Simulation B: Navigation With Lateral Wind

The same flight objective has been simulated in the presence of an external lateral wind of 2 m s^{-1}, which according to experiments constitutes a strong perturbation. The objective was to demonstrate the control robustness with respect to external perturbations.

As before, the takeoff phase has been performed manually, but this time by facing the wind. The reference points have been determined following the same procedure as for the first experiment.

Figure 25 shows the airship's 3D trajectory. The trajectory projection within the lateral plane and the variation of altitude are described in Figure 26. As shown, the trajectory passes a little distance away from points B and C. This error is mainly due to the delay of definition of these points. The altitude's variation differs from the previous experiment (Figure 20). This is due to the fact that, although the velocity with respect to the Earth is regulated, the aerodynamic velocity varies. When the airship is facing the wind V_a increases by 2 m s^{-1}. The downwards pitching torque induced by the gondola is greater than the upwards pitching torque induced by the vectored thrust. As the pitch angle is lower than the incidence angle α, the airship moves downwards. Note that, despite this pitching phenomenon, the altitude variation remains quite moderate, showing the feasibility of the method.

Figure 27 represents the evolution of the components of the velocity with respect to the Earth expressed in the body-fixed frame. As before, the longitudinal component constitutes the essential part of the velocity. The maximal value of the lateral component v is 2 m s^{-1}. Note that for the highest values of v a static error of about 0.4 m s^{-1} occurs in the regulation of u. As shown in Figure 29, these variations induce a thrust increase from 2 to 14 N. Note that the initial thrust saturation corresponds to the acceleration phase, allowing us to reach the cruising speed.

Figure 28 shows the evolution of the airship's attitude. As before, the roll angle behaves as a stable well-damped mode. The pitch angle θ appears to be strongly perturbed. This is due to the presence of the strong lateral wind which acts differently on the airship during the motion. Following the vertical

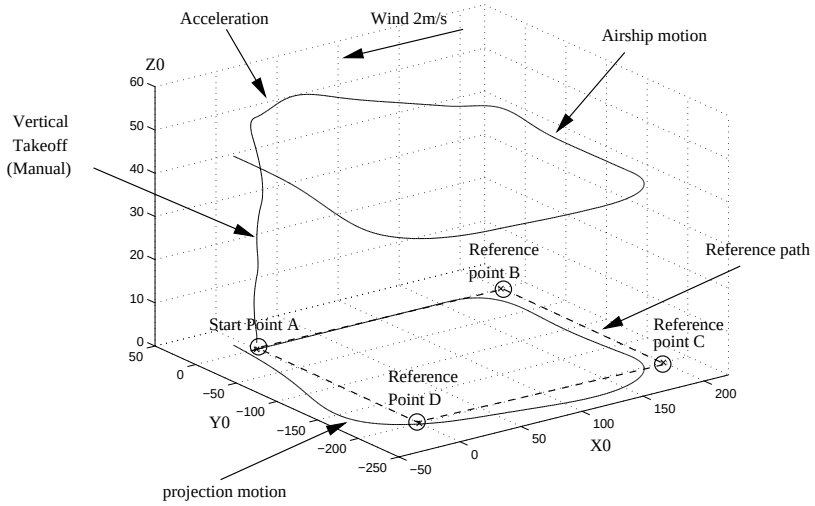

Fig. 25. 3D navigation with lateral wind.

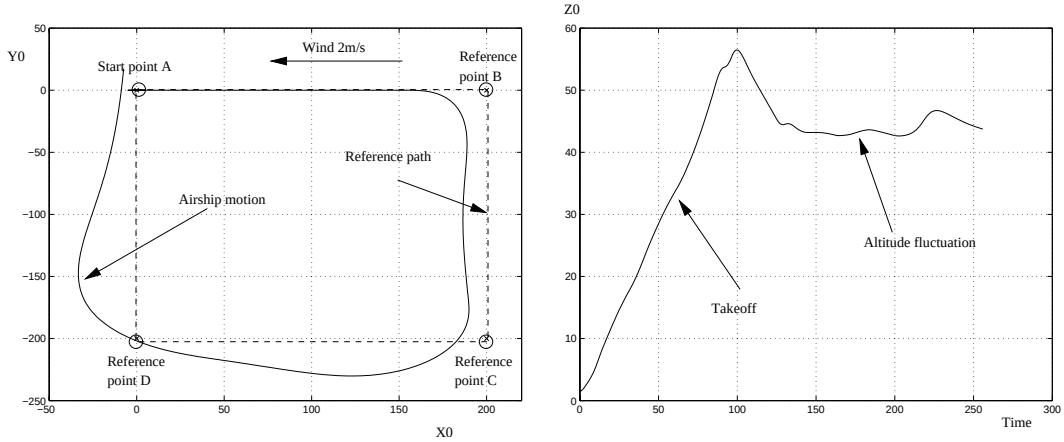

Fig. 26. Lateral motion and time evolution of the altitude.

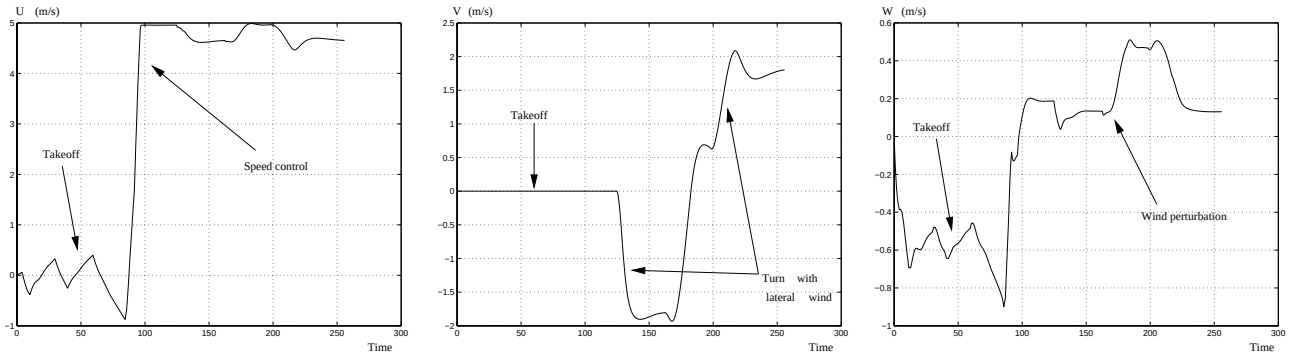

Fig. 27. Evolution of the velocity components in the body-fixed frame.

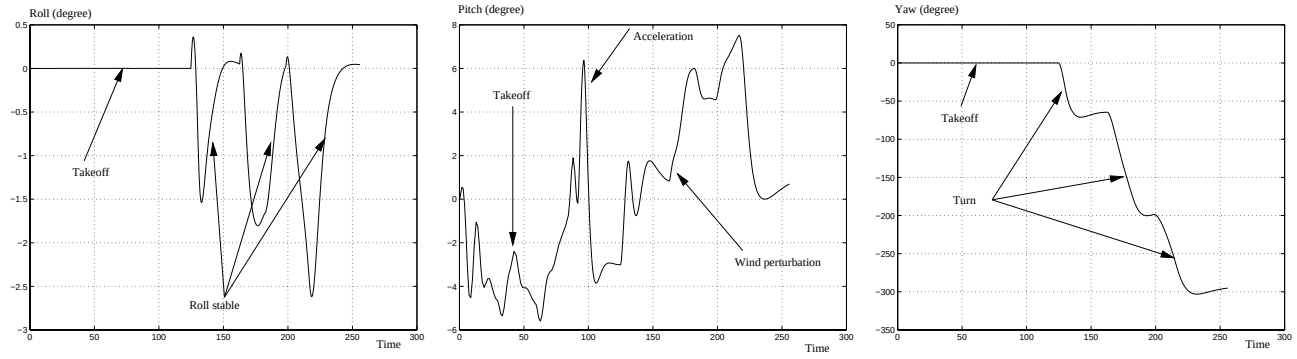

Fig. 28. Airship attitude.

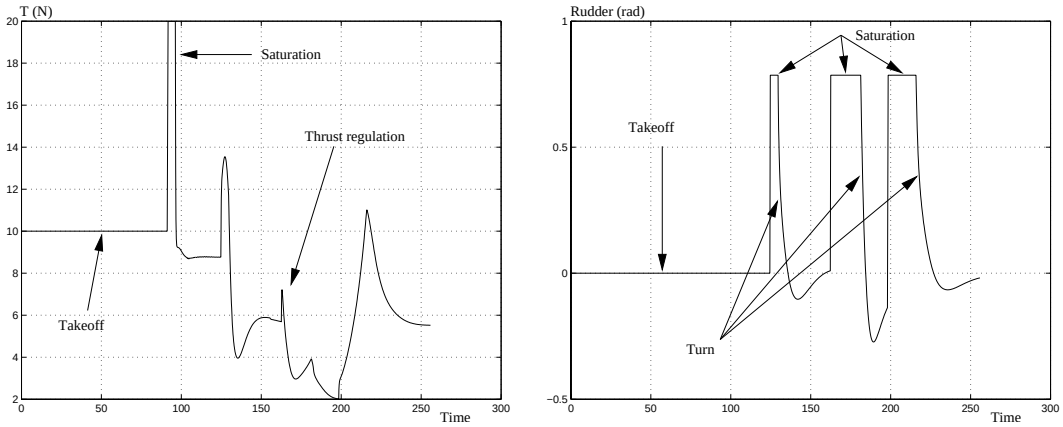

Fig. 29. Actuator activity.

takeoff, the pitch angle ranges between -4° and 8°, inducing moderate changes of the altitude. The variation of the yaw angle ψ differs from the nominal case (22). Depending on the relative wind direction, the time required for the turns and the saturation time of rudders vary (see Figure 29). During the first turn, the airship was facing the wind and the efficiency of rudders was maximal with a relative wind of 7 m s^{-1}. This is the reason why this turn required a shorter time than the following next turns. Note that the wind was blowing from the back with a relative velocity of only 3 m s^{-1} for the second turn, while its lateral action facilitates the last turn. The evolution of lateral position and angular error is shown in Figure 30 (Extension 4).

5. Discussion

The work presented in this first part of the paper is based on the synthesis of a complete model of the airship Karma. This model is issued from a careful analysis of the dynamic and aerodynamic forces and torques acting on it. On this basis, the proposed strategy involves the decoupling of the lateral and longitudinal dynamics by considering two submodels. This approach allows us to construct a global control strategy by addressing the control problem differently for each flight phase. The presentation focused on the steady lateral navigation which constitutes the central flight phase and for which the need for autonomy is predominant (for instance, in the case of exploration tasks as described in the second part of the paper). The proposed controllers have been simulated by considering the complete model with and without wind perturbations. The decoupled approach, which consists of executing in parallel the velocity regulation and the tracking of the reference path, provides a robust and efficient way to control the lateral motion. Although the roll angle and the altitude are not directly controlled during this phase, their behavior is compatible with the control approach. The roll turns out to be stable and well damped, while only a slight variation of the altitude occurs. To answer the problem of altitude drift, the path-following control must be switched off when

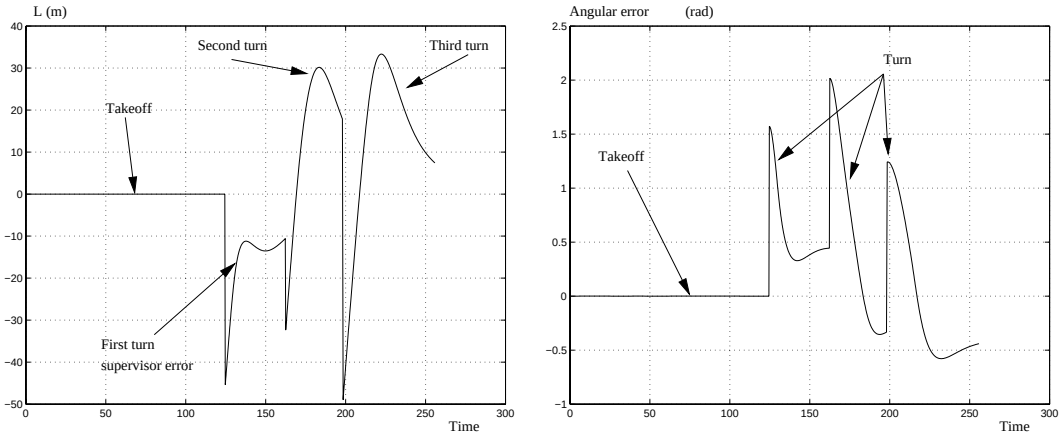

Fig. 30. Evolution of position and angular errors.

necessary to adjust the altitude according to the diagram of Figure 10]. In parallel, thanks to the second control loop, the velocity remains continuously regulated during both phases. The control switch is activated by the supervisor on the base of the prescribed navigation constraints (landmark visibility, for instance).

This theoretical study is currently pursued in two directions. A first part of the work concerns the application of advanced control techniques to extend the control performances. One important objective is to better take into account actuator saturation and uncertainties due to perturbations in designing the control laws. A second part of the work, carried out in collaboration with the CEMIF Laboratory of the University of Evry, concerns the trajectory planning problem.

However, the main effort is now directed towards experiments. The autonomous lateral control is expected to be experimented in the very near future. It will constitute the first part of the longer-term objective, which is to execute complete autonomous flights from takeoff to landing.

