JOURNAL AerospaceLab

ONERA
THE FRENCH AEROSPACE LAB

return on innovation

Issue 8
December 2014

DOI : 10.12762/2014.AL08

Aerial Robotics

On line
www.aerospacelab-journal.com
Webmaster ONERA

Contact
E-mail: aerospacelab@onera.fr

Produced by
ONERA - BP 80100
Chemin de la Hunière
et des Joncherettes
FR-91123 PALAISEAU CEDEX
FRANCE
www.onera.fr

ISSN: 2107-6596

Aerial Robotics: a Bird's-Eye View

Pascal Morin
(UPMC-CNRS)
Holder of the RTE-UPMC
Research Chair
on "Autonomous Mini-Drones"

Philippe Bidaud
(Onera)
Scientific Director
of the Information Processing
and System Branch

DOI : 10.12762/2014.AL08-00

After manufacturing, ground transportation and medicine, robotics has now made an incursion in the field of aerial applications. Several domains, such as mapping, shooting, monitoring of indoor and outdoor 3D environments, agriculture and traffic monitoring, surveillance of sensitive areas, structure inspection, handling and carrying of heavy loads, and physical interventions now seek to exploit what are commonly called "drones". While these unmanned aerial vehicles (whether called UAV, UAS, or RPAS) have reached a fair degree of maturity, as witnessed by their success in entirely new aerial missions (dangerous, long, tedious, etc.), their capabilities and their performance generally remain limited. These systems are still endowed with scant autonomy capabilities, in particular with regard to their capacity for sensing and interacting with their environment, and significant progress is expected in this direction. Other topics of practical importance concern energetic autonomy (*i.e.*, the capacity to fly longer), or avionic architecture in relation with security issues. Robotics will undoubtedly play a major role in replacing humans onboard these aerial vehicles. Robotics is one of the scientific fields of information science that relies on computer science, automatic control and signal and image processing. The involvement of several of these different aspects in the development of next generation unmanned aerial vehicles is discussed in the articles contained in this special issue.

Introduction

After manufacturing, ground transportation and medicine, robotics has now made an incursion in the field of aerial applications. Several domains, such as mapping, shooting, monitoring of indoor and outdoor 3D environments, agriculture and traffic monitoring, surveillance of sensitive areas, structure inspection, handling and carrying of heavy loads and physical interventions, now seek to exploit what are commonly called "drones". These Aerial Robotics systems can also be a flexible alternative to satellites and antennas for optimal network territory coverage. Today, their application potential is recognized as quite considerable.

While these unmanned aerial vehicles (whether called UAV, UAS, or RPAS) have reached a fair degree of maturity, as witnessed by their success in entirely new aerial missions (dangerous, long, tedious, etc.), their capabilities and their performance generally remain limited. These systems are still endowed with scant autonomy capabilities, in particular with regard to their capacity for sensing and interacting with their environment, and significant progress is expected in this direction. Other topics of practical importance concern energetic autonomy (*i.e.*, the capacity to fly longer), or avionic architecture in relation with security issues. For now, the bottom line seems to be that, while research and development efforts are underway to mitigate barriers to the safe and routine integration of unmanned aircraft into the national airspace, these efforts cannot be completed and validated without safety, reliability and performance standards.

Robotics will undoubtedly play a major role in replacing humans onboard these aerial vehicles. Robotics is one of the scientific fields of information science that relies on computer science, automatic control and signal and image processing. It also draws on engineering for modeling at various levels of fidelity and for mechatronical design. It helps to address design and control systems problems, in order to develop a form of intelligence enabling very different tasks to be carried out autonomously (or with some human assistance) in highly dynamic environments.

The science and technology of Robotics, as such, cover a relatively wide range. They focus on aspects related to:

- Structural systems design;
- Control systems, computer architecture (hardware and software) for control;
- Perception (often multi-modal) of the environment, detection and information retrieval at various semantic levels;
- Localization and navigation in complex environments,
- Development of complex functions in autonomous forms and/or interaction with humans or other robotic agents;
- Planning of these functions for performing tasks/complex missions, their dynamic linking, the execution control and adaptation activities with "cognitive" artificial systems;
- Development of learning methods for the representation and solution of problems relating to particular tasks or missions, based on machine learning.

The involvement of several of these various aspects in the development of next generation unmanned aerial vehicles is discussed in the articles contained in this special issue.

Mechanical design and pre-sizing of unmanned aerial vehicles pre-sizing

Most mini-drones are rotary-wing aircraft, due to the ability of this family of systems to perform vertical take-off and landing. There is a very large variety of existing concepts and the development of new platforms is still an active research topic. Among the various possible objectives, obtaining a good trade-off between good hovering capacity and good energy efficiency in cruising flight for a given speed range, maximum weight, payload, dimension, etc., is a complex optimization problem that can involve multidisciplinary models. An article in this special issue discusses these various aspects and describes research projects conducted at Onera for rotary-wing aircraft design and pre-sizing.

Reactive control

Once a UAV (Unmanned Aerial Vehicle) has been built, the first step is to design low-level feedback control laws that ensure good flight stability. For VTOL (Vertical Take-Off and Landing) aircraft, this starts with attitude control, since these systems are inherently unstable. Higher level control objectives include, for example, velocity/position control based on GPS data, which is now part of the functionalities available on most commercial mini-drones. More recently, progress in the miniaturization, cost and energy consumption of micro-processors have made possible the on-board processing of complex sensory data, such as camera data. This opens the door to feedback control strategies based on exteroceptive sensors (cameras, laser scanners), which are very well suited to inspection applications. Several articles in this issue address these various topics from different viewpoints: modeling of strapdown IMU (Inertial Measurement Unit) measurements in relation with quadrotor UAV aerodynamics, design of attitude observers, design of non-linear feedback control laws to ensure stability over large flight envelopes and perturbation rejection, and vision-based control.

Planning and mapping

Reactive control may be sufficient for the remote operation of an aerial vehicle by a human pilot. To further increase the autonomy and envision fully autonomous flight in cluttered environments, a high level of perception and planning capability must be reached. First, the vehicle must be able to build a map of its environment and locate itself on this map. This is a major problem in robotics, usually referred to as the SLAM problem (Simultaneous Localization and Mapping). Once this map is available, planning strategies can be developed so as to ensure safe navigation in the environment. Results on visual SLAM conducted at Onera are presented in one paper in this special issue and planning issues are also covered in two articles. The first, which also describes research work conducted at Onera, addresses planning and training flight for a fleet of vehicles. The second describes research conducted at the University of Sidney on autonomous soaring, with the purpose of achieving long endurance flight. This article also discusses other means to achieve this goal.

Avionic architecture and security

A high level of autonomy can only be achieved via the use of many different sensors and the on-board processing of the associated sensory data. This leads to complex systems, at both the hardware and software levels. Such complexity is a challenge to the design of certified avionics, especially for small UAVs, which are subjected to very strong payload constraints. On the other hand, with the rapid increase in the number of UAVs operating close to inhabited areas, security demands will increase. Based on existing methods that were developed for commercial fixed-wing aircraft, work is being conducted in research labs to develop certified avionics architectures tailored to UAVs. An article in this special issue is dedicated to these aspects ■

J.-M. Biannic, L. Burlion,
H. de Plinval
(Onera)

E-mail: jean-marc.biannic@onera.fr

DOI : 10.12762/2014.AL08-01

Robust Control Design over Large Flight Envelopes: a Promising Approach for Aerial Robotics

In the coming years, unmanned combat air vehicles will be designed to outperform most pilot-in-the-loop systems. The absence of a pilot onboard allows larger flight envelopes to be considered with, for example, higher load factors. From a control perspective, the above remark induces new needs for improved design methods to obtain robust controllers that will automatically adapt to extremely varying flight conditions. Based on the well-known concept of nonlinear dynamic inversion (NDI), the approach of this paper introduces an original and rather generic "robustification" framework, which leads to a multi-objective H_∞ design problem. The latter is now easily solved by existing and efficient numerical tools based on nonsmooth optimization. The proposed design methodology is illustrated by a combat aircraft control problem.

Introduction

Unlike civilian aircraft, unmanned combat air vehicles are expected to operate in extended flight domains with high performance requirements. In this challenging context, control systems have to cope with rapidly varying and possibly nonlinear dynamics, induced by large variations of load factors, angle-of-attack or sideslip angles for example. As a result, popular gain scheduling techniques essentially based on linear interpolations of static gains are no longer applicable, since no guarantee exists in case of rapid changes in the gain scheduling variables [35, 36]. This weakness was the main motivation behind initial research on LPV control techniques in the 1990's [23, 6, 3, 2, 46, 8, 17, 20]. More recently, over the past decade, many efforts were devoted to various improvements, in order to reduce the conservatism of standard LPV design approaches [34, 1, 44, 32, 42, 45, 33]. Unfortunately, most of the proposed extensions introduce many additional variables and lead to (possibly non-convex) optimization problems, which rapidly become intractable beyond three parameters. This certainly explains why more standard gain-scheduling design approaches are still experiencing a large success in the aerospace industry today. This has also been a strong motivation for many researchers who have explored connections between LPV control and gain-scheduling methods [36, 31] or have provided frameworks [30, 35, 13] thanks to which the latter can be theoretically justified. Based on such frameworks, the development of specific implementation strategies for gain-scheduled controllers was made possible. A possible strategy uses the concept of velocity-based linearization [19, 21, 18], but the most popular ones are certainly stability preserving interpolation methods [37, 38, 25, 39, 40]. Within the specific context of aerospace applications, for which analytical models are often available, LPV or gain-scheduled control laws can be interestingly replaced by nonlinear dynamic-inversion (NDI) based controllers. Moreover, as is clarified in [26, 43, 24], strong links can be established between LPV and NDI control.

These links offer interesting ways to study the stability of nonlinear closed-loop systems with NDI controllers. They also provide new possibilities for improving the design of robust NDI controllers, as is further clarified below.

The outline of this paper is as follows. The connections between the LPV framework and dynamic-inversion (DI) based techniques are preliminarily explored. Next, an application to a generic aircraft control problem is considered and an original tuning procedure is detailed. Results of the tuning procedure are then presented, and finally some concluding comments end the paper.

Dynamic-inversion based LPV control

Since the early 1990s, nonlinear dynamic-inversion (NDI) based (or feedback linearization) control techniques have received considerable attention in the literature regarding their application to the design of flight control systems for missiles and aircraft. Given a possibly time-varying nonlinear plant:

$$\dot{x} = f(x,\theta) + g(x,\theta)\,u \qquad (1)$$

a control law that achieves the desired response characteristics may be formulated as follows

$$u = g(x,\theta)^{-1}\ (v - f(x,\theta)) \tag{2}$$

where v specifies the desired response and is generally produced as the output of a linear controller, remarking that – in the ideal case – the closed-loop system now simply reads:

$$\dot{x} = v \tag{3}$$

Interestingly, when an accurate model is available, the control structure (2) compensates not only for the nonlinearities of the plant, but also for its parametric variations, as is further clarified below. Unfortunately, an exact compensation is never achieved in practice because of uncertainties in the model, because of noises in the measurements, because part of the states might not be available, because the control efficiency $g(x,\theta)$ might be temporarily non-invertible and, finally, because of control saturations. For these reasons, making the desired response v requires special attention.

Within the context of Linear Parameter Varying (LPV) plants, the nonlinear differential equation (1) can be rewritten as:

$$\begin{cases} \dot{x} = A(\theta)x + B(\theta)u \\ z = Lx \end{cases} \tag{4}$$

where z denotes the signal to be tracked and $B(\theta)$ is a square and non-singular matrix throughout the operating domain of the plant. Thus, the above "linearizing" control law (2) can be adapted as follows:

$$u = B(\theta)^{-1}\ (B_0 v - (A(\theta) - A_0)x) \tag{5}$$

so that the LPV plant now becomes LTI:

$$\begin{cases} \dot{x} = A_0 x + B_0 v \\ z = Lx \end{cases} \tag{6}$$

for which the new control input v in the following format:

$$v = H(s)\ z_c + K(s)x \tag{7}$$

is easily designed by any standard approach. It is easily verified that the combination of equations (5) and (7) defines a standard LPV control law:

$$u = B(\theta)^{-1}\left[B_0 H(s) z_c + (B_0 K(s) + A_0 - A(\theta))x\right] \tag{8}$$

It is very interesting here to point out that any difficulty related to the size of the parametric vector θ has been removed, which, for many reasons, is clearly a weak point of most LPV or gain scheduling techniques. Note that the selection of the "central" matrices A_0 and B_0 is completely free. A standard choice consists in setting $A_0 = A(\tilde{\theta})$ and $B_0 = B(\tilde{\theta})$ where $\tilde{\theta}$ denotes a mean value of the varying parameter. However, in some cases, it might not be the best. Rather than considering mean values of the parameters, an interesting alternative consists in focusing on worst case combinations, for which the instability degree of A_0, for example, is maximized, or for which the control efficiency is minimized. The central idea is that the LTI system (6) should not necessarily capture the mean behavior of the LPV plant, but rather a worst case behavior. Unfortunately, there are still no general rules for the selection of the "central" model. This is still an open issue and it seems that the best choice will highly depend on the application. Another difficulty of the proposed approach is related to the assumptions regarding the matrix $B(\theta)$, which must be square and invertible. In practice, these two requirements are rarely met. However, within the context of aerospace systems, a time scale separation technique can be used to bypass such difficulties [26]. This point is clarified in the following.

Application to a generic aircraft control problem

Let us consider a longitudinal aircraft short-term dynamics control problem, where the objective is to track the angle-of-attack over a large flight envelope. Using standard flight dynamics notation (see [10] for further details), the equations of interest are as follows, where α and q denote the angle-of-attack and the pitch rate, respectively:

$$\begin{cases} \dot{\alpha} = q + w_\alpha(\theta_p) \\ J_{yy}\ \dot{q} = w_q(\theta_p) + \lambda(\theta_p)\delta m \end{cases} \tag{9}$$

with:

$$\begin{cases} w_\alpha(\theta_p) = \dfrac{g}{V\cos\beta}(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta \\ \qquad\qquad + \tilde{a}_z\cos\alpha - \tilde{a}_x\sin\alpha) - p_i\tan\beta \\ w_q(\theta_p) = q_d SL(C_{M_\alpha}(M_\alpha) + qLC_{M_q}/V) \\ \qquad\qquad + (J_{zz} - J_{xx})pr + J_{xz}(r^2 - p^2) \\ \lambda(\theta_p) = q_d SLC_{M_\delta}(M_\alpha) \end{cases} \tag{10}$$

and:

$$\begin{cases} p_i = p\cos\alpha + r\sin\alpha \\ \tilde{a}_x = ga_{xm} + L_a g(q^2 + r^2) \\ \tilde{a}_z = ga_{zm} + L_a g(\dot{q} - pr) \end{cases} \tag{11}$$

In the above representation, all parametric variations of the system (mainly induced by the variations of velocity) are captured by the parametric vector θ_p. The notation θ, widely used in the LPV control literature, is no longer appropriate here, since it now denotes the pitch angle (see Equation 10). The longitudinal short-term motion of the aircraft is essentially controlled through the second equation in (9), via the control input δ_m denoting the elevator deflection. The latter also impacts the first equation through a small effect on the accelerations $\tilde{a}_x$ and $\tilde{a}_z$. However, the latter is small enough to be neglected.

