# The Navigation and Control technology inside the AR.Drone micro UAV

**Pierre-Jean Bristeau** * **François Callou** ** **David Vissière** ***
**Nicolas Petit** *

\* *Centre Automatique et Systèmes, Unité Mathématiques et Systèmes,*
*MINES ParisTech, 60 bd Saint-Michel, 75006 Paris, FRANCE,*
*pierre-jean.bristeau@mines-paristech.fr*
*\*\* Parrot, 174 quai de Jemmapes, 75010 Paris, FRANCE*
*\*\*\* SYSNAV, 1 rue Jean de Becker Rémy, 27940 Aubevoye, FRANCE*

**Abstract:** This paper exposes the Navigation and Control technology embedded in a recently commercialized micro Unmanned Aerial Vehicle (UAV), the AR.Drone, which cost and performance are unprecedented among any commercial product for mass markets. The system relies on state-of-the-art indoor navigation systems combining low-cost inertial sensors, computer vision techniques, sonar, and accounting for aerodynamics models.

Keywords: Flying robots, data fusion, low cost MEMS

## 1. INTRODUCTION

In 2004, the Parrot company started a project named AR.Drone aiming at producing a micro Unmanned Aerial Vehicle (UAV) for the mass market of videos games and home entertainment. The project was publicly presented at the 2010 Consumer Electronics Show, and, starting on August, 18th, 2010, the AR.Drone has been released on the market. This project has involved from 5 to 12 engineers from Parrot with the technical support of SYSNAV and his academic partner MINES Paris-Tech for navigation and control design. One of its unique features is that it is a stabilized aerial platform, remotely controlled through a user-friendly graphical interface running on an Apple iPhone, iPad or iTouch. It is available from numerous retail stores in various countries and the on-line Apple store at a price below 300 euros. The aim of this paper is to present the control technology embedded inside this product.

The targeted goals of the AR.Drone project go way beyond conventional usages commonly considered in both civilian and military applications. They encompass augmented reality (AR), video games, and interactivity. As is discussed in this article, this landmarking project is a prime example of sophisticated use of low-cost sensors (MEMS and cameras) for mass markets where retail price is of major importance. This project and the embedded algorithms have the particularity of being highly stable, robust and very user-friendly. In other words, the technology yields the way to the enjoyment of playing. The underlying complexity can be completely forgotten.

The UAV platform is a quadrotor which is a very popular and has attracted much attention from academia (Hamel et al. [2002], Castillo et al. [2004], Pounds et al. [2006], Tayebi and McGilvray [2006], Romero et al. [2007], Guénard [2007], Hoffmann et al. [2007], Efe [2007], Cunha et al. [2009], Martin and Salaün [2010], Lupashin et al. [2010]). It is light and affordable, and is capable of hover and fast forward flight in narrow spaces. Yet, in open-loop, the quadrotor is an unstable vehicle. To provide the customer with an easy to pilot platform, the embedded control systems have to be very effective and plays the role of an enabling technology for the whole project.

What are the main problems to be solved in the domain of automatic control here ? The vehicle must be easy to fly, fun and safe. Ease of flying means that the end-user shall only provide high level orders which must be handled by an automatic controller dealing with the complexity of low-level sub-systems. Because the system is unstable, feedback is needed. In turn, this raises the issue of state estimation. Enjoyment is guaranteed by the capability of the control system to handle relatively aggressive maneuvers. Safety means that the vehicle control system must be robust to the numerous disturbances that can be met in practice as the UAV is used in various and unknown environments. Redundancy in the state estimation is the solution in this case.

For these reasons, that will be developed further in this article from the automatic control theorist point of view, the critical points are the accuracy and the robustness of the vehicle state estimation. These points have received much consideration in the academic world over the recent years. While absolute position estimation is not a strict requirement (at the exception of the altitude for safety reasons), it is of paramount importance to know the translational velocity during all the flight phases, so that it is possible to stop the vehicle and to prevent it from drifting. The capability of stopping the vehicle is a security requirement, while cancellation of the drift of the vehicle –which is particularly annoying— has a large added value in terms of end-user experience. Other key questions are stabilization and robustness. Finally, one shall realize that the UAV under consideration must be plug-and-play, in the sense, that it is not a laboratory experiment, and must fly autonomously once it is handed out of its package by the end-user and its the battery is loaded. No sophisticated calibration or tuning procedure can be performed by the end-user who is usually totally unfamiliar with control technology.

To address the problem of state estimation, UAV are usually equipped with embedded inertial sensors (gyrometers and accelerometers, see Titterton and Weston [2004]), a sonar altitude sensor (or a barometer), and, often, an absolute position or velocity sensor such as a GPS or a camera feeding vision algorithms (see e.g. Hamel and Mahony [2007], Rondon et al. [2009]). Combined into data fusion algorithms, these devices have allowed to obtain relatively good results of state estimation and stabilization on rotary wing (ducted fan, see Naldi et al. [2008], Hua et al. [2008], Naldi et al. [2009], small-scale helicopters, see Hua [2009], Bristeau et al. [2010]) and fixed wing UAVs, thus guaranteeing stabilized flights with residuals errors of the order of magnitude of the sensors.