Part B: High-Resolution Terrain Mapping

High-resolution terrain mapping can be the main payload of flying devices in a wide variety of applications: fine geographic survey, environmental analysis, mine detection and localization, etc. However, terrain mapping is also a way to achieve precise localization of the flying robot, by providing environment references, thus enabling a position estimation with bounded errors as the robot flies. Finally, mapping is a prerequisite to the development of cooperative air/ground robotics ensembles. Indeed, whatever the cooperation scenario, one of the most important issues to address in order to foster the development of such ensembles is the building and exploitation of common environment representations, using data provided by all possible sources. This is the case in loose cooperation schemes, e.g., where the ground rover operates after the aerial robot. The map built can then be used to prepare the rover mission, but also online, to localize the rover as it navigates for instance. Also, in tighter cooperation schemes, i.e., when both types of robots operate jointly, the ability to build, share and maintain a common environment model is, of course, a key functionality.

This part of the paper presents our approach to the SLAM problem, using only a set of non-registered low-altitude stereovision image pairs. The approach is presented in the following section, and in Section 7 we present the basic algorithms on which it relies: stereovision, interest point detection and matching, and visual motion estimation. In Section 8 we detail our implementation of the extended Kalman filter (EKF), with a focus on the identification of the various errors. Localization results and the building of digital elevation maps are then presented and discussed in Section 9.

6. Simultaneous Localization and Map Building

The main difficulty in building high-resolution terrain maps is to precisely determine the sensor position and orientation as it moves. Dead-reckoning techniques, which integrate over time the data provided by motion estimation sensors, such as wheel encoders for rovers or inertial sensors, are not sufficient for that purpose. Indeed, not only may the position estimate they provide between successive data acquisitions not be precise enough, but they are intrinsically prone to generate position estimates with unbounded error growth. Visual motion estimation techniques that use stereovision and visual feature tracking or matching have recently been proposed in the context of ground rovers (Mallet, Lacroix, and Gallo 2000; Olson et al. 2001). They allow us to obtain a more precise

position estimate between successive data acquisitions, but their errors also cumulate over time, since they do not memorize any environment feature.

The only solution to guarantee bounded errors on the position estimates is to rely on stable environment features that are detected and memorized as the robot moves. It has early been understood in the robotic community that the problems of mapping such features and estimating the robot location are intimately tied together, and that they must therefore be solved in a unified manner (Chatila and Laumond 1985; Smith, Self, and Cheeseman 1987).

This problem, known as the "SLAM problem", has now been widely studied in robotics, mostly in the case of robots moving on planes, i.e., whose position is totally determined by three parameters (a historical presentation of the main contributions can be found in the introduction of Dissanayake et al. 2001). Among the different approaches to solve it, the Kalman filter based approach, or variants such as the information filter, is undoubtedly the most popular. It is theoretically well grounded, and it has been proved that its application to the SLAM problem converges (Dissanayake et al. 2001). Some contributions cope with its main practical drawback, i.e., its complexity which is cubic in the dimension of the considered state (Leonard and Feder 2000; Guivant and Nebot 2001); such developments are necessary when the robot navigates in large areas.

Other approaches to the SLAM problem have been proposed, mainly to overcome the assumption that the various error probability distributions are Gaussian, which is required by the Kalman filter. Set membership approaches just need the knowledge of bounds on the errors (Kieffer et al. 2000; Di Marco et al. 2001), but they are practically difficult to implement when the number of position parameters exceeds three, and are somehow suboptimal. Expectation minimization (EM) algorithms have also been successfully adapted to the SLAM problem (Thrun, Fox, and Burgard 1998), and an approach that address incremental SLAM in this context can be found in Thrun, Burgard, and Fox (2000).

In terms of sensor modality, solutions to the SLAM problem have mainly been experimented with range sensors in indoor environments: sonar sensors (Leonard and Durrant-Whyte 1991; Wijk and Christensen 2000), laser range finders (Moutarlier and Chatila 1991; Thrun, Burgard, and Fox 2000), and recently millimeter wave radars in outdoor environments (Guivant and Nebot 2001; Dissanayake et al. 2001). To our knowledge, there are many fewer contributions to the SLAM problem based on vision. In Deans and Herbert (2000), monocular vision is used as a bearing sensor, with a combination of a Kalman filter and a bundle adjustment technique. In Se, Lowe, and Little (2001), an approach that uses stereovision and visual scale-invariant feature transforms for a robot evolving on a plane is presented, the data association problem being solved a Hough transform hashing.

6.1. Overview of our Approach

Our approach is an application of a Kalman filter based solution to the SLAM problem, in which the robot position is totally 3D (i.e., determined by three translation parameters and three orientation parameters) and that exclusively uses vision. Vision has the great advantage to allow both a very precise determination of the orientation parameters and the detection and association of stable environment features. Moreover, using a stereovision bench, range estimates of the features are directly available, although much less precise than the data provided by a laser range finder. We will, however, see that thanks to the Kalman filter it is possible to achieve extremely precise localization of the stereovision bench, without the aid of any other positioning sensor.

In our approach, landmarks are "interest points", i.e., visual features that can be matched when perceived from various positions, and whose 3D coordinates are provided by stereovision. The key algorithm that allows both motion estimation between consecutive stereovision frames and the observation and matching of landmarks is a robust interest point matching algorithm. We use an EKF as a recursive filter; the state vector of the EKF is the concatenation of the stereo bench position (six parameters) and the landmark's positions (three parameters for each landmark). The visual motion estimation between consecutive stereovision frames is used to predict the filter state, and is fused with the observations provided by landmark matchings.

The various algorithmic stages achieved every time a stereovision image pair is acquired are depicted in Figure 31.

1. Stereovision. A dense 3D image is provided by stereovision (Section 7.1), along with an estimate of the covariances on the coordinates of the computed 3D points (Section 8.2.1).

2. Interest point detection and matching. Interest points are detected in one of the acquired images, and are matched with the interest points detected in the previous step (Section 7.2).

3. Landmark selection. A set of selection criteria are applied to the matched interest points, in order to partition them in three sets: an observed-landmark set, a non-landmark set, and a candidate-landmark set (Section 8.3). The observed landmarks are the detected points that match already mapped landmarks, non-landmark points will solely be used to estimate the elementary motion between the current and the previous step, and candidate landmarks are points that may be added to the filter state, if they pass through the selection criteria during the next steps.

4. Visual motion estimation (VME). The interest points retained as "non-landmarks" are used to estimate the six motion parameters between the previous and current

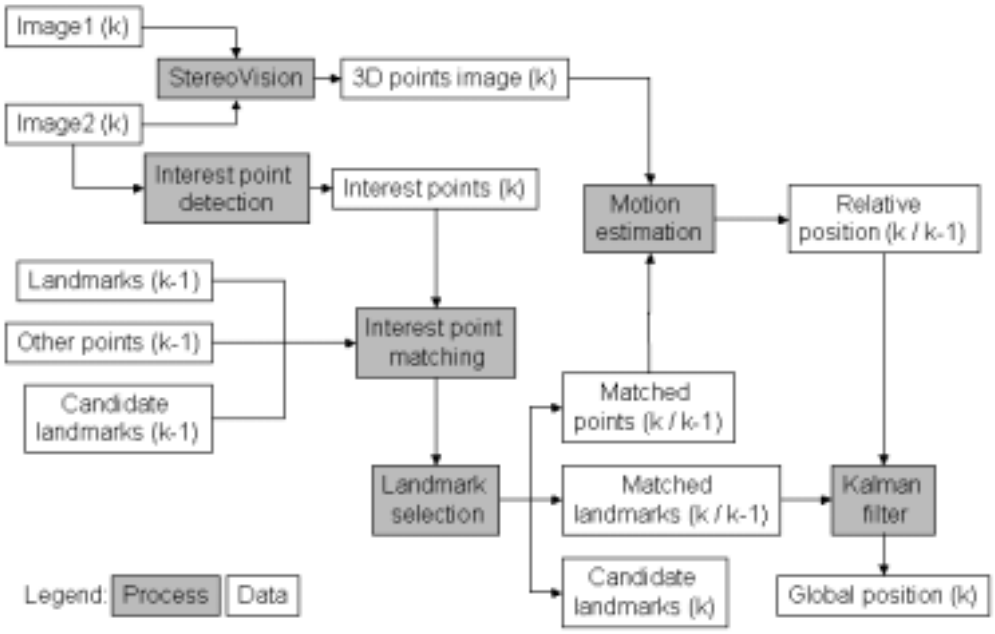

Fig. 31. Functional architecture of our approach to the SLAM problem on the sole basis of stereovision.

steps (Section 7.3), using a least-squares minimization. The associated covariances are also estimated, by propagating stereo and matching errors (Section 8.2.3).

5. Position refinement. This is the update of the Kalman filter state (Section 8).

After every SLAM cycle defined by these steps, a digital elevation map is updated with the acquired images (Section 9.2).

There is an important point to mention here. Indeed, the stereovision bench being the only sensor used in our approach, its data are used both for the prediction stage (visual motion estimation) and the observation stage of the Kalman filter. The prediction and observation are therefore not fully independent, which violates a necessary condition for the filter to be valid. However, in the absence of calibration errors of the stereovision bench, applying the prediction and the observation stages on two separate sets of points does not induce any correlation (which is clear when we consider that the points are perceived by different cameras). This is why the interest points are separated in different sets during the landmark selection. Still, the assumption that there is no calibration error is of course never satisfied, and the errors of the prediction and observation stages are therefore correlated, which is a limitation of our approach.

7. Basic Algorithms

Our vision-based SLAM approach is based on three basic algorithms. Dense stereovision computes the 3D coordinates of most of the perceived pixels, providing thousands of 3D points in the environment. The interest point detection and matching algorithm finds and identify visual landmarks, and allows both the estimation of elementary motions and the observation of already mapped landmarks. Finally, visual motion estimation rejects the outliers produced by the matching algorithm and computes an accurate estimate of the motion between consecutive stereovision frames with the remaining inliers.

7.1. Stereovision

We use a classical pixel-based stereovision algorithm, now widely used in field robotics. It relies on an offline calibrated binocular stereovision bench; the images are first warped (rectified) so that epipolar lines are horizontal, which allows a dramatic optimization to compute similarity scores between the pixels (Faugeras et al. 1993). A dense disparity image is then produced from the warped image pair thanks to a correlation-based pixel matching algorithm (we use either the ZNCC criteria or the census matching criteria; Zabih and Woodfill

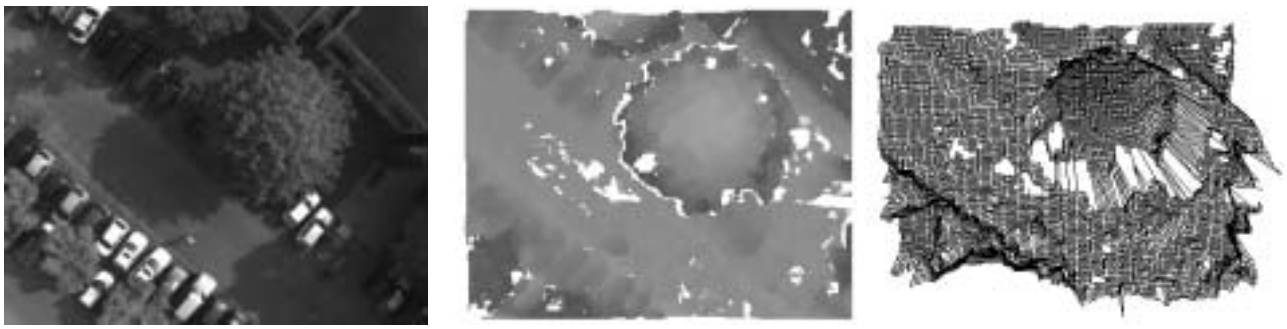

Fig. 32. A result of the stereovision algorithm, with an image pair taken at about 30 m altitude. From left to right: one of the original images, disparity map (the disparities are inversely proportional to the depth of the pixels, and are shown here in a blue/close-red/far-color scale), and 3D image, rendered as a mesh for readability purposes.

1998), false matches are removed thanks to a reverse correlation. Finally, the 3D coordinates of all the matched pixels are determined, using the relative 3D position between the two cameras of the bench provided by the offline calibration stage. In Section 8.2.1 we present how a covariance error matrix is associated with these coordinates.

With the digital cameras of Karma, this algorithm works well for all the scenes we tested. In Figure 32, we can see that pixels have been properly matched in all the perceived areas, even the low textured ones.

7.2. Interest Point Detection and Matching

Visual landmarks should be invariant to image translation, rotation, scaling, partial illumination changes and viewpoint changes. Interest points, such as detected by the well-known Harris detector (Harris and Stephens 1988), has proven to have good stability properties; their repeatability is over 50% when the scale change is no greater than 1.5 times (Schmid, Mohr, and Bauckhage 1998; Jung and Lacroix 2001). If there is a prior knowledge on the scale change, even approximate, a scale adaptive version of the Harris detector yields a repeatability high enough to allow robust matches (Dufournaud, Schmid, and Horaud 2000). When no information on scale change is available, scale adaptation is not possible. In such cases, scale-invariant feature detection algorithms have recently been proposed (Lindeberg 1998; Lowe 1999; Mikolajczyk and Schmid 2001). However, these methods generate many fewer features than the standard or scale-adaptive detectors. Also, matching features in such contexts is quite time-consuming, scale being an additional dimension to search through.