Interestingly, the expressions of the nonlinear inputs (w_α and w_q) and of the control efficiency ($\lambda(\theta_p)$) given in (10) depend on known and on-line measurable data. Consequently, these three parameter-varying terms can be used by the controller. Observing that the control efficiency verifies $\lambda(\theta p) < 0$ (and is thus invertible) over the entire flight envelope, a standard NDI approach consists in inverting the moment equation, so as to control the pitch rate. Assuming that the actuator dynamics are much faster than the desired response on q, they are temporarily neglected. It is then readily verified that the following control law:

$$\delta_{mc} = \lambda(\theta_p)^{-1}(J_{yy}\tau_q^{-1}(q_c - q) - w_q) \quad (12)$$

yields:

$$\dot{q} \approx \tau_q^{-1}(q_c - q) \quad (13)$$

Since the pitch rate evolves much faster than the angle-of-attack, (this also can be enforced by choosing τ_q small enough), one further considers that $q \approx q_c$, so that the first equation in (9) is now controlled via q_c and the following choice for the commanded pitch rate:

$$q_c = \omega_r^2 \int_o^t (\alpha_c - \alpha)\, d\tau - 2\xi_r\omega_r\alpha - w_\alpha \quad (14)$$

enforces a second-order behavior on the angle-of-attack:

$$\frac{\alpha}{\alpha_c}(s) \approx \frac{\omega_r^2}{s^2 + 2\xi_r\omega_r s + \omega_r^2} \quad (15)$$

where the desired closed-loop pulsation ω_r is chosen as a function of the calibrated airspeed. Combining the above equations, one observes that the nonlinear parameter-varying control law may be summarized as:

$$u_c = \lambda(\theta_p)\delta_{mc} = H(s)[\alpha_c \;\; w_\alpha \;\; w_q]' + K(s)[\alpha \;\; q]' \quad (16)$$

with

$$H(s) = \left[\frac{J_{yy}\omega_r^2}{\tau_q s} \quad -\frac{J_{yy}}{\tau_q} \quad -1\right],\; K(s) = -\frac{J_{yy}}{\tau_q}\left[\frac{\omega_r^2}{s} + 2\xi_r\omega_r \quad 1\right] \quad (17)$$

Note that Equation (16) can be viewed as an alternative formulation of (8). With this approach, rather sophisticated parameter-varying control laws covering the entire flight domain of interest are then easily obtained after a very short design process. However, the efficiency of such control laws relies strongly on the availability of the nonlinear inputs w_α and w_q. In practice, uncertainties affect these two signals and the control efficiency $\lambda(\theta_p)$ is not known precisely. Moreover, because of the actuator dynamics, the actual deflection δ_m may differ sometimes significantly from the commanded variable δ_{mc}. Within the context of dynamic inversion, several techniques have been proposed to cope with these limitations, by improving the robustness of the controller [26]. The central idea of these techniques consists in mixing the concept of dynamic inversion with robust control theory. Based on this idea, in the design procedure which is detailed next, a multi-objective H_∞ framework is proposed to generalize the above control structure and to better optimize the gains $K(s)$ and $H(s)$.

A novel tuning procedure

As mentioned above, the desired elevator deflection is not produced instantaneously, but rather is delivered by an actuator of limited capacity. Within our context, its dynamics are accurately described by a linear second-order transfer function. Here, to further simplify the following discussion, let us temporarily reduce it to a first-order system so that $\dot{\delta}_m = \tau^{-1}\left(\delta_{m_c} - \delta_m\right)$ and let us define the new variable $u = \lambda(\theta_p)\delta_m$. Thus, one obtains:

$$\dot{u} = \lambda(\theta_p)\dot{\delta}_m + \dot{\lambda}(\theta_p)\delta_m = \lambda(\theta_p)\dot{\delta}_m + w_u = \tau^{-1}(u_c - u) + w_u \quad (18)$$

where the commanded input u_c is defined in (16) and the perturbation term wu may further be characterized as:

$$w_u = \frac{\dot{\lambda}(\theta_p)}{\lambda(\theta_p)}u = \mu(\theta_p)u \quad (19)$$

Hence, the nonlinear aircraft model of equation (9), including the actuator, can be drawn as shown in figure 1, where the state-space data of the linear system $G(s) = C_G(sI - A_G)^{-1}B_G$ is initially given by:

$$A_G = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\; B_G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & J_{yy}^{-1} & J_{yy}^{-1} \end{bmatrix},\; C_G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (20)$$

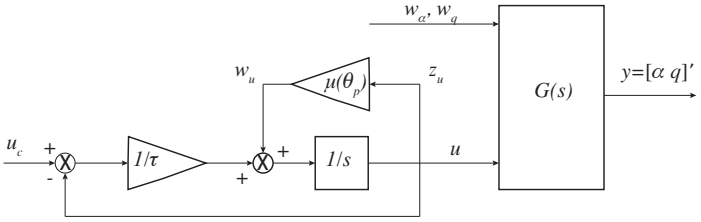


Figure 1 - Description of a nonlinear plant as a linear system with nonlinear inputs

Remark *As already discussed above, the linear system $G(s)$ should at least locally represent a realistic behavior of the aircraft, which is definitely not the case in (20), which corresponds to a double-integrator. For given values $\tilde{\theta}_p$ of the varying parameters, a linearization technique yields:*

$$\begin{cases} w_\alpha = \dfrac{\partial w_\alpha}{\partial \alpha}(\tilde{\theta}_p).\alpha + \dfrac{\partial w_\alpha}{\partial q}(\tilde{\theta}_p).q + \tilde{w}_\alpha \\ w_q = \dfrac{\partial w_q}{\partial \alpha}(\tilde{\theta}_p).\alpha + \dfrac{\partial w_q}{\partial q}(\tilde{\theta}_p).q + \tilde{w}_\alpha \end{cases} \quad (21)$$

Thus, the nonlinear inputs of $G(s)$ become $\tilde{w}_\alpha$ and $\tilde{w}_q$ and its A_G matrix is updated as follows:

$$A_G(\tilde{\theta}_p) = \begin{bmatrix} \dfrac{\partial w_\alpha}{\partial \alpha}(\tilde{\theta}_p) & 1 + \dfrac{\partial w_\alpha}{\partial q}(\tilde{\theta}_p) \\ \dfrac{\partial w_q}{\partial \alpha}(\tilde{\theta}_p) & \dfrac{\partial w_q}{\partial q}(\tilde{\theta}_p) \end{bmatrix} \quad (22)$$

Formulation as a multi-channel H_∞ design problem

Merging the above actuator description and $G(s)$, an augmented linear interconnection model $M(s)$ can be obtained as the linear multivariable transfer matrix from inputs $w_u, \tilde{w}_\alpha, \tilde{w}_q$ and u_c to the outputs $z_\alpha = \alpha$, $z_u = u$ and $z_y = y$. This linear interconnection is the central element of the H∞ design diagram visualized in figure 2.

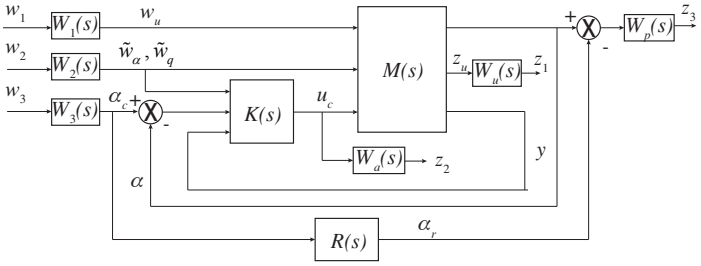


Figure 2 - Design model

Basically, the general idea consists in computing the best controller $K(s)$, such that a few relevant weighted transfers from w_i to z_i are minimized. More precisely, the transfer from w_3 to z_3 can be associated with the nominal performance, since w_3 corresponds to a control input on the angle-of-attack while z_3 denotes the error between the actual output and that of the reference model $R(s)$. Thus, the problem to be solved takes the form of the following multi-channel H_∞ optimization program:

$$\min_{K(s)} \left\| T_{w_3 \to z_3}(s) \right\|_\infty \quad \text{with} \quad \begin{cases} \left\| T_{w_1 \to z_1}(s) \right\|_\infty \le c_{11} \\ \left\| T_{w_2 \to z_3}(s) \right\|_\infty \le c_{23} \\ \left\| T_{w_3 \to z_2}(s) \right\|_\infty \le c_{32} \\ \left\| T_{w_2 \to z_2}(s) \right\|_\infty \le c_{22} \end{cases} \tag{23}$$

where constant terms c_{ij} can be tuned to quantify various robustness levels:

- c_{11}: stability robustness against the neglected term $w_u = \mu(\theta)u$
- c_{23}: performance robustness against perturbations $\tilde{w}_\alpha$ and $\tilde{w}_q$
- c_{32}: bound on the nominal control activity
- c_{22}: bound on the "perturbed" (by $\tilde{w}_\alpha$ and $\tilde{w}_q$) control activity

Interpretation and controller structure

As is visible in the diagram of figure 2 and from (23), the nonlinear control design problem has been re-formulated as a rejection problem of on-line estimated nonlinear input perturbations. Assume that a controller $K(s)$ has been computed, then the control law to be implemented will read:

$$\delta_{mc} = \frac{1}{\lambda(\theta_p)} K(s)[\tilde{w}_\alpha, \tilde{w}_q, \alpha_c - \alpha, y]' \tag{24}$$

The above expression generalizes (16). Here, the unique compensator $K(s)$ includes both the feedforward (previously denoted $H(s)$) and feedback (previously denoted $K(s)$) paths. Note that the signal w_u is not used by the controller, since its estimation might be very poor (because of possibly fast variations). Moreover, it has been observed in practice that this signal is often very small. As is clear from (19), its magnitude is directly related to the rate-of-change in the varying parameters.

Weighting function tuning procedure

The most challenging task in H_∞ design approaches, once the controller structure has been defined, consists in tuning the weighting functions correctly. This step is often very tricky. Fortunately, the situation is quite favorable here, since an initial solution can be obtained by a standard dynamic inversion approach (see Equations (16) and (17)). As a result, the weighting functions can be tuned by a frequency-domain analysis of the design model in feedback loop with this standard solution. Thus, all that remains is to iterate from this starting point, to improve the standard controller. Alternatively, the tuning procedure may also be started from scratch when the standard dynamic inversion based controller fails to satisfy any of the desired closed-loop properties.

Dealing with saturations

A key improvement axis in the above procedure consists in minimizing the control activity by "playing" on the weighting function $W_a(s)$, for example. It is then expected that magnitude and rate limits will no longer induce loss of performance or stability. In a next step, further improvements can be obtained by plugging in an anti-windup compensator $J(s)$, which can also be optimized by H_∞ norm minimization, together with the previous feedback gain $K(s)$. Further details on such an approach can be found in [7] and [9].

Resolution aspects

Last but not least, the resolution of the multi-channel H_∞ optimization problem (23) deserves a few comments. Unlike standard full-order H_∞ design problems, the latter is non-convex, because of the multi-channel aspect. Moreover, as shown by Figure 2, numerous weighting functions have been introduced in the design model, which contributes to a significant increase in its number of states. As a result, the optimization of a full-order controller would certainly result in non-implementable control laws. It is thus strongly recommended here to search for fixed-order and structured controllers, which is a second source of non-convexity. Until recently, these problems were very hard to solve, which certainly explains why formulations such as those stated by (23) have rarely been considered. However, over the last few years, thanks to very recent progress on nonsmooth optimization techniques [4, 5, 11], new efficient tools dedicated to the local optimization of fixed-order and fixed structure H_∞ controllers have appeared. Let us first cite the public domain software HIFOO [15, 16, 14] for use with MATLAB, which is backed up by the theoretical advances described in [11]. Then, appeared the routine HINFSTRUCT, whose theoretical foundations are described in [4]. The latter has been directly integrated to MATLAB by Mathworks Inc., and is available with the Robust Control Toolbox [22].

Results on the flight control problem

The above strategy is now applied to the flight control problem, but here both the longitudinal and the lateral axes are considered. Baseline controllers are preliminarily designed for each axis. Next, both controllers are plugged into design models, as shown in figure 2, where all weighting functions are first set to identity. A singular value analysis is then performed in order to initialize the weighting functions and the fixed-order multi-channel H_∞ (23) is preliminarily solved for each axis. In each case, the initial set of weighting functions is chosen so that all constraints c_{ij} are normalized. The best achieved H_∞ norm of the main transfer associated to the nominal performance also verifies: $\left\| T_{w_3 \to z_3}(s) \right\|_\infty < 1$. Then, an iterative procedure is applied to decrease the constants c_{ij}, while preserving the nominal performance constraint. During this procedure, one essentially tries to minimize c_{23}, which reflects the capacity of the controllers to reject the nonlinear input signals, thus extending their operating domains. For each axis, this value is approximately divided by 3. During this iterative tuning procedure, the transfer $T_{w_3 \to z_1}(s)$ from w_3 to z_1 will also receive particular attention, as well as the selection of the weighting function $W_u(s)$, in order to minimize the control activity.

Point #	Mach number	Altitude (ft)
1	0.25	5000
2	0.5	20000
3	0.7	10000
4	0.9	36000
5	0.9	5000

Table 1: Test points in the flight domain

Both longitudinal and lateral controllers are then implemented in a nonlinear SIMULINK diagram including a complete description of the aircraft, which remains valid over the entire subsonic flight domain. Ten simulations are then performed to evaluate both the longitudinal and lateral controllers throughout the flight domain, in which 5 points are selected (see table 1).

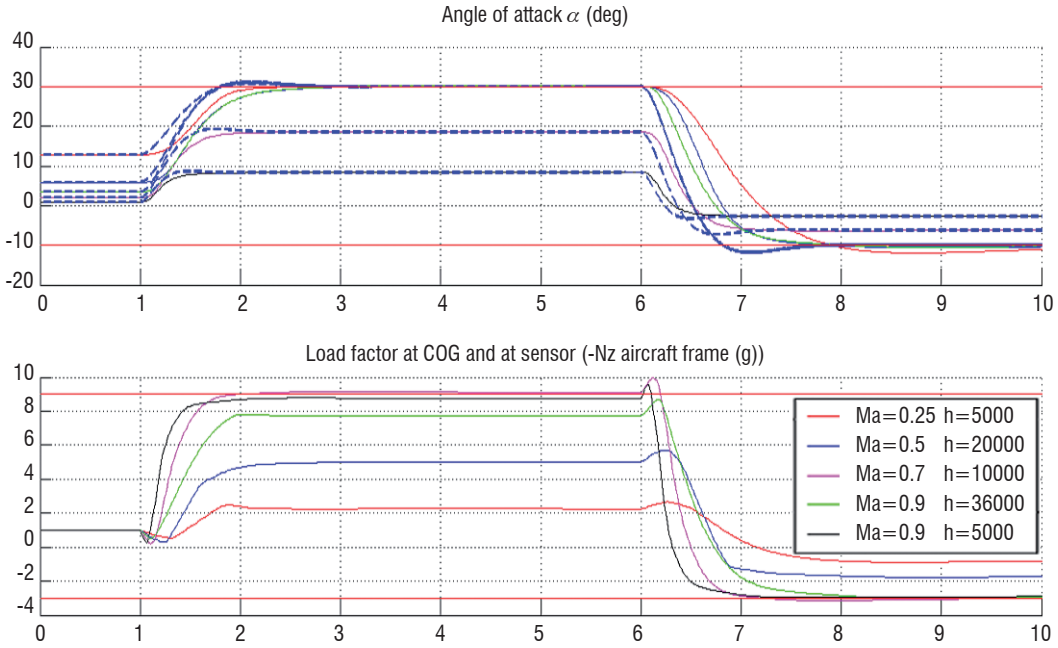


Figure 3 - Nonlinear simulations for various flight conditions: longitudinal axis

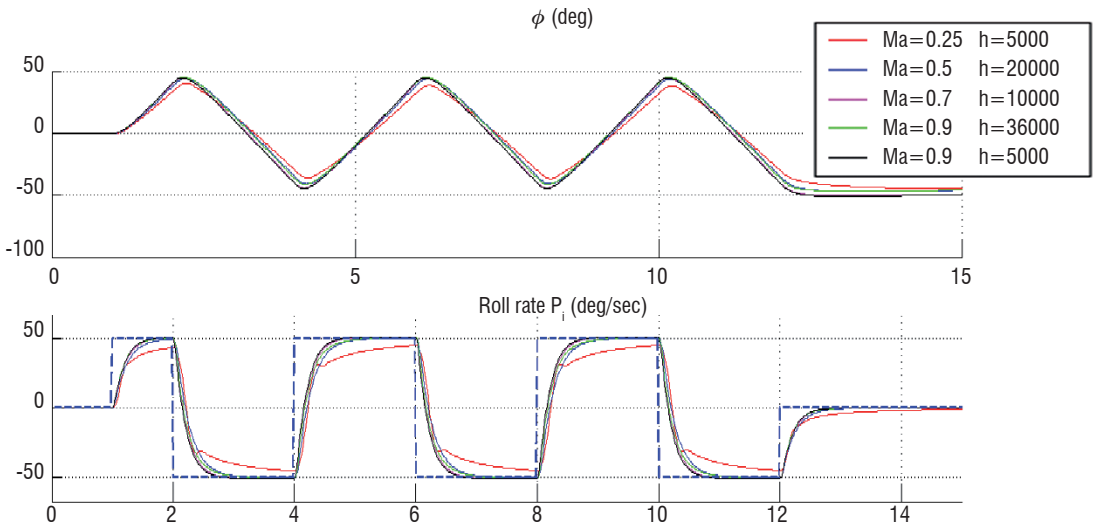


Figure 4 - Nonlinear simulations for various flight conditions: lateral axis

For each of these points, the aircraft is preliminarily trimmed (initial thrust and elevator deflections are set to ensure steady state flight conditions) and two maneuvers are performed:

• **Longitudinal maneuver:** this sequence consists of two steps on the angle-of-attack. The flrst is applied after $1s$. Its magnitude is tuned according to the flight point, so that the vertical load factor does not exceed the maximum value of $9g$. Next, after $6s$ a new step is applied, so that the final angle-of-attack is now between 0 and $-10deg$. Here again, the magnitude is adapted as a function of the flight point, so that the vertical load factor remains above its minimum value, which is fixed to $-3g$. The total length of this maneuver is $10s$. The simulation results are visible in figure 3.