The constraints under consideration for the applications envisioned by Parrot in the AR.Drone project are much more restrictive that the ones usually bearing on micro UAVs applications. Here, it is required to handle both indoor and outdoor autonomous flights, irrespective of the potential unavailability of the GPS or the camera. This point is of importance since these sensors can fail in numerous cases such as GPS signal unavailability, low-light flight, or weakly textured visual environments.

In summary, the AR.Drone project has consisted in designing a micro rotary wing UAV which cost is lower by several orders of magnitude than its competitors, while providing the end-user with unprecedented motion control accuracy, robustness and ease of handling. Numerous videos are available on the Parrot website, and interested readers can simply judge by themselves by trying one of these UAV at a nearby retail store, or contact the authors. In this article, we present the technology included in this system. To obtain the presented results, the algorithms embedded in the AR.Drone contains state-of-the art low-cost navigation methods, video processing algorithms, combined with an innovative usage of an aerodynamics model which previous fine theoretical studies (see Bristeau et al. [2009]) permit to stress crucial information serving as a robust estimate of the air speed of the vehicle.

The paper is organized as follows. In Section 2, the AR.Drone experimental platform is presented. The system includes a MEMS inertial measurement unit (IMU), two cameras, a sonar, and a two processors board. One of the processors (PIC) is used to gather the I/Os, the second one (Parrot P6 ARM based) is in charge of running the algorithms. In Section 3, the main lines of the navigation algorithms are explained. An estimate of the instantaneous rotation is used. The IMU needs to be calibrated. This is done in an automatic way, in two steps. A first calibration is performed, at the factory where the UAVs are constructed, to compensate for sensors misalignment, scale factors, and bias. The second one is performed onboard and used during flights to determine drift and align the sensor board and the aerial platform. Timestamping and synchronization techniques of the inertial measurements and the cameras are discussed. In Section 4, we discuss some key aspects of a dynamics model for the AR.Drone. The velocity estimate obtained through the vision system is unbiased, but can be faulty or unavailable due to algorithms convergence issues. It relates the translational and rotational velocities and the accelerations. The effect of the flexibility of the propellers plays a key role here. By exploiting this effect, through the presented model, it is possible to estimate *at all times* the velocity of the vehicle with respect to the air. The quality of this estimate solely depends on the accuracy of the inertial sensors biases estimates. Therefore, it can slightly drift



Fig. 1. Parrot AR.Drone, with its outdoor hull.

over time. In Section 5, the global data reconciliation algorithm is presented. It takes advantage of the complementarities of the various sources of information discussed earlier. Experimental results stress the obtained accuracy in state reconstruction. Importantly, the presented technique are totally consistent with the desirable plug-and-play nature of the AR.Drone. We present the principles of the control architecture which combines low-level controllers and high-level system management through a state machine. Finally, we give conclusions and sketch future directions in Section 6.

## 2. THE PARROT AR.DRONE PLATFORM

### 2.1 Aerial vehicle

The Parrot AR.Drone is based on a classic quadrotor design. Four brushless motors are used to power the four (fixed) propellers. Together, they constitute variable thrust generators. Each motor is mounted on a highly resistant PA66 plastic feet carrying the brushless control board (BLCB). Each BLCB uses its own ATMEGA8L 8bit microcontroller and cutout system to turn-off the engine in the event an obstacle gets in the propeller path. Extensive testing have been performed to assess the reliability of this safety system. The four thrust generators are attached together using four carbon fiber tubes and a plastic fiber reinforced central cross. The carbon structure supports an Expanded Poly Propylene (EPP) structure which is carrying the Lithium-Polymer (LiPo) battery in a plastic basket. The 3 cells battery supplies 11.1 V and 1000 mAh with a discharge capacity of 15 C, and weights 80 g. The battery capacity corresponds to a 10 to 15 minutes flight. The basket lies on a foam on the EPP structure which filters the motors vibrations. Two hulls can be plugged on the system. The first hull is a removable cover body can be used to prevent propellers from scratching the walls for indoor applications. The second hull is a shield-case for the battery (see Figure 1).

### 2.2 On-board electronics

The on-board electronics consists of two boards screwed to a main plastic basket. The first board is the mother-board which is connected to the navigation board. We now detail these elements.

*Mother-board* The mother board embeds a Parrot P6 processor (32bits ARM9-core, running at 468 MHz), a Wi-Fi chip, a vertically oriented camera and a connector to the front camera.

A Linux based real-time operating system and all the calculations software are run on the P6 processor. The P6 processor is also in charge of acquiring data flows from two video cameras and integrates its own hardware DCT (discrete cosines transform). To maximize the quality of the immersive view, the front camera uses a 93 degrees wide-angle diagonal lens camera. Its output is a VGA resolution (640x480) signal at a framerate of 15 frames per second. The vertical camera is used, in the navigation algorithms, to measure the vehicle speed. It is a 64 degrees diagonal lens camera producing data at a framerate of 60 frames per second. Both camera can be used to perform object recognition for gaming purposes. A miniUSB connector is included for software flashing purposes and to allow add-ons to be plugged (such as GPS, and laser beamer, among others).

