An adaptive vision-based autopilot for mini flying machines guidance, navigation and control

**Farid Kendoul · Kenzo Nonami · Isabelle Fantoni ·
Rogelio Lozano**

Abstract The design of reliable navigation and control systems for Unmanned Aerial Vehicles (UAVs) based only on visual cues and inertial data has many unsolved challenging problems, ranging from hardware and software development to pure control-theoretical issues. This paper addresses these issues by developing and implementing an adaptive vision-based autopilot for navigation and control of small and mini rotorcraft UAVs. The proposed autopilot includes a Visual Odometer (VO) for navigation in GPS-denied environments and a nonlinear control system for flight control and target tracking. The VO estimates the rotorcraft ego-motion by identifying and tracking visual features in the environment, using a single camera mounted on-board the vehicle. The VO has been augmented by an adaptive mechanism that fuses optic flow and inertial measurements to determine the range and to recover the 3D position and velocity of the ve-

hicle. The adaptive VO pose estimates are then exploited by a nonlinear hierarchical controller for achieving various navigational tasks such as take-off, landing, hovering, trajectory tracking, target tracking, etc. Furthermore, the asymptotic stability of the entire closed-loop system has been established using systems in cascade and adaptive control theories. Experimental flight test data over various ranges of the flight envelope illustrate that the proposed vision-based autopilot performs well and allows a mini rotorcraft UAV to achieve autonomously advanced flight behaviours by using vision.

Keywords Visual navigation · Adaptive control · Rotorcraft UAV · Visual odometry · Visual servoing

F. Kendoul (✉) · K. Nonami
Robotics and Control Lab., Electronics and Mechanical
Engineering Dept., Chiba University, 263-8522, Chiba City, Japan
e-mail: fkendoul@restaff.chiba-u.jp

K. Nonami
e-mail: nonami@faculty.chiba-u.jp

I. Fantoni · R. Lozano
Laboratoire Heudiasyc, UMR CNRS 6599, Computer Science
Dept., Universite de Technologie de Compiegne, 60200,
Compiègne, France

I. Fantoni
e-mail: ifantoni@hds.utc.fr

R. Lozano
e-mail: rlozano@hds.utc.fr

1 Introduction

Recently, there is a growing interest in developing fully autonomous UAVs and Micro Air Vehicles (MAVs) for military and civil applications. The design of sensing, navigation and control systems is a crucial step in the development of such autonomous flying machines. Position, altitude and orientation measurements are usually sufficient for the control of UAVs operating at high altitudes. Therefore, conventional avionics that include GPS and IMU provide the required information for flight control and waypoint navigation. On the other hand, mini and micro UAVs are designed to operate at low altitudes in cluttered environments. To achieve realistic missions in such complex environments, the flight controller requires precise estimation of both the vehicle pose and its surrounding environment. Among many sensors for environment mapping and obstacles detection, ultrasonic sensors, Laser Range Finder (LRF), radar and vision have distinct advantages when applied to UAVs. For medium-size

UAVs such as the RMAX helicopter, LRF is widely used for obstacles avoidance (Scherer et al. 2008). We are however, interested in enabling mini UAVs and MAVs to achieve advanced flight behaviors in cluttered environments. The limited payload of these small platforms precludes the use of conventional and standard sensors. Therefore, small cameras, which are available today at low-cost, are attractive sensors for mini UAVs flying near the ground in hazardous environments. Indeed, vision provides a viable and useful solution to sensing on UAVs because cameras are too light to fit the limited payload capabilities of MAVs. Furthermore, visual sensors are passive and contain rich information about the vehicle motion and the environment structure. Unlike GPS, which does not work in the shadow of satellite visibility, vision works in a cluttered urban environment or even in indoor spaces.

1.1 Related work on visual aerial navigation

Although computer vision has many benefits for UAVs control and guidance, it presents many challenges too. Indeed, the design of reliable and real-time vision algorithms is a complex problem. Moreover, synthesizing robust flight controllers that are based on visual cues is another challenge. Despite these challenging issues, promising results have been obtained using computer vision for aerial vehicles control and navigation. Potential applications of computer vision for UAVs include pose estimation, landing, objects detection and tracking, mapping and localization, obstacles avoidance, etc.

In several UAVs projects, vision is used as a method to sense the aircraft ego-motion. Stereo visual odometers for relative position estimation were implemented in the CMU helicopter (Amidi et al. 1999) and the USC's robotic helicopter. In Kendoul et al. (2009a) and Kanade et al. (2004), structure from motion approach has been applied to recover the ego-motion of autonomous micro air vehicles. Vision has also been applied to recover UAVs pose with respect to some artificial marks like the work presented in Altug et al. (2005). Some researchers have also developed vision systems that can estimate the UAV attitude by detecting the horizon line (Ettinger et al. 2002).

The use of vision for autonomous landing has been actively researched. In the BEAR project at the university of California Berkeley, a vision system that uses multiple view geometry has been developed to land an autonomous helicopter on a moving deck (Shakernia et al. 2002). In Saripalli et al. (2003), the authors proposed a vision-based strategy that allows the AVATAR helicopter to land on a slowly moving helipad with known shape. Stereo vision is also used to detect safe landing area and to achieve soft landing (Johnson et al. 2005; Yu et al. 2007).

There are also some applications of vision for UAVs simultaneous localization and map building (visual SLAM)

(Lacroix et al. 2002; Caballero et al. 2009; Kima and Sukkarieh 2007).

Computer vision is also used as the primary sensor for objects detection and tracking like for windows tracking (Mejias et al. 2006; Proctor et al. 2006), moving target tracking (Calise et al. 2007), road following (Frew et al. 2004), etc.

Most of previous techniques rely on the reconstruction of the vehicle's state vector which is then used in the control loop. It is also important to note that most reported experimental results have been obtained using helicopters that weigh several kilograms.

Recently, many researchers have been interested in using image motion or optic flow for UAVs control and navigation without recovering the explicit motion of the vehicle. These techniques are inspired from insects like honeybees and flies which rely heavily on optic flow for navigation and flight control (Srinivasan et al. 1996; Tammero and Dickinson 2002).

Barrows (1999) designed a VLSI optic flow sensor for use in a MAV, which was tested on a small glider, and later applied for altitude control of a small aircraft (Green et al. 2003). Ruffier and Franceschini (2005) developed an optic flow-based autopilot to control a tethered 100-g helicopter. In later work (Serres et al. 2008), another vision-based system was used for speed control and lateral obstacle avoidance. Another interesting work has been done by Zufferey and Floreano (2006) who have developed 30-g and 10-g microflyers that can avoid walls using optic flow. A combined optic flow and stereo-based navigation system has been implemented on the AVATAR helicopter by Hrabar et al. (2005) to demonstrate obstacles avoidance in an urban environment. In (2004), Chahl et al. designed and implemented an optic flow-based system for UAVs landing. In recent work, Garratt and Chahl (2008) conducted experiments on a Eagle helicopter and used optic flow and GPS velocity for height estimation, terrain following and lateral motion control.

While most of reviewed works have been done using fixed-wing UAVs or small-scale helicopters, little experimental results have been reported on vision-based control of mini rotorcraft UAVs that weigh less than one kilogram like the quadrotor helicopter. Most existing works on visual servoing of mini quadrotor UAVs have focused on basic stabilization or hovering (Altug et al. 2005; Fowers et al. 2007; Guenard et al. 2008) and indoor flights in structured environments (Zufferey and Floreano 2006; Ahrens et al. 2009; Herisse et al. 2008). There are also other good examples of hovering miniature aerial robots using vision (Yamada et al. 2001; Mori et al. 2007).

There also has been some work that addressed the navigation of mini quadrotor UAVs in indoor environments by

using active range sensors. In Roberts et al. (2007), infrared and ultrasound sensors are used to control a quadrotor in a structured test environment (a 7×6-m room). He et al. (2008) developed an autonomous quadrotor that uses a lightweight LRF for localization and navigation in GPS-denied indoor environment. Their system allows a miniature quadrotor to reach a given location by following a path which is computed from a predefined map.

In contrast to this approach, our system uses a passive imaging sensor and is suitable to be used indoors and outdoors on less structured environments.

1.2 Description of the proposed vision-based autopilot

In this paper, we present an embedded autopilot which uses information extracted from real-time images for navigation and flight control (Fig. 1). Our objective in this research is to extend previous findings on vision-based navigation in order to allow mini rotorcraft UAVs to navigate within unknown indoor and outdoor environments and to achieve more advanced flight behaviors like trajectory tracking, precise hovering, vertical landing, target tracking, etc. Furthermore, the

Fig. 1 Visual navigation and control of small rotorcraft UAVs using vision

system is needed to meet the payload requirements of miniature rotorcraft that weigh less than 1 kg. In order to reach this goal, we have dealt with both theoretical and practical aspects of vision-based control of UAVs. Indeed, we have designed and implemented a vision-based autopilot that includes a real-time vision algorithm for features tracking and image motion integration, an adaptive visual odometer for range and motion estimation, and a hierarchical nonlinear controller for guidance and 3D flight control.

Functionally, the vision algorithm computes in real-time optic flow, tracks visual features and estimates the travelled flight distance in terms of total image displacement (in pixels). These visual measurements are then fused with IMU data in order to overcome the non-desired rotation effects. Indeed, it is difficult to sense UAV translation with vision alone since image displacements also occur with aircraft rotation (translation-rotation ambiguity). Another ambiguity is the scale factor or the unknown range that can not be estimated from visual measurements alone. This ambiguity is resolved by exploiting the adaptive control tools to identify the scale factor or range from optic flow and accelerometers data. The real aircraft position, velocity and height are then recovered. These estimates are then, used by a multipurpose nonlinear controller for autonomous 3D flight control.

Although previous works have shown that visual cues can lead to an autonomous flight, our work has extended this finding in a number of areas. The proposed autopilot is based on only two lightweight sensors, a small single camera and a low-cost IMU without using GPS. A major novelty in our work is that we have demonstrated the possibility and feasibility of recovering the range and aircraft motion from optic flow and IMU data, thereby leading to advanced flight behaviors. Unlike other works, our system has been implemented and demonstrated outdoors and indoors using a miniature unstable rotorcraft that weighs less than 700 grams. A final new contribution of this work is that we have developed a 3D flight nonlinear controller and proved its asymptotic stability and robustness even in the presence of estimation errors in the visual odometer.

The remainder of this paper is organized as follows. Section 2 describes the vision algorithm and its implementation. In Sect. 3, we explain the mathematical basis of the adaptive visual odometer and discuss how range and UAV motion can be estimated by accurately fusing optic flow with IMU data. The design of a nonlinear flight controller is presented in Sect. 4, followed by the stability analysis of the closed-loop system. The platform hardware and real-time implementation of the vision and control algorithms are described in Sect. 5. Experimental results for a quadrotor MAV are presented and discussed in Sect. 6. Finally, we provide in Sect. 7 some conclusions and a summary of the obtained results.

2 Vision algorithm for features tracking and image motion integration

The main objective of this vision algorithm is to extract useful information, from the video of an on-board camera, about the MAV motion and its environment. The proposed approach is an incremental procedure that relies on features tracking in order to provide visual measurements about the aircraft velocity and its position relative to its initial location.

2.1 Features selection and tracking

The tracking algorithm consists of a Lucas-Kanade (LK) tracker which is based on gradient-based optic flow computation. In our implementation, we have used the OpenCV library that contains efficient implementation of the Shi-Tomasi (Shi and Tomasi 1994) algorithm for good features selection and the pyramidal LK (Lucas and Kanade 1981) algorithm for features tracking.

The outputs of this algorithm are the positions ($x_i(t)$, $y_i(t)$) of the tracked features in the image frame. We have slightly modified that algorithm in order to provide also estimates about the optic flow at each feature location.

2.2 Estimation of the rotorcraft pseudo-motion

A vision system can estimate rotorcraft pseudo-motion[1] by tracking stationary objects (features) in the surrounding environment. Features are displaced in consecutive images as the rotorcraft moves. In the previous subsection, we have briefly described an effective algorithm for accurately measuring this image displacement (feature position and feature velocity or optic flow). There are many approaches for interpreting these visual estimates in order to extract useful information for navigation. The traditional approach, known as the Structure-From-Motion (SFM) problem consists in recovering both the camera ego-motion and scene structure using correspondences of tracked features (Kanade et al. 2004) or computed optic flow (Kendoul et al. 2009a). Another approach, inspired from biology, uses directly optic flow for reactive navigation without any reconstruction of the motion and structure.

Here, we propose another approach which consists in tracking few features in a small image area and integrating over space and time these measurements in order to provide one robust estimate about optic flow and the total image displacement. The approach is motivated by the fact that only few useful and reliable measurements will be processed by

[1]Pseudo-motion means here, motion in the image frame which is expressed in pixels.
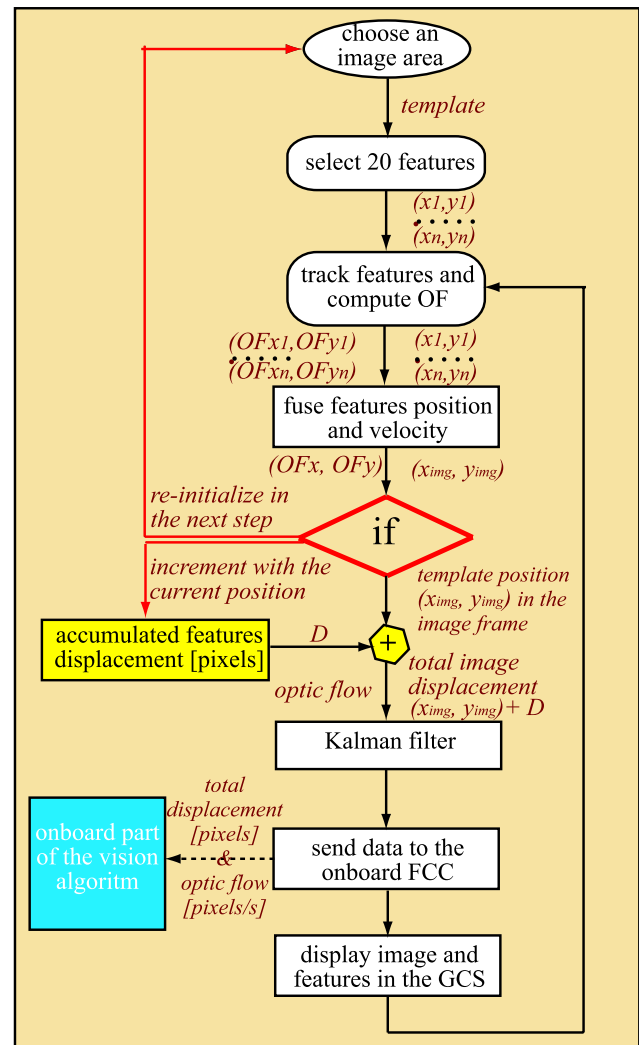
Fig. 2 Block diagram of the real-time vision algorithm

the onboard flight computer, thereby reducing the computation time. Figure 2 shows the main steps of our vision algorithm.