• **Lateral maneuver:** during this sequence, the lateral behavior of the aircraft control laws is evaluated through their capacity to track roll-rate commands. For this purpose, a series of roll-rate steps is applied, as shown in figure 4. During these steps, the objective is to maintain the sideslip angle around 0.

Concluding remarks

In this paper, an original control design methodology combining the concepts of dynamic inversion and LPV control techniques has been described. The proposed strategy, which essentially consists in revisiting NDI control as a linear control problem with measured (or estimated) nonlinear disturbing inputs, is particularly well-suited to aerospace applications. The proposed design approach has been validated on a realistic and complete aircraft control problem over a large flight envelope.

A key advantage of this last parameter-varying control strategy resides in its capacity for handling many parameters without critical impact during the design process.

However, when the nonlinear input signals – which in most cases have to be estimated on-line – differ significantly from reality, it becomes difficult to predict whether the closed-loop properties will be guaranteed or not. A controller validation phase is then required. Such validations generally consist of extensive nonlinear simulations for many flight conditions, many parametric configurations and many different types of maneuvers. This unavoidable process takes a lot of time. This is why many efforts have been recently devoted to the development of numerically cheaper validation techniques for parameter-varying flight control laws. The interested reader may consult references [28, 27, 29, 12] and the book [41] and the references therein ■

Acronyms

LPV (Linear Parameter Varying)
LTI (Linear Time Invariant)
NDI (Nonlinear Dynamic Inversion)

References

[1] P. APKARIAN, R. ADAMS - *Advanced Gain Scheduling Techniques for Uncertain Systems*. IEEE Transactions on Control Systems Technology, 6(1):21–32, January 1998.

[2] P. APKARIAN, J-M. BIANNIC - *Self-Scheduled H_∞ Control of Missiles Via Linear Matrix Inequalities.* Journal of Guidance Control and Dynamics, 18(3):532–538, May-June 1995.

[3] P. APKARIAN, P. GAHINET - *A Convex Characterization of Gain Scheduled H_∞ Controllers*. IEEE Transactions on Automatic Control, 40(5):853–864, May 1995.

[4] P. APKARIAN, D. NOLL - *Nonsmooth H_∞ Synthesis.* IEEE Transactions on Automatic Control, 51(1):71–86, 2006.

[5] P. APKARIAN, D. NOLL - *Nonsmooth Optimization for Multidisk H_∞ Synthesis*. European Journal of Control, 12(3):229–244, 2006.

[6] G. BECKER, A. PACKARD - *Robust Performance of Linear Parametrically Varying Systems Using Parametrically-Dependent Linear Feedback*. Systems and Control Letters, 23(3):205–215, September 1994.

[7] J-M. BIANNIC, P. APKARIAN - *Anti-Windup Design Via Nonsmooth Multi-Objective H_∞ Optimization*. In Proceedings of the ACC, San Francisco, CA. USA, June 2011.

[8] J-M. BIANNIC, P. APKARIAN, W. GARRARD - *Parameter-Varying Control of a High Performance Aircraft.* Journal of Guidance Control and Dynamics, 20(2):225–231, March-April 1997.

[9] J-M. BIANNIC, L. BURLION, S. TARBOURIECH, G. GARCIA - *On Dynamic Inversion with Rate Saturations*. In Proceedings of the ACC, Montreal, June 2012.

[10] J.-L. BOIFFIER - *The dynamics of flight : The equations*. Wiley, New York, 1998.

[11] J.-V. BURKE, D.HENRION, A.-S. LEWIS, M.-L. OVERTON - *Stabilization Via Nonsmooth Nonconvex Optimization*. IEEE Transactions on Automatic Control, 51(11):1760–1769, 2006.

[12] F. DEMOURANT - *New Algorithmic Approach Based on Integral Quadratic Constraints for Stability Analysis of high Order Models*. In Proceedings of the ECC, pages 359–364, Zurich, Switzerland, July 2013.

[13] V. FROMION, G. SCORLETTI - *A Theoretical Framework for Gain Scheduling*. International Journal of Robust and Nonlinear Control, 13(10):951–982, August 2003.

[14] S. GUMUSSOY, D. HENRION, M. MILLSTONE, M.-L. OVERTON - *Multi-Objective Robust Control with HIFOO 2.0.* In Proceedings of the IFAC Symposium on Robust Control Design, Haifa, Israel, July 2009.

[15] S. GUMUSSOY, M. MILLSTONE, M.-L. OVERTON - *H_∞ Strong Stabilization via HIFOO, a Package for Fixed-Order Controller Design.* In Proceedings of the IEEE CDC, pages 4135–4140, Cancun, Mexico, 2008.

[16] S. GUMUSSOY, M.L. OVERTON - *Fixed-Order H_∞ Controller Design via HIFOO, a Specialized Nonsmooth Optimization Package*. In Proceedings of the ACC, Seattle, USA, June 2008.

[17] A. HELMERSSON - *μ Synthesis and LFT Gain Scheduling with Real Uncertainties*. International Journal of Robust and Nonlinear Control, 8(7):631–642, June 1998.

[18] C.-D.-C. JONES, M.-H. LOWENBERG, T.-S. RICHARDSON - *Tailored Dynamic Gain Scheduled Control*. Journal of Guidance Control and Dynamics, 29(16):1271–1281, November-December 2006.

[19] I. KAMINER, A.M. PASCOAL, P. KHARGONEKAR, E.E. COLEMAN - *A Velocity Algorithm for the Implementation of Gain Scheduled Controllers*. Automatica, 31(8):1185–1191, August 1995.

[20] I.E. KOSE, F. JABBARI - *Control of LPV Systems with Partly Measured Parameters*. IEEE Transactions on Automatic Control, 44(3):658–663, March 1999.

[21] D.-J. LEITH, W.-E. LEITHAD - *Gain Scheduled & Nonlinear Systems: Dynamic analysis by velocity-based linearization families*. International Journal of Control, 70(2):289–317, May 1998.

[22] MathWorks - *Matlab R2010b, Robust Control Toolbox*. http://www.mathworks.fr/fr/help/ robust/ref/hinfstruct.html, September 2010.

[23] A. PACKARD - *Gain Scheduling via Linear Fractional Transformations*. Systems and Control Letters, 22(2):79–92, February 1994.

[24] G. PAPAGEORGIOU, R.-A. HYDE - *Analyzing the Stability of NDI-Based Flight Controllers with LPV Methods*. In Proceedings of the AIAA GNC Conference, Montreal, Canada, August 2001.

[25] P.-C. PELLANDA, P. APKARIAN, D. ALAZARD - *Gain Scheduling Through Continuation of Observer Based Realizations - Applications to H_∞ and μ Controllers*. In Proceedings of the IEEE CDC, Sidney, Australia, December 2000.

[26] J. REINER, G.-J. BALAS, W. GARRARD - *Flight Control Design Using Robust Dynamic Inversion and Time-Scale Separation*. Automatica, 32(11):1493–1504, November 1996.

[27] C. ROOS, J-M. BIANNIC - *Efficient Computation of a Guaranteed Stability Domain for a high Order Parameter Dependent Plant.* In Proceedings of the ACC, pages 3895–3900, Baltimore, MD, July 2010.

[28] C. ROOS, C. DOLL, J-M. BIANNIC - *Flight Control Laws: Recent Advances in the Evaluation of their Robustness Properties*. AerospaceLab Journal, 4, May 2012. Special issue on Mastering Complexity. http://www.aerospacelab-journal.org/al4/flight-control-laws.

[29] C. ROOS, F. LESCHER, J-M. BIANNIC, C. DOLL, G. FERRERES - *A Set of μ Analysis Based Tools to Evaluate the Robustness Properties of high Dimensional Uncertain Systems.* In IEEE multiconference on systems and control, page 644?649, Denver, Co, July 2011.

[30] W.-J. RUGH - *Analytical Framework for Gain Scheduling*. IEEE Control System Magazine, 11(1):79–84, January 1991.

[31] W.-J. RUGH, J.S. SHAMMA - *Research on Gain Scheduling.* Automatica, 36(10):1401–1425, October 2000.

[32] C.-W. SCHERER - *LPV Control and Full Block Multipliers*. Automatica, 37(3):361–375, March 2001.

[33] C.-W. SCHERER, I.E. KOSE - *Gain Scheduling Synthesis with Dynamic D-Scalings*. In Proceedings of the IEEE CDC, New Orleans, LA, USA, December 2007.

[34] G. SCORLETTI, L. EL GHAOUI - *Improved LMI Conditions for Gain Scheduling and Related Control Problems*. International Journal of Robust and Nonlinear Control, 8(10):845–877, August 1998.

[35] J.-S. SHAMMA, M. ATHANS - *Analysis of Gain Scheduled Control for Nonlinear Plants*. IEEE Transactions on Automatic Control, 35(8):898–907, August 1990.

[36] J.-S. SHAMMA, M. ATHANS - *Guaranteed properties of gain scheduled control for linear parameter-varying plants*. IEEE Transactions on Automatic Control, 27(3):559–564, March 1991.

[37] D.-J. STILWELL, W.-J. RUGH - *Interpolation of observer state feedback controllers for gain scheduling.* IEEE Transactions on Automatic Control, 44(6):1225–1229, June 1999.

[38] D.-J. STILWELL, W.-J. RUGH - *Stability preserving interpolation methods for the synthesis of gain scheduled controllers*. Automatica, 36(5):665–671, May 2000.

[39] D.-J. STILWELL, W.-J. RUGH - *Stability and L2 gain properties of LPV systems*. Automatica, 36(9):1601–1606, September 2002.

[40] P. TEPPA, J. BERNUSSOU, G. GARCIA, H. KHANSAH - *Scheduling of local robust control laws for nonlinear systems.* Studies in informatics and control, 16(2), February 2007.

[41] A. VARGA, A. HANSSON, G. PUYOU, editors - *Optimization based clearance of flight control laws, volume 416 of LNCIS*. Springer-Verlag, Berlin, Heidelberg, 2012.

[42] F. WANG, V. BALAKRISHNAN - *Improved stability analysis and gain-scheduled controller synthesis for parameter-dependent systems*. IEEE Transactions on Automatic Control, 47(5):720– 734, May 2002.

[43] W.-C. REIGELSPERGER, K.-D. HAMMETT, S.-S. BANDA - *Robust control law design for lateral directional modes of an f-16/matv using μ-synthesis and dynamic inversion.* International Journal of Robust and Nonlinear Control, 7(8):777–795, 1997.

[44] F. WU - *A generalized LPV system analysis and control synthesis framework*. International Journal of Control, 74(7):745–759, July 2001.

[45] F. WU, K. DONG - *Gain scheduling control of LFT systems using parameter-dependent Lyapunov functions*. Automatica, 42(1):39–50, January 2006.

[46] F. WU, A. PACKARD, G.-J. BALAS - *LPV control design for pitch-axis missile autopilots*. In Proceedings of the IEEE CDC, New Orleans, LA, USA, December 1995.

AUTHORS

Jean-Marc Biannic is a senior research scientist at Onera, the French Aerospace Lab. He has supervised 6 PhD students and has authored a tutorial book on automatic control, more than 50 papers, several book chapters, teaching documents and Matlab toolboxes. He graduated from Supaero in 1992, received his PhD in control theory in 1996 and his HDR degree(french habilitation to supervise PhD strudents) in 2010. He received in 2011 the "ERE" distinction from Isae (Aeronautics and Space Institute) with which he is recognized as a full professor in PhD committees

Laurent Burlion is currently working as a research scientist in the Systems Control and Flight Dynamics Department at the French Aerospace Lab. He graduated from the French engineering school Ensieta (now Ensta Bretagne) in 2003 and received the same year the M.Sc. degree in Control at the IRRCyN in Nantes. Working on hybrid systems and nonlinear sampled-data controlled systems within the L2S-CNRS, Dr. Burlion received his Ph.D degree from the University of Paris Sud in 2007. His current research interests are focused on the development of nonlinear control design methods with recent applications to visual servoing, flexible modes attenuation or manoeuvre load control.

Henry de Plinval graduated from Ecole Polytechnique, France, in 2006 and received his MSc in Aeronautics and Astronautics in 2006 from MIT (USA) and his PhD in automatic control from the University of Toulouse in 2014. Since 2008, he is with the Systems Control and Flight Dynamics Department within Onera - the French Aerospace Lab. His areas of interest include guidance, navigation and control, mostly for UAVs, with a particular focus on visual servoing for VTOL UAVs.

Towards Modular and Certified Avionics for UAV

F. Boniol, V. Wiels
(Onera)

E-mail: frederic.boniol@onera.fr

DOI : 10.12762/2014.AL08-02

This paper proposes a review of the current state and forthcoming evolutions for UAV avionics architecture and software. It provides an outlook of the specific technical issues arising in the design of embedded systems for UAV.

Introduction

The Unmanned Aerial Vehicle (UAV) industry has been rapidly growing over the last decade. New UAV are being developed for military applications and also for civil usage. There is a strong correlation between the mission of a UAV and the avionics necessary to implement it. For this reason, the design, development and verification of UAV avionics, including hardware and software architecture, have been the subject of considerable research [23, 21, 16, 22].

UAV avionics, like those of traditional aircraft, are in charge of implementing flight control and flight navigation. However, they should also ensure a desired level of autonomy and the control of the payload (if any).

For flight navigation, a UAV includes embedded means for estimating, at any time and anywhere, its position, speed and acceleration. This requires navigation sensors (such a GPS) and robust estimation algorithms. For flight control, the UAV must generate the steering commands and subsequent control surface deflections to stabilize the vehicle and to adequately follow the flight plan. This again requires robust control algorithms. These computations are relatively simple compared to the flight planning algorithms. They however require the use of accurate real-time processors and operating systems.

In addition to the flight control and navigation part, UAV require specific autonomy means. The autonomy requirement is the main difference between UAV and manned aircraft. Autonomy is the ability to operate without direct control from a ground operator. Complex or faraway missions without ground infrastructure (for instance data link means and ground stations) would necessitate making the UAV increasingly autonomous. Autonomy requires specific sensors, such as optical devices, and complex software, such as image processing software and intelligent flight planning. Ideally, the UAV must have the capability to plan and re-plan its own flight plan. This results in the requirement for an on-board high-performance computing architecture where flight-planning algorithms can be run. These algorithms require knowledge of the UAV's surroundings, including other traffic, weather, obstacles, fuel usage, flight time, etc. Furthermore, in the event of failure, the UAV must have the capability to reconfigure itself and re-plan its trajectory or its mission. These autonomy requirements result in complex software, which requires high performance computing means without compromising safety: efficient techniques are necessary to verify and validate software.

The aim of this article is to discuss new challenges for future UAV avionics architectures and software : current state and forthcoming evolutions of UAV avionics, use of IMA (Integrated Modular Avionics) for UAV and certification issues.

Current state and forthcoming changes

Current state

The main challenge encountered by UAV avionics is to safely operate on-board two types of computation: flight control/navigation and flight planning/re-planning, including the reconfiguration of the avionics itself in case of mission re-planning.