*Navigation board*     The navigation board uses a 16bits PIC micro-controller running at 40 MHz, and serves as an interface with the sensors. These sensors are a 3-axis accelerometers, a 2-axis gyroscope, a 1-axis vertical gyroscope, and 2 ultrasonic sensors.

The ultrasonic sensors are 2 Prowave ultrasonics sensors which are used for altitude estimation. The PIC micro-controller handles the ultrasonic transmitter, and digitalizes the signal from the ultrasonic receiver. Ultrasonic sensors are used to estimate the altitude and the vertical displacements of the UAV. They can also be used to determine the depth of the scene observed by the vertical camera. The ultrasonic sensors have a 40 kHz resonance frequency and can measure distances as large as 6 m at a 25 Hz rate.

The accelerometers and gyroscopes constitute a low-cost inertial measurement unit (IMU). The cost of this IMU is less than 10 USD. A Bosch BMA150 3-axis accelerometer using a 10 bits A/D converter is used. It has a +/- 2g range. The two axis gyro is an Invensense IDG500. It is an analog sensor. It is digitalized by the PIC 12 bits A/D converter, and can measure rotation rates up to 500 degrees/s. On the vertical axis, a more accurate gyroscope is considered. It is an Epson XV3700. It has an auto-zero function to minimize heading drift. The IMU is running at a 200Hz rate.

*Embedded software*     The real time operating system is a custom embedded Linux. Simultaneously, it manages multiple threads: Wi-Fi communications, video data sampling, video compression (for wireless transmission), image processing, sensors acquisition, state estimation and closed-loop control. The data acquisition and control thread is running at a 200 Hz rate and is sequenced by the A/D converter. The image processing is sequenced by the vertical camera frames.

## 3. VISION AND INERTIAL NAVIGATION ALGORITHMS

### 3.1 Vision algorithms

In view of the video gaming applications, the vehicle is equipped with two video cameras. They are also used onboard to infer some information on the vehicle velocity. The speed estimation from the images provided by the vertical camera (pointing to the ground in hovering flight) is achieved using two complementary algorithms. Depending on the scene content or the expected quality of their results, one is preferred to the other. The first algorithm is a multi-resolution scheme and computes the optical flow over the whole picture range. The algorithms implements the methodology described in Lucas and

Kanade [1981], and uses a kernel to smooth spatial and temporal derivatives, following Horn and Schunck [1993]. In details, attitude change between two images is ignored during the first resolution refinement steps. The induced error is finally efficiently canceled in most cases by subtracting the displacement of the optical center induced by the attitude change alone. The second algorithm, usually named "corner tracking", estimates the displacement of several points of interest (trackers). The motion of the camera is deduced from these computed displacements through an iteratively weighted least-squares minimization procedure. The algorithm uses a first processing of data with a FAST type corner detector, as detailed in Trajkovic and Hedley [1998]. A fixed number of trackers are placed over the corner positions after a second analysis of the underlying image content. Some erroneous tracker candidates positions are eliminated. Then, the trackers updated positions are searched in the image, within a frame whose radius is adapted to the expected trackers screen speed. The IRLS estimation (see Michaelsen and Stilla [2004]) is carried out by assuming the depth of the scene is uniform. The first method reveals usually slightly less robust in common experimentations, but it can handle scenes that have a very low contrast [1]. This multi-resolution (coarse-to-fine) scheme allows the estimation of small and large screen speeds with a constant computational cost. This algorithm is used by default. One switches to the second algorithm for sake of increased accuracy, when the scene is deemed suitable for the corner detector and the speed is low. One can switch back to the first algorithm when the estimated speed reaches a certain threshold value, or when the number of detected trackers becomes too low.

### 3.2 Inertial sensors calibration

*Factory calibration*     Using low-cost inertial sensors implies dealing with bias, misalignment angles, and scale factors are not negligible and differ from one AR.Drone sensor board to another. Despite construction mechanical tolerances are indeed small, a manual board calibration has to be performed at the factory as follows. Consider a sensor (e.g. a triaxial accelerometer) and note $Y_m$ its measurement and $Y_v$ the true value of the sensed variable, the following model is considered

$$Y_m = \alpha R Y_v + \beta \text{ with } \beta = [\beta_1, \beta_2, \beta_3]^T,$$

$$R = \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix}, \quad \alpha = \text{diag}(\alpha_1, \alpha_2, \alpha_3)$$

where $R$ represents a first order misalignment matrix (micro-rotator) between the frame of the camera to the frame of the sensor board, $\alpha$ stands for scale factors and $\beta$ for accelerometer bias. Additionally to this model, a non-orthogonality error is considered so that one can model the whole calibration process by computing the unknown parameters $A$, $B$ as the minimizers of a least-square cost where $A$ takes into account scale factors, misalignment, and non-orthogonality, and $B$ is the bias.

$$Y_m = A Y_v + B$$

Two practical methods have been developed to solve this problem. The simplest uses a two-axis calibration table that provides a reference for $Y_v$ such that the problem reduces to a simple least-square minimization. A second method, used during the research and development process, is presented in Dorveaux

---

[1] The Lucas-Kanade equation coefficients provide a cheap but useful estimation of the overall picture contrast, the trace of the picture matrix being the sum of the squared norm of the gradients.

et al. [2009]. Interestingly, it does not need any calibration hardware.