In the beginning, about 20 features are selected in some given 50×50 image area that can be considered as a "target template" which is initially chosen at the image center. These features are then tracked in the successive images, and the position of the "target template" is simply computed by taking the mean of the tracked features position. The "target template" velocity or optic flow is also calculated by taking the mean of the tracked features velocity. As the rotorcraft moves, older features leave the Field Of View (FOV) and new features enter the FOV. Therefore, an effective mechanism is required to deal with features disappearance and appearance and to continue motion estimation. The proposed approach consists in selecting a new set of features under the following conditions:

◇ *Target template is at the image border*: when the target template is about to go out of view causing features disappearance, a new target template is chosen at the image center and new features are selected in this template. In order to provide a pseudo position estimate relative to the initial MAV location, a displacement offset is incremented and added to the new template position measured in the image frame (see Figs. 2 and 3).

◇ *Features dispersion*: in an ideal tracking, the geometric configuration of tracked features should be almost constant. However, because of large attitude changes, poor matches and erroneous correspondences due to image noise, features may be scattered, dispersed and go out of the template. To overcome this problem, we compute the variance of features position and re-select new features in the same template when this variance exceeds some threshold.

◇ *Unreliable feature correspondences*: during real-time experiments, we have noticed that from time to time there are false feature correspondences which are mainly due to image quality and video transmission. We have thus, implemented a simple strategy which consists in rejecting feature matches when the difference between the previous and actual position (or optic flow) exceeds some threshold. This simple strategy turned out to be very effective in practice and it improved significantly the performance of the visual odometer.

◇ *Arbitrary target selection by the operator at the ground control station* (*GCS*): the developed GCS software allows the operator to chose any target or area in the image by just clicking on the desired location. Consequently, new features are selected in that area which is then tracked over time.

The obtained visual measurements are filtered and sent through WiFi to the onboard Flight Control Computer (FCC) where further processing is performed. More precisely, six data are sent to the FCC: the two components of optic flow $(\dot{x}, \dot{y})$ [pixels/s]; the target template position (x_{img}, y_{img}) [pixels] in the image frame; and the two components of the total image displacement (x, y) [pixels].

2.3 Rotation effects compensation

Image displacements occur with rotorcraft translation and orientation. Therefore, in order to sense aircraft translation which is essential for flight control, rotation effects must be eliminated from the measured image displacement and optic flow. Furthermore, this translation-rotation ambiguity is more significant in rotorcraft UAVs since the vehicle translation is a direct result of its attitude change. To overcome this problem and compensate the rotational components of op-

tic flow and image displacement, onboard IMU data (Euler angles (θ, ϕ) and angular rate data $(\omega_x, \omega_y, \omega_z)$) are used.

$$\begin{cases} x_t = x - (-f \tan \theta) \\ y_t = y - (f \tan \phi) \end{cases} \tag{1}$$

and

$$\begin{cases} \dot{x}_t = \dot{x} - \left(\frac{x_{img} y_{img}}{f} \omega_x - \frac{f^2 + x_{img}^2}{f} \omega_y + y_{img} \omega_z \right) \\ \dot{y}_t = \dot{y} - \left(\frac{f^2 + y_{img}^2}{f} \omega_x - \frac{x_{img} y_{img}}{f} \omega_y - x_{img} \omega_z \right) \end{cases} \tag{2}$$

where f is the camera focal length, (x_t, y_t) and $(\dot{x}_t, \dot{y}_t)$ are the translational components of image displacement and optic flow, respectively.

For effective compensation of the image displacement caused by rotation, the latency problem between the IMU measurements and visual estimates should be addressed. From experimental tests, we have identified a time-delay of about 0.25 s between the IMU and visual data. This delay is mainly due to the fact that images are processed at the ground station and the visual data are sent back to the embedded microprocessor for further processing. To overcome this problem, we applied a low-pass filter to the IMU data to filter the measurements from noise but also to introduce a 0.25 s delay in the IMU data. By using this simple strategy, the filtered (or delayed) IMU measurements were almost identical to vision estimates during pure rotation movements, thereby resulting in effective compensation of rotation effects (see Fig. 7).

At this stage, we have visual information $(x_t, y_t, \dot{x}_t, \dot{y}_t)$ about the rotorcraft position and velocity which are expressed in terms of incremented image displacement in [pixels] and optic flow in [pixels/s]. The true UAV position and velocity can not be directly deduced because of the range ambiguity. Indeed, the translational image displacement depends on both aircraft translation and relative distance (range) to the perceived objects.

3 Adaptive visual odometer for range and UAV motion estimation

This section presents an adaptive visual odometer that can estimate the true MAV position in [m] and velocity in [m/s] using the visual estimates $(x_t, y_t, \dot{x}_t, \dot{y}_t)$ and accelerometers measurements. Furthermore, the proposed odometer allows to recover the range, which is used for height control when the camera is looking downwards or obstacles avoidance for any other camera orientation.

3.1 Mathematical formulation of the adaptive visual odometer

Here, we show theoretically that it is possible to recover the range and the real UAV motion using only visual information from a single camera and accelerometers data.

Let us write the relation between the available visual measurements (x_t, y_t [pixels], $\dot{x}_t$, $\dot{y}_t$ [pixels/s]) and the state variables (X, Y, Z [m], V_x, V_y, V_z [m/s]) expressed in the inertial frame (see Fig. 3). If the camera is looking downwards, then we can write:

$$\begin{cases} x_t = f\frac{X}{Z} \\ y_t = f\frac{Y}{Z} \end{cases} \tag{3}$$

and

$$\begin{cases} \dot{x}_t = f\frac{V_x}{Z} + x_{img}\frac{V_z}{Z} \\ \dot{y}_t = f\frac{V_y}{Z} + y_{img}\frac{V_z}{Z} \end{cases} \tag{4}$$

It is clear from the above equations that if the range Z (here equivalent to height) can be estimated then the true MAV position and velocity can be recovered using (3) and (4). As a first step, we propose a real-time identification algorithm that will estimate the range Z under the following assumptions:

1. The height changes are small ($Z \approx cst$ and $V_z \approx 0$) when applying the identifier.
2. The terrain or relief is smooth such that it can be decomposed into flat segments.

In order to satisfy the first assumption, a static pressure sensor was added to the platform for enhancing the vertical motion estimation and control. However, this sensor does not provide height above the ground objects or relative distances to the perceived objects. Like GPS, it indicates the height with respect to the sea level, and when calibrated it can estimate the height with respect to the initial point. We would like to highlight that pressure sensor is very light (few grams), cheep (few dollars) and works in indoor and outdoor environments. Thus, it can be easily integrated into MAV and combined with vision to improve the control of vertical motion.

By considering the previous assumptions, equation (4) becomes:

$$\begin{cases} \dot{x}_t \simeq f\frac{V_x}{Z} \\ \dot{y}_t \simeq f\frac{V_y}{Z} \end{cases} \implies \begin{cases} \ddot{x}_t \simeq f\frac{a_x}{Z} \\ \ddot{y}_t \simeq f\frac{a_y}{Z} \end{cases} \tag{5}$$

where (a_x, a_y) are the MAV linear accelerations expressed in the inertial frame.

Many types of on-line parameter estimation techniques can be applied to estimate Z using the derivatives of op-
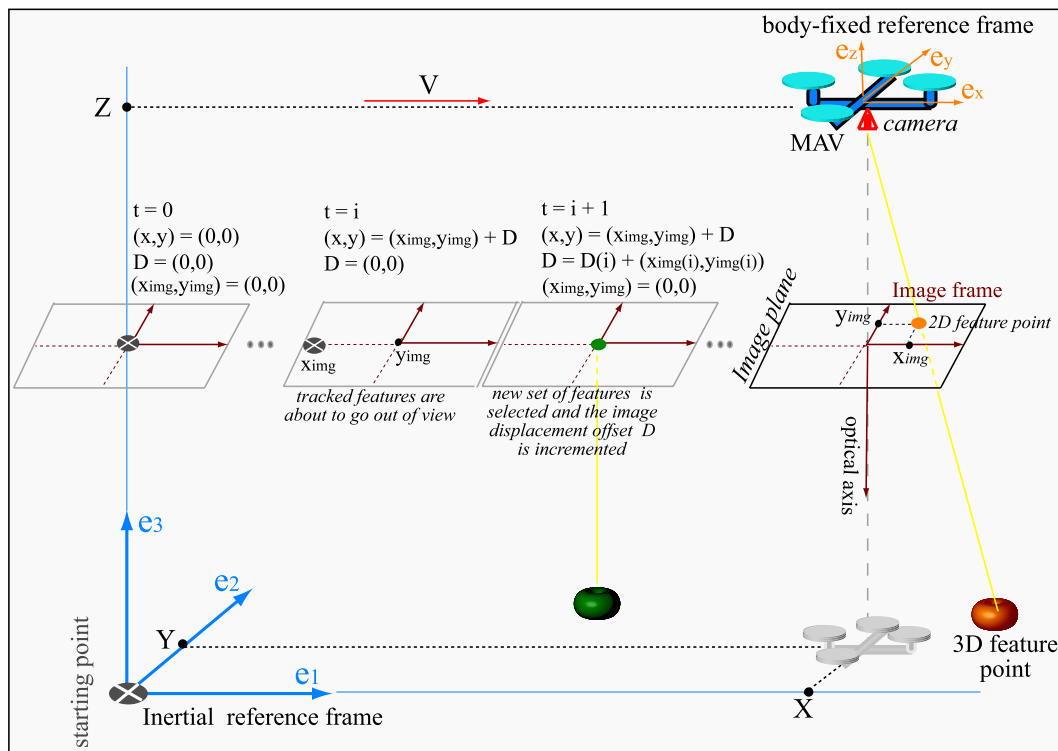
Fig. 3 The rotorcraft displacement in the inertial and the image frames. The horizontal MAV position (X, Y) in the inertial frame can be expressed in terms of the translational image displacement ((x_t, y_t) = (x, y) − (x_{rot}, y_{rot})) accumulated along the flight path

tic flow and linear accelerations (see (5)). Our odometer is based on the Recursive-Least-Squares (RLS) algorithm which presents many advantages for our application.

3.2 Generalities on the Recursive-Least-Squares algorithm

The choice of this method is essentially motivated by the fact that it is easy to implement and appropriate for real-time applications. In addition, it is robust with respect to noise and it has been widely used in parameter estimation in a recursive and nonrecursive form mainly for discrete-time systems (Ioannou et al. 1996; Goodwin and Sin 1984; Astrom and Wittenmark 1989; Landau et al. 1998).

The RLS method is simple to apply in the case where the unknown parameters appear in a linear form, such as in the linear parametric model:

$$y = b^T \gamma$$

where b^T is a constant vector which contains the unknown parameters and (y, γ) are the available output and input signals.

The RLS algorithm formula are obtained by minimizing some cost function and its expression is as follows (Ioannou et al. 1996):

$$
\begin{aligned}
\dot{\hat{b}} &= P \varepsilon \gamma \\
\dot{P} &= \beta P - P \frac{\gamma \gamma^T}{m^2} P, \quad P(0) = P_0 \\
m^2 &= 1 + \gamma^T \gamma \\
\varepsilon &= \frac{y(t) - \hat{b}^T \gamma(t)}{m^2}
\end{aligned}
\tag{6}
$$

The stability properties of the RLS algorithm depend on the value of the forgetting factor β.

In our implementation, we have used a robust version of the standard RLS algorithm that includes:

– *RLS with forgetting factor*: equation (6) has been modified to avoid an unbounded growth of he gain matrix $P(t)$ when the forgetting factor $\beta > 0$.
– *RLS with projection*: in many practical problems where b represents the physical parameters of a plant, we may have some a priori knowledge like the upper and/or lower bounds (b_{max}, b_{min}). Thus, we have modified the RLS algorithm to constrain the estimates search, thereby resulting in the following advantages: (1) speed-up the convergence, (2) reduce large transients, and (3) avoid some singular cases like $\hat{b} = 0$ which is very important for control design.
– *RLS with dead-zone*: the principal idea behind the dead-zone is to monitor the size of the estimation error and adapt only when the estimation error is large relative to the modelling error or disturbance. This means that adaptation or identification process is switched off when the normalized estimation error εm is small.

More details about robust RLS algorithms including their stability analysis can be found in Ioannou et al. (1996) and the references therein.

3.3 Application of RLS algorithm to range (height) estimation

Now, let us apply a robust RLS algorithm to estimate the height Z in (5). In a deterministic case, Z can be estimated using one of the following SISO subsystems "$\ddot{x}_t = f \frac{a_x}{Z}$" or "$\ddot{y}_t = f \frac{a_y}{Z}$". An interesting solution could consist in estimating Z for each SISO subsystem and then, fuse the two obtained estimates $(\hat{Z}_1, \hat{Z}_2)$ in a favorable manner. For more robustness against noise and external disturbances, we have applied the RLS algorithm with projection and dead-zone in order to estimate $\hat{Z}_1, \hat{Z}_2$ from available measurements $(\dot{x}_t, a_x)$, $(\dot{y}_t, a_y)$, respectively. The two estimates $(\hat{Z}_1, \hat{Z}_2)$ have then been fused using a weighted averaging method where the weights depend on the *Persistency Excitation* (PE) property of each signal $a_i, i = x, y$. This idea can be mathematically described as follows:

$$
\hat{Z} = \begin{cases} \frac{1}{w_1 + w_2}(w_1 \hat{Z}_1 + w_2 \hat{Z}_2) & \text{if } w_1 + w_2 \neq 0 \\ \hat{Z}_{-1} & \text{otherwise} \end{cases}
\tag{7}
$$

where the weights (w_1, w_2) are functions of the *PE* condition.

In order to apply the RLS method to estimate Z in (5), we need to write the preceding system in the following linear parametric form: $y = b^T \gamma$, and then the application of the modified robust RLS algorithm becomes straightforward. This can be done using the following formula:

$$
\begin{cases} y_1 = \frac{s \dot{x}_t}{(s + \lambda_1)}, & \gamma_1 = \frac{a_x}{(s + \lambda_1)}, & b_1 = \frac{f}{Z} \\ y_2 = \frac{s \dot{y}_t}{(s + \lambda_2)}, & \gamma_2 = \frac{a_y}{(s + \lambda_2)}, & b_2 = \frac{f}{Z} \end{cases}
\tag{8}
$$

where $\frac{1}{(s + \lambda_1)}$ and $\frac{1}{(s + \lambda_2)}$ are fist-order low-pass filters, which are used to avoid direct signals differentiation.