In order to respond to this challenge, the first generation of UAV avionics architectures were divided into three loosely coupled physical parts. The first one is dedicated to navigation and flight control; the second one offers sensors, hardware and software components ensuring the desired level of autonomy; while the third part controls the payload of the UAV. The second and third parts are generally specific to the operational role that the UAV is supposed to carry out. In most cases, each part is implemented by a monolithic dedicated platform composed of the simplest possible processor with its own resources (memory and communication bus) (figure 1). UAV developed in the 90 s and 2000 s were based on this principle (see for instance appendix A of [27], and the Piccolo architecture in [33]).

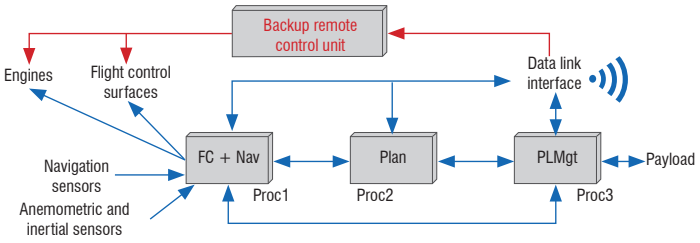


Figure 1 – Typical UAV 1st-generation architecture

Such a physical segregation between the three parts ensures that they operate (nearly) independently from each other. More precisely, the autonomy sensors and algorithms do not interfere with the control loop and vice-versa. Likewise, the payload software does not affect the rest of the avionics. Thus, a failure in the payload part should not affect the safety of the vehicle. Thus, each part can be designed, developed, dimensioned and certified separately, without considering interferences coming from the other parts. Finally, in the event of failure in one of the first two parts, a human operator can directly control the vehicle from the ground by means of a set of data-links and specific components. The vehicle becomes in that case a remote-controlled plane.

However, these first architectures based on the principle of separation of concerns have many limitations.

Forthcoming changes

Firstly, safety is guaranteed in the final degraded mode (i.e., the mode in which the automatic flight control part is lost) by the ability to pilot the vehicle from the ground. As stated above, it assumes (1) safe onboard mechanisms to commute from the automatic flight mode to the remote-controlled flight, (2) a data-link between the vehicle and the group operator, which guarantees that orders and data are transmitted in real-time (in less than 10 or 100 milliseconds for flight control orders) and, (3) a ground infrastructure able to present to the human operator the complete situation of the vehicle (position, speed, attitude, obstacles in front of the vehicle, etc.). Such requirements are not consistent with complex and faraway missions, or with missions in a hostile environment. In that case, contrarily to manned aircraft, the UAV must ensure its own safety without waiting for backup orders sent by a human pilot. However, first generation UAV avionics architectures do not offer the appropriate safety level. This is their first limitation.

Secondly, the continual development of UAV applications results in an ever-increasing demand on embedded algorithms. On board computational resources must meet this demand, while at the same time providing robustness, reliability and a small footprint, both in physical size, mass and power consumption.

Thirdly, new applications may necessitate the integration in a more coupled way of the three parts of the UAV avionics. In particular, the payload management may depend on flight data, such as position, speed, attitude, etc. Conversely, navigation and flight control may depend on the state of the payload. This requires an appropriate mechanism providing navigation data to the payload and conversely, in such a manner that the payload activity cannot interfere with flight control and planning. Failure of the payload must not compromise the safety of the UAV (for example, denying access to the on-board data-bus by saturating it with payload messages). First generation UAV avionics architectures do not offer such a mechanism.

A first solution could be to continue with the segregation principle (each part has its own sub-architecture), while increasing the number of computing resources. It should lead to the duplication of several components (for instance, the flight data calculation for the payload). However, this first architecture principle reaches its natural limit when the weight and volume of the dedicated sub-architectures encounter the envelope restrictions of the UAV. This issue becomes central in the case of small UAV able to carry only a few kilograms (generally less than 10 kg) including payload and avionics. Another drawback becomes obvious: the huge number of different resources has significantly increased the maintenance costs in terms of component spare part provisioning and handling.

Another approach, called Integrated Modular Avionics (IMA) [2, 3] has been suggested to address this issue for manned aircraft, such as Airbus A380 / A350 and Boeing B787.

Towards modular integrated avionics for UAV

Modular Integrated Avionics

Resource sharing and robust partitioning are the central ideas of the IMA concept. They are based on two principles: partitioning principles in processing modules and partitioning principles for communications between functions.

Processing module partitioning

As has already been explained, an UAV avionics architecture implements several software functions (flight control, navigation, planning and payload management), each of them possibly divided into sub-functions. Initially running on different processors, the first IMA idea is to place these functions on processing modules partitioned with respect to space (resource partitioning) and time (temporal partitioning).

- *Resource partitioning.* A processing module is divided into partitions. Each partition is seen as a virtual processing module. It is allocated a set of private spatial resources (memory, non-volatile memory, I/O resources, etc.) in a static manner. Low-level mechanisms (at the operating system level) provide protection for partition data against any modification from the other partitions. They monitor function activity with reference to allowed resources, which are statically allocated through configuration tables.
- *Temporal partitioning.* Each function is allocated a partition. The scheduling of partitions on each module is defined off-line by a periodic sequence of slots, statically organized in a time-frame. Each partition is allocated a time slot for execution. At the end of this time slot, the partition is suspended and execution is given to the next partition (running another function). Thus, each function is periodically executed at fixed times.

Thanks to partitioning mechanisms, functions become independent. A faulty function can be isolated without affecting functions placed on the same module.

Communication resource partitioning

Initially routed onto different physical links, the second IMA idea is to place communications between functions on shared communication networks. The network is divided into Virtual Links (VL). Each VL is dedicated to the traffic coming from a single function. It is characterized

by a bounded bandwidth. Similarly to processing modules, a low-level mechanism (at the network level) guarantees that no function can go beyond its contract, that is, produce more communication than the permitted bandwidth. Such a mechanism can be implemented by a traffic shaper, which separates two successive emissions on the VL by at least a fixed time interval called Bandwidth Allocation Gap (BAG). This principle has been implemented in the Avionics Full Duplex Ethernet (AFDX) architecture embedded in the Airbus A380 and A350 [3].

A typical IMA platform is described in figure 2. Its hardware architecture consists of 3 generic computing processing modules (called CPM) that are connected to a communication network. The network is composed of two identical redundant parts (Part A in blue, and Part B in red). CPM1 and 2 are connected to Switches 1 (A and B), while CPM3 is connected to Switches 2 (A and B). Flight control and navigation sensors and actuators are reached through a redundant gateway (Gtw1) connected to Switches 1. Similarly, the payload and the data link interface are reached through a second redundant gateway (Gtw2). As shown in the figure, the critical functions Flight Control (FC) and Navigation (Nav) are triplicated and the Planning Function (Plan) is duplicated, while the Payload Management Function (PLMgt) is implemented by a single occurrence (*i.e.*, without any redundancy). Each CPM is divided into four partitions (*e.g.*, FC1 is hosted in the first partition of CPM1). On each module, partitioning and scheduling are ensured by a partition manager, while bandwidth communication from each function is controlled by a VLs manager. Figure 3 shows the time-triggered scheduling of the four partitions hosted by CPM1. As has already been explained, this scheduling is organized as a sequence of time slots. It is composed of two minor frames (MiF), the duration of which is 10 ms. FC1 runs in the first time slot of each MiF. The duration of this time slot is 2ms. FC1 is then supposed to execute every 10 ms within an execution time of less than 2 ms. Nav1 runs only in the first MiF, while PLMgt runs in the second MiF. The aim of the partition manager is to unroll this sequence and monitor each partition. For instance, if PLMgt tries to continue after the end of its time slot, the partition manager stops it and starts the next partition (FC1). Hence, a software failure in PLMgt does not affect FC1. Note that in the CPM1 scheduling, a spare time slot is reserved for hosting potential new functions without affecting other functions.

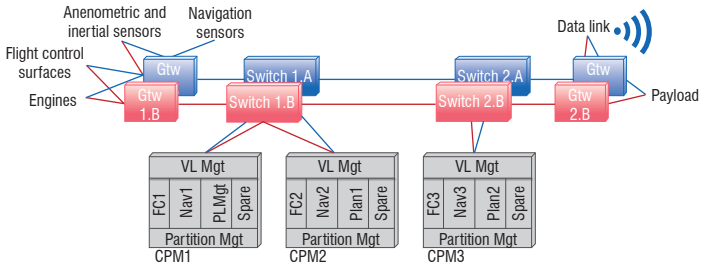


Figure 2 – Example of UAS IMA architecture

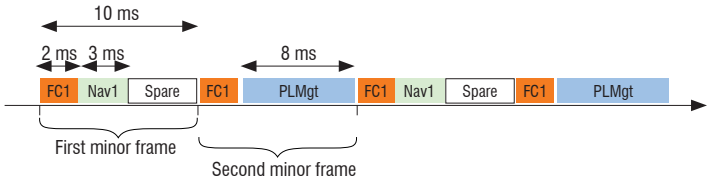


Figure 3 – Temporal scheduling of CPM1

Benefits and effects for UAV avionics

The benefits of such a new architecture are mainly: safety level improvement, as well as weight and power consumption reduction. Let us again consider the architectures given in figure 1 and figure 2. These two architectures implement the same functions FC, Nav, Plan and PLMgt. They are both composed of 3 processors. However, the first architecture is not fault tolerant. For instance, loss of Processor 1 leads to total loss of the vehicle. Conversely, the IMA architecture (figure 2) is fault tolerant with the same number of processors. For instance, despite of the loss of CPM1, the flight control and navigation functions still run properly on CPM2 and CPM3. Only the payload management is lost. It is obvious that the total loss of the vehicle is consecutive to at least two failures: for instance, loss of Switches 1.A and 1.B or loss of Gateways 1.A and 1.B, etc. In that sense, with (nearly) the same number of resources, the IMA architecture is safer than the first one.

Globally, IMA results in a reduction of the required physical resources. Reduced physical resources translate into global weight and power savings for the UAV. The same trend has been observed in aircraft architectures: for instance, the number of processing units in the A380 is half that of previous generations. Reductions in operating costs are expected to be significant, with the decrease in the number of computers and cables (for power supply or communication), contributing to a reduction of vehicle weight leading to better fuel consumption efficiency and then to a greater autonomy.

Past and recent experiments on applying IMA to UAV

Considering these expected benefits, recent research has been conducted on the integration of IMA architecture into modern UAV. A preliminary work has been proposed by Elston *et al.* [17]. They are developing a distributed modular architecture concept for small UAV (about 10 kg). This architecture is composed of a set of computing modules communicating through a CAN bus. Similarly, Ellen *et al.* investigated in [16] an architecture for the QUT research UAV, still based on a cluster of small dedicated processors communicating through CAN buses. They show that the performances of this architecture, in terms of power consumption, size and weight, are better than those for the legacy architecture (based on a centralized PC104 computer). However, contrary to the full IMA concept, computing modules in these proposed architectures still own their private sensors and actuators and host only one function. There is no partitioning mechanism.

Following this direction, Lopez *et al.* investigated in [25] a middleware-based architecture suitable to operate as a flexible payload and mission controller in a UAV. The architecture is composed of low-cost computing devices connected by a network. The functionality is divided into reusable services distributed over a number of nodes, with a middleware partitioning their lifecycle and communication. However, the middleware does not take into account real-time issues. Thus, flight control and navigation cannot run on this platform and still require a dedicated real-time architecture.

In order to respond to the real-time issue, [29] proposes an architecture platform based on a Time-Triggered network. Functions, including flight control and navigation, run on dedicated PC/104 computers and communicate in a deterministic way through the network. Thanks to the time-triggered protocol, the network guarantees fixed time slots for each function. This solution has been implemented on large UAV, such as the R-MAX Helicopter (about 10 kg).

More recently, [22] has developed a complete IMA solution, based on an ARINC 653 processing module, for a small quad-rotor helicopter. Due to the UAV size (70 cm in diameter and 1 kg payload), the avionics only include a single embedded processor hosting the flight control, navigation and planning functions. The processing unit is partitioned in a deterministic way according to the IMA principle. To our knowledge, this experiment is the first attempt to apply the IMA concept, here reduced to the processing module part, to small UAV.

All of these experiments clearly show the trend to embed an IMA execution platform for flight control and the navigation function, as well as for planning and payload management, in both small and larger UAVs.

Certification issues

Given that the use of drones for different kinds of mission is spreading, and will continue to do so, their safety and security will become a crucial issue. For aircraft, safety and security are assessed using certification standards. The ARP 4754 [4] is the standard for systems, DO-178 [14] is the standard for software, while Common Criteria (ISO 15408) [12] handles security aspects.

These standards are bound to also be applied for UAV systems and software. Application of ISO 15408 will enforce security requirements and help to prevent the hacking of UAV. However, it will only be effective if safety-critical requirements are also taken into account. Ensuring the safety of embedded software is paramount, since there is no human pilot onboard. In this section, we consider the challenges at stake for the certification of this software. We distinguish the domains where aircraft solutions can be adapted to UAV without too much trouble and UAV specific certification issues.

[28] gives a broader overview of the challenges and a roadmap for the certification of Remotely-Piloted Aircraft Systems. We focus on software aspects, but also consider autonomous systems (even though their operational use is further away). [28] also tackles insertion into the airspace; we do not consider this certification issue.

Issues similar to those for aircraft

Flight control and navigation systems are certified for aircraft, using classical means or more recently innovative verification techniques. We will not provide an exhaustive overview of existing work here, but rather only list the various aspects that should be considered together with a few references, mainly of Onera work in this domain. We focus on aspects related to avionics; safety and security assessment at the aircraft and system levels are also essential, but will not be discussed here (see [7,5]).

Real-time analysis

Certification objectives regarding real time are scattered around in various certification standards (IMA, software) but they are essential for the correctness of software and systems. Regarding real-time behaviors, the first requirement is to guarantee that each function located in an IMA partition terminates properly before the end of the partition. For instance, let us consider the flight control function FC1 hosted by CPM1 (figure 2). FC1 runs in a partition of which the duration is 2 ms (figure 3). Thus, it must be shown that all sub-functions involved in FC1 are scheduled in such a way that they will all terminate before 2 ms have elapsed. Several techniques and tools have been developed to analyze worst-case execution time [1, 26], and worst-case response time [30, 11] for IMA software, or generate a correct scheduling from different constraints within a partition [30].

The second certification requirement deals with worst-case traversal time through a communication network. Let us again consider the example in figure 2. FC1 periodically sends orders to actuators through the network. Note that the payload also sends and receives data through the same network. It could then happen that, if a failure occurs somewhere in the payload, it may begin to send a huge amount of data to PLMgt, overloading the communication network, leading to delays in the flight control orders. Such a scenario may lead to a catastrophic situation, despite the initial single failure being of minor importance. An interesting benefit of the IMA principle is that, if functions are statically allocated in modules and partitions, and if the network hosts mechanisms enforcing functions to respect their communication contract (*e.g.*, traffic shapers), then it is possible to mathematically prove that the end-to-end delay of any message is bounded and it is possible to evaluate an over-approximation of this bound. This proof is based on the network calculus theory [8, 9]. Network calculus has been used for certification of the A380 and A350 avionics network. It contributes an adequate mathematical technique for UAV avionics network certification as well.

Software verification

DO-178/ED-12 [14] does not prescribe a specific development process for software, but rather identifies important activities and design considerations throughout a development process and defines objectives for each of these. DO-178 [14] distinguishes development processes from “integral” processes that are meant to ensure correctness, control and confidence in the software life cycle processes and their outputs. The verification process is part of the integral processes, along with configuration management and quality assurance. Version C of this standard, which was published in 2011, includes technical supplements to take into account and facilitate the appropriate use of new software engineering techniques. DO-333/ED-216 [15] is the formal method supplement. Formal methods can be applied to many of the development and verification activities required for software. The supplement proposes guidance for the use of formal methods. It describes the activities that are needed when using formal methods, new or modified objectives and the evidence needed for meeting those objectives.

Formal verification techniques have already been used for the certification of aircraft avionics software [32] and a lot of work is underway in this field [34]. Specific work on the verification of stability and safety properties of flight control software could be of special interest for UAV [10].