Gyroscope triedra calibration is done using various angular position and integrating the angular rate between those two position according to the following model, similar to accelerometer one, $\Omega_m = A\Omega_v + B$. To determine the angular position, the calibration table may be used.

*Onboard calibration*    Misalignment between the AR.Drone frame and the sensor-board is not completely compensated at the factory stage. A simple way to resolve this issue is to consider that the mean acceleration of the UAV must be equal to zero on a stationary flight (e.g. one minute long) and to invoke another least-square problem to determine the micro-rotation that keeps the camera direction constant in the horizontal plane and that rotates the vertical reference (using only small roll and pitch variations). The micro-rotator is then transformed to a rotation matrix by reintroducing sine and cosine functions with the small calculated angles to keep the calibration valid. After each landing, this onboard calibration is automatically performed again to compensate for the misalignment due to the displacement of the damping foam during the stopping shock.

### 3.3 Attitude estimation

The main role of the inertial sensors is to provide attitude and velocity estimates to be used in the closed-loop stabilizing control algorithm. Inertial navigation is performed using the following facts and principles

*Principles*

- the accelerometers and gyroscopes can be used together as inputs in the motion dynamics, which, once integrated, give the estimates of attitude angles and velocities.
- Noting $V = \begin{bmatrix} u & v & w \end{bmatrix}^T$ the vector velocity of the center of gravity of the IMU in the body frame, $Q = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ the Euler angles (roll-pitch-yaw), i.e. the angles between the inertial frame and the body, and $\Omega = \begin{bmatrix} p & q & r \end{bmatrix}^T$ the angular rate of turn in the body frame, and $F$ the external forces, the governing equation is

$$\dot{V} = -\Omega \times V + F$$
$$\dot{Q} = G(\Omega, Q)$$

with

$$G(\Omega, Q) = \begin{bmatrix} p + (q\sin\phi + r\cos\phi)\tan\theta \\ q\cos\phi - r\sin\phi \\ (q\sin\phi + r\cos\phi)\sec\theta \end{bmatrix}$$

*Facts*    The inertial sensors suffer from the following flaws.

- the accelerometers, which do not measure the body accelerations but its acceleration minus the gravity, expressed in the body frame, are biased and misaligned. Classically, we consider that the accelerometer signal $Y_V$ has a bias $B_V$ (independently on each axis) and suffers from additive white noise $\mu_v$. The measurement equations are

$$Y_V = F - Rg + B_V + \mu_V$$

where $g$ stands for the gravity acceleration and $R$ is the rotation matrix from the inertial frame to the body frame

$$R = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \begin{array}{c} \sin\phi\sin\theta\cos\psi \\ -\cos\phi\sin\psi \end{array} & \begin{array}{c} \sin\phi\sin\psi\sin\theta \\ +\cos\phi\cos\psi \end{array} & \cos\theta\sin\phi \\ \begin{array}{c} \cos\phi\sin\theta\cos\psi \\ +\sin\phi\sin\psi \end{array} & \begin{array}{c} \cos\phi\sin\psi\sin\theta \\ -\sin\phi\cos\psi \end{array} & \cos\theta\cos\phi \end{bmatrix}$$

- the gyroscopes are biased and noisy. The measurement equation is

$$Y_\Omega = \Omega + B_\Omega + \mu_\Omega$$

where $\mu_\Omega$ is a noise vector, and $B_\Omega$ is a bias vector.

To overcome the preceding fundamental issues, the following procedure is used. The accelerometer bias are not dealt with by the attitude estimation algorithm. They are in fact estimated and compensated thanks to the vision system as presented in Section 4. Under this assumption, various algorithms for attitude estimation can been considered: Kalman filter, extended Kalman filter (see Vissière [2008]) or complementary filter in both linear and nonlinear implementations (see Mahony et al. [2005], Metni et al. [2006], Jung and Tsiotras [2007], Mahony et al. [2008], Martin and Salaün [2008]).

### 3.4 Inertial sensors usage for video processing

As briefly mentioned earlier, the inertial sensors are used to compensate for the so-called *micro-rotations* in the images of the camera. This is a significant help for the determination of the optical flow in the vision algorithm. In details, the gyroscopes and the accelerometers are used as follows. Consider two successive frames at 60Hz, from which it is desired to deduce the 3D linear velocity of the UAV from the pixels displacement. The displacement of tracked objects is related to the linear velocity on the horizontal plane once the vertical linear velocity and angular velocity are compensated for, according to the estimate of the altitude. Interestingly, a specific linear data fusion algorithm combining sonar and accelerometer information is implemented to give accurate vertical velocity and position estimates above the obstacle. Thanks to the attitude estimation algorithm, one obtains estimates of orientation at time $t$, $R_t$ and of orientation at time $t + \delta t$, $R_{t+\delta t}$. The micro-rotation between the two successive frames is equal to $R_{t+\delta t}R_t^T$. Alternatively, due to the low-pass nature of the attitude estimation algorithm, good results can be obtained using a direct integration gyrometer measurement (de-biased by the attitude estimation algorithm). Then, the micro-rotation matrix is obtained as follows

$$\delta R = \begin{bmatrix} 1 & r\delta t & -q\delta t \\ -r\delta t & 1 & p\delta t \\ q\delta t & -p\delta t & 1 \end{bmatrix}$$