Theorem 1 *For $i = 1, 2$, the modified RLS algorithm with projection and dead-zone, applied to system (5) and (8), guarantees the following properties (Ioannou et al. 1996):*

(a) $\hat{b}_i, \dot{\hat{b}}_i \in \mathcal{L}_\infty$ *(bounded)*.
(b) $b_{min} \leq \hat{b}_i(t) \leq b_{max}, \forall t \geq 0$, *where $b_{min} > 0$ and $b_{max} > 0$ are a priori known lower and upper bounds of the parameter b (projection principle)*.
(c) $\lim_{t \to \infty} \hat{b}_i(t) = \bar{b}_i$ *where $\bar{b}_i$ is a positive constant*.
(d) *If $\gamma_i \in \mathcal{L}_\infty$ and γ_i is PE, then the estimate $\hat{b}$ converges exponentially to its true value[2] b.*

---

[2]In the presence of external disturbances, $\tilde{b}_i = b - \hat{b}_i$ converges exponentially to the residual set: $\{\tilde{b}_i / |\tilde{b}_i| \leq c(\rho_0 + \bar{d})\}$, where $\bar{d}$ is the disturbance upper bound, c is a positive constant and ρ_0 characterizes the dead-zone.

Rigorous and detailed proof of this theorem can be found in any book on identification techniques such as Ioannou et al. (1996).

Since the two estimates $(\hat{b}_1, \hat{b}_2)$ satisfy the properties in Theorem 1, it is then trivial to show that $\hat{Z}_1 = \frac{f}{\hat{b}_1}$, $\hat{Z}_2 = \frac{f}{\hat{b}_2}$ and $\hat{Z}$ in (7) are positive and satisfy also the properties (a)–(d) listed in Theorem 1. Once the height Z is identified, the aircraft horizontal position vector (X, Y) and velocity vector (V_x, V_y) can be recovered using (3) and (5):

$$
\begin{cases}
\hat{X} = \frac{x_t}{b} = \hat{Z}\frac{x_t}{f} \\
\hat{Y} = \frac{y_t}{b} = \hat{Z}\frac{y_t}{f} \quad \text{and} \\
\hat{Z} = \frac{f}{b}
\end{cases}
\quad
\begin{cases}
\hat{V}_x = \frac{\dot{x}_t}{b} = \hat{Z}\frac{\dot{x}_t}{f} \\
\hat{V}_y = \frac{\dot{y}_t}{b} = \hat{Z}\frac{\dot{y}_t}{f}
\end{cases}
\tag{9}
$$

Proposition 1 *The adaptive observer* (9) *is stable and the convergence of the state variables* $(\hat{X}, \hat{Y}, \hat{Z}, \hat{V}_x, \hat{V}_y)$ *is guaranteed. Furthermore, if the* PE *property is satisfied, then the identification errors* $(\tilde{X} = X - \hat{X}, \dots \tilde{V}_y = V_y - \hat{V}_y)$ *converge to zero.*

In fact, the observer stability is a direct consequence of the parameter identifier (RLS algorithm) stability. Indeed, the obtained position and velocity estimates satisfy the following properties:

– Since $\hat{b}(t) \in [b_{min}, b_{max}] \to \bar{b}$, then, from (9) we deduce that

$$
(\hat{X}, \hat{Y}, \hat{Z}, \hat{V}_x, \hat{V}_y) \to (X, Y, Z, V_x, V_y)b/\bar{b}
\tag{10}
$$

– Furthermore, if (a_x, a_y) satisfy the *PE* property, then,

$$
\begin{cases}
\hat{b} \to b \Rightarrow \tilde{b} \to 0 \\
(\tilde{X}, \tilde{Y}, \tilde{Z}, \tilde{V}_x, \tilde{V}_y) \to (0, 0, 0, 0, 0)
\end{cases}
\tag{11}
$$

3.4 Fusion of visual estimates, inertial and pressure sensor data

This last step of the visual odometer consists in fusing the visual estimates $(\hat{X}, \hat{Y}, \hat{Z}, \hat{V}_x, \hat{V}_y)$, inertial data (a_x, a_y, a_z) and pressure sensor measurements Z_{ps} in order to improve the odometer accuracy and robustness, reduce the noise and estimate the vertical velocity V_z. The data fusion is performed using a linear Kalman filter with choosing (X, Y, Z, V_x, V_y, V_z) as a state vector, $(\hat{X}, \hat{Y}, \hat{Z}, Z_{ps}, \hat{V}_x, \hat{V}_y)$ as a measurement vector and (a_x, a_y, a_z) as an input vector. The implementation of such Kalman filter is straightforward and thus further details are omitted here.

4 Nonlinear 3D flight controller design and stability

This section presents the design of a nonlinear flight controller and the analysis of the closed-loop system stability.

The objective is to design an effective and practical controller that considers plant system nonlinearities and coupling, but also the characteristics and specificities of the adaptive visual odometer estimates. Furthermore, the controller is required to guarantee good flight performance and robustness even in the presence of identification errors.

4.1 Rotorcraft dynamics modelling

Let us denote by $\xi = (X, Y, Z)$, $\upsilon = (V_x, V_y, V_z)$, $\eta = (\phi, \theta, \psi)$, $\dot{\eta} = (\dot{\phi}, \dot{\theta}, \dot{\psi})$ the position, translational velocity, orientation and angular rate vectors, respectively. Therefore, the dynamics of a rotorcraft UAV such as the quadrotor helicopter, used in this research, can be represented by the following mathematical model (Olfati-Saber 2001; Kendoul 2007):

$$
\begin{cases}
\ddot{\xi} = \frac{1}{m}uRe_3 - ge_3 \\
M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \Psi(\eta)^T \tau
\end{cases}
\tag{12}
$$

where u is the total thrust, τ is the torque vector, R and Ψ are the rotation and Euler matrices, respectively. The pseudo inertial matrix M is defined as $M(\eta) = \Psi(\eta)^T J \Psi(\eta)$, and C is given by $C(\eta, \dot{\eta}) = \Psi(\eta)^T J \dot{\Psi}(\eta) - \Psi(\eta)^T sk(\Psi(\eta)\dot{\eta}) J \Psi(\eta)$.

4.2 Flight controller design

Control design for rotorcraft UAVs is already a challenging task, especially when it comes to deal with unknown parameters and state variables estimation errors. To cope with these issues, we have proposed a hierarchical flight controller that exploits the structural properties of rotorcraft UAVs model given by (12). Our objective is to develop a 3D flight controller that performs well in practice[3] as well as in theory.[4]

To reach this goal, we have separated the aircraft model into two connected subsystems by decoupling the translational dynamics (outer-loop) and rotational dynamics (inner-loop). The time-scale separation between the two subsystems is made by considering the following transformations (or change of variables):

$$
\tau = J\Psi(\eta)\tilde{\tau} + \Psi^{-1}C(\eta, \dot{\eta})\dot{\eta}
$$
$$
\mu = \frac{1}{m}uR(\phi_d, \theta_d, \psi_d)e_3 - ge_3
\tag{13}
$$

where $\tilde{\tau}$ is a new torque vector, μ is an intermediary force vector and $(\phi_d, \theta_d, \psi_d)$ are desired roll, pitch and yaw angles.

---

[3]It is easy to implement and guarantees good flight performance.

[4]It considers system nonlinearities/coupling and guarantees the stability of the closed-loop system.

By defining the tracking errors $\chi = (\xi - \xi_d, \upsilon - \upsilon_d)^T \in \mathbb{R}^6$ and $e = (\eta - \eta_d, \dot{\eta} - \dot{\eta}_d)^T \in \mathbb{R}^6$, replacing η by $\eta_d + e$ in (12) and considering (13), the system (12) can be written in the following form (Kendoul et al. 2009b; Kendoul et al. 2008):

$$\begin{cases} \dot{\chi} = \underbrace{A_1\chi + B_1(\mu - \ddot{\xi}_d)}_{f(\chi,\mu,\ddot{\xi}_d)} + \underbrace{\frac{1}{m}uH(\eta_d, e_\eta)}_{\Delta(u,\eta_d,e_\eta)} \\ \dot{e} = A_2 e + B_2(\tilde{\tau} - \ddot{\eta}_d) \end{cases} \quad (14)$$

where $H(\eta_d, e_\eta) = (0, 0, 0, h_x, h_y, h_z)^T$ is a nonlinear interconnection term. The matrices $A_1 \in \mathbb{R}^{6\times6}$, $B_1 \in \mathbb{R}^{6\times3}$, $A_2 \in \mathbb{R}^{6\times6}$ and $B_2 \in \mathbb{R}^{6\times3}$ are defined in Kendoul et al. (2008).

The rotorcraft control problem is thus, formulated as the control of two cascaded subsystems which are coupled by a nonlinear term $\Delta(u, \eta_d, e_\eta)$. Some techniques have been proposed in the literature to control systems in cascade (Sontag 1988; Sepulcre et al. 1997). Here, we use partial passivation design to synthesize two stabilizing feedbacks $\mu = \alpha(\hat{\chi}, \ddot{\xi}_d)$, $\tilde{\tau} = \beta(e, \ddot{\eta}_d)$ such that the tracking errors (χ, e) are asymptotically stable for all initial conditions. The idea is to consider $\Delta(u, \eta_d, e_\eta)$ as a disturbance on the χ-subsystem which must be driven to zero, and stabilize independently the χ- and e-subsystems.

Since the χ-subsystem without the coupling $\Delta(.)$ and the e-subsystem in (14) are linear, we can use simple linear controllers such as PD or PID. Therefore, we synthesize two control laws

$$\begin{cases} \mu = -K_\chi\hat{\chi} + \ddot{\xi}_d, \quad K_\chi \in \mathbb{R}^{3\times6} \\ \tilde{\tau} = -K_e e + \ddot{\eta}_d, \quad K_e \in \mathbb{R}^{3\times6} \end{cases} \quad (15)$$

such that the matrices $A_\chi = A_1 - B_1K_\chi$ and $A_e = A_2 - B_2K_e$ are Hurwitz.

It is important to note that the control $\mu(\hat{\chi})$ is computed using the estimates of the adaptive visual odometer. Indeed, $\hat{\chi} = (\hat{\xi} - \xi_d, \hat{\upsilon} - \upsilon_d)^T$ where $\hat{\xi} = (\hat{X}, \hat{Y}, \hat{Z})^T$ and $\hat{\upsilon} = (\hat{V}_x, \hat{V}_y, \hat{V}_z)^T$.

Although A_χ and A_e are Hurwitz, the asymptotic stability of the closed-loop system can not be directly deduced because of the interconnection term $\Delta(\hat{\chi}, \eta_d, e_\eta)$ and the odometer estimation errors. The asymptotic stability of the closed-loop system and its robustness with respect to $\Delta(\hat{\chi}, \eta_d, e_\eta)$ are analyzed in the next subsection.

4.3 Closed-loop system stability and robustness

By substituting (15) into (14), the closed-loop system dynamics become

$$\begin{cases} \dot{\chi} = A_1\chi - B_1K_\chi\hat{\chi} + \Delta(\hat{\chi}, e) \\ \dot{e} = A_e e \end{cases} \quad (16)$$

In this section, the stability of the closed-loop system (16) is established, considering the interconnection term $\Delta(\hat{\chi}, e)$ and errors in the state vector estimation.

If the visual odometer estimates $(\hat{\xi}, \hat{\upsilon})$ converge to their true values (*no estimation errors*), then the tracking error $\hat{\chi}$ will become χ and system (16) can be written as follows:

$$\begin{cases} \dot{\chi} = (A_1 - B_1K_\chi)\chi + \Delta(\chi, e) = A_\chi\chi + \Delta(\chi, e) \\ \dot{e} = A_e e \end{cases} \quad (17)$$

Even for this simplified system it is difficult to prove the stability property because of the complex interconnection term $\Delta(\chi, e)$. However, a detailed demonstration of the stability property for system (17) can be found in our previous paper (Kendoul et al. 2009b).

This paper extends the previous results by proving the stability property of system (16) even in the presence of identification and estimation errors (i.e., $\hat{b} \to \bar{b} \neq b$ and $(\hat{\xi}, \hat{\upsilon}) = (\xi, \upsilon)b/\hat{b}$ which means that $\hat{\chi} \neq \chi$).

First, let us express the tracking error $\hat{\chi} = (\hat{\xi} - \xi_d, \hat{\upsilon} - \upsilon_d)^T$ as a function of the original tracking error $\chi = (\xi - \xi_d, \upsilon - \upsilon_d)^T$:

$$\hat{\chi} = \left(\frac{b}{\hat{b}}\xi - \xi_d, \frac{b}{\hat{b}}\upsilon - \upsilon_d\right)^T = \frac{b}{\hat{b}}\chi + \frac{b-\hat{b}}{\hat{b}}(\xi_d, \upsilon_d)^T \quad (18)$$

So, the outer-loop dynamics in (16) become:

$$\dot{\chi} = A_1\chi - B_1K_\chi\left[\frac{b}{\hat{b}}\chi + \frac{b-\hat{b}}{\hat{b}}(\xi_d, \upsilon_d)^T\right] + \Delta(\hat{b}, \chi, e)$$

$$= \underbrace{\left(A_1 - \frac{b}{\hat{b}}B_1K_\chi\right)}_{\hat{A}_\chi}\chi \underbrace{- \frac{b-\hat{b}}{\hat{b}}B_1K_\chi(\xi_d, \upsilon_d)^T}_{\varepsilon(\hat{b},\xi_d,\upsilon_d)}$$

$$+ \Delta(\hat{b}, \chi, e) \quad (19)$$

Now, the closed-loop system (16) can be expressed in the following form:

$$\begin{cases} \dot{\chi} = \hat{A}_\chi\chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d) + \Delta(\hat{b}, \chi, e) \\ \dot{e} = A_e e \end{cases} \quad (20)$$

The term $\varepsilon(\hat{b}, \xi_d, \upsilon_d)$ is mainly due to the parameter estimate error $b - \hat{b}$ (height estimation error). Indeed, if $\hat{b}$ converges to its true value b, then ε will converge to zero.

According to Sontag's theorem (Sontag 1988), the closed-loop system (20) is Globally Asymptotically Stable (GAS) provided that the outer-loop "$\dot{\chi} = \hat{A}_\chi\chi + \varepsilon(\hat{b}, z_d, \upsilon_d)$" and the inner-loop "$\dot{e} = A_e e$" are GAS and the trajectories $(\chi(t), e(t))$ are bounded. One of the major tools usually used to show the boundedness of connected system trajectories is the Input-to-State-Stability (ISS) property (Sontag 1988). The ISS property is a strong condition which

is often difficult to verify. For the system (20), the complexity is due to the interconnection term $\Delta(\hat{b}, \chi, e)$. Consequently, we propose a theorem that guarantees the GAS of the connected system (20) provided that the interconnection term $\Delta(\hat{b}, \chi, e)$ satisfies some relaxed conditions.

Theorem 2 *If the following three conditions hold, then all the solutions $\chi(t)$ and $e(t)$ of (20) are bounded.*

A1. *The equilibrium point $e = 0$ is GAS and Locally Exponentially Stable (LES).*

A2. *There exist a positive semi-definite radially unbounded function $V(\chi)$ and positive constants c_1 and c_2 such that for $\|\chi\| \geq c_1$*

$$
\begin{cases}
\frac{\partial V}{\partial \chi}[\hat{A}_\chi \chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d)] \leq 0 \\
\|\frac{\partial V}{\partial \chi}\| \|\chi\| \leq c_2 V(\chi)
\end{cases}
\tag{21}
$$

A3. *There exist a positive constant c_3 and one class-$\mathcal{K}$ function $\gamma(.)$, differentiable at $e = 0$, such that*

$$
\|\chi\| \geq c_3 \quad \Longrightarrow \quad \|\Delta(\hat{b}, \chi, e)\| \leq \gamma(\|e\|)\|\chi\|
\tag{22}
$$

If in addition, $\dot{\chi} = \hat{A}_\chi \chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d)$ is GAS, then the equilibrium point $(\chi, e) = (0, 0)$ is GAS.