To match interest points, we use an algorithm that we originally described in Jung and Lacroix (2001). We briefly presents its principle here, with an adaptation to roughly known scale variations.

Interest points are local features for which the signal changes in a two-directional (2D) way. The precise Harris detector computes the autocorrelation matrix with gradients of signal on each image point, the two eigenvalues of this matrix being the principle curvatures (Schmid, Mohr, and Bauckhage 1998). When the principle curvatures are significant and locally maximum, the point is declared as an interest point (or corner). In the precise version of the Harris detector, Gaussian functions are used to compute the derivatives. To stabilize the derivatives in scale space, the Gaussian functions are normalized with respect to scale changes. The autocorrelation matrix of scale adaptive Harris detector is then

$$M(\mathbf{x}, s, \tilde{s}) = G(\mathbf{x}, \tilde{s}) \otimes \begin{pmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{pmatrix} \tag{12}$$

$$I_u = sG_u(\mathbf{x}, s) * I(\mathbf{x}) \qquad I_v = sG_v(\mathbf{x}, s) * I(\mathbf{x}) \tag{13}$$

where G is the Gaussian kernel, $G_{u,v}$ is the first-order derivative in the u, v direction, and $\mathbf{x} = (u, v)$. When scale change is not significant, s is set to 1 and the autocorrelation matrix is the same as in the precise version of the Harris detector.

Our matching algorithm relies on local interest point group matching, imposing a combination of geometric and signal similarity constraints, thus being more robust than approaches solely based on local point signal characteristics. Its steps are as follows.

1. Starting with a randomly selected interest point in the first image, matching hypotheses are generated with a similarity measure based on the computed curvatures; a set of candidate matching points is determined in the second image.

2. Local interest point groups are constructed around the studied point and its candidate matches, considering the n closest neighboring points (n being of the order of six).

3. Point-to-point match hypotheses are generated for the neighboring points with the same similarity measure as

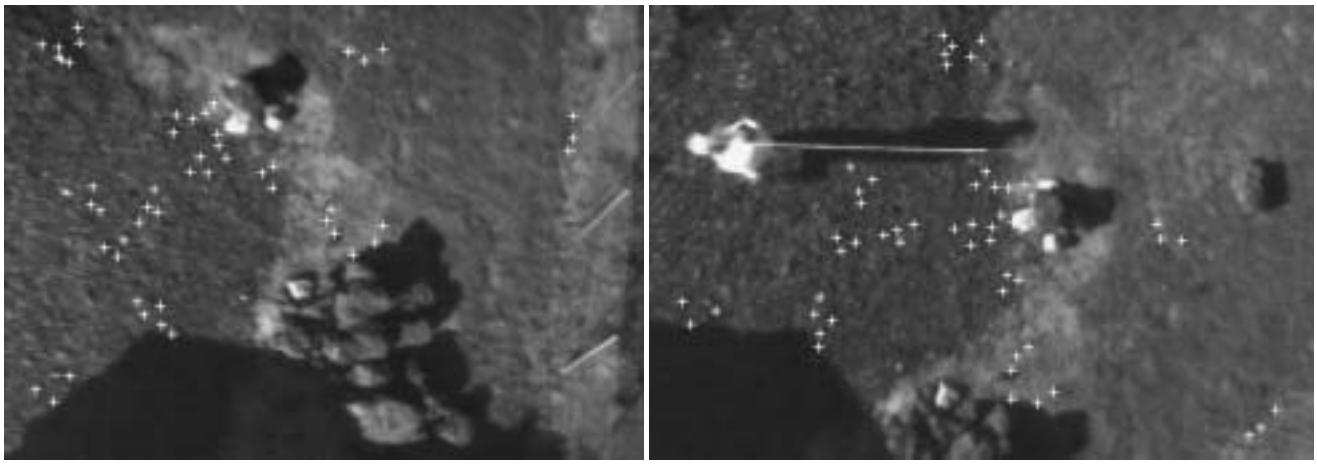

Fig. 33. A result of interest point matching between two non-registered aerial images.

above. The 2D affine group transformation hypotheses are established on the basis of these hypotheses. The transformation yielding the highest point repeatability is confirmed by another similarity measure computed on all the points of the groups.

4. Once a group hypothesis is generated, steps 1–3 are reiterated, starting with the closest point to the first matched group in the first image, and using the estimated 2D affine transformation to focus the search in the second image.

The local affine transformation is updated each time a new group match is found, and these steps are iterated until no more matches are found. Figure 33 shows that this interest point matching algorithm can generate many good matches, even when the viewpoint change between the considered images is quite high.

Between consecutive images, and in the absence of any external motion estimation, no information is available on the scale change. This change is however always small, and the precise version of the point detector is used (i.e., s is set to 1). When flying over a previously perceived area, the altitude of the blimp might have changed significantly, and so the scale. However, an estimate of this altitude change is known thanks to the SLAM algorithm; an estimation of the scale change s is available, and is used to match the already mapped landmarks in the current image.

7.3. Visual Motion Estimation

The interest points matched between consecutive images and the corresponding 3D coordinates provided by stereovision are used to estimate the six displacement parameters between the images. This is achieved by the least-squares minimization technique presented in Haralick et al. (1989). The important point here is to remove the outliers (wrong matches), as they considerably corrupt the minimization result. The interest point matching outliers could be rejected using the epipolar constraint defined by the fundamental matrix computed on the basis of the matches. However, the computation of this matrix is very sensible to the small errors in the point positions and to the outliers themselves. Also, such an outlier removal technique will not deal with stereovision errors, such as those that occur along depth discontinuities for instance; inlier matches in the image plane might become outliers when considering the corresponding 3D coordinates.

Therefore, we have developed a specific outlier rejection method that considers both matching and stereovision errors. First, matches that imply a 3D point whose coordinates uncertainties are over a threshold are discarded (the threshold is empirically determined by statistical analysis of stereovision errors). Then, the remaining matches are analyzed according to the following procedure.

1. A 3D transformation is determined by least-squares minimization. The mean and standard deviations of the residual errors are computed.

2. A threshold is defined as k times the residual error standard deviation. k should be at least greater than 3.

3. The 3D matches whose error is over the threshold are eliminated.

4. k is set to $k - 1$ and the procedure is reiterated until $k = 3$.

This outlier rejection algorithm guarantees a precise 3D motion estimation (see results in Sections 8.2.3 and 9.1), which can then be used during the prediction stage of the Kalman filter (Section 8.1).

8. Kalman Filter Setup

We present here in detail our Kalman filter setting, using the results of the three algorithms sketched above. After the problem formulation and the description of the estimation process, we detail the identification of the various errors, which is the key problem in setting up a Kalman filter to solve the SLAM problem. An active selection of the landmarks is then proposed, which allows us to minimize the state dimension growth and to optimize the precision of the estimations.

8.1. Extended Kalman filter

The EKF is an extension of the standard linear Kalman filter, which linearizes the nonlinear prediction and observation models around the predicted state. The goal of the EKF is to estimate the state of a stochastic nonlinear dynamic system, which evolves under control inputs.

8.1.1. General Framework

A general discrete nonlinear system is modeled as

$$x(k+1) = \mathbf{f}(x(k), u(k+1)) + \upsilon(k+1) \tag{14}$$

where $u(k)$ is a control input, and υ is a vector of temporally uncorrelated process noise with zero mean and covariance $\mathbf{P}_{\upsilon}(k)$.

The nonlinear observation model of the system is modeled as

$$z(k) = \mathbf{h}(x(k)) + w(k) \tag{15}$$

where h maps the state space into the observation space, and w is a vector of temporally uncorrelated observation errors with zero mean and covariance $\mathbf{P}_{w}(k)$.

In the Kalman filter framework, the state estimation encompasses three stages: prediction, observation and update of the state and covariance estimates.

Prediction. The state and observations are predicted using eqs. (14) and (15), and the state covariance is obtained through the linearization of eq. (14):

$$\hat{x}(k+1 \mid k) = \mathbf{f}(\hat{x}(k), u(k+1)) \tag{16}$$

$$\hat{z}(k+1 \mid k) = \mathbf{h}(\hat{x}(k+1 \mid k)) \tag{17}$$

$$\mathbf{P}_{\hat{x}}(k+1 \mid k) = \nabla\mathbf{f}\mathbf{P}_{\hat{x}}(k)\nabla\mathbf{f}^{\mathrm{T}} + \mathbf{P}_{\upsilon}(k+1). \tag{18}$$

Observation. The true state $x(k+1)$ is observed, yielding the innovation $\upsilon(k+1)$, the corresponding covariance being obtained by linearizing eq. (15):

$$\upsilon(k+1) = z(k+1) - \hat{z}(k+1 \mid k) \tag{19}$$

$$\mathbf{S}(k+1) = \nabla\mathbf{h}\mathbf{P}_{\hat{x}}(k+1 \mid k)\nabla\mathbf{h}^{\mathrm{T}} + \mathbf{P}_{w}(k+1). \tag{20}$$

Update. The update stage fuses the prediction and the observation to produce and estimate of the state and its associated covariance, according to the following formulae

$$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + \mathbf{K}(k+1)\upsilon(k+1) \tag{21}$$

$$\begin{aligned}\mathbf{P}_{\hat{x}}(k+1 \mid k+1) = \mathbf{P}_{\hat{x}}(k+1 \mid k) \\ - \mathbf{K}(k+1)\mathbf{S}(k+1)\mathbf{K}^{\mathrm{T}}(k+1)\end{aligned} \tag{22}$$

in which $\mathbf{K}(k+1) = \mathbf{P}_{\hat{x}}(k+1 \mid k)\nabla\mathbf{h}^{\mathrm{T}}\mathbf{S}^{-1}(k+1)$ is the Kalman filter gain matrix.

8.1.2. Filter Setup for Stereovision-Based SLAM

In our approach, the state of the filter is composed of the six positioning parameters $\mathbf{x}_p = [\phi, \theta, \psi, t_x, t_y, t_z]$ of the stereovision bench (or the robot; the notations are the same as in the first part of the paper) and of a set of N landmarks 3D coordinates $\mathbf{m}_i = [x_i, y_i, z_i]$, $0 < i \leq N$:

$$\mathbf{x}(k) = [\mathbf{x}_p, \mathbf{m}_1 \cdots \mathbf{m}_N]. \tag{23}$$

The associated state covariance has the following form

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{pp}(k) & \mathbf{P}_{pm}(k) \\ \mathbf{P}_{pm}{}^{\mathrm{T}}(k) & \mathbf{P}_{mm}(k) \end{bmatrix},$$

where $\mathbf{P}_{pp}$ represents the robot pose covariance, $\mathbf{P}_{mm}$ denotes the landmark covariance and $\mathbf{P}_{pm}$ is the cross-covariance between the robot pose and landmark estimates.

Prediction. Under the assumption that landmarks are stationary, the state prediction is

$$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{f}(k+1)(\hat{\mathbf{x}}(k), \mathbf{u}(k+1)) \tag{24}$$

where $\mathbf{u}(k+1) = (\Delta\phi, \Delta\theta, \Delta\psi, \Delta t_x, \Delta t_y, \Delta t_z)$ is the visual motion estimation result between k and $k+1$ positions. The predicted state covariance (eq. (18)) is written as

$$\begin{aligned}\mathbf{P}_{pp}(k+1 \mid k) = \nabla_p\mathbf{f}(k+1)\mathbf{P}_{pp}(k)\nabla_p\mathbf{f}^{\mathrm{T}}(k+1) \\ + \nabla_u\mathbf{f}(k+1)\mathbf{R}_u(k)\nabla_u\mathbf{f}^{\mathrm{T}}(k+1) \\ + \mathbf{P}_{\upsilon}(k+1)\end{aligned} \tag{25}$$

$$\mathbf{P}_{pm}(k+1 \mid k) = \nabla_p\mathbf{f}(k+1)\mathbf{P}_{pm}(k) \tag{26}$$

$$\mathbf{P}_{mm}(k+1 \mid k) = \mathbf{P}_{mm}(k) \tag{27}$$

where $\mathbf{R}_u$ represents the error covariance of the visual motion estimation result. Note that the covariance of landmarks is not changed in the prediction stage.

Observation. When observing the ith landmark, the observation model and the Jacobian of the observation function are

written as

$$\hat{\mathbf{z}}_i(k+1 \mid k) = \mathbf{h}_i(k+1)(\hat{\mathbf{x}}(k+1 \mid k)) \tag{28}$$

$$\nabla \mathbf{h}_i(k) = [\nabla_p \mathbf{h}_i(k), 0 \cdots 0, \nabla_{m_i} \mathbf{h}_i(k), 0 \cdots 0] \tag{29}$$

where $\mathbf{h}_i(k+1)(\hat{\mathbf{x}}(k+1 \mid k))$ is a function of the predicted robot state and the ith landmark in the state vector of the filter. It can then also be written as $\mathbf{h}_i(k+1)(\hat{\mathbf{x}}_p(k+1 \mid k), \hat{\mathbf{m}}_i(k+1 \mid k))$. The innovation and the associated covariance are written as

$$\boldsymbol{\nu}_i(k+1) = \mathbf{z}_i(k+1) - \hat{\mathbf{z}}_i(k+1 \mid k) \tag{30}$$

$$\begin{aligned} \mathbf{S}_i(k+1) = {} & \nabla \mathbf{h}_i(k+1) \mathbf{P}(k+1 \mid k) \nabla \mathbf{h}_i{}^{\mathrm{T}}(k+1) \\ & + \mathbf{R}_i(k+1) \end{aligned} \tag{31}$$

where $\mathbf{R}_i$ represents the error covariance of ith landmark observation.