## 4. AERODYNAMICS MODEL FOR AN INNOVATIVE VELOCITY ESTIMATION

### 4.1 Inertial navigation strategy

A key feature of the AR.Drone is its accurate velocity estimation technique. For safety (i.e. its ability to go in hovering mode when the user does not prescribe any order) as for the normal flying mode (i.e. when the velocity setpoint is proportional to the handheld device orientation), it is of prime importance to have an accurate and robust velocity estimate in all conditions.
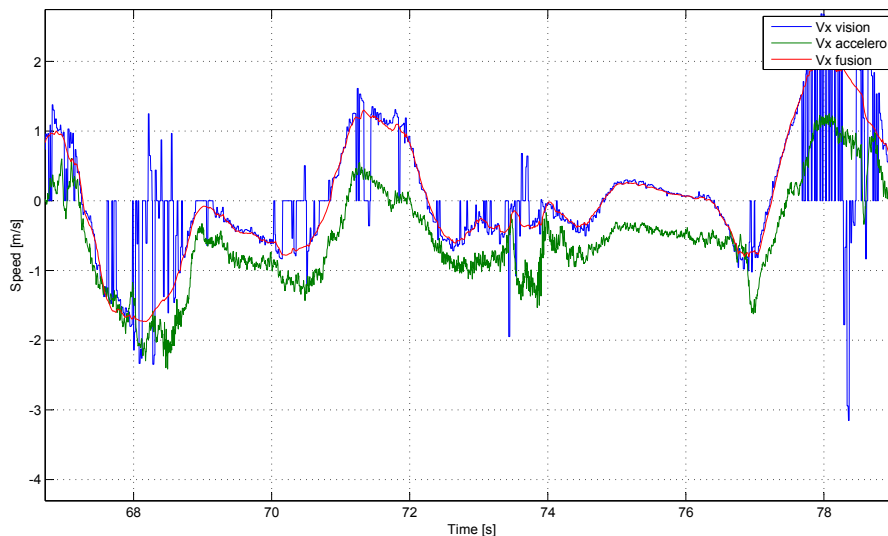
Fig. 2. Velocity estimates: computer vision velocity estimate (blue), aerodynamics model velocity estimate from direct accelerometer reading (green), fused velocity estimate (red).

It implies redundancy and mutual drift-compensation. Vision-based velocity estimation algorithms described above are particularly efficient when the ground is sufficiently textured. Yet, even in that case, the estimate is noisy and relatively slowly updated compared to the UAV dynamics. A main innovation is the construction of a reliable velocity estimate thanks to an aerodynamics model (presented below). Typically obtained results are presented in Figure 2. The two sources of information are fused as follows. When the aerodynamics velocity and the vision velocity are simultaneously available, accelerometer bias are estimated and vision velocity is filtered. Alternatively, when vision velocity is unavailable, only the aerodynamics estimate is used with the last updated value of the bias.

All these techniques form a tightly integrated vision and inertial navigation filter that can be described the following way: after inertial sensors calibration, sensors are used inside a complementary filter to estimate the attitude and de-bias the gyros; the de-biased gyros are used for vision velocity information combined with the velocity and altitude estimates from a vertical dynamics observer; the velocity from the computer vision algorithm is used to de-bias the accelerometers, the estimated bias is used to increase the accuracy of the attitude estimation algorithm; eventually the de-biased accelerometer gives a precise body velocity from an aerodynamics model.

We now expose the aerodynamics model under consideration.

### 4.2 Velocity estimation thanks to aerodynamics model

While the mechanical structure of a quadrotor is simple (four rotors with simple propellers and rigid frame), the dynamic behavior is surprisingly involved. In particular, the aerodynamics of the propellers and the motion of the rigid frame can interfere to produce a coupled dynamical model. This fact has been discussed earlier in the literature, e.g. Pounds et al. [2006], Hoffmann et al. [2007] and studied analytically in particular in Bristeau et al. [2009]. As has been shown in Bristeau et al. [2009], the observed aerodynamics drag induced by the rotors during a translational motion are non negligible.

The analytical computations can be performed following the works on large-size helicopter rotors (Johnson [1980], Prouty [1990], Leishman [2002]), and models proposed specifically for quadrotors (Metni et al. [2006], Pounds et al. [2006], Romero et al. [2007], Guénard [2007]), one can transpose the large-size rotors modeling techniques to small-size rotors, taking into account angular rates, which are negligible at larger scales. By contrast with Mettler [2003], Vissière et al. [2008], one should not neglect the forward flight speed $(u, v)$. The aerodynamic effects applied to the rotor are evaluated by integrating, along each rotor blade, the aerodynamic resultant force per surface increment. The obtained expressions can be rendered more precise by adding a degree of freedom to the rotor dynamics. It is relevant to take into account the flexibility of the rotor blade, and, implicitly, allow it to rotate out of the plane normal to the rotation axis. The angle between the blade and this plane is called flapping angle. The flapping dynamics of the rotor blade can be approximately determined using Fourier expansions, and the conservation of angular momentum around the flapping axis. The flapping angle is determined by the equilibrium between aerodynamic moment, centrifugal moment and stiffness moment. Additionally, the flapping dynamics can be considered, but in this paper, we simply use the expressions of stabilized flapping angles. Then, one shall consider the whole vehicle with its four contrarotative rotors (out of ground effects). Applying Newton's second law, the efforts created by the four rotors are incorporated into the rigid body dynamics. Accounting for the coupling between the rigid body motion and the aerodynamic effects, relatively unexpected terms appears in the dynamics, which are visible in the matrix of linearized model about steady flight $\dot{X} = AX + BU$. A diagonal stabilizing term in the matrix $A$ appears which is due to the drag forces induced by the rotors.