The proof of Theorem 2 is done in Kendoul (2007).

Proposition 2 *The closed-loop system (20) is stable and the tracking error χ is bounded. Furthermore, if $\hat{b}(t)$ converges to b then, the equilibrium point $(\chi, e) = (0, 0)$ is GAS.*

Proof In order to prove the stability of the closed-loop system (16) or (20), we need to prove that the e-subsystem, the χ-subsystem and the coupling term $\Delta(.)$ satisfy the conditions A1, A2 and A3 respectively.

- A1. *e-subsystem stability*: Since the matrix A_e is Hurwitz, then, the e-subsystem (inner-loop) is GES which is stronger than the GAS property.
- A2. *χ-subsystem stability*: We are interested here, in the stability of the subsystem $\dot{\chi} = \hat{A}_\chi \chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d)$ without the interconnection term. It is clear from (19) that if $\hat{b} \to b$, the system $\dot{\chi} = \hat{A}_\chi \chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d)$ will be GAS. Indeed, for $\hat{b} \to b$, we have $\varepsilon(.) \to 0$ and $\hat{A}_\chi \to A_\chi$, which is Hurwitz. Consequently, the inequalities (21) hold.

Now, let us analyze the stability of the χ-subsystem in the presence of parameter estimate errors, i.e., $\hat{b} \to \bar{b} \neq b$. So, in the following, we will show that the term $\varepsilon(\hat{b}, \xi_d, \upsilon_d)$ is bounded, the matrix $\hat{A}_\chi = A_1 - \frac{b}{\hat{b}} B_1 K_\chi$ is Hurwitz and the tracking error $\chi(t)$ is bounded. Furthermore, we will prove that the assumption A2 of Theorem 2 is satisfied.

Let us recall the expression of the identification error-related term $\varepsilon(\hat{b}, \xi_d, \upsilon_d) = -\frac{b-\hat{b}}{\hat{b}} B_1 K_\chi (\xi_d, \upsilon_d)^T$. Since $\hat{b}(t) \in [b_{min}, b_{max}]$ (see Theorem 1) and the desired trajectories $(\xi_d(t), \upsilon_d(t))$ are bounded, then it is trivial to deduce that $\varepsilon(.)$ is also bounded. Then, there exists a positive constant d such that $d = \|\varepsilon(.)\|_\infty$.

We recall that the matrix $A_1 - B_1 K_\chi$ is Hurwitz and the term $\frac{b}{\hat{b}}$ is positive and bounded. Due to the structure of the matrices A_1 and B_1, then it is easy to show that the matrix $\hat{A}_\chi = A_1 - \frac{b}{\hat{b}} B_1 K_\chi$ is also Hurwitz. Therefore, the subsystem $\dot{\chi} = \hat{A}_\chi \chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d)$ is equivalent to the following differential equation: $\dot{\chi} - \hat{A}_\chi \chi = \varepsilon(t)$. The solution of the preceding differential equation is:

$$
\chi(t) = \chi(0)e^{\hat{A}_\chi t} + \int_0^t e^{\hat{A}_\chi(t-\tau)} \varepsilon(\tau) d\tau
\tag{23}
$$

From (23), we can show that after a finite time T, we obtain

$$
\|\chi(t)\| \leq \|\varepsilon(t)\| \leq d, \quad \forall t \geq T
\tag{24}
$$

This means that given any initial condition $\chi(0)$, the trajectories $\chi(t)$ of the subsystem "$\dot{\chi} = \hat{A}_\chi \chi + \varepsilon(\hat{b}, \xi_d, \upsilon_d)$" converge exponentially to a bounded ball with a radius d. Hence, inequalities (21) hold for $\|\chi\| \geq c_2 \triangleq d$.

- A3. $\Delta(.)$ *growth restriction*: Now, let us analyze if the interconnection term $\Delta(.)$ satisfies the growth condition (22). By recalling (14), we have

$$
\|\Delta(\hat{\chi}, e)\| = \frac{1}{m} |u(\hat{\chi})| \|H(\hat{\chi}, e)\|
\tag{25}
$$

where

$$
\begin{cases}
\|H(\hat{\chi}, e)\| = \sqrt{h_x^2 + h_y^2 + h_z^2} \\
|u(\hat{\chi})| = m\|\mu(\hat{\chi}) + ge_3\| \\
\quad = m\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2}
\end{cases}
\tag{26}
$$

Lemma 1 *Assume that the desired trajectories $(\xi_d(t), \upsilon_d(t))$ and their time-derivatives are bounded. Then, there exist positive constants r, k_1 and k_2 such that the collective thrust $u(\hat{\chi})$ and the coupling term $H(\hat{\chi}, e)$ satisfy the following properties*:

$$
\begin{cases}
|u(\chi)| \leq \begin{cases} k_1 \|\hat{\chi}\|, & \text{for } \|\hat{\chi}\| \geq r \\ k_1 r, & \text{for } \|\hat{\chi}\| < r \end{cases} \\
\|H(\hat{\chi}, e)\| \leq k_2 \|e\|
\end{cases}
\tag{27}
$$

The proof of Lemma 1 is given in Kendoul (2007).

Recalling (25) and (27), the interconnection term $\Delta(.)$ verifies

$$
\|\Delta(\hat{\chi}, e)\| \leq \frac{k_1 k_2}{m} \|e\| \|\hat{\chi}\|, \quad \text{for } \|\hat{\chi}\| \geq r
\tag{28}
$$

Substituting (18) into (28) and recalling that $\hat{b}$ and (ξ_d, υ_d) are bounded, then one can write

$$\|\Delta(\hat{b}, \chi, e)\|$$

$$= \|\Delta(\hat{\chi}, e)\| \leq \frac{k_1 k_2}{m} \|e\| \left\| \frac{b}{\hat{b}} \chi + \frac{b - \hat{b}}{\hat{b}} (\xi_d, \upsilon_d)^T \right\|$$

$$\leq \frac{k_1 k_2}{m} \|e\| (k_3 \|\chi\| + k_4 \|\chi\|)$$

$$\leq \frac{(k_1 k_2)(k_3 + k_4)}{m} \|e\| \|\chi\| \qquad (29)$$

for $\|\chi\| \geq 1$, $k_3 = \frac{b}{b_{min}}$ and k_4 is the upper bound of $\|\frac{b-\hat{b}}{\hat{b}}(\xi_d, \upsilon_d)^T\|$. By defining $\gamma(\|e\|) = \frac{(k_1 k_2)(k_3+k_4)}{m} \|e\|$, which is a class-$\mathcal{K}$ function, the interconnection term $\Delta(\hat{b}, \chi, e)$ satisfies the growth restriction (22), that is, $\Delta(\hat{b}, \chi, e) \leq \gamma(\|e\|) \|\chi\|$ for $\|\chi\| \geq c_3 \triangleq \max(\frac{r - k_4}{k_3}, 1)$.

Finally, the stability of the closed-loop system (20) and the boundedness of trajectories $(\chi(t), e(t))$ are direct consequences of Theorem 2. Furthermore, if the *PE* property is satisfied, then the estimate $\hat{b}(t)$ converges exponentially to the true value b. Therefore, the χ-subsystem is GAS, thereby ensuring the GAS of the equilibrium point $(\chi, e) = (0, 0)$. $\qquad \square$

These theoretical results are very interesting because they state that the aircraft position and velocity can be accurately recovered and controlled using only visual cues and IMU data. Furthermore, in the presence of significant external disturbances that may probably induce bounded errors in the parameter estimate $\hat{b}$ (height since $\hat{b} = f/\hat{Z}$), the stability of the connected closed-loop system holds and the position tracking errors remain within a small bounded region (see (24)). The tracking errors can be significantly reduced by exciting the system (*PE* property) in order to improve the estimation of the unknown parameter b or height.

5 Aerial robotic platform and software implementation

The proposed adaptive vision-based autopilot was implemented on a quadrotor MAV platform that we have developed at *Chiba University*. This section provides information about the air vehicle, its avionics and real-time architecture, and the implementation of vision and control algorithms.

5.1 Description of the aerial robotic platform

Our testbed is based on a quadrotor vehicle which is manufactured by the *Ascending Technologies GmbH* company (Germany). The original kit includes the airframe, four brushless motors, four propellers and some electronics for manual flight. The vehicle is 53 cm rotor-tip to rotor-tip and weighs 400 grams including battery. It has a 300-g payload and an endurance of approximately 22 minutes without payload and 12 minutes with full payload. The original electronics of the AscTech quadrotor were kept but its attitude controller has been disabled except internal gyros feedback control which is running at 1 kHz.

In order to demonstrate vision-based autonomous flight, we have fitted the quadrotor helicopter with an embedded autopilot, which is designed and built by the authors. As shown in Fig. 4, the hardware components that make up the basic flight avionics of our platform include a small micro-controller from *Gumstix Inc.*, the MNAV100CA sensor from *Crossbow Inc.* and a vision system from *Range Video Inc.*

– *Gumstix micro-controller*: the Gumstix micro-controller is a light-weight and powerful computing unit that is very appropriate for our needs and micro UAVs applications in general. Indeed, it weighs only 8 grams and has a 400-MHz CPU with 16-MB flash memory and 64-MB SDRAM memory. Two expansion boards (*console-st* and *wifistix*) have been mounted on the Gumstix motherboard, thereby providing two RS-232 ports and WiFi communication with the GCS. The total weight of the obtained Linux-running computer is about 30 grams including interface card, WiFi module and antenna.

The Gumstix-based Flight Control Computer (FCC) performs sensor data acquisition and fusion, implements vision and control algorithms and generates the required control inputs (thrust, pitching torque, rolling torque and yawing torque).

– *AVR micro-controller*: this auxiliary micro-controller has two main tasks:

1. It reads the control inputs from the main FCC and encodes them into a PPM signal which is then used to drive the vehicle motors.
2. It implements the flight termination system which consists in achieving an emergency landing when some hardware or software problem occurs (see Kendoul et al. 2009b for more details).

– *MNAV sensor*: The MNAV100CA sensor is manufactured by *Crossbow* and designed for miniature ground and air vehicles. It includes a digital IMU, a ublox GPS receiver and a pressure sensor in one compact sensor. It is a low-cost (1500 USD) and light-weight (35 grams without GPS antenna) sensor, with a low power consumption, making it ideal for mini and micro UAVs applications. The IMU outputs raw data from 3 accelerometers, 3 gyrometers and 3 magnetometers at the rate of 50 Hz to the FCC. The GPS data are updated at 4 Hz and the static pressure sensor measurements are provided at a rate of 50 Hz. All these sensor data are sent to the FCC through an RS-232 serial link.
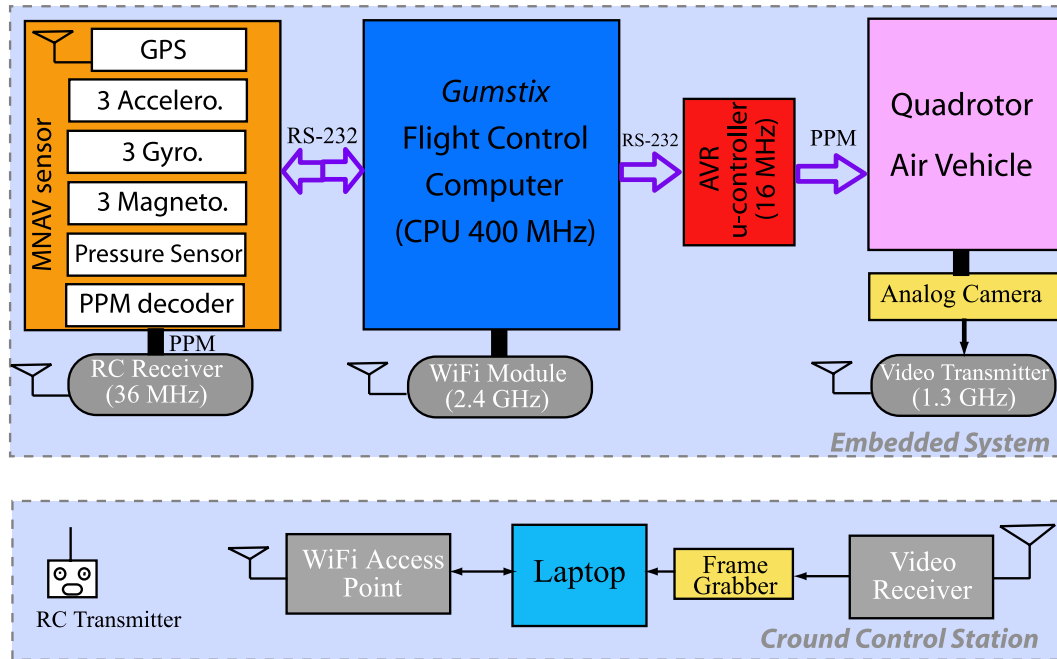
Fig. 4 Avionics architecture of our aerial robotic system (GPS is used here for comparison only)

– *Vision system*: for this project, we have used an imaging system from *RangeVideo* that includes an analog camera (KX-171), a 1.3 GHz wireless video transmitter, a hight gain antenna, a video receiver and a grabber card. The choice of this vision system was based primarily on range (about 1 km) and frequency (to avoid interferences with the 2.4 GHz of the WiFi module). The framegrabber is used to digitize individual frames from the analog video signal coming from the onboard imaging system. This enables frame acquisition at speeds up to 25 frames/s to obtain images of 320×240 pixels resolution.

5.2 Implementation of the real-time software

The developed real-time software can be divided into two main parts, the ground control station software and the embedded navigation and control algorithms. Figure 5 illustrates the interaction between the different algorithms and systems.

– *The Ground Control Station (GCS) software*: the GCS software is implemented on a standard laptop using C++, MFC, OpenGL and OpenCV libraries. It includes three different threads that are running at different frequencies. The first thread runs at 5 Hz and handles wireless communication with the onboard FCC for receiving and decoding flight data (telemetry), and sending visual estimates and other navigation commands to the onboard FCC. The communication program uses *socket* and *UDP* protocol. The second thread is also running at 5 Hz and implements

the GCS interface that allows to display flight data in real-time and to send flight commands by clicking on the appropriate buttons as shown in Fig. 6. The third and last thread implements the first part of the vision algorithm which is described in Sect. 2. This vision algorithm runs at 12 Hz and exploits some functions of the OpenCV library.