Update. The update stage of the state and associated covariance estimates is made through the applications of eqs. (21) and (22), in which the gain matrix $\mathbf{K}$, innovation $\boldsymbol{\nu}$ and associated covariance $\mathbf{S}$ are respectively replaced by $\mathbf{K}_i$, $\boldsymbol{\nu}_i$, and $\mathbf{S}_i$.

If no observation are made (i.e., if no already mapped landmarks are re-perceived), the observation and update stages are not activated; the state and its covariance are just updated by the prediction stage.

When detecting a new landmark, it is added to the state vector of the filter, which becomes $\hat{\mathbf{x}}(k) = [\hat{\mathbf{x}}_p(k), \hat{\mathbf{m}}_1(k) \cdots \hat{\mathbf{m}}_N(k), \hat{\mathbf{m}}_{N+1}(k)]$ (its size increased by three units). The landmark initialization model is

$$\hat{\mathbf{m}}_{N+1}(k) = \mathbf{g}(k)(\hat{\mathbf{x}}_p(k), \mathbf{z}_{\mathbf{N+1}}(k)) \tag{32}$$

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{pp}(k) & \mathbf{P}_{pm}(k) & (\nabla_p \mathbf{g}(k) \mathbf{P}_{pp}(k))^{\mathrm{T}} \\ \mathbf{P}_{pm}{}^{\mathrm{T}}(k) & \mathbf{P}_{mm}(k) & (\nabla_p \mathbf{g}(k) \mathbf{P}_{pm}(k))^{\mathrm{T}} \\ \nabla_p \mathbf{g}(k) \mathbf{P}_{pp}(k) & \nabla_p \mathbf{g}(k) \mathbf{P}_{pm}(k) & \nabla_p \mathbf{g}(k) \mathbf{P}_{pp}(k) \nabla_p \mathbf{g}^{\mathrm{T}}(k) + \nabla_z \mathbf{g}(k) \mathbf{R}_m(k) \nabla_z \mathbf{g}^{\mathrm{T}}(k) \end{bmatrix} \tag{33}$$

where $\mathbf{z}_{\mathbf{N+1}}(k)$ denotes the new landmark, $\mathbf{g}(k)$ represents the initialization function using the current robot pose estimate, and $\mathbf{R}_m$ is the error covariance of the new landmark.

8.2. Error Identification

To implement the Kalman filter in our context, the following errors must therefore be estimated:

- the landmark initialization error (covariance matrix $\mathbf{R}_m$);
- the landmark observation error (covariance matrix $\mathbf{R}_i$ for the observed landmark i);
- the error of the input control u, which is the visual motion estimation result (covariance matrix $\mathbf{R}_u$).

This is important, as a precise determination of these errors will avoid the empirical "filter tuning" step. Note that, in our approach, the lumped process noise υ is set to 0, landmarks being stationary and the robot pose prediction being directly computed with the current pose and the result of the visual motion estimation.

8.2.1. Landmark Initialization Errors

Landmarks are detected and matched on the video images, their 3D coordinates being computed by stereovision. When a new landmark is detected, its identity is given by the corresponding interest point, and the covariance matrix $\mathbf{R}_m$ on its state coordinates is totally defined by the stereovision error. Once identified, a new landmark is added in the filter state vector according eq. (32), and its uncertainties are propagated into the state estimate covariance matrix according to eq. (33).

An error model of stereovision. During the stereo matching phase, disparities are computed for integer values, the matching disparity d_m being the one that maximizes the similarity score s. In order to obtain a finer disparity estimate, a subpixellic disparity d'_m is determined by fitting a parabola to the similarity score curve at its peak, the parabola being defined by the similarity scores computed at disparities $d_m - 1$, d_m and $d_m + 1$. The subpixellic disparity is the disparity that maximizes the found parabola[2]:

$$d'_m = d_m + \frac{s(d_m - 1) - s(d_m + 1)}{2[(s(d_m) - s(d_m - 1) + (s(d_m) - s(d_m + 1)]}. \tag{34}$$

The sources of errors in the disparity estimates are the image noise, the slight viewpoint change of the two cameras, the spatial sampling of the scene induced by the cameras, the size of the correlation window used, and the interpolation of the similarity score curve. Thorough studies of these phenomena can be found in the vision literature, but they lead to complex algorithms that are not tractable online.

In order to have an estimate of the disparity errors, we studied the distribution of the disparities on a set of 100 stereo image pairs acquired from the same position. As in Matthies (1992), it appeared that the distribution of the disparity computed on any given pixel can be well approximated by a Gaussian (Figure 34). However, a much more interesting fact is that there is a strong correlation between the shape of the similarity score curve around its peak and the standard deviation on

2. Note that there does not exist any theoretical ground that justifies the use of a parabolic interpolation. It is only simple to compute, and it shifts the value of the integer disparity towards the neighbor that gives the highest similarity score, which is intuitive.

the disparity; the sharper the peak, the more precise the disparity found (Figure 34). This rather intuitive relation is the basis of our error model; online, during the stereo matching phase, a standard deviation σ_d is associated to each computed disparity d, using the curvature of the similarity score curve at its peak. This is done at no extra computing cost, as this curvature is that of the interpolating parabola at its peak.

Once matches are established, the coordinates of the 3D points are computed with the usual triangulation formula

$$z = \frac{b\alpha}{d}, \qquad x = \beta_u z, \qquad y = \gamma_v z, \tag{35}$$

where z is the depth, b is the stereo baseline, and α, β_u, and γ_v are calibration parameters (the latter two depending on (u, v), the position of the considered pixel in the image). Using a first-order approximation, it becomes

$$\sigma_z^2 \simeq \left(\frac{\partial z}{\partial d}\right)^2 \sigma_d^2 = \frac{(b\alpha)^2}{d^4}\sigma_d^2. \tag{36}$$

Substituting the definition of z defined in eq. (35), we have

$$\sigma_z = \frac{\sigma_d}{b\alpha} z^2, \tag{37}$$

which is a well-known property of stereovision, i.e., that the errors on the depth grow quadratically with the depth, and are inversely proportional to the stereo baseline. (This is the reason why we choose a 2.2 m wide stereo bench; with the new gondola, we will adapt a rigid 3 m wide bench.) The covariance matrix of the point coordinates is then

$$\mathbf{R}_m = \begin{bmatrix} 1 & \beta_u & \gamma_v \\ \beta_u & \beta_u^2 & \beta_u\gamma_v \\ \gamma_v & \beta_u\gamma_v & \gamma_v^2 \end{bmatrix} \left(\frac{\sigma_d}{b\alpha} z^2\right)^2. \tag{38}$$

When a new landmark is observed, its coordinates are added to the filter state, and the state covariance is updated according to eqs. (32) and (33).

8.2.2. Observation Error

In our case, landmark observation is based on interest point matching; matching errors on the image plane is the first error source to consider to define the observation error. Two types of error can occur: wrong matches (outliers), and interest point location errors. Outliers are rejected by the rejection algorithm presented in Section 7.3; only interest point location errors are considered to determine the matching error.

With the precise Harris detector, the subpixellic coordinates of an interest point $\mathbf{p}$ belong to one pixel. When perceived again from a very close point of view (e.g., in two consecutive images), most of the area corresponding to this pixel is mapped into another pixel, and the matched interest point $\mathbf{p}'$ coordinates lie within this pixel. The expected matching error is therefore set to 0.5 pixel. However, when the viewpoint is very different (e.g. when re-perceiving a landmark after a long loop), the projective deformation of the 3D scene and possible occlusion effects are much more important. In such cases, it might happen that the matched point $\mathbf{p}''$ does not lie within the pixel which covers the first point $\mathbf{p}$ (and it might also happen that no matches are detected in the worst cases); the expected matching error value is then set to 1 pixel, which is consistent with the maximum error of 1.5 pixel generally tolerated to assess good matches using the epipolar constraint (Schmid, Mohr, and Bauckhage 1998). Figure 35 illustrates this phenomenon. The two leftmost images were consecutively taken during a flight and the matching errors are less than 0.5 pixel, whereas in the rightmost image that has been taken from a very different viewpoint, the matching location error exceeds 0.5 pixel, but is not greater than 1 pixel.

Now that interest point matching error is defined, it is necessary to combine it with the errors on the corresponding 3D estimates to define the observation error. The principle of this combination is illustrated in Figure 36; the observation matching error is defined by the reprojection of the matching error in the 3D scene. When the 2D matching error is set to 1 pixel, the expected value of a matching point $\mathbf{p}_0$ is defined by its eight closest neighbors $\mathbf{p}_k$, $k = 1, 2...8$. The stereo error distribution being a zero mean normal one, the expected 3D coordinate and associated variance of the matching point is computed as follows

$$\bar{\mathbf{X}} = \frac{1}{9}\sum_{k=0}^{8} \mathbf{X}_k, \qquad \boldsymbol{\sigma}_{\bar{\mathbf{X}}}^2 = \frac{1}{9}\sum_{k=0}^{8} (\bar{\mathbf{X}} - \mathbf{X}_k)^2 + \boldsymbol{\sigma}_k^2 \tag{39}$$

where $\mathbf{X}_0$ and $\mathbf{X}_k$ are the 3D point coordinates of $\mathbf{p}_0$ and its neighbors, and $\boldsymbol{\sigma}_0$ and $\boldsymbol{\sigma}_k$ are the corresponding variances.

When the expected matching error is set as 0.5 pixel, the 3D coordinates being only computed on integer pixels by stereovision, we assume the 3D surface variation is locally linear, and the expected 3D coordinate and corresponding variances of the observed point $\mathbf{p}_0$ are then

$$\begin{aligned} \bar{\mathbf{X}} &= \frac{1}{9}\left(\mathbf{X}_0 + \sum_{k=1}^{8}\left(\frac{\mathbf{X}_0 + \mathbf{X}_k}{2}\right)\right), \\ \boldsymbol{\sigma}_{\bar{\mathbf{X}}}^2 &= \frac{1}{9}\sum_{k=0}^{8}\left(\frac{\bar{\mathbf{X}} - \mathbf{X}_k}{2}\right)^2 + \left(\frac{\boldsymbol{\sigma}_0 + \boldsymbol{\sigma}_k}{2}\right)^2. \end{aligned} \tag{40}$$

These coordinates and the associated variances are used in eqs. (30) and (31).

8.2.3. Motion Estimation Errors

Given a set of 3D matched points $\hat{\boldsymbol{\mathcal{Q}}} = [\mathbf{X}_1...\mathbf{X}_N, \mathbf{X}_1...\mathbf{X}'_N]$, the function which is minimized to determine the corresponding motion is the following (Haralick et al. 1989)

$$J(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}}) = \sum_{n=1}^{N} (\mathbf{X}'_n - R(\hat{\phi}, \hat{\theta}, \hat{\psi})\mathbf{X}_n - [\hat{t}_x, \hat{t}_y, \hat{t}_z]^{\mathrm{T}})^2 \tag{41}$$

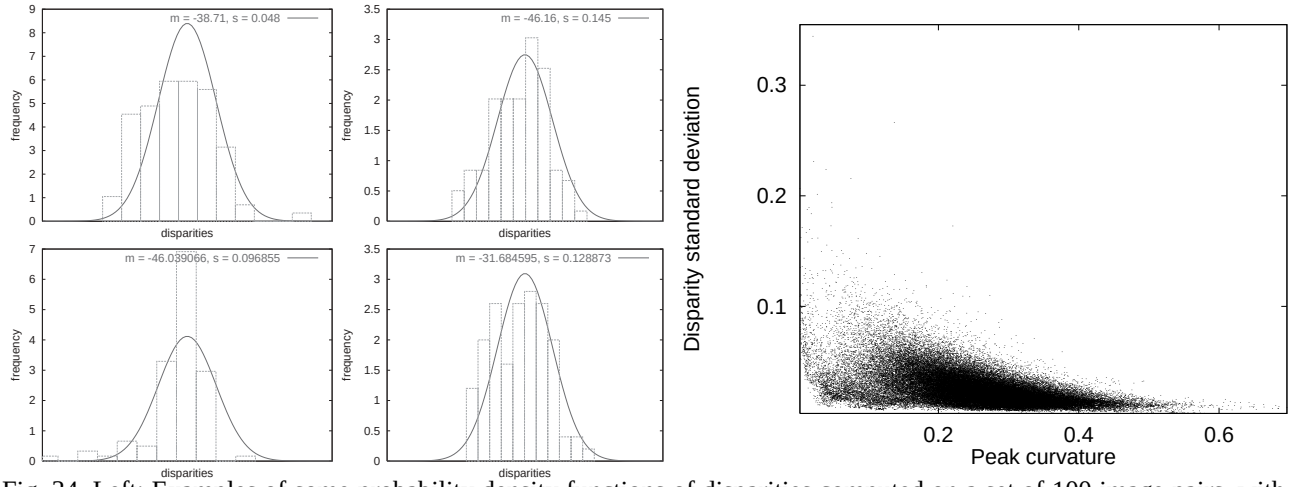

Fig. 34. Left: Examples of some probability density functions of disparities computed on a set of 100 image pairs, with the corresponding Gaussian fit. Right: Standard deviation of the disparities as a function of the curvature of the similarity score curve at its peak.