Further, by incorporating the flexibility of the rotor blades about stationary flight, the matrix $A$ is changed. The stabilizing term is reinforced by the tilt phenomenon of the rotor discs. This phenomenon takes place as follows. During a forward flight, the advancing blade experiences stronger lift than the retreating one. Under the rigid modeling assumption, one would obtain
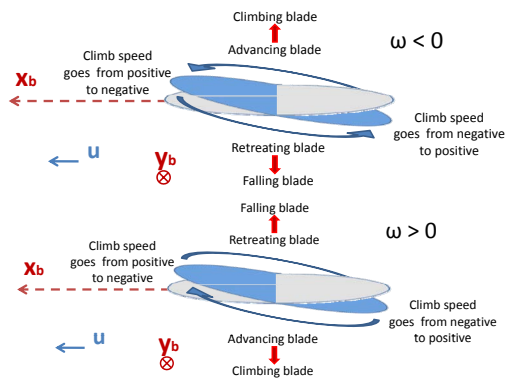
Fig. 3. Tilt phenomenon in case of forward flight: counter-clockwise rotation (top) and clockwise rotation (bottom).

roll moments on each rotor which would cancel on the whole vehicle. By contrast, by considering the flexibility, stronger lifts do not cause any moment but induces flapping speeds. The flapping angle has the same extrema for all the rotation direction (Figure 3). In fact, one has to notice that, as well as in the rigid model, on a quadrotor (four contrarotative rotors), axial effects are added while side effects cancel.

In summary, linear drag term exists from the interaction between the rigid body and the rotors and this term is reinforced by tilt phenomenon which changes a lift force component in drag. These induced-drag effects are non negligible and they yield interesting information on the velocity of the system. The induced forces are directly measured by the accelerometers, and through the model, can be used to reliably estimate the velocities of the UAV. The match between the measured forces and the model is illustrated in Figure 2 where the linear relationship between the accelerometer measurement and the velocity is visible, up to time-varying bias which is accounted for as previously discussed in Section 4.1.

## 5. CONTROL ARCHITECTURE

The data fusion and control architecture consists of several loops in which the human pilot is embedded. The pilot uses his handheld device, through a Wi-Fi connection, to remotely send his high-level orders, and, simultaneously, to visualize the video stream from the front camera or the AR.Drone. A typical view from the graphical user interface displayed on the screen of the iPhone is reported in Figure 4.



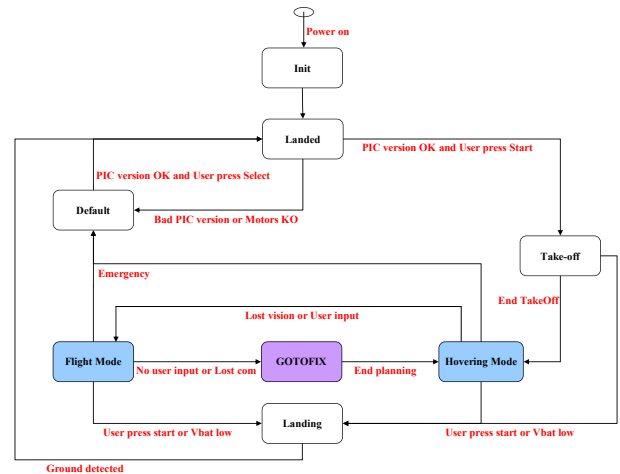Fig. 4. Screen-shot of the Graphical User Interface.



Fig. 6. State Machine to govern the various flying and control modes.

The architecture is illustrated in Figure 5. The switches between the various modes (hovering flight, take off, landing, forward flight) are handled by a finite state machine which accounts for the orders of the pilot. This point is illustrated in Figure 6. By default, i.e. when the user does not touch the screen of his hand-held device, the AR.Drone goes in hovering mode, where the altitude is kept constant and attitude and velocity are stabilized to zero. A double-click on the screen makes it go in landing mode, while the touch-screen is usually meant to specify velocity setpoints (in two directions) and yaw rate.

The control is realized by two nested loops, the *Attitude Control Loop* and the *Angular Rate Control Loop*. The first loop computes an angular rate setpoint from the difference between the estimated attitude and the attitude setpoint (zero in case of hovering, prescribed by the user in flight mode). This angular rate is tracked with a proportional integral (PI) control. The Angular Rate Control Loop controls the motors with simple proportional controllers.

From the state machine description in Figure 6, it can be seen that there are two guidance modes. In Flying mode, the attitude setpoint is prescribed by the pilot. In Hovering mode, the attitude setpoint is zero but the transition from Flying mode to Hovering mode is realized by the *Gotofix* motion planning technique. The Hovering mode (zero speed and zero attitude) is maintained by the *Hovering Control Loop* which consists of a PI controller on the speed estimate.