UAV specific issues

In this section, we point out the specific certification issues arising for UAV in the various domains considered previously.

In-flight reconfiguration

As explained above, IMA architectures are based on a strict principle: static and fixed allocations. However, it could be interesting, in the event of a hardware failure or in the event of loss of the communication link, for example, to be able to reconfigure the system, which means reallocating

functions to safe processors. Let us consider the example in figure 2 and let us imagine that CPM1 fails. Then PLMgt is lost. It could be interesting to reallocate it in the spare partition of CPM2. Such a mechanism could allow a reduction of the number of on-board processors, thereby saving weight, particularly for small UAV. Unfortunately, current IMA architectures do not allow in-flight reconfiguration. Recent research work conducted by Onera with Thales and Airbus has explored the reconfiguration issue for aircraft IMA architectures in the European SCARLETT project (http://www.scarlettproject.eu/) [6]. The solution is limited to on-the-ground reconfigurations, which seems to be enough for aircraft architectures. However, small UAV can only include a small number of embedded resources. Safe in-flight reconfiguration remains a strong challenge for UAV architectures.

Software verification

A significant difference between aircraft and UAS resides in mission management software. As stated by [23], mission management software may be quite complex, in order to be able to respond to various situations; it may include various concurrent tasks, etc. Moreover, the development of mission management software typically does not follow stringent processes, such as those used for flight critical software; the verification of this software is currently mostly done through simulations and flight tests. The proliferation of UAV will call for the use of more rigorous means of verification for mission management software.

Requirements for this kind of software will first have to be identified and formalized. It may not be an easy task, due to the very nature of the software. In order to ensure the autonomy of the UAV, mission management software is designed to be “intelligent”, to be able to respond to many different situations by analyzing available information. An exhaustive enumeration of all possible situations might be a tedious and difficult task. Once the requirements have been expressed, formal verification techniques will also have to be adapted, or extended, to handle the specificities of mission software. A family of techniques that could be useful for the verification of mission software is runtime verification. The principle is to monitor the software with respect to a given set of formalized properties [18, 20].

Conclusion

In this paper, we have described the current state of avionics for UAV, identified challenges in this domain and proposed directions for future work. In conclusion, we would also like to mention an Onera initiative, called FORC3ES (Formal engineering for certified control-command embedded systems). This initiative is aimed at defining a set of techniques and tools for the formal development and verification of control-command systems. The framework is experimented with on a UAV and its associated Iron Bird (an Iron Bird is a system test bench; it includes the same sensors, actuators and avionics as the real aircraft; see pictures in figure 4). The first part of this project is dedicated to flight control software development and verification, but in the long run we also intend to study the verification of mission management software and to experiment with new concepts of IMA architectures for UAV ■

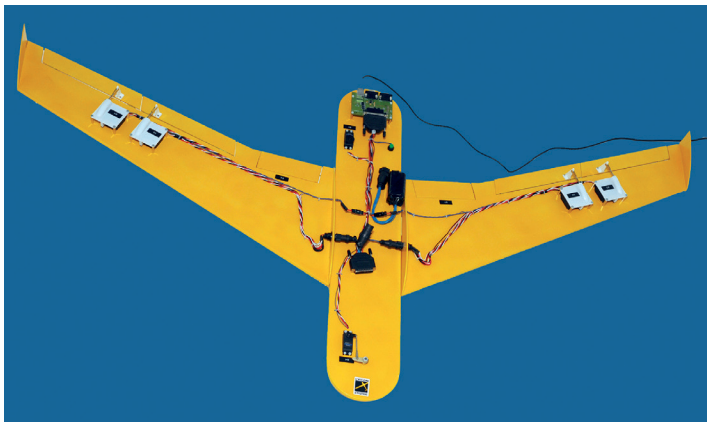


Figure 4 – UAV (left) and associated Iron Bird (right)

References

[1] *aiT Worst-Case Execution Time Analyzers*. http://www.absint.com/ait.
[2] Aeronautical Radio Inc – *ARINC 653: Avionics Application Software Standard Interface*. 1997.
[3] Aeronautical Radio Inc – *ARINC 664: Aircraft Data Network, Part 1: Systems Concepts and Overview*. 2002.
[4] ARP64754A – *Guidelines for Development of Civil Aircraft and Systems*. 2010.
[5] P. BIEBER, J.-P. BLANQUART, G. DESCARGUES, M. DULUCQ, Y. FOURASTIER, E. HAZANE, M. JULIEN, L. LÉONARDON and G. SAROUILLE – *Safety and Security Assurance in Aerospace Embedded Systems*. ERTSS, Toulouse, February 2012.
[6] P. BIEBER, E. NOULARD, C. PAGETTI, T. PLANCHE, F. VIALARD – *Preliminary Design of Future Reconfigurable IMA Platforms.* SIGBED Review 6(3): 7, 2009.
[7] P. BIEBER and C. SEGUIN – *Safety Analysis of the Embedded Systems with the AltaRica Approach*. Industrial Use of Formal Methods: Formal Verification, Wiley, 2012.
[8] J.-Y. LE BOUDEC and P. THIRAN – *Network Calculus: a Theory of Deterministic Queuing Systems for the Internet*. LNCS, Vol. 2050, Springer, 2001.
[9] M. BOYER, N. NAVET, X. OLIVE and E. THIERRY – *The Pegase Project: Precise and Scalable Temporal Analysis for Aerospace Communication Systems with Network Calculus*. LNCS, Vol. 6415, pp. 122–136, Springer, 2010.
[10] A. CHAMPION, R. DELMAS, M. DIERKES, P.-L. GAROCHE, R. JOBREDEAUX and P. ROUX – *Formal Methods for the Analysis of Critical Control Systems Models: Combining Non-linear and Linear Analyses*. FMICS, pp. 1-16, 2013.
[11] Cheddar, Université de Brest, http://beru.univ-brest.fr/~singhoff/cheddar.
[12] Common Criteria, http://www.commoncriteriaportal.org.
[13] R.P.G. COLLINSON – *Introduction to Avionics Systems.* http://books.google.fr/books?id=aU8SMhzrScgC, Springer, 2011.
[14] RTCA/DO-178C, EUROCAE/ED-12C – *Software Considerations in Airborne Systems and Equipment Certification.* 2011.
[15] RTCA/DO-333, EUROCAE/ED-216 – *Formal Methods Supplement to DO-178C and DO-278A.* 2011.
[16] R. ELLEN, P. ROBERTS and D. GREER – *An investigation into the Next Generation Avionics Architecture for the QUT UAV Project.* Goh, Roland & Ward, Nick (Eds.) Smart Systems 2005 Postgraduate Research Conference, Brisbane,15 December, 2005.
[17] J. ELSTON, B. ARGROW and E. FREW – *A Distributed Avionics Package for Small UAVs.* AIAA Conference, Arlington, Viginia. 2005.
[18] Y. FALCONE, K. HAVELUND and G. REGER – *A Tutorial on Runtime Verification. Book chapter four: Summer School Marktoberdorf 2012 - Engineering Dependable Software Systems. July 31 to August 12, 2012.* IOS Press book, NATO Science for Peace and Security Series - D: Information and Communication Security, Vol. 34, 2013.
[19] V. GAVRILETS – *Avionics Systems Development for Small Unmanned Aircraft.* Master's Thesis, MSc in Aeronautics and Astronautics, MIT, 1998.
[20] A. GROCE, K. HAVELUND, G. HOLZMANN, R. JOSHI and R-G. XU – *Establishing Flight Software Reliability: Testing, Model Checking, Constraint-Solving, and Monitoring.* Annals of Mathematics and Artificial Intelligence, March 2014.
[21] G.Y. IMMANUEL and N. JOHNSON – *New Architectures for UAV Flight Control Avionics.* 21st Digital Avionics Systems Conference, 2002.
[22] H.C. JO, S. HAN, S.H. LEE and H.W. JIN – *Implementing Control and Mission Software of UAV by Exploiting Open Source Software-Based Arinc 653.* Digital Avionics Systems Conference (DASC), IEEE/AIAA 31st, pp. 8B2–1–8B2–9, October 2012.
[23] T. JOHNSON, R. KONECK and S.F. BUSH – *Improving UAV Mission Success Rate through Software Enabled Control Design.* IEEE Aerospace Conference, 2000.
[24] M. LAUER, J. ERMONT, F. BONIOL and C. PAGETTI – *Worst Case Temporal Consistency in Integrated Modular Avionics System.* HASE, T. M. KHOSHGOFTAAR Ed., IEEE Computer Society, pp. 212–219, 2011.
[25] J. LÓPEZ, P. ROYO, E. PASTOR, C. BARRADO and E. SANTAMARIA – *A Middleware Architecture for Unmanned Aircraft Avionics.* Middleware'07, Newport Beach, California, November 26-30 2007.
[26] OTAWA, IRIT Université de Toulouse, http://www.otawa.fr.
[27] S. PARK – *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles.* PhD Thesis, MIT, 2004.
[28] *Roadmap for the Integration of Civil Remotely-Piloted Aircraft Systems into the European Aviation System.* Final report from the European RPAS steering group, June 2013.
[29] A. SAMUEL, N. BROWN, R. COLGREN and G. KELLY – *Subsystem Design and Integration of A Robust Modular Avionics Suite for UAV Systems Using the Time Triggered Protocol (TTP).* SAE Aerospace Technology Conference, 2007.
[30] SchedMCore, Onera, http://sites.onera.fr/schedmcore.
[31] A. SHAHSAVAR – *Defining a Modular, High Speed and Robust Avionic Architecture for UAV's.* Master's Thesis Programmes in Engineering Space Engineering, 2008:174, Lulea University of Technology. 2008.
[32] J. SOUYRIS, V. WIELS, D. DELMAS and H. DELSENY – *Formal Verification of Avionics Software Products*. Formal Methods, 2009.
[33] B. VAGLIENTI and R. HOAG – *A Highly Integrated UAV Avionics System.* Technical Report, A cloud cap Technology, 2003.
[34] V. WIELS, R. DELMAS, D. DOOSE, P.-L. GAROCHE, J. CAZIN and G. DURRIEU – *Formal Verification of Critical Aerospace Software.* AerospaceLab, Issue 4, 2012.

Acronyms

AFDX	(Avionics Full Duplex Ethernet)
BAG	(Bandwidth Allocation Gap)
CPM	(Computing Processing Module)
FC	(Flight Control)
IMA	(Integrated Modular Avionics)
MiF	(Minor Frame)
PL Mgt	(Payload Management)
UAS	(Unnamed Aerial System)
UAV	(Unnamed Aerial Vehicle)
VL	(Virtual Link)

AUTHORS

Frédéric Boniol graduated from a French High School for Engineers in Aerospace Systems (Suapero) in 1987. He holds a PhD in computer science from University of Toulouse (1997). He has a research position at Onera/DTIM. His research interests include modeling languages and performance analysis methods for embedded real-time systems.

Virginie Wiels is research scientist at Onera/DTIM since 1998. She is working on formal verification of embedded systems and software. Before joining Onera, she was research associate at NASA/WVU IV&V Facility in Fairmont (USA). She holds a PhD in computer science from University of Toulouse (1997).

I. Fantoni, G. Sanahuja
(Université de Technologie
de Compiegne)

E-mail: isabelle.fantoni@hds.utc.fr

DOI : 10.12762/2014.AL08-03

Optic Flow-Based Control and Navigation of Mini Aerial Vehicles

This paper concerns recent work on the application of optic flow for control and navigation of small unmanned aerial vehicles. Bio-inspired strategies, such as the use of optic flow, have always motivated researchers in the control community. Recent methodologies for active and passive navigation of aerial vehicles using optic flow, such as obstacle detection or terrain following, are presented. Applications to path following that have been achieved at Heudiasyc Laboratory (CNRS-UTC) in Compiègne are described, in order to illustrate the strength of the concept.

Introduction

Nowadays, we see an increased interest in mini and micro-UAVs. This growing activity in regard to small flying machines is motivated by recent advances in miniaturized electronics, but also by the increasing demand for military and civilian applications. Autonomous functions become more and more essential for UAVs and vision-based approaches offer undeniable assets for the autonomous navigation of such systems. In this paper, we will show the interest and the contribution of vision in the navigation of UAVs and especially the use of optic flow for control and navigation.

Ideally, in order to improve its autonomy, a vehicle has to perceive, model and interpret its environment to adapt its actions. In the same way as a man relies on his senses, an autonomous vehicle uses data provided by sensors. Among the sensory resources, we find proprioceptive sensors, which perform their actions in relation to what they locally perceive of the robot movement, or exteroceptive sensors, which are based on measurements taken in relation to their overall environment. The eye is indeed a perfect illustration of an exteroceptive sensor. Physiologically, vision is the ability that seems to give to living beings the most information about their environment. Motivated by work on insects, principles inspired on their behavior were developed in [5], [16] and have allowed natural ideas to solve localization issues, navigation, obstacle detection, etc., to be brought to robotics. Insects move and act according to exterior movements caused by their own movement. This visual characterization, called optic flow, is increasingly used for guidance assistance in robotics. We are particularly interested in the contribution of optic flow for UAVs navigation, which is the purpose of this article.

Optic flow

Definition

The optic flow is defined as the apparent motion of the image intensities (or brightness patterns) caused by the 2D projection onto a retina of the relative 3D motion of scene points. Ideally, it corresponds to the approximation of the velocity field in the image plane (or visual displacement field of image points) obtained by the 2D projection of the speed of moving objects in 3D space (geometric concept). This movement helps to explain variations in displacement of a moving picture. The optic flow is formulated as a vector field over two dimensions, where the domain is either the two-dimensional visual field or a focal plane and the range is a vector describing how fast and in which direction the texture is moving. The optic flow is created by the translational and the rotational movements of a point $P\ (X, Y, Z)$ in the camera reference frame. Considering the projection of this point in the image plane $p(x, y)$ (figure 1), we can express it according to point P and the focal length f.

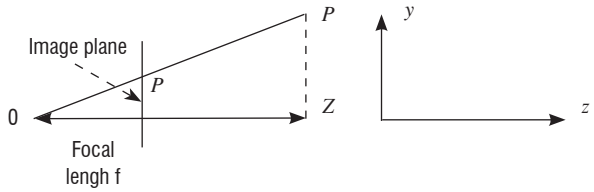


Figure 1 - Scheme of projective geometry for optic flow computing

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \quad (1)$$

The velocity of p is found through differentiation of (1). Thus, the optic flow computed at an image point (x, y) can be expressed as a summation of translational and rotational parts, as follows:

$$\begin{bmatrix} OF_x \\ OF_y \end{bmatrix} = T_{OF} + R_{OF} \tag{2}$$

with the translational part

$$T_{OF} = \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{3}$$

and the rotational part

$$R_{OF} = \begin{bmatrix} \frac{xy}{f} & -(f+\frac{x^2}{f}) & y \\ (f+\frac{y^2}{f}) & -\frac{xy}{f} & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4}$$

where OF_j is the optic flow component in the coordinate j of the point p, V_k and ω_k are the translation velocities and rotation rates, respectively, of the point P at the coordinate k. This expression represents the optic flow defined on the image plane. Note that the optic flow can be defined on projection surfaces other than image planes, such as spheres, which are often used for their passivity properties [18]. Furthermore, relationships exist between different expressions of optic flow, so there is no information loss when the projection surface is changed.

Computation methods

Standard techniques for calculating optic flow can be classified into four groups: differential or gradient methods based on intensity [22], [24], [26], [35], correlation or block matching methods [1], [32], methods based on energy [19] and those based on phase [15], [37]. Block matching techniques present very good accuracy and performance against aperture problems (figure 2) and large displacements, but they are computationally expensive, less accurate in the presence of deformation and displacements of less than one pixel are not detectable. Energy-based and phase-based methods are also very expensive in terms of processing time and also very complex to implement. On the other hand, differential methods suffer from sensitivity issues due to changing lighting and noise due to the calculation of derivatives. They are nevertheless precise, with less computing time necessary compared with all cited techniques. Differential or gradient-based methods are thus well known, are the most implemented techniques because of their properties and are widely used in the literature for optic flow computations.

Despite their differences, many of the gradient-based techniques can be conceptually viewed in terms of a set of three processing stages:
1) pre-filtering or smoothing;
2) computation of spatiotemporal derivatives or local correlation surfaces
3) integration of these measurements to produce a two-dimensional flow field.