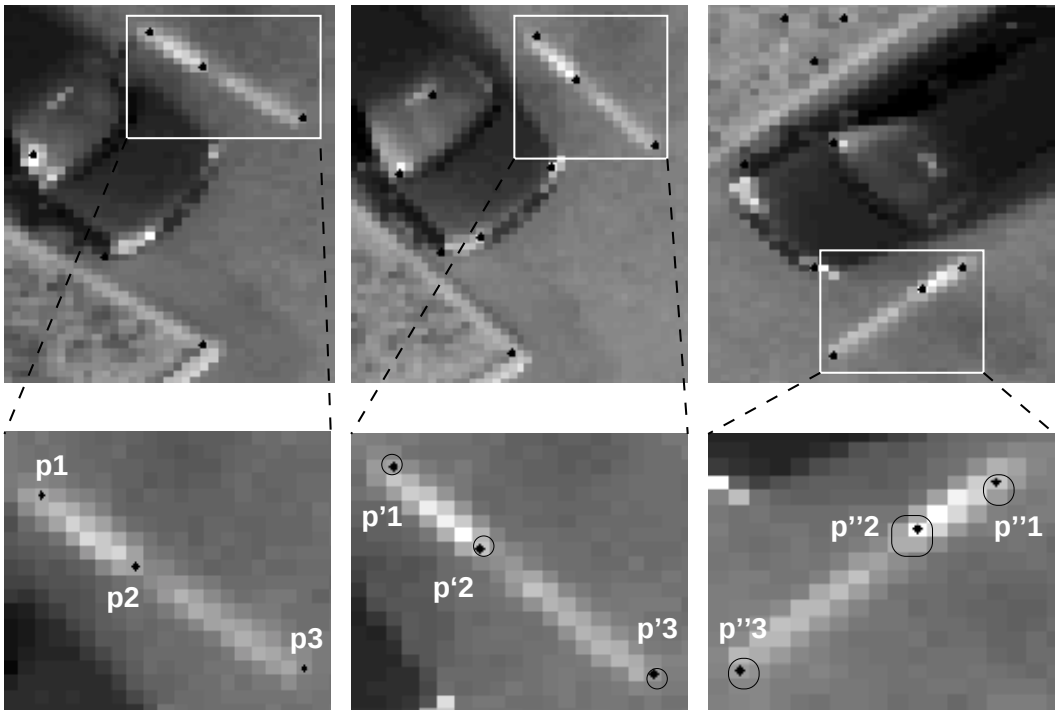

Fig. 35. Illustration of the matching point location error, in the case of small and big viewpoint changes. Top images show the detected interest points, the uncertainty of the matched ones being represented by black circles in the close-up bottom images.

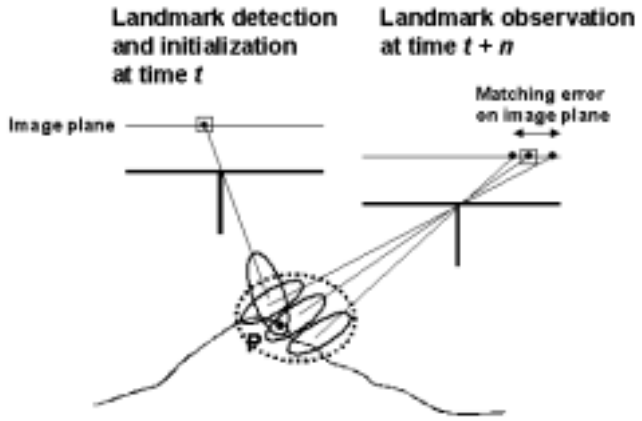

Fig. 36. Principle of the combination of the matching and stereovision errors. The points located in the square box are the projection of P on the image plane. Small ellipses indicate stereovision errors, and the large dotted ellipsoid is the resulting observation error.

where $\hat{\mathbf{u}} = (\hat{\phi}, \hat{\theta}, \hat{\psi}, \hat{t}_x, \hat{t}_y, \hat{t}_z)$. $\hat{\mathbf{u}}$ and $\hat{\boldsymbol{\mathcal{Q}}}$ can be written with random perturbations

$$\hat{\mathbf{u}} = \mathbf{u} + \Delta\mathbf{u}, \qquad \boldsymbol{\mathcal{Q}} = \hat{\boldsymbol{\mathcal{Q}}} + \Delta\hat{\boldsymbol{\mathcal{Q}}}$$

where the true $\mathbf{u}$ and $\hat{\boldsymbol{\mathcal{Q}}}$ are not observed. In order to measure the uncertainty of local motion estimation, the uncertainties of 3D matching points set are propagated to the optimal motion estimate $\hat{\mathbf{u}}$. Assuming the optimal motion estimate minimizes the cost function, the Jacobian of the cost function is 0, and the uncertainties of landmarks and their observation can be propagated by taking Taylor series expansion of the Jacobian around $\mathbf{u}$ and $\hat{\boldsymbol{\mathcal{Q}}}$, as shown in (Haralick 1994)

$$\begin{aligned} g(\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}}) = g(\mathbf{u} + \Delta\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}} + \Delta\boldsymbol{\mathcal{Q}}) \\ - \frac{\partial g}{\partial \mathbf{u}}(\mathbf{u} + \Delta\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}} + \Delta\boldsymbol{\mathcal{Q}})\Delta\mathbf{u} \\ - \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\mathbf{u} + \Delta\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}} + \Delta\boldsymbol{\mathcal{Q}})\Delta\boldsymbol{\mathcal{Q}} \end{aligned} \tag{42}$$

where $g = \frac{\partial J}{\partial \hat{\mathbf{u}}}$ is the Jacobian of the cost function, and $\frac{\partial g}{\partial \mathbf{u}}$ is the Hessian of the cost function with respect to $\mathbf{u}$. The Hessian $\frac{\partial g}{\partial \mathbf{u}}$ is positive definite for all $(\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}})$ because the relative extremum of the cost function is a relative minimum; this guarantees the existence of the reciprocal of the Hessian. Since $\hat{\mathbf{u}}$ and $\mathbf{u}$ minimize $J(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})$ and $J(\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}})$, $g(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})$ and $g(\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}})$ are set to 0 in eq. (42). The random perturbation $\Delta\mathbf{u}$ and its covariance are then computed as follows

$$\begin{aligned} \Delta\mathbf{u} = -\left(\frac{\partial g}{\partial \mathbf{u}}(\mathbf{u} + \Delta\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}} + \Delta\boldsymbol{\mathcal{Q}})\right)^{-1} \\ \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\mathbf{u} + \Delta\mathbf{u}, \hat{\boldsymbol{\mathcal{Q}}} + \Delta\boldsymbol{\mathcal{Q}})\Delta\boldsymbol{\mathcal{Q}} \end{aligned} \tag{43}$$

$$\begin{aligned} \hat{\Sigma}_{\Delta\mathbf{u}} = \left(\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})\right)^{-1} \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})\Sigma_{\Delta\hat{\boldsymbol{\mathcal{Q}}}} \frac{\partial g}{\partial \boldsymbol{\mathcal{Q}}} \\ (\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})^{\mathrm{T}} \left(\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})\right)^{-1}. \end{aligned} \tag{44}$$

Considering that $\mathbf{X}_n$ and $\mathbf{X}'_n$ are not correlated, the covariance estimate $\mathbf{P}_{\hat{\mathbf{u}}}$ can be also written as

$$\mathbf{P}_{\hat{\mathbf{u}}} = \left(\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})\right)^{-1} (\Lambda_{\mathbf{X}} + \Lambda_{\mathbf{X}'}) \left(\frac{\partial g}{\partial \mathbf{u}}(\hat{\mathbf{u}}, \hat{\boldsymbol{\mathcal{Q}}})\right)^{-1} \tag{45}$$

where

$$\Lambda_{\mathbf{X}} = \sum_{n=1}^{N} \frac{\partial g}{\partial \mathbf{X}_n}(\hat{\mathbf{u}}, \mathbf{X}_n)\mathbf{P}_{\mathbf{X}_n} \left(\frac{\partial g}{\partial \mathbf{X}_n}(\hat{\mathbf{u}}, \mathbf{X}_n)\right)^{\mathrm{T}}$$

$$\Lambda_{\mathbf{X}'} = \sum_{n=1}^{N} \frac{\partial g}{\partial \mathbf{X}'_n}(\hat{\mathbf{u}}, \mathbf{X}'_n)\mathbf{P}_{\mathbf{X}'_n} \left(\frac{\partial g}{\partial \mathbf{X}'_n}(\hat{\mathbf{u}}, \mathbf{X}'_n)\right)^{\mathrm{T}}.$$

$\mathbf{P}_{\hat{\mathbf{u}}} = \mathbf{R}_u$ is the input covariance matrix which is used in eq. (25) to estimate the state variances during the filter prediction stage. Results of the visual motion estimations and

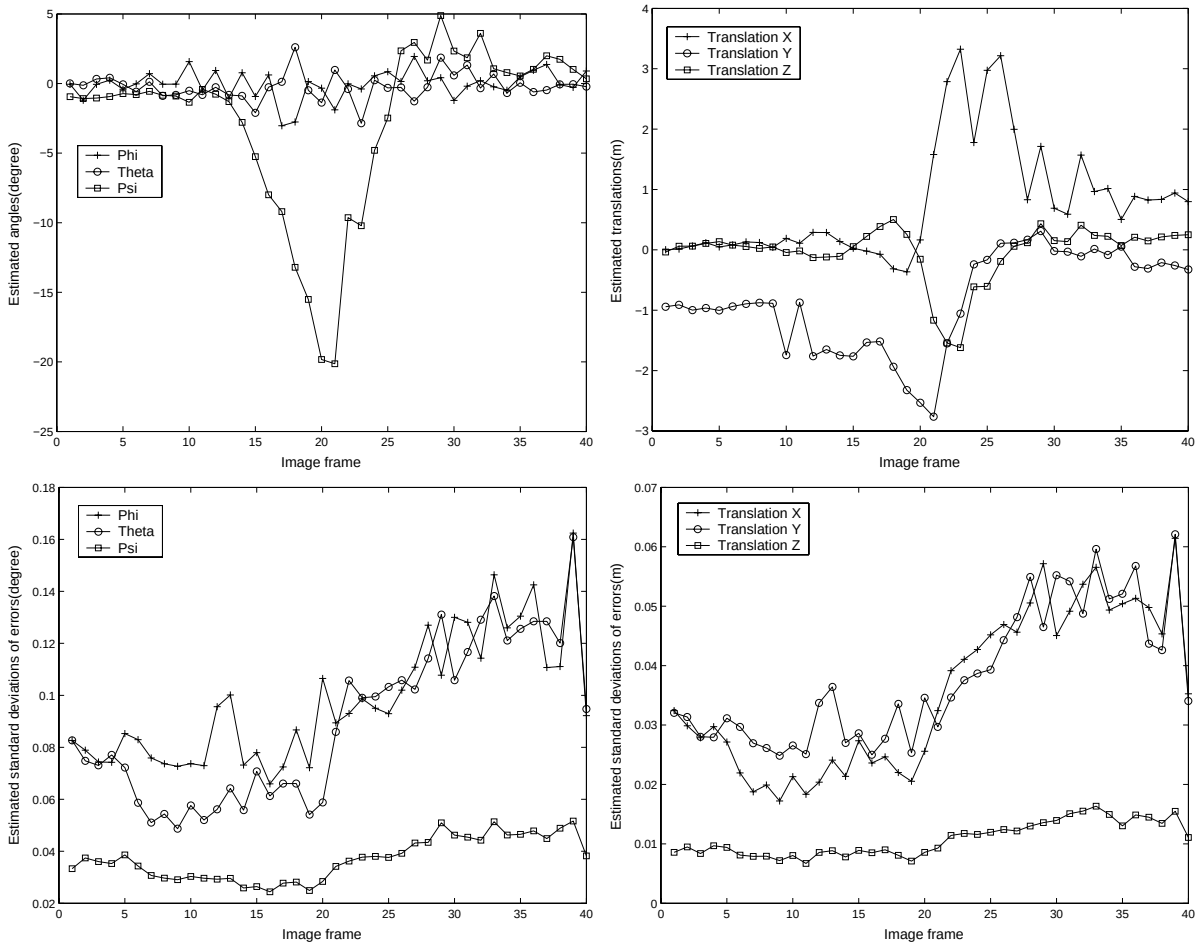

Fig. 37. The motion parameters computed by the visual motion estimation algorithm between consecutive frames along a 40 image sequence (top), and corresponding estimated errors (bottom). Note that there are no obvious correlations between the image angular and linear distance and the computed errors.

Table 3. Statistics on the Estimated Errors of the VME of Figure 37

	Φ	Θ	Ψ	t_x	t_y	t_z
Average of estimated VME errors	0.098^o	0.089^o	0.037^o	0.036 m	0.038 m	0.011 m
σ of VME estimated errors	0.024^o	0.030^o	0.008^o	0.013 m	0.011 m	0.003 m

the corresponding error estimates are presented in Figure 37, along a sequence of 40 images taken from Karma. The mean computed variances on the six motion parameters, and their dispersion are summarized in Table 3; the visual motion estimation measures translations of a few meters with an accuracy of a few centimeters, and measures rotations with a precision of the order of 0.1°, with quite good regularity.