– *The embedded software*: the adaptive visual odometer described in Sect. 3 and the flight controller presented in Sect. 4 are implemented on the *Gumstix* FCC using multi-thread programming. Other navigation algorithms for sensor data acquisition and fusion are implemented in the onboard FCC. In total, the embedded software is composed of six different threads that are running at different frequencies: (1) thread 1 for communication with the GCS (10 Hz); (2) thread 2 for sensor data acquisition (50 Hz); (3) thread 3 for attitude estimation (50 Hz); (4) thread 4 for GPS-INS fusion and position estimation (10 Hz); (5) thread 5 that implements the adaptive visual odometer (10 Hz); (6) guidance and control algorithms (50 Hz). Figure 5 shows the different programs that are running on the GCS laptop and the onboard microcontroller.

The values of the nonlinear controller gains are shown in Table 1. The parameters of the RLS algorithm used for height estimation are chosen as follow: $\beta = 0.4$, $P(0) = 5$, $P_{max} = 10000$, $Z_{min} = 0.5$ m, $Z_{max} = 20$ m, *dead-zone* $= 0.1$, $\lambda_1 = \lambda_2 = 0.1$, $Z(0) = 0.5$ m.
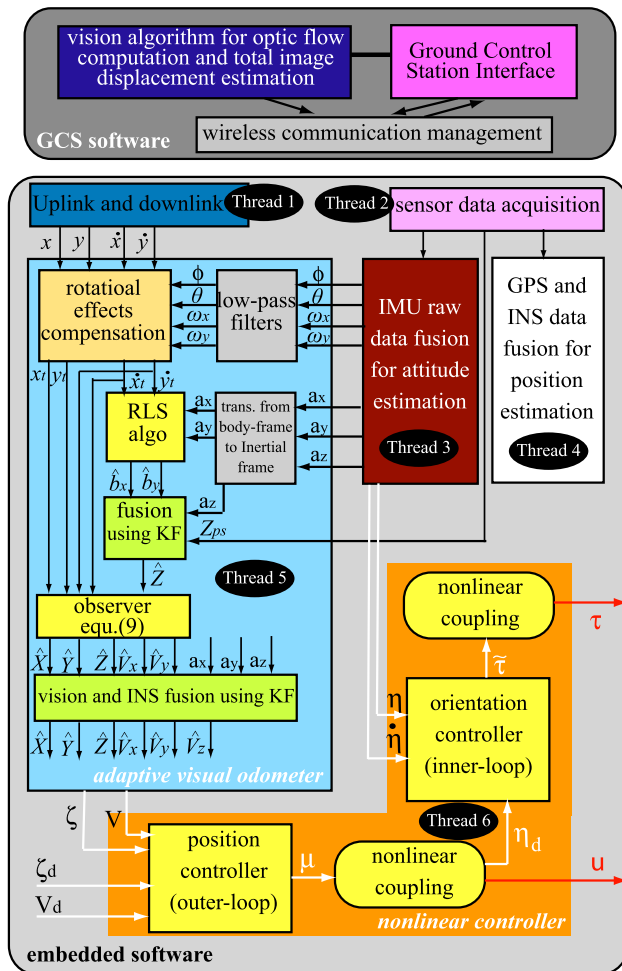
Fig. 5 Block diagram of the real-time software

Table 1 Control system gains

Parameter	Value	Parameter	Value
kp_x, kp_y	0.6	kp_ϕ, kp_θ	28
ki_x, ki_y	0.02	ki_ϕ, ki_θ	0.5
kd_x, kd_y	0.8	kd_ϕ, kd_θ	1
kp_z	0.8	kp_ψ	3
ki_z	0.03	ki_ψ	0.05
kd_z	1	kd_ψ	0.2

6 Experimental results of vision-based flights

The performance of the adaptive vision-based autopilot was demonstrated in real flights using the quadrotor MAV described in Sect. 5. We have performed various indoor and outdoor flight tests under autonomous control for take-off, landing, hovering, trajectory tracking, stationary and moving target tracking. All the outdoor flight tests, described in this Section, have been conducted at the play-ground of the Chiba University Campus which was not prepared in any

way for flight experiments. As it can be seen in Fig. 6 and flight videos, the field is almost homogenous without rich texture.

Remark 1 For the experimental flight tests, described in this Section, the flight controller used the adaptive visual odometer estimates for position, height and velocity feedback. GPS measurements are recorded during flights and plotted here for comparison only. It is important to note that GPS data can not be considered as the *ground-truth* since we are using a low-cost GPS with ± 2 m accuracy for horizontal position and ± 5 m error for height.

6.1 Static tests for rotation effects compensation and height estimation

These static tests aim at demonstrating the effectiveness of the approach (1)–(2) for compensating the rotational component of optic flow and image displacement using IMU data. We also analyze the performance of the adaptive visual odometer for height estimation using optic flow and accelerations.

We have conducted two static tests under different conditions. In *test A*, we have moved the MAV by hand in the following sequence: (1) vertical movement from 0 m to 1 m height; (2) horizontal movement along the X-axis at different heights (1 m, 0.5 m, 1 m); (3) pure rotational movement (pitch movement); and (4) vertical landing. In *test B*, we have repeated almost the same maneuvers with two differences: (1) the MAV is moved along the X-axis (zig zag) when performing vertical landing; (2) a small fan or ventilator is used to simulate wind.

For both tests, we have plotted in Fig. 7, the total image displacement and OF computed by the vision algorithm (blue line), the rotational image motion obtained from IMU data (red line), and the resulted translational image motion after compensation (black line). We can clearly see in Fig. 7 that rotation effects are effectively cancelled.

Concerning height estimation by the pressure sensor and the adaptive visual odometer, several remarks can be made.

– The pressure sensor works well in good weather conditions and it is sensitive to wind and temperature.
– The adaptive visual odometer is able to estimate the height when the translational OF is not very small.
– Compared to the pressure sensor, the visual odometer takes some time to converge to the true value when the MAV experiences vertical motion (height changes).

These tests prove the feasibility and possibility of estimating the height using optic flow and IMU data. For more robustness and accuracy, the height estimate used in flight tests is obtained by fusing the odometer measurement, the pressure sensor data and INS using a linear Kalman filter.

Fig. 6 The interactive interface of the ground control station during vision-based autonomous hovering
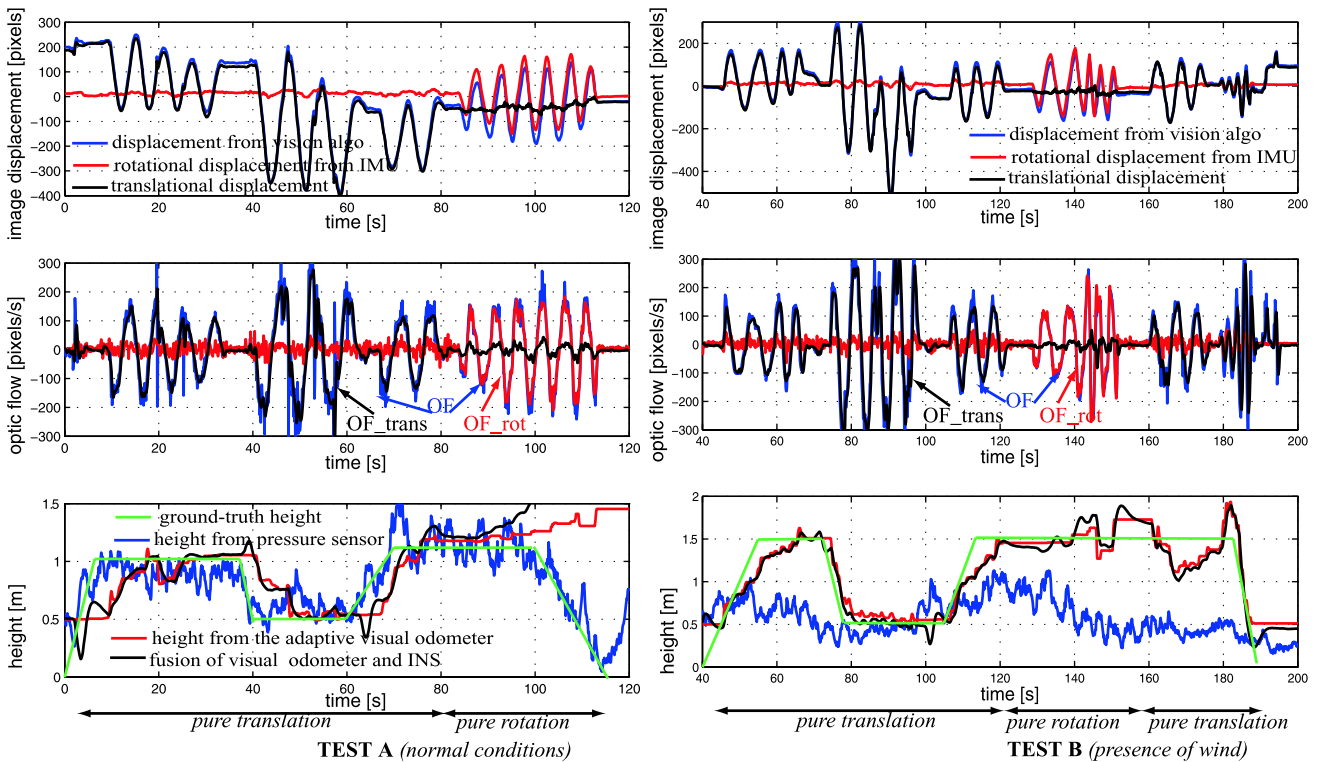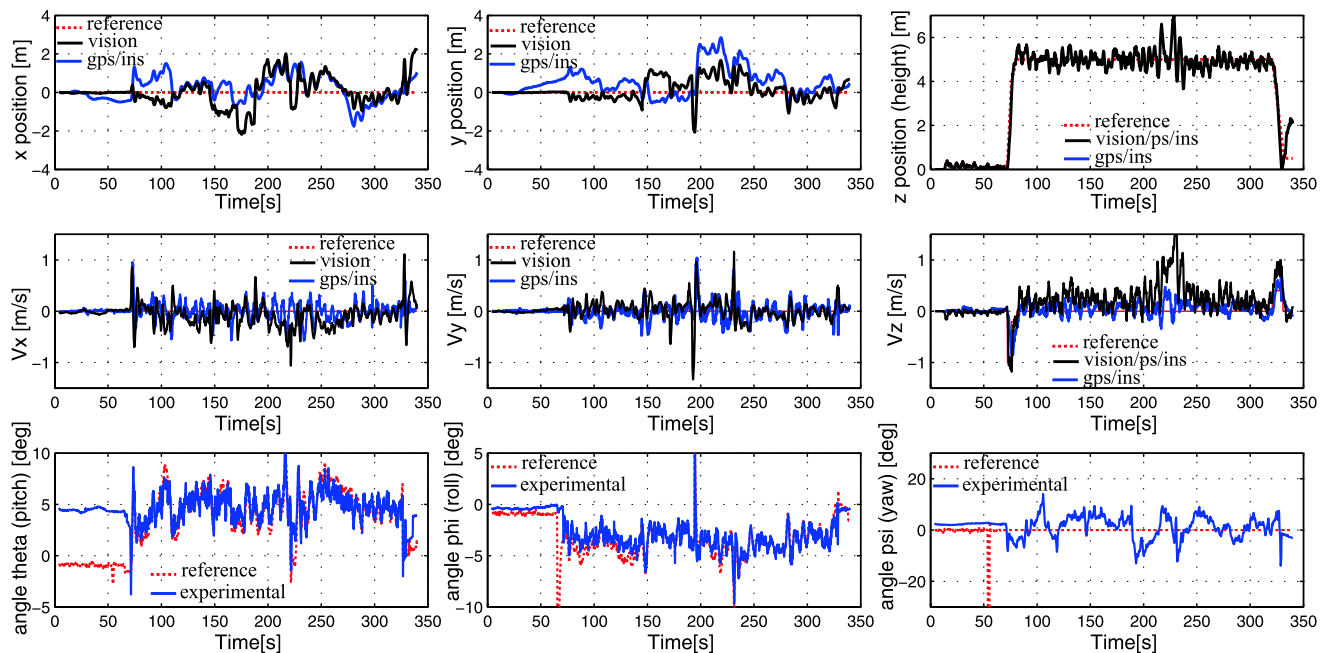

Fig. 7 Static tests showing rotation effects compensation and comparison of pressure sensor and visual odometer for height estimation

Fig. 8 Outdoor autonomous hovering with automatic takeoff and landing using the visual odometer

6.2 Outdoor autonomous hovering with automatic take-off and landing

The objective of this experiment is to check the robustness and accuracy of the developed vision-based autopilot for achieving stationary flights in natural environments with poor texture. The MAV is tasked to take-off autonomously, to hover at a desired altitude of 5 m and then to achieve a vertical auto-landing.

As shown in Fig. 8, this task was achieved successfully with good performance. Indeed, the MAV maintained its 3D position with good accuracy (±2 m maximum error) using the adaptive visual odometer estimates. The small errors in position control are mainly due to wind which was about 3.5 m/s during the flight test. We can also see on Fig. 8 that reference height trajectories are tracked accurately (±1 m maximum error) during take-off, hovering and landing phases. The inner-loop controller performs also well and tracks the reference angles.

We conclude from this test that all the components of the proposed vision-based autopilot (vision algorithm, adaptive visual odometer, nonlinear controller) perform stably and robustly despite the textureless environment.

Video clips of autonomous vision-based hovering can be found at: http://jp.youtube.com/watch?v=9I8BXtbrDQM&feature=channel_page.