The Gotofix motion planning is an important feature of the AR.Drone. The motion planning starts from the current attitude and speed of the quadrotor when the pilot leaves the Flying mode. The transient trajectory generation technique has been carefully designed off-line to obtain zero speed and zero attitude in short time (without overshoot), under the form of a feed-forward control determined by inversion of the governing dynamics.

Concentrating on the longitudinal velocity $u$ and the corresponding pitch angle $\theta$, the reference signal $\theta_{ref}(t)$ is computed inversing the following two dynamics

$$\dot{u} = -g\theta - C_x u \qquad (1)$$

where $C_x$ is the identified linear drag force coefficient, and the second order filter which results from a closed-loop identification
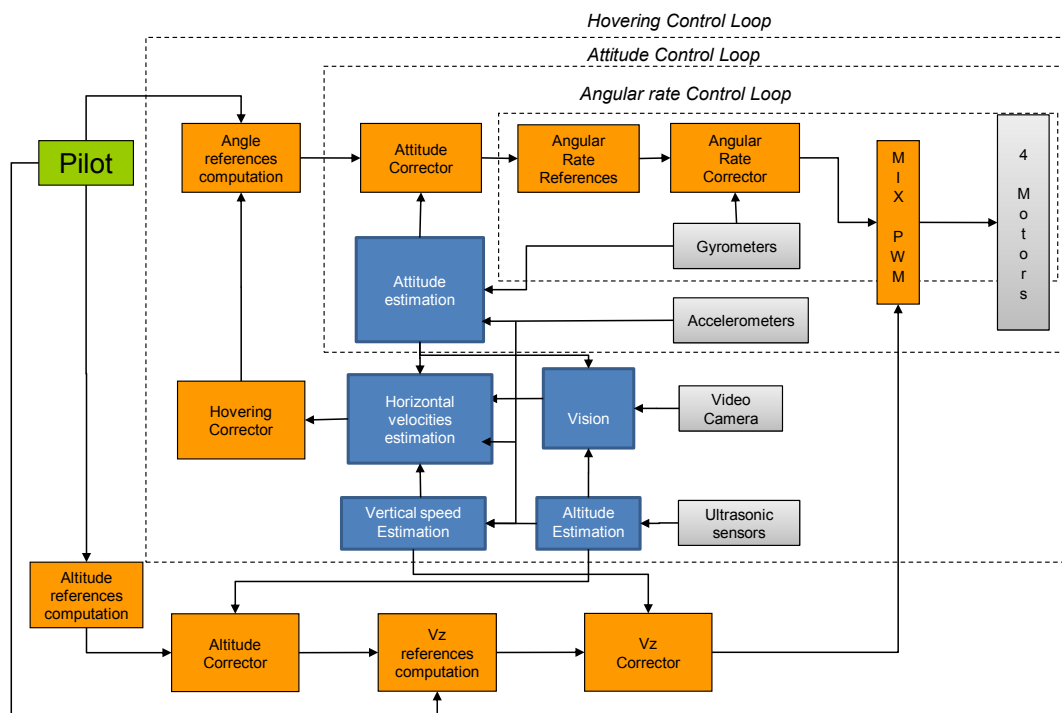
# Control principle



Fig. 5. Data fusion and control architecture.

$$\frac{\theta(s)}{\theta_{ref}(s)} = \frac{K}{s^2/\omega_0^2 + 2\xi s/\omega_0 + 1} \qquad (2)$$

The transients are tuned such that the following performances can be obtained (see Table 1 where various stop times are reported for varying initial longitudinal velocity).

| Initial speed | Outdoor hull | Indoor hull |
|---|---|---|
| $u_0 < 3$ m.s$^{-1}$ | 0.7 s | 1.5 s |
| $3 < u_0 < 6$ m.s$^{-1}$ | 1.0 s | 2.2 s |
| $u_0 > 6$ m.s$^{-1}$ | 1.5 s | 2.4 s |

Table 1. Stop times for different initial speed.

## 6. CONCLUSIONS AND PERSPECTIVES

The purpose of this paper is to present the navigation and control technologies embedded in the commercial micro UAV AR.Drone. As it appears, a main problem is the state estimation which has required to embed numerous sensors of various types. Among these are inertial sensors and cameras. The resulting estimation architecture is a complex combination of several principles, used to determine, over distinct time-horizons, the biases and other defects of each sensor. The outcome is a sophisticated system but this complexity is not visible by the user. This stresses the role of automatic control as an enabling but hidden technology.

Current developments are now focused on video games to use this aerial platform in interactive augmented reality gameplays. Now that low-level control algorithms have been developed and have proven to stabilize the system in sufficiently various flight modes, one can expect control design to play a key-role by means of guidance in such applications. This a subject for future works.

## REFERENCES

P.-J. Bristeau, P. Martin, E. Salaün, and N. Petit. The role of propeller aerodynamics in the model of a quadrotor uav. In *Proc. European Control Conference*, 2009.

P.-J. Bristeau, E. Dorveaux, D. Vissière, and N. Petit. Hardware and software architecture for state estimation on an experimental low-cost small-scaled helicopter. *Control Engineering Practice*, 18:733–746, 2010.