8.3. Landmark Selection

As explained in the overview of our approach (Section 6.1), the 3D matches established after the interest point matching step are split into three groups: observed landmarks, non-landmarks, and candidate landmarks. The observed-landmark set is simply the points that correspond to landmarks already in the state vector of the EKF. New candidate landmarks should be cautiously added to the filter state, in order to avoid a rapid growth of its dimension and to obtain a regular landmark coverage of the perceived scenes. The candidate-landmark selection procedure is made according to the following three criteria.

- **Observability.** Good landmarks should be observable in several consecutive frames: it guarantees that they are salient.
- **Stability.** The 3D coordinates of good landmarks must be precisely estimated by stereovision.

- **Representability.** Good landmarks must efficiently represent a 3D scene. The robot state estimation will be more stable if landmarks are regularly dispatched in the perceived scene, and this regularity will avoid a rapid growth of the EKF state vector size.

The number of candidate landmarks that are checked is determined on the basis of the number of new interest point matches (i.e., those that do not match with an already mapped landmark). This number is a percentage of the new interest points; we actually use 10%, as the visual motion estimation technique requires many matches to yield a precise result. The landmark selection is made through the following steps.

1. The found number of new landmark candidates is first selected using the observability criterion. The observability of a landmark candidate is evaluated during several frames. When the selected landmarks at time t are observable up to time $t + k$, the candidates discovered at time $t + 1$ should be observable at least until time $t + k + 1$.
2. The candidates that pass through the observability test are then checked for stability. Their position and observation errors during the observability check are memorized, and they are ranked according to the maximum of these errors.
3. Finally, the candidate representability is checked. The ranked candidate list is scanned, starting from the most precise one; every time a candidate is located at a distance greater than a given threshold from the already mapped landmarks, it is added to the mapped landmarks.

The observability criterion requires that candidates are evaluated through several successive frames; the EKF is therefore activated a few image frames later as the images are gathered.

9. Results

Our developments have been tested with hundreds of images taken on-bard Karma, at altitudes ranging from 20 to 35 m. The digital cameras of the 2.2 m wide stereo bench are 1/2 inch CCD sensors with 1024×768 pixels, and are equipped with a 4.8 mm focal length lens ($67^\circ \times 53^\circ$ field of view). The cameras have been calibrated at full resolution, and images are processed after being subsampled by a factor of 2, to save stereovision computing time (which is cubic in the dimension of the images).

9.1. Positioning Errors

The GPS on-board Karma is a differential code GPS with a 3σ accuracy of 2 m; it cannot be used as a ground truth reference to validate the position estimates of the stereo bench. However, when Karma flies over an already perceived area (i.e., when it "closes a loop"), the visual motion estimate can provide an estimate of the relative positions between the first and last images of the sequence that overlaps. This reference is precise enough, as compared to the cumulation of errors induced with the visual motion estimation applied on consecutive frames.

Figure 38 presents a comparison of the reconstructed loop trajectory with a set of 40 images, while Figure 39 shows the evolution of the standard deviation of the six position parameters of the stereo bench when applying the EKF. Two phases can be seen on this latter figure: until image 25, the standard deviation grows, however, much more slowly than when propagating only the errors of the VME. A few landmarks detected in the beginning of the sequence are re-perceived in the following images; the standard deviation decreases, and stabilizes for the subsequent images where some "old" landmarks are still observed.

The quantitative figures summarized in Table 2 are much more informative. They compare the results of the final position estimate with respect to the reference defined by the VME applied between images 1 and 40. The precision enhancement brought by the EKF is noticeable, and the absolute estimated errors are all bounded by twice the estimated standard deviations. The translation errors are below 0.1 m in the three axes after a trajectory about 60 m long, and angular errors are all below half a degree.

Figure 40 shows another trajectory reconstructed with a set of 100 images, and finally Figures 41 and 42 show results integrating 400 images, the corresponding numerical results being presented in Table 3. In this latter case, a strong position refinement is provided by the filter when the blimp flies again over data acquired at the beginning of the trajectory.

9.2. Digital Elevation Maps

Thanks to the precise positioning estimation, the processed stereovision images can be fused after every update of the EKF into a digital elevation map (DEM), which describes the environment as a function $z = f(x, y)$, determined on every cell (x_i, y_i) of a regular Cartesian grid.

Our algorithm to build a DEM simply computes the elevation of each cell by averaging the elevations of the 3D points that are vertically projected on the elementary surface it defines. The standard deviation on the cell elevation is also straightforwardly computed, and, since a luminance value is associated to each 3D point produced by stereovision, it is also possible to compute a mean luminance value for each map cell. Figure 43 shows a digital elevation built from the 100 images during the trajectory shown in Figure 40; the resolution of the grid is here 0.1 m, and no map discrepancies can be detected in the corresponding orthoimage, which is the vertical orthogonal projection of the luminance information encoded in the DEM grid (Extension 3).

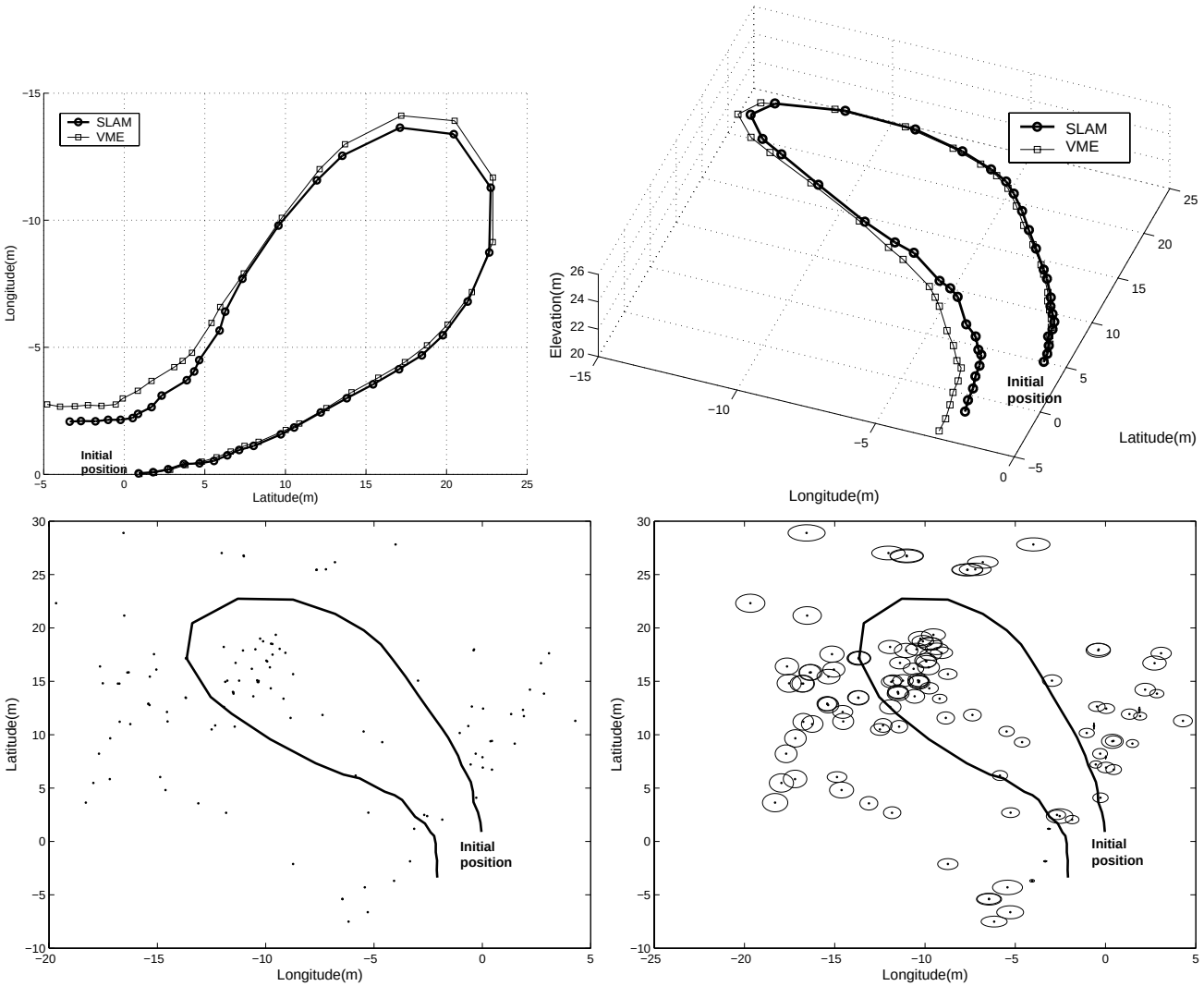

Fig. 38. A first result of our SLAM implementation with a sequence of 40 stereovision pairs, taken at altitudes ranging from 20 to 25 m. The top images show the reconstructed trajectory in orthogonal projection and in 3D. The bottom images show the 120 landmarks mapped, with 1σ uncertainty ellipses (left, real scale; right, magnified by a factor of 40).

Table 4. Comparison of the Errors Made by the Propagation of the VME Alone and With the SLAM EKF Approach, Using as a Reference the VME Applied Between Images 1 and 40

	Frame 1/40 Reference	Reference Standard Development	VME Result	VME Absolute Error	SLAM Result	SLAM Standard Development	SLAM Absolute Error
Φ	$-7.08°$	$0.10°$	$-10.38°$	$3.30°$	$-7.29°$	$0.21°$	$0.20°$
Θ	$-1.50°$	$0.09°$	$-3.90°$	$2.40°$	$-2.11°$	$0.28°$	$0.61°$
Ψ	$-105.71°$	$0.04°$	$-105.15°$	$0.56°$	$-105.82°$	$0.09°$	$0.11°$
t_x	-1.84 m	0.04 m	-2.75 m	0.91 m	-2.08 m	0.11 m	0.24 m
t_y	-3.31 m	0.04 m	-4.78 m	1.47 m	-3.38 m	0.08 m	0.07 m
t_z	-1.73 m	0.01 m	-2.64 m	0.91 m	-1.83 m	0.04 m	0.10 m

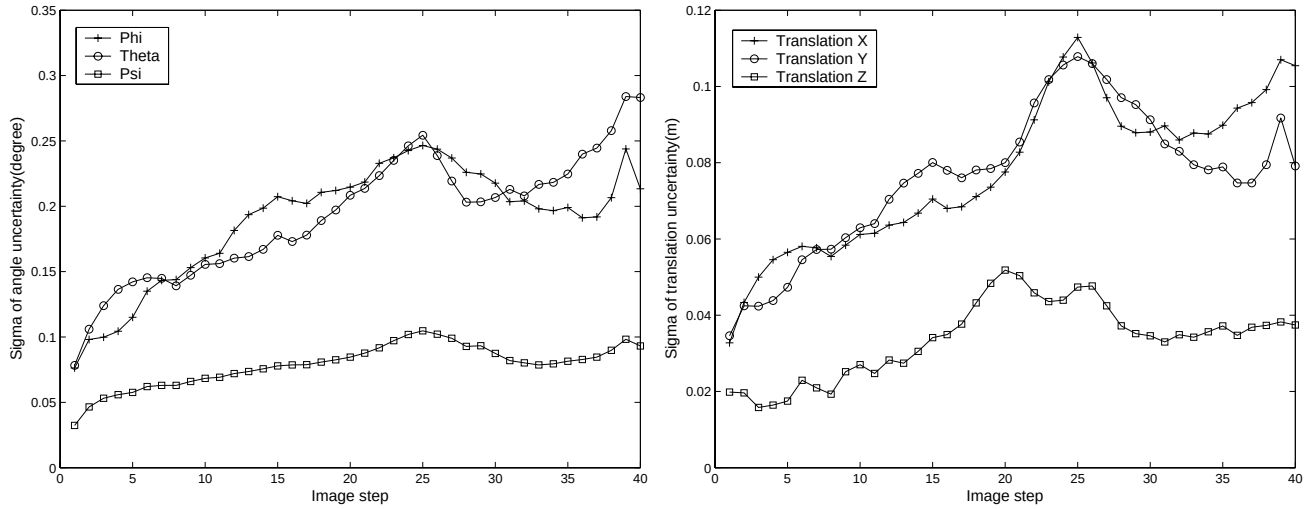

Fig. 39. Evolution of the standard deviations of the robot position parameters during the flight shown in Figure 38.

Table 5. Comparison of the Errors Made by the Propagation of the VME Alone and With the SLAM EKF Approach for the Trajectory of Figure 41, Using as a Reference the VME Applied Between Images 1 and 400

	Frame 1/40 Reference	Reference Standard Development	VME Result	VME Absolute Error	SLAM Result	SLAM Standard Development	SLAM Absolute Error
Φ	−0.12°	0.87°	−0.26°	0.15°	−1.05°	0.40°	0.94°
Θ	2.87°	1.14°	−6.16°	9.02°	1.74°	0.44°	1.12°
Ψ	105.44°	0.23°	101.87°	3.57°	104.77°	0.14°	0.67°
t_x	−4.93 m	0.57 m	6.29 m	11.22 m	−2.81 m	0.23 m	1.12 m
t_y	0.14 m	0.46 m	3.22 m	3.08 m	1.53 m	0.24 m	1.39 m
t_z	3.89 m	0.15 m	20.17 m	16.27 m	3.48 m	0.08 m	0.41 m

9.3. Discussion

The first results described here show that thanks to the application of an EKF on the sole basis of stereovision, it is possible to achieve a positioning of the blimp with a precision of a few centimeters after a flight of several tens of meters, thus enabling the possibility of building very high resolution environment maps. To our knowledge, this is the first attempt to tackle a SLAM problem in 3D space, using exclusively information provided by vision.