Differential methods are based on calculations of the derived spatiotemporal image intensity over a region of the image. The conservation of intensity is reflected by the following equation (see [4])

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \tag{5}$$

where $I(x, y, t)$ is the intensity of the image, t is the time and (dx, dy) is the image displacement. From equation (5) it is possible to obtain the intensity conservation constraint, also called optic flow or gradient constraint equation [4], expressed by

$$\nabla I(x,y)(v_x, v_y) + I_t = 0 \tag{6}$$

where $\nabla I(x, y)$ is the spatial gradient, (v_x, v_y) is the image velocity and I_t is the temporal derivative of the image intensity. Giving this constraint, B. Lucas and T. Kanade [24] have constructed a technique for estimating the optic flow, based on a weighted least squares minimization of the intensity conservation constraint in each small spatial neighborhood S.

$$\min \sum_{(x,y)\in S} W^2(x,y)[\nabla I(x,y,t)\vec{v} + I_t(x,y,t)]^2 \tag{7}$$

where W is a window that gives more importance to the constraints near the center of the chosen neighborhood. The solution of equation (7) is given by

$$\vec{v} = [A^T W^2 A]^{-1} A^T W^2 b \tag{8}$$

where

$$A = [\nabla I(x_1, y_1), \ldots, \nabla I(x_n, y_n)]^T \tag{9}$$

$$W = diag[W(x_1, y_1), \ldots, W(x_n, y_n)] \tag{10}$$

$$b = -[I_t(x_1, y_1), \ldots, I_t(x_n, y_n)] \tag{11}$$

One advantage of the Lucas-Kanade algorithm is that it provides a measure of the estimation error, given that the matrix $[A^T W^2 A]^{-1}$ is consistent with a covariance matrix. Unreliable estimates can be identified using the inverse eigenvalues of this matrix. However, this method is not suitable for displacements larger than one pixel per frame, which causes estimation to fail. Nevertheless, an extension of this method has been proposed, resulting in a solution that computes optic flow via a hierarchical coarse-to-fine process, called pyramidal representation. This coarse-to-fine method is based on the Lucas-Kanade algorithm, which is complemented by a pseudo-iterative scheme that allows optic flow to be computed by propagating the flow in lower resolutions to the larger resolutions [9]. Despite all of the advantages of the Lucas-Kanade algorithm, it is important to remember that optic flow estimated by this method represents the apparent movement of objects in the scene, which may not correspond to real object movement. Indeed, a number of assumptions are made, in order to be able to compute optic flow. These hypotheses can be summarized as follows:

- Lighting must remain constant over time. This hypothesis is the basis of optic flow computation. Without it, equation (5) could not be written. If the lighting in the scene changes from one instant to another, objects may seem to move when in reality they are static over time. Indeed, shadows are different depending on the position of the light source and while the object is stationary the camera can detect shadow movement;
- Rich textured features are another important condition for accurate estimation of optic flow. For an effective differential approach, such as the Lucas-Kanade algorithm, there must be contrasts in the scene and thus contrasting textures must be chosen. Furthermore, in order to increase image contrasts, low pass filtering can be applied. There are two ways of achieving this filtering; one is to apply

a Gaussian filter to create a sort of blurring of the images. On the other hand, a simple way to apply such a filter is to de-focus the lens (the image becomes blurred). Nevertheless, this is prejudicial to the computation of the optic flow, since it requires a good texture but must not be too discontinuous;

• Finally, the well-known aperture problem must be taken into account. In general, a differential method alone does not solve what is called the aperture problem (figure 2). However, this entails increased computing time. One solution to this problem is to eliminate inconsistent points by means of the covariance matrix previously defined.

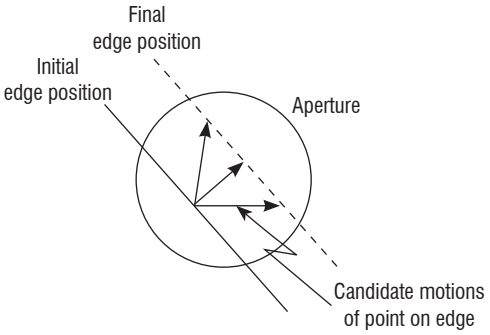


Figure 2 - Aperture problem

Navigation using optic flow

Active navigation

In order to ensure UAV autonomy with respect to unexpected changes in its environment, several approaches using optic flow can be found in the literature. Indeed, the main techniques developed have sought to provide the necessary information for the vehicle, enabling it to react against obstacles. Among reactive navigation modes using optic flow algorithms, we can cite: the effect of centering (Urban Canyon), wall monitoring, frontal obstacle detection and altitude centering for navigation within buildings.

Centering in a corridor

L. Muratet et al. [25] developed a centering strategy by equaling pixel motion calculated on the right and left of a helicopter. Optic flow rotational components were compensated using inertial unit (IMU) data. Indeed, at each point having an estimated optic flow, the rotational components were subtracted. These components were predicted using the camera projective model and the rotational speeds were calculated by the inertial measurement unit.

$$\overrightarrow{OF_i}^{rot} = f\begin{bmatrix} x_i y_i \omega_y - (1+x_i^2)\omega_z + y_i \omega_x \\ (1+y_i^2)\omega_y - x_i y_i \omega_z - x_i \omega_x \end{bmatrix} \quad (12)$$

where $\overrightarrow{OF_i}^{rot}$ is the optic flow rotational component, ω_k are rotational velocities around the axes and x_i and y_i are coordinates in the image plane. The optic flow resulting from the difference between the estimation and the prediction of the rotational component was considered as the optic flow due only to translational movement. The difference of the translational optic flows on either side was used to construct a proportional controller that provides the centering effect. Note that these methods that compensate for the rotational components require accurate calibration between the camera and the inertial measurement unit.

A. Argyros and F. Bergholm [2] achieved a centering effect in real time, for a mobile robot using a trinocular vision system: a camera pointing toward the front of the robot and two lateral cameras. The central camera was used to remove the optic flow rotational components calculated by peripheral cameras. The difference of both remaining optic flows was fed back, so as to produce a centering effect. Moreover, in [3], Argyros et al. used a panoramic camera to develop a control strategy for centering a mobile robot in a hallway. By using a coordinate transformation (polar to Cartesian) to find a set of cylindrical images, a large number of perspective images were approximated, while each column of pixels defined a steering angle. The central region was considered to be the one representing the angle $\theta = 0$, the optic flow was used to compensate offset components caused by the rotation of the robot. By taking symmetrical windows with respect to the central region, a centering effect was produced by performing a proportional control strategy using the optic flow calculated in each window. The embedded implementation was performed using two Pentium III (800 MHz) and Lucas-Kanade algorithm [24] for estimating optic flow.

S. Serres *et al.* [31] studied insects, in order to develop a stabilized binocular system comprising optic flow sensors instead of a standard camera system and an estimation algorithm. Each sensor had only two pixels and they were always aligned to the direction of displacement with gyro-compass servoing. Lateral flows measured by these sensors were then used in an optic flow regulator that ensured the stabilization of a hovercraft in the middle of a corridor. This strategy was only tested in simulation.

In Conroy *et al.* [14], a bio-inspired optic flow navigation system was implemented on a quad-rotorcraft and demonstrated in an indoor textured corridor. The inner-loop pitch and roll stabilization was accomplished using rate gyros and accelerometers. Altitude control was achieved via the fusion of sonar and accelerometer measurements. A ventrally located optic flow sensor from Centeye was used to increase the longitudinal and lateral damping, thus improving vehicle stability. An omnidirectional visual sensor, based on a Surveyor camera board and a parabolic mirror, was used for optic flow estimation and outer-loop control. The navigation strategy consisted in decomposing translational optic flow patterns (magnitude, phase, and asymmetry) with weighting functions, in order to extract signals that encoded relative proximity and speed, with respect to obstacles in the environment, which were used directly for outer-loop navigation feedback. The flight tests were performed in a textured corridor about 1.5m wide and 9m long. The quadrotor successfully avoided corridor walls and finished its course without collisions.

Obstacle detection

T. Camus used [10] a method for estimating optic flow in real time, in order to calculate the time-to-contact or time-to-collision of an obstacle. Time-to-contact is mathematically defined as the time until an object crosses an infinite plane defined by the image plane. T. Camus showed that taking a series of circles with different radii and center as the focus of image expansion, the time-to-contact may be calculated as the ratio between the distance between each point and the circle center, and the divergence from this point.

$$TTC = \frac{\sqrt{(x-x_{FOE})^2 + (y-y_{FOE})^2}}{\sqrt{V_x^2 + V_y^2}} \quad (13)$$

where (x_{FOE}, y_{FOE}) are the coordinates of the expansion focus (the point in the center of the horizon from which, when we are in motion, all points in the perspective image seem to emanate), V_x et V_y are the components of the velocity vector at point (x, y). The time-to-contact was calculated in each region defined by means of a circle of a given radius. The global result, in the entire image, was taken as the estimate of Least Squares of means for each region. The expansion focus position was estimated at any time from the average displacement of pixels. Indeed, T. Camus only considered full frontal obstacles, whose initial FOE estimate was the image center. The average of each coordinate of the optic flow vector was considered as an offset of the initial estimate.

In [25], L. Muratet et al. implemented, in a helicopter, a simplified method of this technique. They considered a fixed focus of expansion in the image center. They developed a control law that operated the opposite of this time-to-contact, commonly known as the relative depth.

J.-C. Zufferey and D. Floreano [38] implemented an obstacle avoidance technique for a 30g autonomous aerial vehicle. In this study, they were inspired by insects and optic flow sensors were used instead of the usual vision system. Optic flow sensors were composed of a 102 pixel vector and a microcontroller containing an estimation algorithm proposed by Srinivasan [33]. In order to compensate for the optic flow rotational components, sensors were first calibrated using a gyroscope. Since the output signal of these sensors was of the same order as that of the gyroscope, the measurement of the latter was directly used to subtract rotational components of the measured optic flow. The proposed obstacle avoidance method was only based on the optic flow divergence. When approaching an obstacle, the optic flow at the sensor extremes had a maximum amplitude and opposite directions. The difference of these two maximum speeds gave an estimate of the optic flow divergence, which is proportional to the inverse of the time-to-contact. This approach was justified by a theoretical study of areas of interest generated by calculating the optic flow, using a camera as it approached an obstacle. Experiments were conducted on a platform on the ground and on the aircraft for indoor flight.

A. Beyeler *et al.* [7] conducted a study on altitude control, considering it as a case of detection and obstacle avoidance. By applying the method developed previously by [38], they built a pseudo-centering method for a vehicle in indoor flight. The floor and roof were therefore taken as a corridor. The authors showed simulation results, but the algorithm was not experimentally tested.
A. Beyeler, J.-C. Zufferey and D. Floreano [8] described another optic-flow-based autopilot (optiPilot) for an outdoor fixed-wing MAV. The optiPilot was based on seven optic mouse sensors, MEMS rate gyroscopes and a pressure-based airspeed sensor. It was validated in simulations and demonstrated in real flights to avoid a group of tall trees (lateral avoidance) and small trees.

Passive navigation: terrain following and automatic landing

The passive navigation is the UAV ability to determine its own motion parameters with respect to the navigation reference. The estimation of these parameters is commonly called egomotion problem. However, there are new approaches used to estimate other important parameters for navigation, such as: altitude, direction of travel, pitch and roll angle, in order to perform terrain following, or altitude control and automatic landing.

J.-C. Zufferey *et al.* [38] used an altitude control using optic flow. They showed that an optic flow sensor pointing downward from the vehicle measured optic flow generated by the translational movement. A regulator allowed a constant optic flow to be maintained, which was equivalent to keeping the vehicle at a constant height. In [6], A. Beyeler *et al.* developed a similar strategy, but directly using raw (unprocessed) data from the optic flow sensor. By applying an optimization technique to the sensor raw data, the pitch angle and the altitude were estimated in real time, without needing to estimate any velocity vector.

The optic floww obtained from a camera pointing downwards may be used to perceive depth (ground). In fact, the optic flow model that appears for a 1D translational movement can be expressed as [29]

$$OF[^\circ] = \frac{V_x}{Z}\sin\alpha - \Omega_y \tag{14}$$

where Ω_y is the pitch angular velocity and α is the elevation angle. Since the optic flow rotational component Ω_y does not contain distance information, it is generally assumed to be equal to zero, or it is compensated for using inertial data. Most techniques based on optic flow for UAVs altitude control refer to equation (14).

The idea of terrain following is to fly at a fixed altitude above the ground by maintaining the optic floww at a constant value and then following the ground profile by regulating the optic flow [29], [12]. Maintaining a constant optic flow in (14) leads to an automatic reduction of the horizontal speed V_x when the height Z decreases. Thus, if the descent speed V_z is controlled proportionally to the forward speed, the descent angle remains constant.

Bees use this simple strategy (two rules) to ensure a smooth landing without explicit measurement or knowledge of their flight speed, or height above the ground [34]. Therefore, the height, the forward and descent speeds will exponentially decrease with time and become null at landing. This landing strategy inspired by insects has been demonstrated by many researchers [29], [13], [17].

Wagter and Mulder [36] constructed an algorithm for performing the 3D terrain reconstruction using optic flow. However, their system considered that the real rotation and translation speeds were known. Since the optic flow component created by the vehicle motion is inversely proportional to the depth and directly proportional to the translation speed, if the translation speed is known, the depth can be deduced. By using a camera pointing downward from an autonomous vehicle, the altitude can be estimated in real time so a reconstruction of the terrain can thus be achieved. The algorithm estimated the optic flow generated by the vehicle rotations from the gyro data, and this estimation was subtracted from the measured optic flow.

In Kendoul, Fantoni, and Nonami [23], the Structure From Motion (SFM) method (which deals with camera ego-motion estimation and the reconstruction of the 3D structure of the scene) was successfully applied for the state estimation and flight control of a small quadrotor system. The proposed implementation was based on three nested Kalman filters (3NKF) and involved real-time computation of the optic flow, fusion of visual and inertial data, and recovery of translational

motion parameters. The scale factor ambiguity was solved by using height measurements from a pressure sensor and assuming flat ground. The system was also augmented by a visual odometer that estimated the horizontal position by integrating image displacements. The developed system was first validated on a ground vehicle and then on a quadrotor system through vision-based autonomous flights, including automatic takeoff and landing, outdoor and indoor hovering, and velocity command tracking.

An optic flow-based terrain-following algorithm for mini quad-rotorcraft was proposed in Hérissé et al. [20]. The developed system computed optic flows at multiple observation points, obtained from two onboard cameras, using the *LK* algorithm and combined this information with forward speed measurements to estimate the height above the ground. A backstepping controller was used to regulate the height to some desired value. Indoor closed-loop flights were performed over a textured terrain using the CEA quadrotor vehicle flying at a forward speed of 0.3-0.4 m/s. The system was able to maintain a desired height of 1.5 m above a ramp and 2D corner textured terrain of about 4 m in length. In an extended work [21], the authors presented a nonlinear controller for a *VTOL UAV* that exploited a measured optic flow to enable hover and landing control on a moving platform. Their first objective concerned the stabilization of the vehicle relative to the moving platform that maintained a constant offset from a moving reference. The second concerned the regulation of automatic vertical landing onto a moving platform. Experimental results were provided for a quadrotor *UAV* to demonstrate the performance of the proposed control strategy.

In Sabiron *et al.* [30], a 6-pixel low-speed Visual Motion Sensor (VMS), inspired by insect visual systems, performed local 1-D angular speed measurements ranging from 1.5°/s to 25°/s. The sensor was tested under free-flying outdoor conditions over various fields, onboard an 80kg unmanned helicopter. The results showed that the optic flow measured closely matched the approximate ground-truth optic flow, despite the complex disturbances encountered. The sensor was also able to accurately sense low visual angular speeds, giving quite frequently refreshed measurements even at great heights over an unknown complex outdoor environment.

Application to path following

The strength of optic flow-based strategies for aerial vehicles has been extensively demonstrated through the work cited above. In the Heudiasyc laboratory, we have addressed the problem of hover flight and velocity regulation of a quad-rotorcraft along a followed path (line). This objective can be viewed as a first step for future work relating to road following, for traffic monitoring, or for power or railway line supervision.