P. Castillo, A. Dzul, and R. Lozano. Real-time stabilization and tracking of a four rotor mini rotorcraft. *IEEE Trans. Control Systems Technology*, 12:510–516, 2004.

R. Cunha, D. Cabecinhas, and C. Silvestre. Nonlinear trajectory tracking control of a quadrotor vehicle. In *Proc. European Control Conference*, 2009.

E. Dorveaux, D. Vissière, A. P. Martin, and N. Petit. Iterative calibration method for inertial and magnetic sensors. In *Proc. 48th IEEE Conf. on Decision and Control*, 2009.

M.O. Efe. Robust low altitude behavior control of a quadrotor rotorcraft through sliding modes. In *Mediterranean Conf. on Control Automation*, 2007.

N. Guénard. *Optimisation et implémentation de lois de commande embarquées pour la téléopération intuitive de micro drones aériens X4-flyer*. PhD thesis, Université Nice Sophia-Antipolis, 2007.

T. Hamel and R. Mahony. Image based visual servo-control for a class of aerial robotic systems. *Automatica*, 43:1975–1983, 2007.

T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. Dynamic modelling and configuration stabilization for an x4-flyer. In *Proc. IFAC Symposium*, 2002.

G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. AIAA Guidance, Navigation, and Control Conference*, 2007.

B. K. P. Horn and B. G. Schunck. Determining optical flow: a retrospective. *Artificial Intelligence*, 59:81–87, 1993.

M.-D. Hua. *Contributions au contrôle automatique de véhicules aériens*. PhD thesis, Université de Nice Sophia-Antipolis, 2009.

M.-D. Hua, T. Hamel, P. Morin, and C. Samson. Control of thrust-propelled underactuated vehicles. Technical report, Institut National de Recherche en Informatique et Automatique, 2008.

W. Johnson. *Helicopter Theory*. Princeton University Press, 1980.

D. Jung and P. Tsiotras. Inertial attitude and position reference system development for a small uav. In *AIAA Infotech Aerospace*, 2007.

J. G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2002.

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Imaging Understanding Workshop*, pages 121–130, 1981.

S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *Proc. IEEE International Conf. on Robotics and Automation*, 2010.

R. Mahony, T. Hamel, and J.-M. Pflimlin. Complementary filter design on the special orthogonal group so(3). In *44th IEEE Conf. on Decision and Control*, 2005.

R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Trans. Automatic Control*, 53:1203 –1218, 2008.

P. Martin and E. Salaün. A general symmetry-preserving observer for aided attitude heading reference systems. In *47th IEEE Conf. on Decision and Control*, 2008.

P. Martin and E. Salaün. Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Engineering Practice*, 18:712–722, 2010.

N. Metni, J.-M. Pflimlin, T. Hamel, and P. Souères. Attitude and gyro bias estimation for a vtol uav. *Control Engineering Practice*, 14:1511–1520, 2006.

B. Mettler. *Identification Modeling and Characteristics of Miniature Rotorcraft*. Kluwer Academic Publishers, 2003.

E. Michaelsen and U. Stilla. Pose estimation from airborne video sequences using a structural approach for the construction of homographies and fundamental matrices. *Lecture Notes in Computer Science*, 3138:486–494, 2004.

R. Naldi, L. Marconi, and A. Sala. Modelling and control of a miniature ducted-fan in fast forward flight. In *Proc. American Control Conference*, 2008.

R. Naldi, L. Marconi, and L. Gentili. Robust takeoff and landing for a class of aerial robots. In *Proc. 48th IEEE Conf. on Decision and Control*, 2009.

P. Pounds, R. Mahony, and P. Corke. Modelling and control of a quad-rotor robot. In *Proc. Australasian Conf. on Robotics and Automation*, 2006.

R. W. Prouty. *Helicopter Performance, Stability, and Control*. Krieger Publishing Company Inc., 1990.

H. Romero, S. Salazar, A. Sanchez, P. Castillo, and R. Lozano. Modelling and real-time control stabilization of a new VTOL aircraft with eight rotors. In *Proc. of IEEE International Conf. on Intelligent Robots and Systems*, 2007.

E. Rondon, I. Fantoni-Coichot, A. Sanchez, and G. Sanahuja. Optical flow-based controller for reactive and relative navigation dedicated to a four rotor rotorcraft. In *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems*, 2009.

A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Trans. Control Systems Technology*, 14:562–571, 2006.

D. H. Titterton and J. L. Weston. *Strapdown Inertial Navigation Technology*. The American Institute of Aeronautics and Aeronautics, 2004.

M. Trajkovic and M. Hedley. Fast corner detection. *Image and Vision Computing*, 16:75–87, 1998.

D. Vissière. *Solution de guidage navigation pilotage pour véhicules autonomes hétérogènes en vue d'une mission collaborative*. PhD thesis, Ecole des Mines de Paris, 2008.

D. Vissière, P.-J. Bristeau, A. P. Martin, and N. Petit. Experimental autonomous flight of a small-scaled helicopter using accurate dynamics model and low-cost sensors. In *Proc. 17th IFAC World Congress*, volume 17, 2008.