The main advantage of our approach relies on the use of interest points as landmarks. Such points are indeed very numerous in any type of environment; no "obvious" landmarks (such as trees, rocks, fences, pebbles, etc.) are required for the algorithms to operate successfully, and this allows an active selection of the landmarks to map. The reliability of our interest point matching algorithm is, of course, a key point here, as it allows robust data associations. Also, the use of the visual estimation technique as a means to achieve the prediction stage of the filter is very efficient; its estimates are precise enough to yield a fast convergence, keeping the filter linearizations as fair approximations. Finally, a thorough study and identification of the various error estimates involved in the filter allowed us to set it up properly, without any empirical tuning stage, which would have been very tedious.

There remain, however, various points to tackle, before having the system integrated on board Karma. First, the high number of landmarks will not allow us to maintain the whole EKF state vector over a few tens of images, because of computation time constraints. An implementation such as the compressed EKF (Guivant and Nebot 2001) is definitely required. Secondly, the fact that the prediction and observation stages

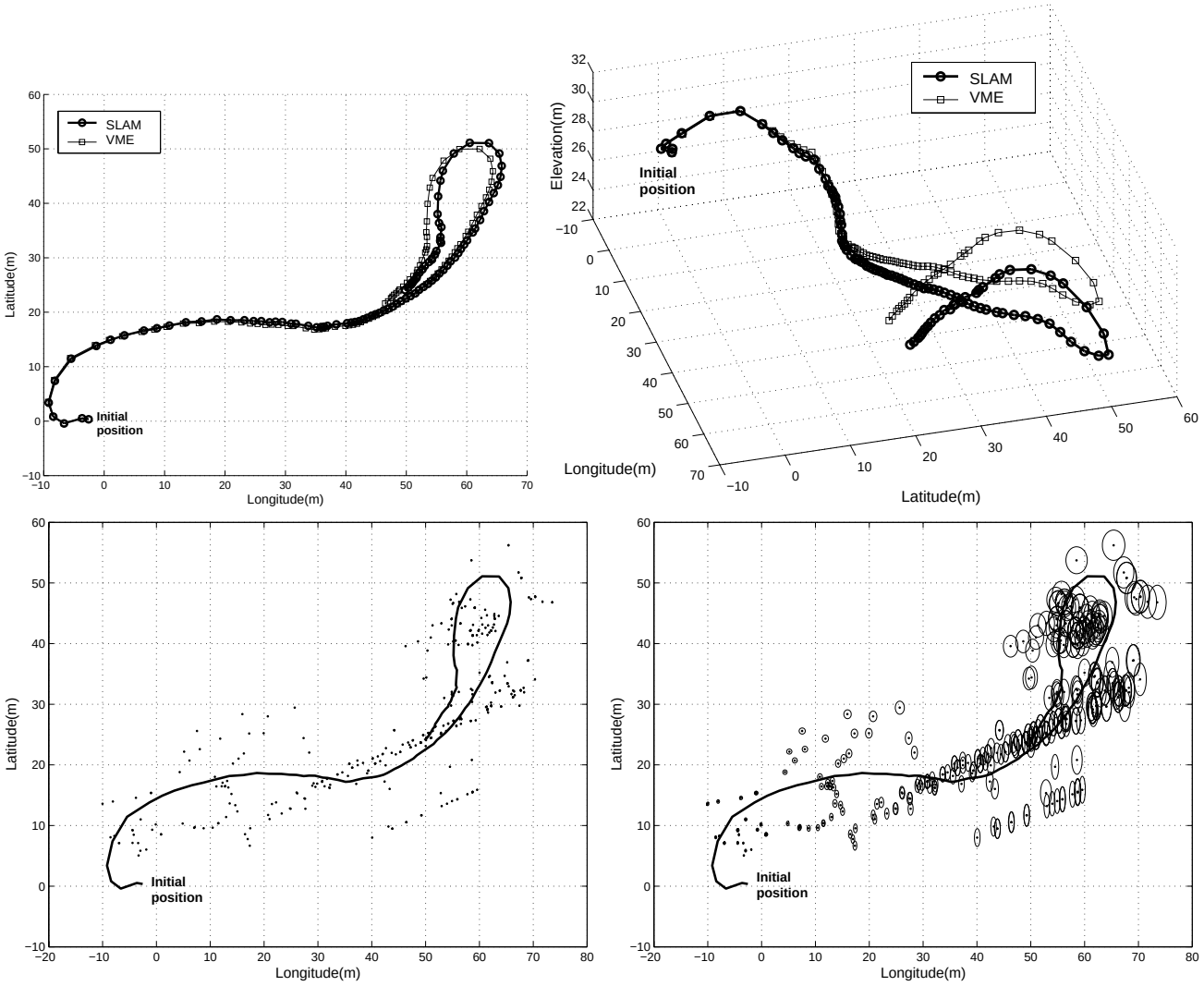

Fig. 40. Another trajectory reconstructed with a sequence of 100 images; 320 landmarks have been mapped (the magnification factor of the uncertainty ellipses in the bottom-right image is here 20).

originate from the same sensor, and are therefore not fully uncorrelated, might raise some convergence problems over a long time, especially as calibration bias is not negligible; we were, however, not able to detect such problems in our experiments. Anyhow, our algorithms for the prediction stage (VME) or the landmark mapping could easily be integrated with any other positioning sensor in a Kalman filter framework. Finally, our approach is intrinsically limited in altitude by the use of a stereovision bench. Our experiments showed that a baseline/depth ratio (the well-known b/H ratio in aerial imagery) of $1/15$ is big enough to allow centimeter accuracy positioning. We believe that our algorithms will have similar performances until b/H ratios as small as $1/30$ (i.e. altitudes around 70 m), but this has to be confirmed experimentally.

Other issues remain to be considered before considering the exploitation of the built maps by a ground rover. Especially, at the considered map resolutions, DEMs are not suited to represent verticals and overhangs present in the environment (see how bad the trees look in Figure 43). Some pre-processing is required in order to detect such situations before fusing the data in the DEM.

10. Conclusions

We have presented the current status of our project, insisting on developments related to flight control and terrain mapping. The project is ongoing, and now focuses on integration.

- Integration of the control laws during actual flights will be achieved in a progressive manner, thanks to

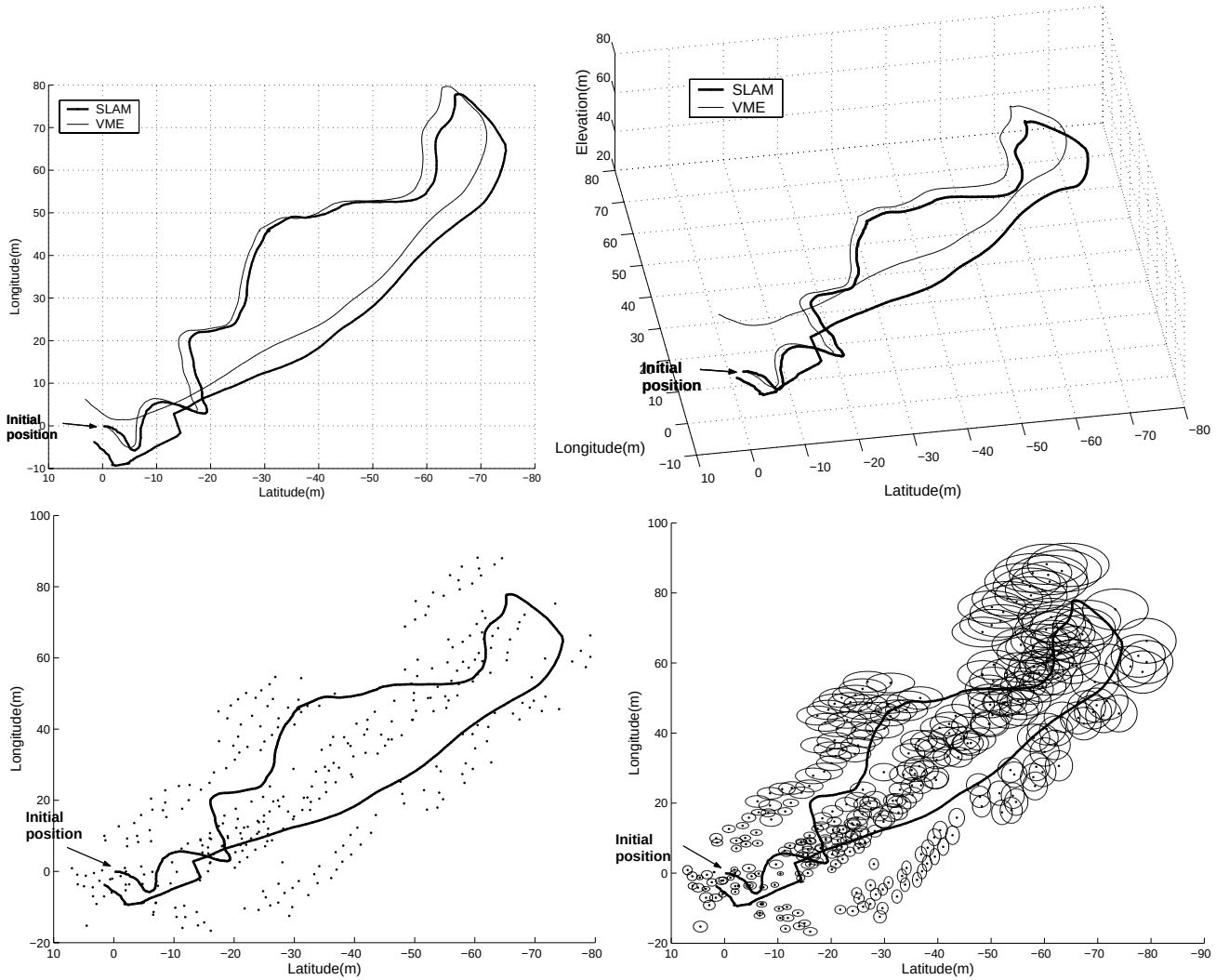

Fig. 41. Another result of our SLAM implementation with a sequence of 400 stereovision pairs. Top images show the reconstructed trajectory in orthogonal projection and in 3D. Bottom images show the 340 landmarks mapped, with 1σ uncertainty ellipses (left, real scale; right, magnified by a factor of 40).

the electronic switch module that allows a mix operator/automatic control. Actual flight data will help to refine some of the involved parameters, in both the model and control laws definition.

- Integration of the mapping algorithms is rather a matter of software engineering (besides the evaluation of a compressed SLAM approach performances).

- However, the most interesting aspects come when tackling the integration of control and mapping issues. Fusion of the blimp state parameters, considering also the control inputs, with the state as measured by the SLAM algorithms will allow us to enhance the precision of absolute positioning, allowing the achievement of long trajectories. Autonomous mapping of a given area is then achievable, thanks to an "exploration planner" that will send reference trajectories to the motion planner, on the basis of the current state of the built map.

Besides these integrations related to the blimp autonomy, the fusion of data acquired from a ground rover with the aerial map is currently under way, as a first step to cooperative air/ground robotics. Most of the difficulties rely here on the registration of the ground data with respect to the existing map; the viewpoints, the resolution, and the precision of the data are indeed extremely different between air and ground data.

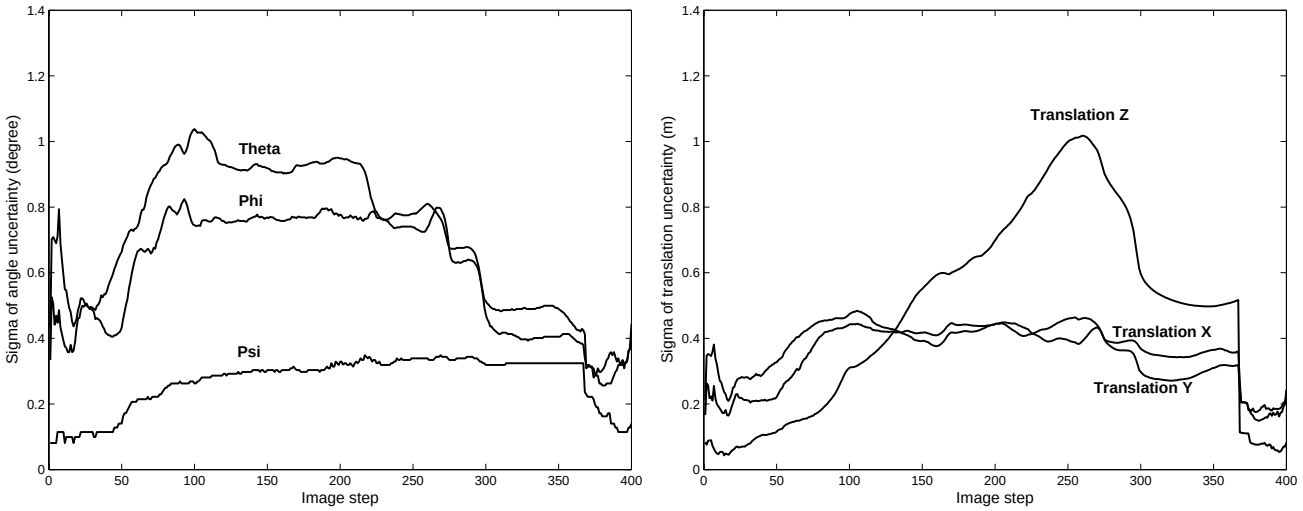

Fig. 42. Evolution of the standard deviations of the robot position parameters during the flight shown in Figure 41. A drastic decrease of the variances appears in the end, when several landmarks are re-perceived.