Line following with velocity regulation

In [27] and [28], the problem of stabilizing the position and translational velocity of a quad-rotorcraft during autonomous flight along a road was considered. The proposed solution was based on a vision algorithm for line detection and optic flow computation. The algorithm used images provided by a monocular camera system embedded onboard the *UAV*. Such research involves two fundamental characteristics for any autonomous navigation system: we seek to accurately measure translational displacements and to eliminate the position drift when hovering. If the translational drift is correctly compensated, the hover flight can be used as an intermediary task between different flying behaviors, each suited to different conditions of the environment. Furthermore, velocity regulation was implemented to establish the different flying modes, such as lateral displacement and forward displacement actions. In order to make use of the optic flow in a very appropriate manner, a vision-based altitude controller was also developed. The combination of these three vision-based controllers (hover flight, velocity regulation and altitude stabilization) allowed the vehicle to navigate autonomously over a road model in a real-time application. Two different kinds of missions were tested: position hold over a road segment and road following at constant velocity. In the work [28], the road model was known (especially the width of the road); this allowed the altitude of the *UAV* above the road to be deduced. A diagram of the quad-rotorcraft above the road model can be seen in figure 3. The performance of the proposed methodologies was validated in a simulation environment. Real-time experimental applications with a quad-rotorcraft aerial vehicle, consisting of autonomous hover and forward flight at constant velocity were successfully achieved. Note that the experiments were achieved through a supervisory PC ground station, while the vision system was embedded on the *UAV*. The communication between the quad-rotorcraft and the ground station was provided by a wireless data link. The image processing computations were done on the ground station and sent to the *UAV*, where the low-level controllers were executed onboard the vehicle.

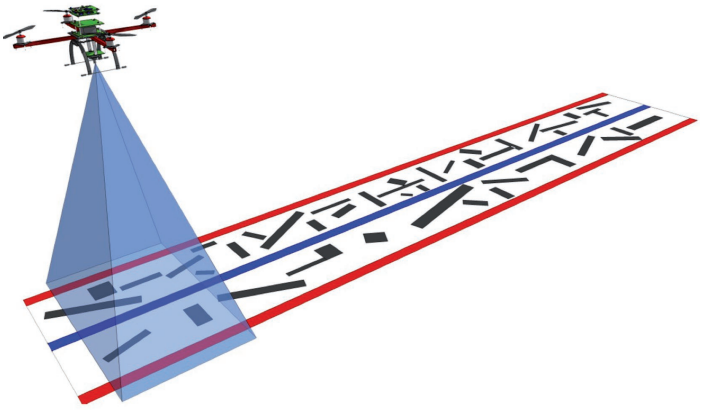


Figure 3 - Quad-Rotorcraft navigation: a diagram of the proposed setup presented in [28]

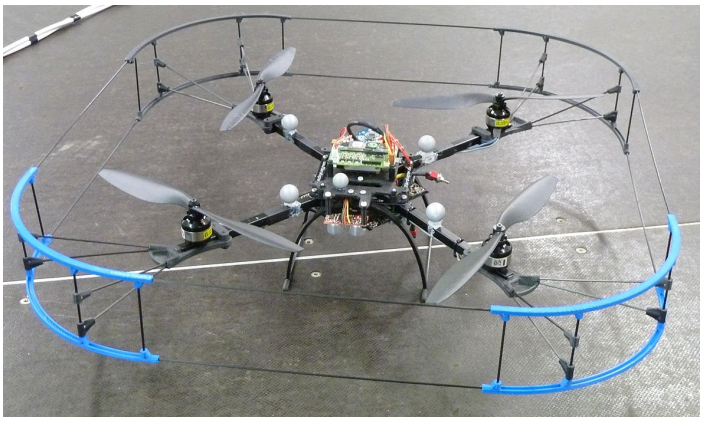


Figure 4: Experimental quadrotor designed and constructed at the Heudiasyc laboratory

Full autonomous line following

Heudiasyc quad-rotorcraft

The Heudiasyc laboratory at the University of Technology of Compiègne, in France, has a long experience with control and navigation of *UAVs*, especially using quadrotors. The latest version of our experimental quadrotor, shown in figure 4, is a *UAV* based on a Mikrokopter frame (with modifications). It has four brushless motors driven by *BLCTRLV2* controllers. The total weight is 1.2 kg, using an 11.1 V *LiPO* battery of 6000 mAh, giving about 15 minutes of flight time. The main electronic board is based on an *IGEP* module, equipped with a System On Chip (*SOC*) DM3730 from Texas Instruments. This *SOC* has one *ARM Cortex A8* core running at 1 GHz and one *DSP C64x+* core running at 800 MHz, which allows embedded image processing. The *ARM* processor allows Linux and its real-time extension Xenomai to be run. Thus, the control law runs in real time at 100 Hz. The *UAV* is also equipped with a Microstrain *3DMGX3-25 IMU* giving Euler angles and rotation speed measurements at 100Hz, a SRF10 ultrasonic range finder that provides the vehicle altitude at 50 Hz in a range between 0 and 2 meters and a PS3Eye camera capable of providing up to 120 images per second, at a resolution of 320x240 pixels. The camera points downwards, which allows the scene below the vehicle to be observed. Note that planarity of the scene is assumed in this work by the Heudiasyc laboratory. The images provided by the camera are processed by computer vision algorithms, in order to estimate the helicopter translational velocity in the x-y plane and the heading angle, as well as the lateral position w.r.t. the road direction. The translational velocity in the x-y plane is obtained from an optic flow algorithm, which is based on the pyramidal Lucas-Kanade method. For this purpose, the algorithm uses two pyramid levels, seeking up to 64 points of interest in the image. A Harris affine region detector was implemented to perform the characteristic feature detection. The DSP allows embedded calculation of the optic flow at about 100 Hz. Finally, the *UAV* is connected to the ground station via a WiFi connection.

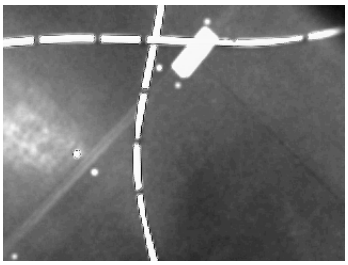
Line following

The strategy described here, is based on the work developed in [11]. A line is placed on the ground and the *UAV* follows it autonomously. This is done thanks to the downward looking embedded camera, which computes a Hough transform to detect lines. Thus, the *UAV* stabilizes itself over the line and can move forward using the optic flow to regulate its speed. Again, the embedded *DSP* is used for all image processing, running at 90 frames per second, to provide high speed measurements. The main contribution was to permit the quadrotor to navigate over a line in a completely autonomous manner with full embedded control and image processing. No external sensors or system (such as a motion capture system) are needed here, despite the presence of markers onboard the quadrotor (figure 4 of the quadrotor).

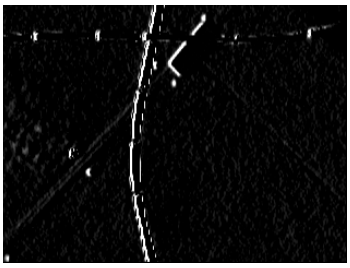
Indeed, we have used markers for other experiments where the knowledge of the vehicle position was needed. The present methodology has resulted in several improvements to the embedded control and image processing and a summary of real-time experiments that are currently being shown in the Heudiasyc Laboratory is presented here.

The quadrotor is able to follow a line without knowing its size, its color or its shape. We perform a demonstration with a closed 8-shape, where the quadrotor can continuously navigate over the path as shown in figure 5. A raw gray scale image obtained from the onboard camera while the vehicle is flying over the road is shown in Figure 6(a). The image obtained by edge detection using Sobel is shown in figure 6(b) and a binary image (black and white image) deduced by thresholding is presented in figure 6(c). Specifically, the grayscale image is used by the optic flow algorithm, while the black and white image is used by the line detection algorithm. A Hough transform computation allows vertical straight lines to be found in the image with an appropriate selection of the followed line. Finally, the post-processed image where the road has been detected is highlighted in green, as shown in Figure 6(d). Note that the Sobel detection is only performed in the x-axis direction to speed up image processing. Therefore, the heading angle of the vehicle is aligned

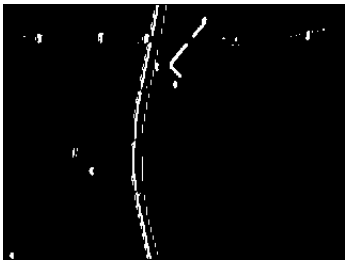
Figure 5 - Line following mission: the quadrotor flies over the line autonomously



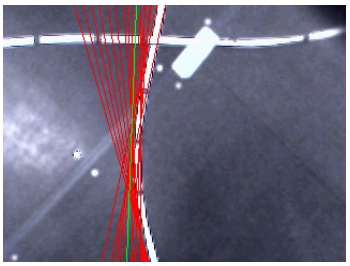
a) Raw grayscale image



b) Sobel detection



c) Binary image



d) Processed image after Hough transform algorithm

Figure 6 - Image processing

with the direction of the line. Using imaging sensors, the lateral distance of the vehicle is then stabilized through the yaw angle, in order to navigate exactly over the road. Optic flow computation also allows the vehicle to navigate at a constant velocity over the path. Figure 7 shows optic flow vectors at points of interest and projected on one image. Figures 8(a) to 8(d) describe the consecutive images along the path. Since the overall objective of such an application would be

to supervise road traffic, power lines or railway infrastructure, the idea was also to consider cases when the line goes out the field of view of the downward pointing camera. Therefore, a switching control strategy was designed in order to consider both operational modes: the nominal case presented above, i.e., the camera allows the lateral position and the yaw angle to be stabilized; and the degraded mode, which is implemented in case the vehicle loses the line. In the nominal case, the forward speed is constant and regulated by optic flow. The degraded mode involves an algorithm integrating the lateral optic flow, in order to estimate the position and return to the line.

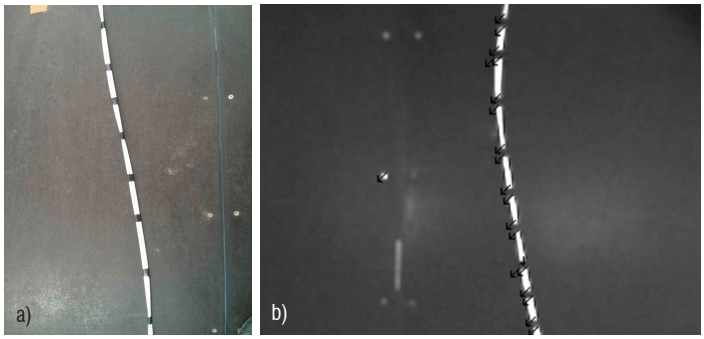


Figure 7 - Optic flow vectors from points of interest projected on one image along the line
a) Example of an original image with pieces of black tapes to improve Harris point detection, in order to avoid the aperture problem
b) Optic flow vectors projected on one image

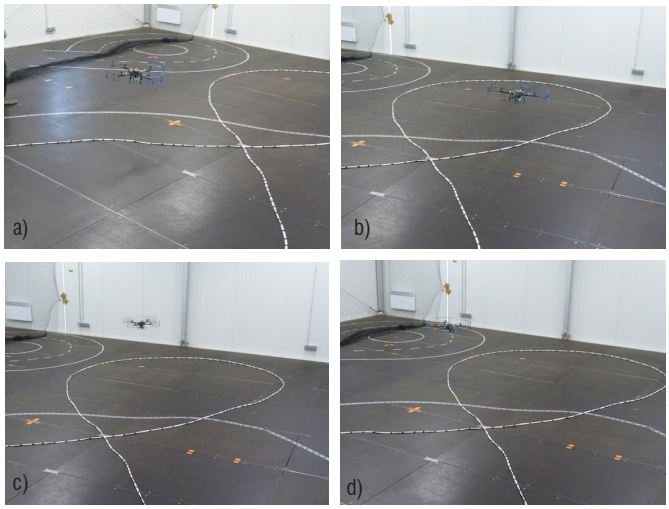


Figure 8 - Consecutive images along the 8-path

In this mode, the yaw angle is pointed using the IMU with the preceding value before the loss of the road detection. Figures 9(a)-9(c) demonstrate the degraded mode when the quadrotor is perturbed. A video of the demonstration can be viewed on the Heudiasyc *UAV* team web site: www.hds.utc.fr/uav-horus/platform/videos/.

Figure 9 - The quadrotor is moved from its initial path by a manual perturbation. It can retrieve the followed line

Conclusion

Significant progress has been made in the development of autonomous *UAVs*. Although there is often a gap between theoretical work and experimental developments, numerous reported experiments are appearing with validation in real time. Vision-based navigation, including bio-inspired strategies such as the use of optic flow, is a very active research topic. Numerous developments are arising, especially indoors. New technologies and extensive engineering work are also needed for outdoor implementations. We believe that bio-inspired visual control and navigation approaches for *UAVs* will increase in the future■

Acronims

VTOL	(Vertical Take-Off and Landing)
UAV	(Unmanned Aerial Vehicule)
IMU	(Inertial Measurement Unit)
FOE	(Focus Of Expansion)
MAV	(Micro Aerial Vehicule)
SFM	(Structure From Motion)
3NKF	(Three Nested Kalman Filter)
LK	(Lucas Kanade(algorithm))
VMS	(Visual Motion Sensor)
SOC	(System On Chip)

Acknowledgements

This work was carried out within the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02) and the Equipex ROBOTEX (Reference ANR-10-EQPX-44-01), which were funded by the French Government, through the program “Investments for the future” managed by the National Agency for Research. The second author is financed by the European Regional Development Fund.
The European Union is investing in your future.

References

[1] P. ANANDAN - *A Computational Framework and an Algorithm for the Measurement of Visual Motion*. International Journal of Computer Vision, 2(3):283–310, 1989.

[2] A. ARGYROS, F. BERGHOLM - *Combining Central and Peripheral Vision for Reactive Robot Navigation*. Computer Vision and Pattern Recognition Conference CVPR'99, pages 2646–2651, Fort Collins, Colorado, USA, June 23-25, 1999.

[3] A. ARGYROS, D.-P. TSAKIRIS, C. GROYER - *Biomimetic Centering Behavior for Mobile Robots with Panoramic Sensors*. IEEE Robotics and Automation Magazine, 11(4):21–68, 2004.

[4] J.-L. BARRON, D.-J. FLEET, S.-S. BEAUCHEMIN - *Performance of Optical Flow Techniques*. International Journal of Computer Vision, 12(1):43–77, 1994.

[5] G.-L. BARROWS, J.-S. CHAHL, M.-V. SRINIVASAN - *Biomimetic Visual Sensing and Flight Control*. Bristol UAVs Conference, Bristol, UK, 2002.

[6] A. BEYELER, C. MATTIUSSI, J.-C. ZUFFEREY, D. FLOREANO - *Vision-Based Altitude and Pitch Estimation for Ultra-Light Indoor Microflyers*. IEEE International Conference on Robotics and Automation, pages 2836–2841, Orlando, Florida, 2006.

[7] A. BEYELER, J.-C. ZUFFEREY, D. FLOREANO - *3D Vision-Based Navigation for Indoor Microflyers.* IEEE international conference on Robotics and Automation, pages 1336–1341, Roma, Italy, 2007.

[8] A. BEYELER, J.-C. ZUFFEREY, D. FLOREANO - *Vision-Based Control of Near-Obstacle Flight*. Autonomous Robots, 27:201–219, 2009.

[9] J.-Y. BOUGUET - *Pyramidal Implementation of the Lucas Kanade Feature Tracker.* Technical report, Intel Corporation, Microprocessor Research Labs, Technical report, 1999.

[10] T. CAMUS - *Calculating Time-to-Contact Using Real-Time Quantized Optical Flow*, 1995.

[11] L.R. GARCÍA CARRILLO, G. FLORES, G. SANAHUJA, R. LOZANO - *Quad Rotorcraft Switching Control: An Application for the Task of Path Following*. IEEE Transactions on Control Systems Technology, 22(4): 1255-1267, July 2014.

[12] J. CHAHL, A. MIZUTANI - *An Algorithm for Terrain Avoidance Using Optical Flow*. IEEE American Control Conference, pages 2742–2747, Minnesota, USA, 2006.