6.3 Indoor autonomous flight

We have conducted several indoor flight tests that demonstrate the ability of our MAV, equipped with the developed
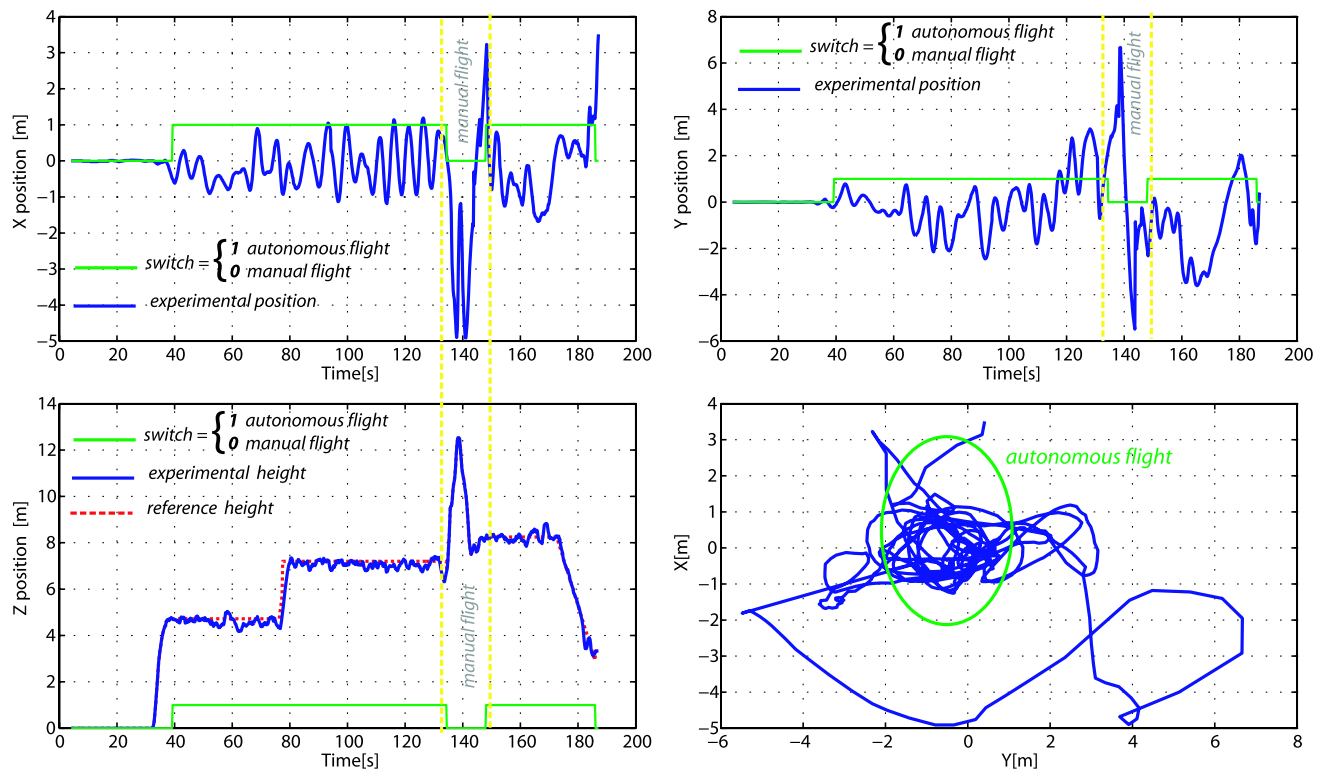

Fig. 9 Indoor fully autonomous flight using optic flow. Demonstration at *Tokyo Big Sight* for an international exhibition

vision-based autopilot, to achieve fully autonomous indoor flight using information extracted from image motion. Here, we present experimental results from a flight test that has been conducted at *Tokyo Big Sight* during an international exhibition.

Remark 2 In this test, optic flow and motion parameters were computed using the vision algorithm presented in our previous paper (Kendoul et al. 2009a). Figures 9 and 10 have already been published in Kendoul et al. (2009a).

Since the floor is homogeneous without any texture, we have put some objects on the ground to provide some texture for optic flow computation (see Fig. 9). The task was similar to the one described above, that is take-off, hover-

Fig. 10 MAV position and height trajectories of indoor autonomous flight using optic flow

ing and automatic landing. The exception or difference here is that during autonomous hovering, we have switched to manual flight for several seconds and then switched back to autonomous flight. The objective of this maneuver was to show to the audience (visitors) that the rotorcraft is naturally unstable and the vision-control system plays the main role in stabilizing the vehicle.

As it can be seen in Figs. 9 and 10, the rotorcraft achieved autonomously the required task with good performance.

A video clip of this flight test can be found at: http://jp.youtube.com/watch?v=Zt2WisDjUY0&feature=channel_page.

6.4 Automatic take-off, accurate hovering and precise auto-landing on some arbitrary target

As described in Sect. 2, the image area where optic flow and image displacement are computed is initially chosen at the image center. However, the developed GCS and embedded softwares allow to choose this target-template at any location of the image by just selecting the desired area/object on the image. This flight test consists in exploiting this useful characteristic to achieve an accurate hovering above some designated ground target and to perform a precise auto-landing on it.

The rotorcraft was put on a small box of about 50 cm ×70 cm which is used as a target. The take-off procedure

is launched from the GCS and the target is selected when it appeared in the camera FOV (about 1 m height during take-off). When the MAV reached the desired height of 10 m, it performed an accurate hovering by detecting the target and keeping it at the image center (see Fig. 11). Finally, the auto-landing procedure is activated and the MAV executed descent flight while controlling its horizontal position to keep the target at the image center. The MAV landed at 25 cm from the target, but it can be seen from the video that the MAV was exactly on the target at 30 cm height and then landed just near the target. This is due to very large image displacements when the MAV is at few centimeters from the target or ground. One approach to solve this problem could consist in deactivating the visual odometer and decrementing the thrust when the aircraft is under some height (50 cm for example).

Figure 12 shows the obtained MAV trajectories (position, height, velocity, orientation). The relative horizontal position between the MAV and the target was regulated to zero with about ±0.5 m maximum error. The height is also estimated and controlled accurately. The MAV was very stable even at 10 m height. Indeed, as shown in Fig. 12, the horizontal velocities, pitch and roll angles are stabilized and kept very small.

The good performance of this flight test can be checked by seeing the associated video clip at: http://jp.youtube.com/watch?v=rbmsivw5luk&feature=channel_page.

Fig. 11 Rotorcraft during vision-based autonomous hovering above a stationary target

6.5 Tracking a moving ground target with automatic take-off and auto-landing

Here, we explore the possibility of our vision-control system to track a ground moving target. For this experiment, we have used a small cart (see Fig. 14) as a target and placed it at about 20 m from the GCS.

First, the MAV performed automatic take-off from the target and hovered above the target (6 m height) for nearly 100 s. Then, the target is continuously moved towards the GCS by pulling some wire attached to the target. The control objective is thus, to keep the moving target at the image center by controlling the relative position between the MAV and the target to zero.

Figure 13 shows that the target is accurately tracked even when it is moving. The GPS ground-track on the first graph shows that the MAV flied about 20 m (which corresponds also to target movement) while controlling the relative position between the MAV and the target to zero with ± 1 m maximum error during tracking.

Figure 14(a) shows the tracked target on an image captured by the onboard camera and displayed at the GCS. We can also see in Fig. 14(b) the rotorcraft tracking the moving target.

Video clip of this flight test is available at: http://jp.youtube.com/watch?v=6obHavVvJyk&feature=channel_page.

6.6 Velocity-based control for trajectory tracking using vision

This flight test involves a velocity control scheme. It aims at evaluating and demonstrating the ability of the MAV to achieve hovering flight and velocity trajectory tracking by relying on velocities computed from optic flow without position feedback. After automatic take-off, the MAV is tasked to hover and then to achieve autonomous translational flight by tracking some reference trajectories, sent in real-time from the GCS. The commands for this test were: take-off, fly left, stop, fly forward, stop, fly right, stop, fly backward, stop, hover, land.

From Fig. 15, the rotorcraft can be seen to clearly respond to commands and to track reference velocity trajectories. Although the closed-loop control of horizontal position is not used in this test, the MAV seems to track also the position reference trajectories with small drifts. In fact, the position reference trajectories are obtained by integrating over time the velocity reference trajectories.

Despite the poor texture (see Fig. 16), the vision algorithm and adaptive visual odometer were able to track features, compute optic flow and recover the MAV motion parameters.

When doing these tests, the GPS signal was not available because of some technical problem of the GPS antenna connector.

This flight behavior is very useful and needed for many real-world applications where GPS signal is not available. Indeed, autonomous velocity control is sufficient to achieve many realistic tasks by just sending high-level commands.

A video clip of this flight test and other tests can be found at: http://jp.youtube.com/watch?v=Zp12GjZzjt4&feature=channel_page.

### 6.7 Position-based control for trajectory tracking using visual estimates

In this experiment, both position and velocity estimates are used by the controller to track arbitrary position and velocity trajectories, pre-programmed in the onboard FCC or sent by the operator from the GCS. The reference trajectories, shown in Fig. 17, consist in vertical climb (take-off) to an altitude of 6 m, 12 m-sideward flight and 16 m-forward flight, simultaneous backward and sideward flight to return to the starting point. The height is then, reduced to 3 m and small displacement commands are given.[5]

---

[5]In fact, height is reduced to 3 m to avoid damaging the platform in case where the MAV crashes because of empty battery (there was no charged battery for this test).
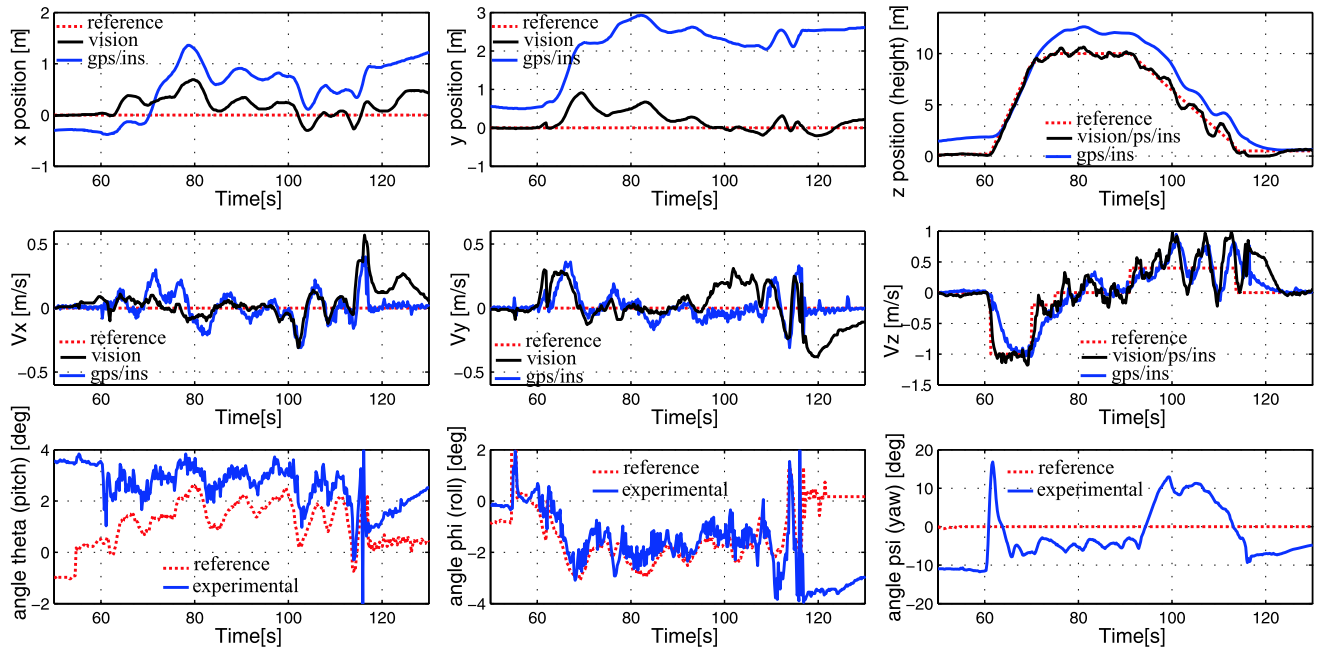
Fig. 12 (Color online) The rotorcraft was tasked to achieve the following maneuvers autonomously by relying on visual estimates: takeoff, hover above the target at a desired height of 10 m, and land on the target. The *black line* (visual estimates used for control) shows that the relative horizontal position between the MAV and the target was regulated to zero with about ±0.5 m maximum error. The commanded height is accurately tracked, resulting in an automatic take-off and landing
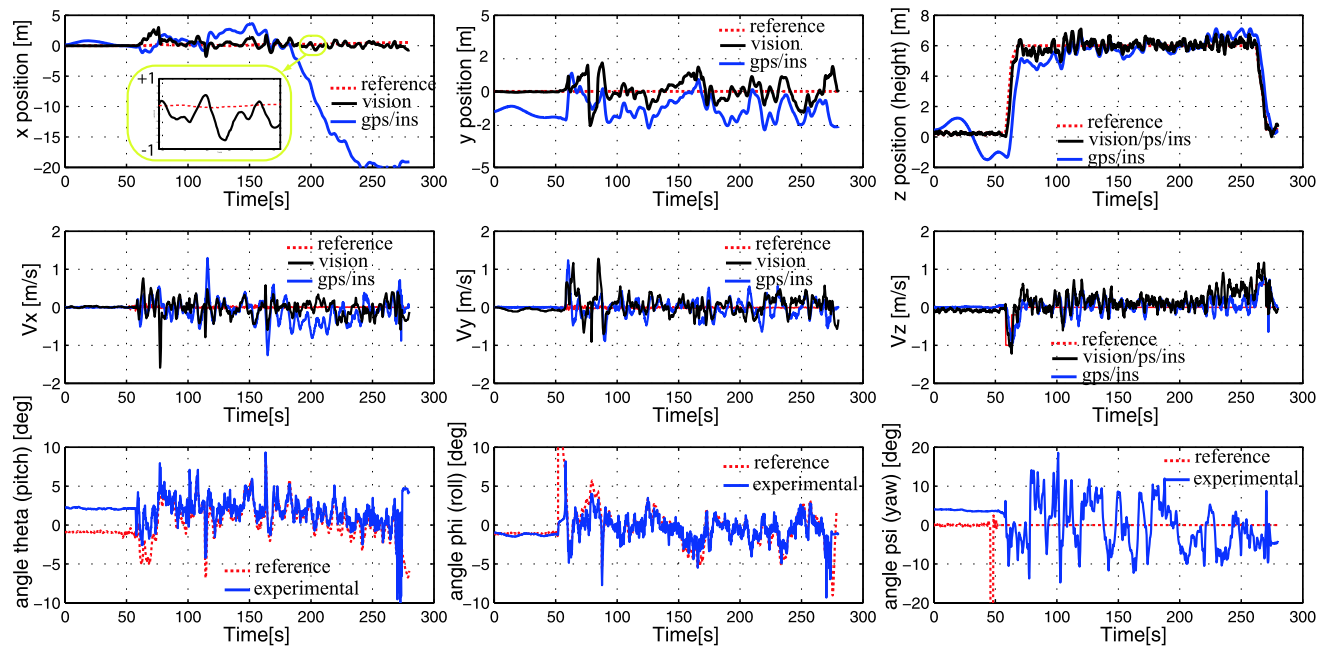

Fig. 13 (Color online) Application of the vision-based autopilot for tracking a moving ground target. The GPS ground-track (*blue line*) shows that the MAV flied approximately 20 m at an average speed of about −0.5 m/s which also corresponds to the target movement. The relative position between the MAV and the target (*black line*) is regulated to zero with ±1 m maximum error which means that the rotorcraft tracked successfully the target and kept it in its field of view

The obtained results, shown in Fig. 17 are very satisfactory. Indeed, the reference position, height, velocity and attitude trajectories are well tracked. The MAV was stable along the flight course.

6.8 GPS-based waypoint navigation and comparison with the visual odometer estimates

In this last test, we have performed waypoint navigation using GPS data for horizontal movement control and adaptive visual odometer estimates for height control. A set of four waypoints were chosen by just clicking the desired locations on the 2D map of the GCS interface (see Fig. 6). The MAV should then, pass the assigned waypoints in a given sequence. The objective of this test is to compare the GPS and the visual odometer for estimating the horizontal po-
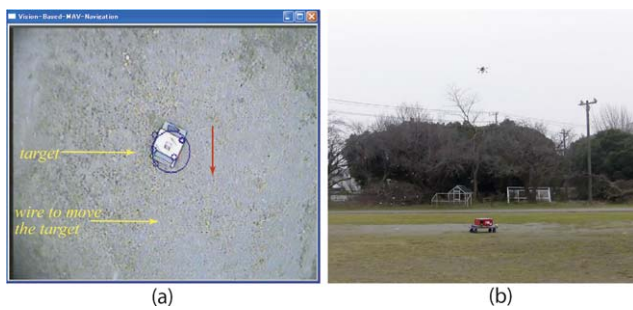
Fig. 14 (**a**) onboard image showing the detection and tracking of a moving target; (**b**) MAV tracking the moving target

sition during waypoint navigation. The obtained results are shown in Fig. 18.