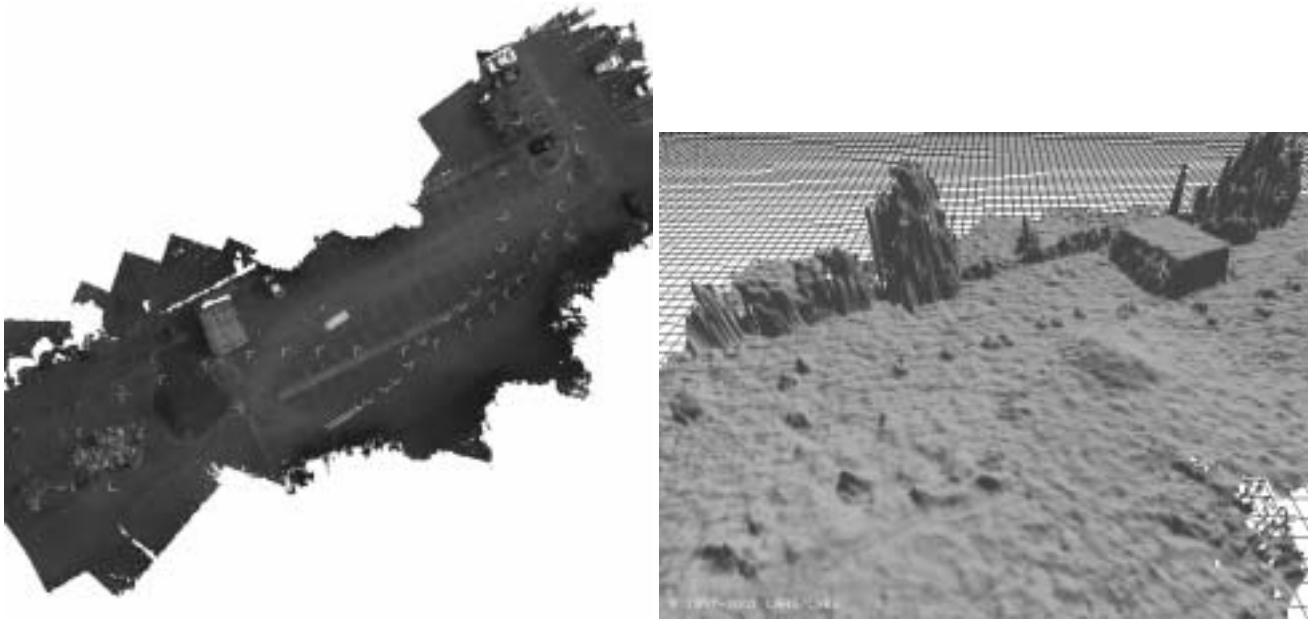

Fig. 43. The DEM computed with 100 images, positioned according to the trajectory of Figure 40: orthoimage and 3D view of the bottom-left area. The map covers an area of about 3500 m^2.

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Video	A teleoperated flights of Karma
2	Video	Replay of a flight from GPS and attitude logged informations
3	Video	Illustration of our simultaneous localization and mapping approach
4	Video	Simulation of autonomous flight control

Acknowledgments

This work is partially supported by the COMETS IST project (No 34304) of the Fifth European Framework Program, and by the Midi Pyrénées region. The authors are particularly grateful to Jean-Paul Berry and Arnaud Jacquet, who helped to conceive and realize all the Karma hardware and software architecture.

References

AIAA. 2001. *Proceedings of the 14th Lighter-Than-Air Convention and Exhibition*, Akron, OH.

Airship Association. 2002. *Proceedings of the 4th International Airship Convention and Exhibition*, Cambridge, UK.

Alami, R., Chatila, R., Fleury, S., Ghallab, M., and Ingrand, F. 1998. An architecture for autonomy. *Special Issue on Integrated Architectures for Robot Control and Programming, International Journal of Robotics Research* 17(4):315–337.

Azinheira, J. R., de Paiva, E., Ramos, J., Bueno, S. S., Bergerman, M., and Varella Gomes, S. B. 2001. Extended dynamic model for Aurora robotic airship. *Proceedings of the 14th AIAA Lighter-Than-Air Conference and Exhibition*, Akron, OH.

Bamberger, Y. 1981. *Mecanique de l'ingenieur 1, systemes de corp rigides*. Hermann, Paris.

Barnes, D., Summers, P., and Shaw, A. 2000. An investigation into aerobot technologies for planetary exploration. *Proceedings of the 6th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, Noordwijk, the Netherlands.

Bestaoui, Y. and Hima, S. 2001. Some insight in path planning of small autonomous blimps. *Archives of Control Sciences, Polish Academy of Sciences* 11(3–4):21–49.

Bonnet, A. 2003. Identification des coefficients aérodynamiques du dirigeable AS500 du LAAS. Technical report, Etude Hydro-Aérodynamique.

Bryson, A. E. 1954. Evaluation of the inertia coefficients of the cross section of slender body. *Journal of the Aeronautical Sciences—Reader's Forum* 21(6):424–427.

Canudas de Wit, C., Khennouf, H., Samson, C., and Sordalen, O. J. 1993. *Nonlinear Control Design for Mobile Robots*, Y.F. Zheng, editor. World Scientific, Singapore.

Chatila, R., and Laumond, J.-P. 1985. Position referencing and consistent world modeling for mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, St Louis, MO, pp. 138–145.

Deans, M., and Herbert, M. 2001. Experimental comparison of techniques for localization and mapping using a bearings only sensor. *Experimental Robotics VII*, Lecture Notes in Computer Science, S. Singh, editor. Springer-Verlag, Berlin.

de Paiva, E., Bueno, S., Varella Gomes, S., Bergerman, M., and Ramos, J. 1999. A control system development environment for Aurora's semi-autonomous robotic airship. *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI.

Di Marco, M., Garulli, A., Lacroix, S., and Vicino, A. 2001. Set membership localization and mapping for autonomous navigation. *International Journal of Robust and Nonlinear Control* 11(7):709–734.

Dissanayake, G., Newman, P.M., Durrant-Whyte, H.-F., Clark, S., and Csorba, M. 2001. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation* 17(3):229–241.

Dufournaud, Y., Schmid, C., and Horaud, R. 2000. Matching images with different resolutions. *International Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, pp. 612–618.

Elfes, A., Bueno, S. S., Bergerman, M., Ramos, J. G., and Varella Gomes, S. B. 1998. Project AURORA: development of an autonomous unmanned remote monitoring robotic airship. *Journal of the Brazilian Computer Society* 4(3):70–78.

Elfes, A., Bergerman, M., Carvalho, J. R. H., Carneiro de Paiva, E., Ramaos, J. J. G., and Bueno, S. S. 1999. Air–ground robotic ensembles for cooperative applications: concepts and preliminary results. *Proceedings of the 2nd International Conference on Field and Service Robotics*, Pittsburgh, PA, pp. 75–80.

Elfes, A., Bergerman, M., and Bueno, S. 2001. The potential of robotic airships for planetary exploration. *Proceedings of the 10th International Conference on Advanced Robotics*, Budapest, Hungary, pp. 131–138.

Faugeras, O., Vieville, T., Theron, E., Vuillemin, J., Hotz, B., Zhang, Z., Moll, L., Bertin, P., Mathieu, H., Fua, P., Berry, G., and Proy, C. 1993. Real-time correlation-based stereo: algorithm, implementations and application. Technical Report RR 2013, INRIA.

Fossen, T. 1996. *Guidance and Control of Ocean Vehicles*. Wiley, New York.

Guivant, J., and Nebot, E. 2001. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation* 17(3):242–257.

Haralick, R. M. 1994. Propagating covariances in computer vision. *International Conference on Pattern Recognition*, Jerusalem, Israel, pp. 493–498.

Haralick, R., Joo, H., Lee, C.-N., Zhuang, X., Vaidya, V. G., and Kim, M. B. 1989. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics* 19(6):1426–1446.

Harris, C., and Stephens, M. 1988. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, Manchester, UK, pp. 147–151.

Hygounenc, E., and Souères, P. 2002. Automatic airship con-

trol involving backstepping techniques. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia.

Hygounenc, E., and Souères, P. 2003. Lateral path following GPS-based control of a small-size unmanned blimp. *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan.

Hygounenc, E., Souères, P., and Lacroix, S. 2001. Developments on autonomous airship control at LAAS/CNRS. *Proceedings of the 14th AIAA Lighter-Than-Air Systems Convention and Exhibition*, Akron, OH.

Jung, I.-K., and Lacroix, S. 2001. A robust interest point matching algorithm. *Proceedings of the 8th International Conference on Computer Vision*, Vancouver, Canada.

Kieffer, M., Jaulin, L., Walter, E., and Meizel, D. 2000. Robust autonomous robot localization using interval analysis. *Reliable Computing* 6(3):337–362.

Kokotovic, P., Krstic, M., and Kanellakopoulos, I. 1995. *Nonlinear and Adaptive Control Design*. Wiley Interscience, New York.

Kungl, P., Schlenker, M., and Kröplin, B. 2001. Research and testing activities with the solar powered airship Lotte within the scope of the airship research group at the University of Stuttgart. *Proceedings of the 14th AIAA Lighter-Than-Air Conference and Exhibition*, Akron, OH.

Leonard, J. J., and Durrant-Whyte, H. F. 1991. Simultaneous map building and localization for an autonomous mobile robot. *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Osaka, Japan.

Leonard, J. J., and Feder, H. J. S. 2000. A computationally efficient method for large-scale concurrent mapping and localization. *Proceedings of the 9th International Symposium on Robotics Research*, Salt Lake City, UT, pp. 169–176.

Lewis, D. J., Lipscombe, J. M., and Thomasson, P. G. 1984. The simulation of remotely operated underwater vehicles. *Proceedings of the ROV '84 Conference and Exposition*, San Diego, CA, the Marine Technology Society.

Lindeberg, T. 1998. Feature detection with automatic scale selection. *International Journal on Computer Vision* 30(2):79–116.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. *Proceedings of the 7th International Conference on Computer Vision*, Kerkyra, Corfu, Greece, pp. 1150–1157.

Mallet, A., Lacroix, S., and Gallo, L. 2000. Position estimation in outdoor environments using pixel tracking and stereovision. *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, pp. 3519–3524.

Matthies, L. 1992. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. *IEEE International Conference on Computer Vision and Pattern Recognition*, Champaign, IL, pp. 765–768.

Mikolajczyk, K., and Schmid, C. 2001. Indexing based on scale invariant interest points. *Proceedings of the 8th International Conference on Computer Vision*, Vancouver, Canada, pp. 525–531.

Moutarlier, P., and Chatila, R. 1991. Incremental free-space modelling from uncertain data by an autonomous mobile robot. *International Workshop on Intelligent Robots and Systems*, Osaka, Japan, pp. 1052–1058.

Munk, M. M. 1922. Notes on aerodynamic forces 3: the aerodynamic forces on airships. Technical Report 106, NACA Technical Notes.

Olson, C., Matthies, L., Schoppers, M., and Maimone, M. 2001. Stereo ego-motion improvements for robust rover navigation. *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 1099–1104.

Schmid, C., Mohr, R., and Bauckhage, C. 1998. Comparing and evaluating interest points. *Proceeding of the 6th International Conference on Computer Vision*, Bombay, India, pp. 230–235.

Se, S., Lowe, D., and Little, J. 2001. Local and global localization for mobile robots using visual landmarks. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, pp. 414–420.

Smith, R., Self, M., and Cheeseman, P. 1987. A stochastic map for uncertain spatial relationships. *Robotics Research: The 4th International Symposium*, Santa Cruz, CA, pp. 468–474.

Thrun, S., Fox, D., and Burgard, W. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots* 5:253–271.

Thrun, S., Burgard, W., and Fox, D. 2000. A real-time algorithm for mobile robot with applications to multi-robot and 3d mapping. *IEEE International Conference on Robotics and Automation*, San Francisco, CA.

Thwaites, B. 1960. *Incompressible Aerodynamics*, Oxford University Press, Oxford, p. 415.

Turner, A. 2000. Development of a semi-autonomous control system for the uva solar airship Aztec. *Proceedings of the 3rd International Airship Convention and Exhibition*, Friedrichshafen, Germany.

Wijk, O., and Christensen, H.I. 2000. Triangulation based fusion of sonar data for robust robot pose tracking. *IEEE Transactions on Robotics and Automation* 16(6):740–752.

Wimmer, D.-A., and Well, K. H. 2001. Instrumentation, identification and control of airship Lotte. *Proceedings of the 14th AIAA Lighter-Than-Air Conference and Exhibition*, Akron, OH.

Zabih, R., and Woodfill, J. 1998. A non-parametric approach to visual correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* submitted.