One can see that the reference trajectories are tracked and the mission is accomplished. It is also important to note that during this translational flight, the adaptive visual odometer was able to estimate the MAV position or the travelled flight distance despite the textureless environment. Therefore, vision-based waypoint navigation is possible by the current system when combined with some landmarks recognition algorithm.

6.9 Discussion

The reported results over different flight scenarios testify on the effectiveness and robustness of the developed vision-based autopilot. However, the proposed system can be further improved especially the adaptive visual odometer. Here, we discuss briefly some weaknesses and limitations of the developed system, which are mainly related to position estimation process and off-board image processing.

Inaccuracies in position estimation are mainly due to two factors, estimation errors growing in long range navigation and identification errors in the adaptive process.

- *Growing estimation errors in Visual Odometry (VO)*: VO pose estimates are produced by tracking visual features in the environment and integrating or accumulating their image displacement. Although VO is an incremental process, our results have shown that positioning accuracy
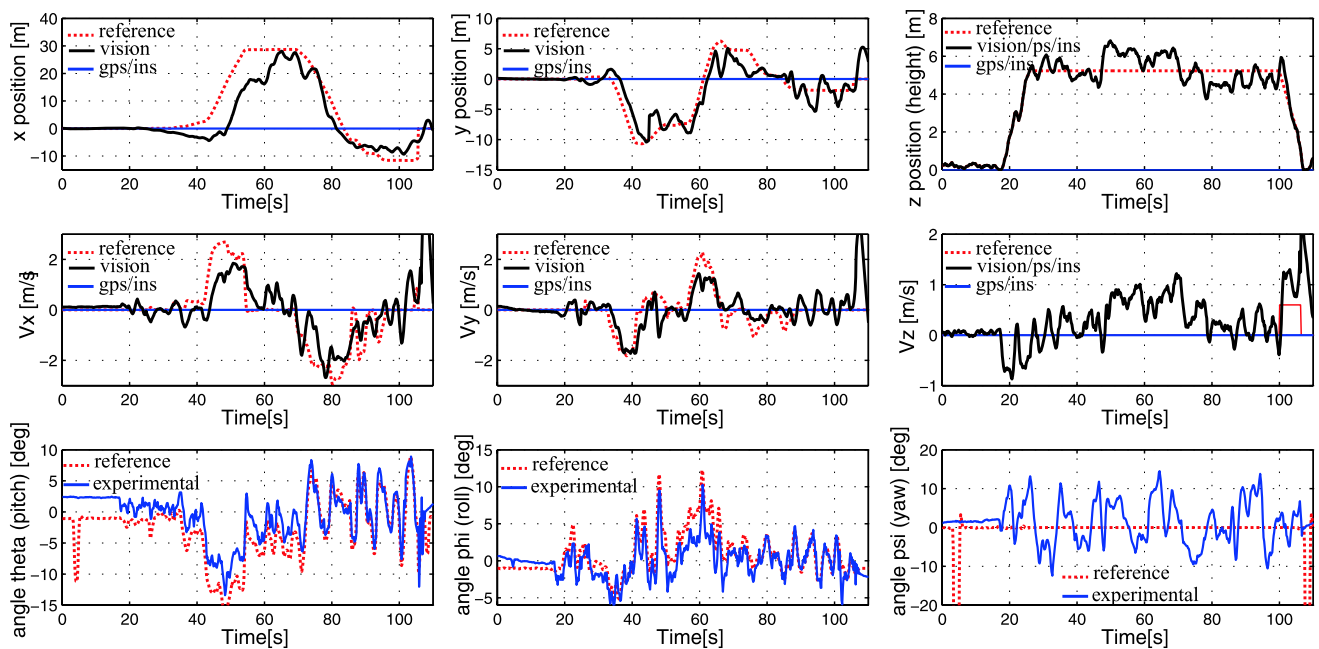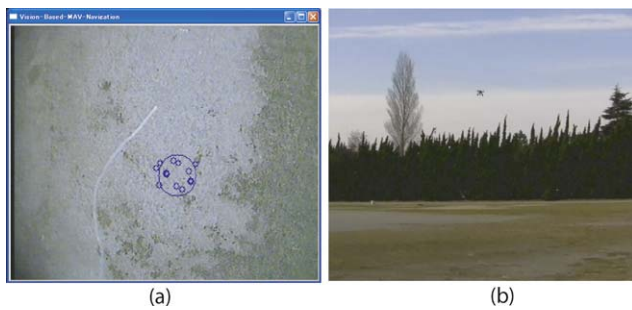

Fig. 15 (Color online) Experimental results from a flight test where the quadrotor tracks autonomously velocity reference trajectories without position feedback. The velocity measurements used for control are computed by the visual odometer using optic flow. GPS signal (*blue line*) was not available during this flight test

to within a few percent of measured ground truth is possible, over distances of about hundred meters. However, the performance of the VO is expected to degrade for longer flight distances due to the accumulation of unbounded errors. Furthermore, the poor texture of some terrain surfaces leads to poorer performance since VO is an exteroceptive sensing method that relies on features tracking.

To enable accurate and robust long range navigation with VO, this latter can be combined with other terrain relative navigation techniques like landmarks recognition. An interesting approach could consist in performing landmarks matching and recognition in a moment-to-moment fashion (or periodically) to estimate the absolute position, and integrating the flight path between landmarks to estimate the relative position. Therefore, computation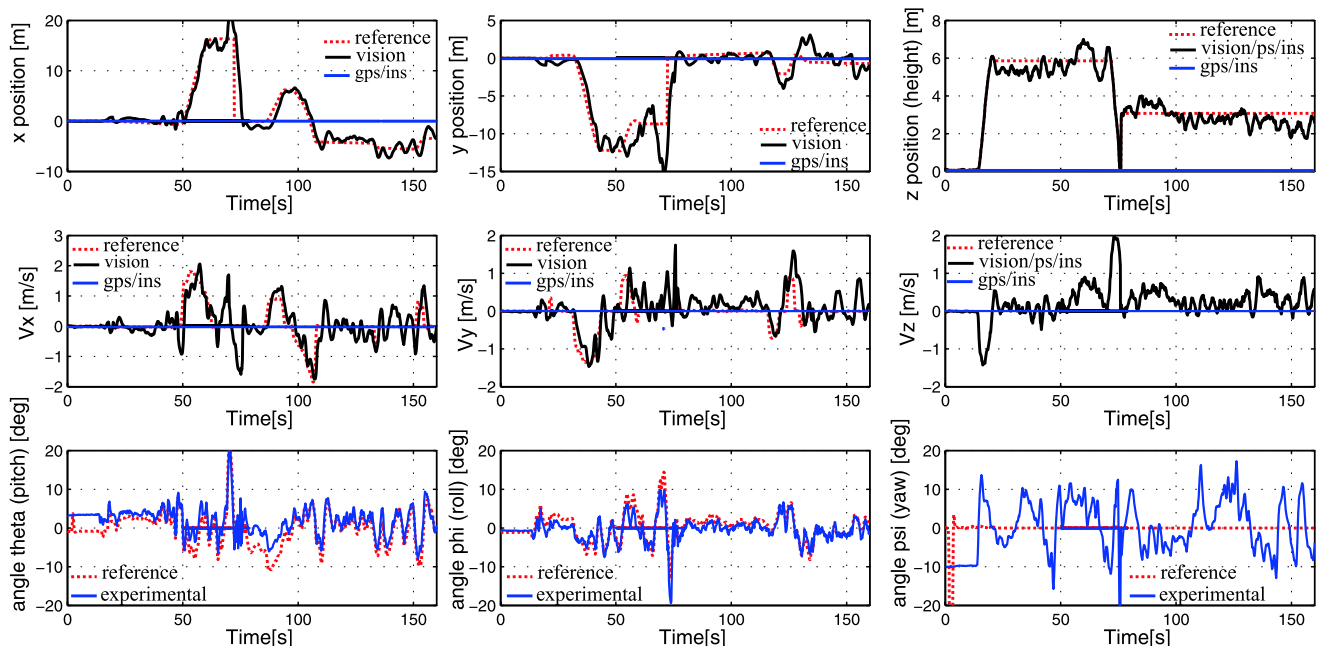 of distance flown is re-commenced whenever a prominent landmark is encountered. Re-setting the VO at each landmark facilitates accurate long-range navigation by preventing excessive accumulation of odometric errors. Indeed, some animals like bees seem to use landmark-based cues as well as visual path integration to navigate to a goal (Srinivasan et al. 1997).

In this paper, we have concentrated on the VO for estimating the vehicle's position relative to some known location like the initial location or recognized landmark.

- *Range estimation errors in the adaptive process*: The visual odometer presented here requires an approximate measure of height above the ground. A RLS algorithm has been used to estimate the height by fusing the derivatives of optic flow and IMU accelerations data. This process suffers from the general well-known drawbacks of identification algorithms like noise and poor input-output signals (non-excited system). During real-time experiments, we have noticed performance degradation in height estimation by the RLS algorithm when the rotorcraft is hovering or flying at very low speeds. In this case, input and output signals are small compared to the noise level and do not contain sufficient information to estimate the height. As stated in Sect. 3, this is related to the persistent excitation (PE) property which is not satisfied in this case.

This issue has been handled by switching off the adaptation process when the input and output signals are small (stationary or slow flights). In this case, height estimate is primary provided by the pressure sensor as a propagation

Fig. 16 Autonomous vision-based translational flight. (**a**) onboard image showing features tracking in a textureless environment; (**b**) rotorcraft during vision-based trajectory tracking

Fig. 17 (Color online) Position trajectory tracking is demonstrated using the visual odometer estimates. The position commands are sent in real-time by the operator from the GCS. GPS signal (*blue line*) was not available during this flight test
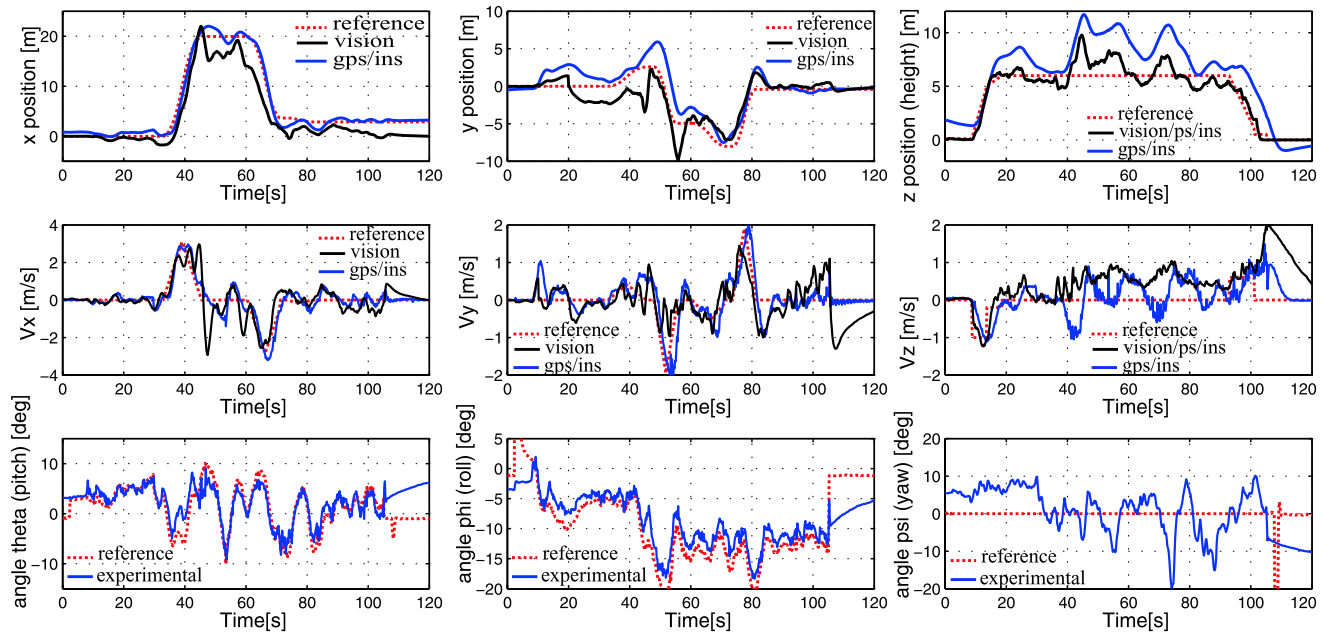
Fig. 18 (Color online) Comparison of GPS measurements with the visual odometer estimates during GPS-based waypoint navigation. The obtained preliminary results demonstrate the capability of the adaptive visual odometer to gauge the travelled flight distance and to estimate the vehicle's position relative to the staring point. Inaccuracies in position estimation may be attributed to range identification and image displacement accumulation errors due to the poor texture of the terrain

of previous estimates using a kinematic Kalman filter. Although the obtained results are satisfactory, range estimation process can be further improved by exciting the system (small lateral oscillations or zig-zag flight) from time to time and using more robust identification algorithms. In fact, some insects like desert locust sways its head laterally (peering behavior) in order to measure target range using image motion (Wallace 1959). In (1993), Srinivasan et al. suggested also that bees deliberately impart a lateral sinusoidal displacement in order to be able to deduce range from the lateral optic flow signal. In Sect. 3, we have showed mathematically that it possible to use optic flow and acceleration measurements to determine height above terrain, provided that the PE property is satisfied (rich signals; see Ioannou et al. 1996 for more details on the PE property).

A such behavior can be implemented on a rotorcraft by applying an open-loop oscillation in attitude which will induce small lateral velocities upon the flight path. The generated optic flow and lateral accelerations can be then used to estimate the height by applying the technique presented in Sect. 3. However, a detailed analysis is needed to determine the best oscillation frequency and amplitude that will result in robust range estimation without spending a lot of energy and deteriorating the flight performance.

- *Off-board image processing*: In this work, the vision algorithm described in Sect. 2 is implemented off-board on

the GCS. This affects the robustness of the vision algorithm since the image quality depends on transmission noise, distance, obstacles, etc. Off-board image processing induces a time delay in visual measurements which limits the system bandwidth. However, this has not significant effect on the overall system performance since all guidance, navigation and control algorithms run on-board the FCC. The main limitation of off-board image processing is that the system requires valid video and WiFi communication links between the aerial vehicle and the GCS in order to transmit images and to send back the visual measurements to the onboard FCC. Experimental tests showed also that there are interferences between the video transmitter and other communication channels.

In order to overcome this problem and to extend the capabilities of our rotorcraft MAV, we are developing a completely embedded system that is based on the Surveyor SRV-1 Blackfin Camera which consists of a digital camera and an integrated 500 MHz processor for image processing.

7 Conclusion

In this paper, we have presented a vision-based autopilot which is designed for miniature UAVs and MAVs flying at low altitudes. The developed system is based on a downward-looking camera and relies heavily on visual cues

to achieve various navigation tasks in unknown environments. Unlike some existing bio-inspired flight controllers which are usually used for reactive navigation, the proposed vision-based autopilot extends optic flow-based control capabilities to complex navigational tasks such as accurate hovering, arbitrary trajectory tracking, stationary and moving target tracking, etc. These capabilities are mainly due to the incorporated adaptive visual odometer that allows the estimation of the height and the recovery of the rotorcraft position and velocity. A practical nonlinear control system is also designed and used for flight control. The combined visual odometer and flight controller result in an effective autopilot that is easy to implement while guaranteeing the asymptotic stability of the entire closed-loop system.

The experimental flight tests, performed indoors and outdoors under realistic conditions, have shown the good performance of the proposed adaptive vision-control system. Indeed, experimental results over various ranges of the flight envelope illustrate that the proposed vision-based autopilot allowed a small quadrotor MAV to achieve automatic take-off, accurate hovering, precise auto-landing, trajectory tracking and moving target detection and tracking. These results demonstrate that visual odometry can enable UAVs and MAVs to navigate in unknown environments where global localization is impossible.

As future work, we plan to combine the visual odometer with other navigation algorithms, including landmarks recognition and vision-based obstacle avoidance. This should enable visual odometry to operate in long range and goal-oriented navigation. We are also implementing a modified version of the system which operates in real-time onboard the MAV.

References

Ahrens, S., Levine, D., Andrews, G., & How, J. P. (2009). Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 2643–2648), Kobe, Japan, May 2009.

Altug, E., Ostrowski, J. P., & Taylor, C. J. (2005). Control of a quadrotor helicopter using dual camera visual feedback. *International Journal of Robotics Research*, *24*(5), 329–341.

Amidi, O., Kanade, T., & Fujita, K. (1999). A visual odometer for autonomous helicopter flight. *Robotics and Autonomous Systems*, *28*(2–3), 185–193.

Astrom, K. J., & Wittenmark, B. (1989). *Adaptive control*. Reading: Addison-Wesley.

Barrows, G. L. (1999). *Mixed-mode VLSI optic flow sensors for micro air vehicles*. Ph.D. Dissertation, Department of Electrical Engineering, University of Maryland.

Caballero, F., Merino, L., Ferruz, J., & Ollero, A. (2009). Vision-based odometry and slam for medium and high altitude flying UAVs. *Journal of Intelligent Robotic Systems*, *54*, 137–161.

Chahl, J. S., Srinivasan, M. V., & Zhang, S. W. (2004). Landing strategies in honeybees and applications to uninhabited airborne vehicles. *International Journal of Robotics Research*, *23*(2), 101–110.

Frew, E., McGee, T., ZuWhan, K., Xiao, X., Jackson, S., Morimoto, M., Rathinam, S., Padial, J., & Sengupta, R. (2004). Vision-based road-following using a small autonomous aircraft. In *Proceedings of the IEEE aerospace conference* (Vol. 5, pp. 3006–3015), March 2004.

Ettinger, S. M., Nechyba, M. C., Ifju, P. G., & Waszak, M. (2002). Towards flight autonomy: Vision-based horizon detection for micro air vehicles. In *Proceedings of the Florida conference on recent advances in robotics*, Miami, May 2002.

Fowers, S. G., Lee, D.-J., Tippetts, B. J., Lillywhite, K. D., Dennis, A. W., & Archibald, J. K. (2007). Vision aided stabilization and the development of a quad-rotor micro uav. In *Proceedings of the IEEE international symposium on computational intelligence in robotics and automation* (pp. 143–148), Florida, USA, June 2007.

Garratt, M. A., & Chahl, J. S. (2008). Vision-based terrain following for an unmanned rotorcraft. *Journal of Field Robotics*, *25*(4–5), 284–301.

Goodwin, G. C., & Sin, K. C. (1984). *Adaptive filtering prediction and control*. Englewood Cliffs: Prentice Hall.

Green, W. E., Oh, P. Y., Sevcik, K., & Barrows, G. (2003). Autonomous landing for indoor flying robots using optic flow. In *Proceedings of the 2003 ASME international mechanical engineering congress*, Washington, 15–21 November 2003.

Guenard, N., Hamel, T., & Mahony, R. (2008). A practical visual servo control for a unmanned aerial vehicle. *IEEE Transactions on Robotics*, *24*(2), 331–341.

He, R., Prentice, S., & Roy, N. (2008). Planning in information space for a quadrotor helicopter in a GPS-denied environment. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 1814–1820), California, USA, May 2008.

Herisse, B., Russotto, F.-X., Hamel, T., & Mahony, R. (2008). Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 801–806), Nice, France, September 2008.

Hrabar, S., Sukhatme, G. S., Corke, P., Usher, K., & Roberts, J. (2005). Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In *Proceedings of the IEEE international conference on intelligent robots and systems* (pp. 302–309), Canada, 2005.

Ioannou, P., & Sun, J., (1996). *Robust adaptive control*. Englewood Cliffs: Prentice Hall Inc.

Johnson, A., Montgomery, J., & Matthies, L. (2005). Vision guided landing of an autonomous helicopter in hazardous terrain. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 4470–4475), Barcelona, Spain, April 2005.

Calise, A. J., Watanabe, Y., Ha, J., Neidhoefer, J. C., & Johnson, E. N. (2007). Real-time vision-based relative aircraft navigation. *AIAA Journal of Aerospace Computing, Information, and Communication*, *4*(4), 707–738.

Kanade, T., Amidi, O., & Ke, Q. (2004). Real-time and 3d vision for autonomous small and micro air vehicles. In *Proceedings of the 43rd IEEE conference on decision and control* (pp. 1655–1662), Atlantis, Bahamas, December 2004.

Kendoul, F. (2007). *Modelling and control of unmanned aerial vehicles, and development of a vision-based autopilot for small rotorcraft navigation*. Ph.D. Thesis Report, CNRS Heudiasyc Laboratory, University of Technology of Compiegne, France.

Kendoul, F., Fantoni, I., & Lozano, R. (2008). Adaptive vision-based controller for small rotorcraft uavs control and guidance. In *Proceedings of the 17th IFAC world congress* (pp. 797–802), Seoul, Korea, 6–11 July 2008.

Kendoul, F., Fantoni, I., & Nonami, K. (2009a). Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robotics and Autonomous Systems (Elsevier)*, *57*, 591–602.

Kendoul, F., Zhenyu, Y., & Nonami, K. (2009b). Embedded autopilot for accurate waypoint navigation and trajectory tracking: Application to miniature rotorcraft uavs. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 2884–2890), Kobe, Japan, May 2009.

Kima, J., & Sukkarieh, S. (2007). Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems*, *55*, 62–71.

Lacroix, S., Jung, I. K., Soueres, P., Hygounenc, E., & Berry, J. P. (2002). The autonomous blimp project of LAAS/CNRS: Current status and research challenges. In *Proceedings of the workshop WS6 aerial robotics, IEEE/RSJ international conference on intelligent robots and systems* (pp. 35–42), Lausanne, Switzerland, 2002.

Landau, I. D., Lozano, R., & M'Saad, M. (1998). *Adaptive control. Communications and control engineering*. Berlin: Springer.

Lucas, B., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the DARPA IU workshop* (pp. 121–130).

Mejias, L., Saripalli, S., Campoy, P., & S Sukhatme, G. (2006). Visual servoing of an autonomous helicopter in urban areas using feature tracking. *Journal of Field Robotics*, *23*(3/4), 185–199.

Mori, R., Hirata, K., & Kinoshita, T. (2007). Vision-based guidance control of a small-scale unmanned helicopter. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 2648–2653), San Diego, CA, USA, Oct. 29–Nov. 2007.

Olfati-Saber, R. (2001). *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*. Ph.D. Thesis Report, Department of Electrical Engineering and Computer Science, MIT, USA, February 2001.

Proctor, A. A., Johnson, E. N., & Apker, T. B. (2006). Vision-only control and guidance for aircraft. *Journal of Field Robotics*, *23*(10), 863–890.

Roberts, J. F., Stirling, T., Zufferey, J. C., & Floreano, D. (2007). Quadrotor using minimal sensing for autonomous indoor flight. In *European micro air vehicle conference and flight competition (EMAV)*, Toulouse, France, September 2007.

Ruffier, F., & Franceschini, N. (2005). Optic flow regulation: the key to aircraft automatic guidance. *Robotics and Autonomous Systems*, *50*(4), 177–194.

Saripalli, S., Montgomery, J. F., & Sukhatme, G. S. (2003). Visually-guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, *19*(3), 371–381.

Scherer, S., Singh, S., Chamberlain, L., & Elgersma, M. (2008). Flying fast and low among obstacles: Methodology and experiments. *International Journal of Robotics Research*, *27*(5), 549–574.

Sepulcre, R., Jankovic, M., & Kokotovic, P. (1997). *Constructive nonlinear control. Communications and control engineering series*. Berlin: Springer.

Serres, J., Dray, D., Ruffier, F., & Franceschini, N. (2008). A vision-based autopilot for a miniature air vehicle: joint speed control and lateral obstacle avoidance. *Autonomous Robots*, *25*, 103–122.

Shakernia, O., Sharp, C. S., Vidal, R., Shim, D. H., Ma, Y., & Sastry, S. (2002). Multiple view motion estimation and control for landing an unmanned aerial vehicle. In *Proceedings of the IEEE conference on robotics and automation* (Vol. 3, pp. 2793–2798).

Shi, J., & Tomasi, C. (1994). Good features to track. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 593–600), Seattle, WA, USA.

Sontag, E. D. (1988). Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, *34*, 435–443.

Srinivasan, M. V., Miles, F. A., & Wallman, J. (1993). *Visual motion and its role in the stabilisation of gaze*. Amsterdam: Elsevier.

Srinivasan, M. V., Zhang, S. W., Lehrer, M., & Collett, T. S. (1996). Honeybee navigation en route to the gaaol: Visual flight control

and odometry. *The Journal of Experimental Biology*, *199*(1), 237–244.

Srinivasan, M. V., Zhang, S. W., & Bidwell, N. J. (1997). Visually mediated odometry in honeybees. *The Journal of Experimental Biology*, *200*, 2513–2522.

Tammero, L. F., & Dickinson, M. H. (2002). The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster. *Journal of Experimental Biology*, *205*, 327–343.

Wallace, G. K. (1959). Visual scanning in the desert locust schistocerca gregaria. *The Journal of Experimental Biology*, *36*, 512–525.

Yamada, H., Ichikawa, M., & Takeuchi, J. (2001). Flying robot with biologically inspired vision. *Journal of Robotics and Mechatronics*, *13*(6), 621–624.

Yu, Z., Nonami, K., Shin, J., & Celestino, D. (2007). 3D vision based landing control of a small scale autonomous helicopter. *International Journal of Advanced Robotic Systems*, *4*(1), 51–56.

Zufferey, J. C., & Floreano, D. (2006). Fly-inspired visual steering of an ultralight indoor aircraft. *IEEE Transactions On Robotics*, *22*(1), 137–146.

Farid Kendoul was born in Béjaia, Algeria in 1980. He received the Engineer Diploma in Robotics and Mechatronic, and the Master's Degree in Automatic Control from the University of Technology of Compiègne (UTC), France in 2004. In 2004, he joined the C.N.R.S. Heudiasyc laboratory and obtained his Ph.D. in 2007 at the University of Technology of Compiègne researching in the area of Unmanned Aerial Vehicles (UAVs). He is currently a JSPS postdoctoral research fellow at Chiba University in Japan. His research interests are modelling and control of UAVs, flight control and guidance, adaptive and nonlinear control, computer vision and insect-inspired navigation. His current research focuses on the development of a cooperative vision-based system for formation flight of several rotorcraft MAVs.

Kenzo Nonami received M.S. degree and Ph.D. degree in Mechanical Engineering in 1976 and 1979 respectively, from Tokyo Metropolitan University. He joined Chiba University in 1979 as a Research Associate, Associate Professor from 1988 to 1994. Since 1994, he has been a full professor in Department of Mechanical Engineering and Department of Electronics and Mechanical Engineering at Chiba University. In 2004, Dr. Kenzo Nonami was a vice dean of faculty of engineering. Also, he has carried out a research in NASA in USA during two years from 1985 to 1988. Now, Dr. Nonami is a vice president from April, 2008. His recent research interests are Fully autonomous unmanned small-scale helicopter, Micro air vehicle, Quad tilt wing unmanned aerial vehicle, Land mine detection robots with multi-functional arm, Walking machines, Master slave manipulator and dual manipulator hand system, Unmanned autonomous boats, Flywheel energy storage system with active magnetic bearing powered electric vehicle, Robust and nonlinear control, Control applications.

Isabelle Fantoni was born in Le Mans, France, on January 31, 1973. She graduated from the University of Technology of Compiègne (UTC), in France and received the Master of Sciences in Electronic System Design at the University of Cranfield in England (double degree), both in 1996. She received the Master in Control Systems, in 1997 and the Ph.D. degree, in 2000 from the University of Technology of Compiègne, in France. Since October 2001, she has been a permanent Researcher at Heudiasyc Laboratory, UTC, in Compiègne, France, employed by the French National Foundation for Scientific Research (CNRS).

Her research interests include non-linear control, underactuated mechanical systems, modelling and control for Unmanned Aerial Vehicles (UAVs), computer vision for navigation of aerial vehicles.

Rogelio Lozano was born in Monterrey Mexico, on July 12, 1954. He received the B.S. degree in electronic engineering from the National Polytechnic Institute of Mexico in 1975, the M.S. degree in electrical engineering from Centro de Investigación y de Estudios Avanzados (CINVESTAV), Mexico in 1977, and the Ph.D. degree in automatic control from Laboratoire d'Automatique de Grenoble, France, in 1981. He joined the Department of Electrical Engineering at CINVESTAV, Mexico, in 1981 where he worked until 1989. He was Head of the Section of Automatic Control from June 1985 to August 1987. He has held visiting positions at the University of Newcastle, Australia, from November 1983 to November 1984, NASA Langley Research Center VA, from August 1987 to August 1988, and Laboratoire d'Automatique de Grenoble, France, from February 1989 to July 1990. Since 1990 he is a CNRS Research Director at University of Technology of Compiègne, France. He was promoted to 1st class CNRS Research Director in 1997. He is head of the Laboratory Heudiasyc, UMR 6599 CNRS-UTC since January 1995. He was Associate Editor of Automatica in the period 1987–2000 and of Int. J. of Adaptive Control and Signal Processing since 1993. His research interests are in adaptive control of linear, non-linear systems, robot manipulators, passive systems, teleoperation and unmanned aerial vehicles (UAV).