[13] J.-S. CHAHL, M.-V. SRINIVASAN, S.-W. ZHANG - *Landing Strategies in Honeybees and Applications to Uninhabited Airborne Vehicles*. International Journal of Robotics Research, 23(2):101–110, February 2004.

[14] J. CONROY, G. GREMILLION, B. RANGANATHAN, J.-S. HUMBERT - *Implementation of Wide-Field Integration of Optic Flow for Autonomous Quadrotor Navigation*. Autonomous Robots, 27:189–198, 2009.

[15] D.-J. FLEET, A.-D. JEPSON - *Computation of Component Image Velocity from Local Phase Information.* International Journal of Computer Vision, 5:77–104, 1990.

[16] N. FRANCESCHINI, J. PICHON, C. BLANES - *From insect Vision to Robot Vision*. Philosophical Transactions of the Royal Society of London B, 4(4):283–294, 1992.

[17] W.-E. GREEN, P.-Y. OH, G.-L. BARROWS - *Flying Insect Inspired Vision for Autonomous Aerial Robot Maneuvers in Near-Earth Environments.* IEEE International Conference on Robotics and Automation ICRA 2004, volume 3, pages 2347–2352, New Orleans, USA, April 2004.

[18] T. HAMEL, R. MAHONY - *Visual Servoing of an Under-Actuated Dynamic Rigid-Body System: An Imagebased Approach.* IEEE Transactions on Robotics and Automation, 18(2):187–198, 2002.

[19] D.-J. HEEGER - *Optical Flow Using Spatiotemporal Filters*. International Journal of Computer Vision, 1(4), 1988.

[20] B. HÉRISSÉ, T. HAMEL, R. MAHONY, F.-X. RUSSOTTO - *A Terrain-Following Control Approach for a VTOL Unmanned Aerial Vehicle Using Average Optical Flow.* Autonomous Robots, 29(3-4), 2010.

[21] B. HÉRISSÉ T. HAMEL, R. MAHONY, F.-X. RUSSOTTO - *Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow.* IEEE Transactions on Robotics, 28(1), 2012.

[22] B.-K.-P. HORN, B.-G. SCHUNCK - *Determining optical Flow*. Artificial Intelligence, 17:185–204, 1981.

[23] F. KENDOUL, I. FANTONI, K. NONAMI - *Optic Flow-Based Vision System for Autonomous 3D Localization and Control of Small Aerial Vehicles.* Robotics and Autonomous Systems, 57:591–602, 2009.

[24] B. LUCAS, T. KANADE - *An Iterative Image Registration Technique with an Application to Stereo Vision*. Proceedings DARPA IU Workshop, pages 121–130, 1981.

[25] L. MURATET, S. DONCIEUX, Y. BRIRE, J.-A. MEYER - *A Contribution to Vision-Based Autonomous Helicopter Flight in Urban Environments*. Robotics and Autonomous Systems, 50(4):195–209, 2005.

[26] H.-H. NAGEL - *Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences*. Computer Vision, Graphics, and Image Processing, 21(1):85–117, 1983.

[27] E. RONDÓN, I. FANTONI-COICHOT, A. SANCHEZ, G. SANAHUJA - *Optical-Flow Based Controller for Reactive and Relative Navigation Dedicated to a Four Rotorcraft.* International Conference on Intelligent Robots and Systems (IROS), pages 684–689, St. Louis, USA, October 10-15 2009.

[28] E. RONDÓN, L.-R. GARCIA-CARRILLO, I. FANTONI - *Vision-Based Altitude, Position and Speed Regulation of a Quadrotor Rotorcraft*. International Conference on Intelligent Robots and Systems (IROS), pages 628–633, Taipei, Taiwan, October 2010.

[29] F. RUFFIER, N. FRANCESCHINI - *Optic Flow Regulation: the Key to Aircraft Automatic Guidance.* Elsevier, Robotics and Autonomous Systems, 50:177–194, 2005.

[30] G. SABIRON, P. CHAVENT, T. RAHARIJAONA, P. FABIANI, F. RUFFIER - *Low-Speed Optic-Flow Sensor Onboard an Unmanned Helicopter Flying Outside Over Fields.* Proc. IEEE International Conference on Robotics and Automation (ICRA), pages 1742–1749, Karlsruhe, Germany, 2013.

[31] J. SERRES, F. RUFFIER, S. VIOLLET, N. FRANCESCHINI - *Toward Optic Flow Regulation for Wall-Following and Centring Behaviours*. International Journal of Advanced Robotic Systems, 3(2):147–154, 2006.

[32] A. SINGH - *An Estimation-Theoretic Framework for Image-Flow Computation.* Proceedings of the third International Conference on Computer Vision ICCV, pages 168–177, Osaka, Japan, 1990.

[33] M.-V. SRINIVASAN - *An Image-Interpolation Technique for the Computation of Optic Flow and Egomotion*. Biological Cybernetics, 71:401–416, 1994.

[34] M.-V. SRINIVASAN, S.-W. ZHANG, J.-S. CHAHL - *Landing Strategies in Honeybees, and Possible Applications to Autonomous Airborne Vehicles.* Biological Bulletin, 200:216–221, April 2001.

[35] S. URAS, F. GIROSI, A. VERRI, V. TORRE - *A Computational Approach to Motion Perception.* Biological Cybernetics, 60(2):79–97, 1988.

[36] C. De Wagter, J.A. Mulder - *Towards Vision-Based UAV Situation Awareness.* AIAA Guidance, Navigation, and Control Conference, pages 1–16, San Francisco, California, 2005.

[37] A.-M. WAXMAN, S. ULLMAN - *Surface Structure and Three Dimensional Motion from Image Flow Kinematics*. International Journal of Robotics Research, 4(3):72–94, 1985.

[38] J.-C. ZUFFEREY, D. FLOREANO - *Toward 30-Gram Autonomous Indoor Aircraft: Vision-Based Obstacle Avoid-Ance and Altitude Control.* IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.

AUTHORS

Isabelle Fantoni received the PhD degree, with the European label, in Non-linear Control for Underactuated Mechanical Systems, in 2000 from the “Université de Technologie de Compiègne”, in France. Since October 2001, she is a permanent Researcher at Heudiasyc laboratory, UTC, in Compiègne, France, employed by the French National Foundation for Scientific Research (CNRS) and CNRS Research Director since October 2013. Her research interests include non-linear control, modelling and control for Unmanned Aerial Vehicles (UAVs), fault-tolerant control for UAVs, vision for navigation of aerial vehicles, cooperation of UAVs, heterogeneous robotic systems in cooperation.

Guillaume Sanahuja has done his PhD thesis at Heudiasyc laboratory, Université de Technologie de Compiègne, in 2010 on the topic of vision for UAVs. He is now working in the same laboratory as a research engineer. His work is related to the French project Equipex (equipment of excellence) ROBOTEX, which aim is to equip Heudiasyc and its partners with robotic platforms. His function is to design and develop embedded architectures at both software and hardware levels. His topics of interests are embedded systems, embedded vision, non-linear control.

M.-D. Hua, G. Ducard, T. Hamel
(CRNS)
R. Mahony
(ANU)

E-mail: hua@isir.upmc.fr

DOI : 10.12762/2014.AL08-04

Introduction to Nonlinear Attitude Estimation for Aerial Robotic Systems

A robust and reliable attitude estimator is a key technology enabler for the development of autonomous aerial vehicles. This paper is an introduction to attitude estimation for aerial robotic systems. First, attitude definition and parameterizations are recalled and discussed. Then, several attitude estimation techniques – ranging from algebraic vector observation-based attitude determination algorithms to dynamic attitude filtering and estimation methodologies – are presented and commented upon in relation to practical implementation issues. Particular attention is devoted to the applications of a well-known nonlinear attitude observer (called explicit complementary filter in the literature) to aerial robotics, using a low-cost and light-weight inertial measurement unit, which can be complemented with a GPS or airspeed sensors.

Introduction

The growing interest of the robotics research community in aerial robotic vehicles is partly related to numerous Applications, such as surveillance, inspection, and mapping. The development of a small-scale low-cost autonomous aerial vehicle requires effective solutions to a number of key technological problems. The avionics system of such a vehicle is arguably the technology that is most closely coupled to the autonomy of the vehicle. Within an avionics system, the attitude estimator provides the primary measurement that ensures robust stability of the vehicle flight. The development of a robust and reliable attitude estimator that can run on low-cost computational hardware and that requires only measurements from low-cost and light-weight sensing systems, is a key technology enabler for the development of such systems.

This paper is an introduction to attitude estimation for aerial robotic systems, with a focus on nonlinear attitude observers. In fact, recent advances in observer theory have led to the development of a significant body of nonlinear attitude observers [13], [25], [26], [32], [40], [47], [51]. These observers are algorithmically simple and can be implemented on low-processing power microprocessors in unit quaternion form. They need only vector measurement inputs from low-cost and light-weight microelectromechanical system (MEMS) strap-down inertial measurement units (IMUs), which can be further complemented with a GPS or airspeed sensors. Typically, the algorithms make use of a measurement of angular velocity, measured by a 3-axis gyroscope, a vector direction estimate of the gravitational direction derived from a 3-axis accelerometer (based on the small acceleration assumption) and where possible, vector measurement of the magnetic field, measured by a 3-axis magnetometer [13], [25], [32]. All low-cost MEMS devices are subject to significant noise effects. Gyroscopes and accelerometers suffer from time-varying bias and noise due to temperature change, vibration and impacts; magntometer readings are corrupted by onboard magnetic fields generated by motors and currents, as well as external magnetic fields experienced by vehicles that maneuver in built environments. Earlier work in the development of attitude observers tackled the question of bias in the gyrometer MEMS devices [13], [25], [32], [47], [49] by introducing an adaptive bias estimate in the algorithm. Decoupling of input signals to ensure that the roll and pitch estimates are not affected by deviations in the magnetometer measurements was considered in [17], [32], [16] and represents an important modification of the basic algorithm to improve the overall quality of the attitude estimate. On the other hand, when the vehicle is subject to important linear accelerations, the attitude estimate provided by conventional solutions can be significantly erroneous, since the vector direction estimate of the gravitational direction is no longer close to that obtained from the accelerometer measurements. To cope with strong accelerations, a complementary GPS measurement of the vehicle's linear velocity can be used to estimate its linear acceleration and, subsequently, to improve the precision of the attitude estimate [31], [15], [38]. In addition, in the case of an aircraft performing a level turn, air pressure sensors, such as pitot tubes that measure the magnitude of the airspeed, can be combined with accelerometer readings in order to derive a more precise estimate of the gravitational direction and, thus, significantly improve the quality of the attitude estimate [23].

The paper is organized as follows. In § "Attitude parameterizations", attitude definition and parameterizations are recalled and discussed. In § "Overview on attitude estimation based on vector observations", existing attitude estimation methods based on vector observations,

including both static and dynamic attitude estimation methodologies, are reviewed with a particular discussion on a nonlinear explicit complementary filter/observer [25] that was proposed by the last two authors of this paper and has become a common solution for most aerial robotic applications. Then, § "Nonlinear attitude observers for aerial robotic systems" presents some relevant nonlinear attitude observers for aerial robotic systems that were developed on the basis of the explicit complementary filter. Finally, conclusions are given.

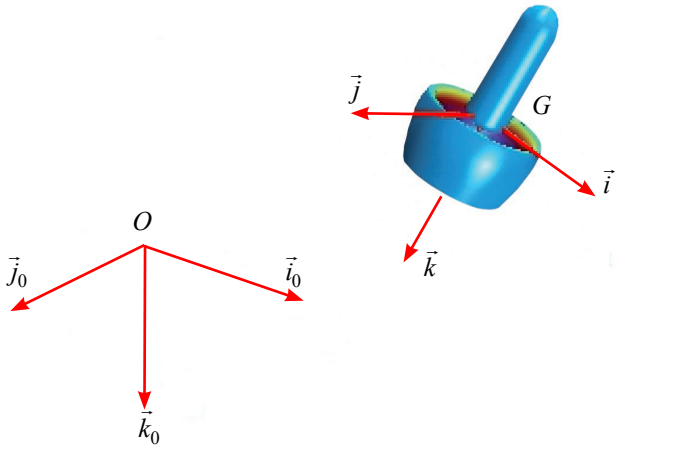


Figure 1- Attitude represents the orientation of body frame $\mathcal{B}=\{O;\vec{i},\vec{j},\vec{k}\}$ w.r.t. inertial frame $\mathcal{I}=\{O;\vec{i}_o,\vec{j}_o,\vec{k}_o\}$

Attitude parameterizations

The attitude represents the orientation of a frame, attached to the moving rigid body (i.e., body frame $\mathcal{B}$), with respect to (w.r.t.) an inertial reference frame $\mathcal{I}$ (see figure 1). It can be described by a rotation matrix, an element of the special orthogonal group $\mathrm{SO}(3)$, where

$$\mathrm{SO}(3)\triangleq\{R\in\mathbb{R}^{3\times3}\,|\,\det(R)=1, R^TR=RR^T=I_3\}$$

By denoting such a rotation matrix as $R\in\mathrm{SO}(3)$, it satisfies the following differential equation

$$\dot{R}=RS(\omega) \qquad (1)$$

where $\omega=[\omega_1,\omega_2,\omega_3]\in\mathbb{R}$ is the angular velocity vector of the body frame relative to the inertial frame, expressed in the body frame and the notation $S(\bullet)$ denotes the skew-symmetric matrix associated with the cross product $\times$, i.e., $\forall a,b\in\mathbb{R}^3, S(a)b=a\times b$.

Studies about the rotation group $\mathrm{SO}(3)$ started in the eighteenth century and the problem of parameterization of the rotation group of the Euclidean 3D-space has received great interest since 1776, when Euler showed that this group is a three-dimensional manifold. A rotation matrix has nine scalar components, but an element of the rotation group can be represented by a set of less than nine parameters. Three is the minimum number of parameters required for this. However, it was shown that no three-dimensional parameterization can be 1-1 (i.e., its transformation to $\mathrm{SO}(3)$ is a global diffeomorphism) [45]. Previously, Hopf showed in 1940 that no four-dimensional parameterization can be 1-1 and that a five-dimensional parameterization can be used to represent the rotation group in a 1-1 global manner. However, the greatest inconvenience of Hopf's five-dimensional parameterization concerns the nonlinearity of the associated differential equations [45]. On the other hand, four-dimensional parameterizations [45], [39], like the quaternion parameterization, only represent the rotation group in a 2-1 manner. Nevertheless, although the quaternion parameterization is not 1-1, no difficulty arises for practical purposes, because the transformation of a unit quaternion to $\mathrm{SO}(3)$ is a local diffeomorphism everywhere. Hereafter, the Euler angles and the quaternion parameterizations are recalled and discussed.

Euler angle parameterization

Among many three-dimensional parameterizations [45], the Euler angles are the most widely-used. Their definition depends on the problem to be solved and on the chosen coordinate systems. A definition commonly used in the aerospace field is the Euler angle parameterization with three angles ϕ, θ and ψ corresponding to roll, pitch and yaw respectively [45], [34]. These Euler angles allow a rotation matrix R to be factorized into a product of three matrices of rotation about three axes of the body frame as follows:

$$R=\begin{bmatrix}C\psi & -S\psi & 0\\ S\psi & C\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}C\theta & 0 & S\theta\\ 0 & 1 & 0\\ -S\theta & 0 & C\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & C\phi & -S\phi\\ 0 & S\phi & C\phi\end{bmatrix},$$

where S and C denote the $\sin(\bullet)$ and $\cos(\bullet)$ operators. They can be computed from the rotation matrix R as

$$\begin{cases}\phi=\mathrm{atan2}(r_{3,2},r_{3,3})\\ \theta=-\mathrm{asin}(r_{3,1})\\ \psi=\mathrm{atan2}(r_{2,1},r_{1,1})\end{cases}$$

where $r_{i,j}$ is the component of row i and column j of R. The Euler angle kinematics satisfy [45], [34]

$$\begin{cases}\dot{\phi}=\omega_1+S\phi T\theta\,\omega_2+C\phi T\theta\,\omega_3\\ \dot{\theta}=C\phi\,\omega_2-S\phi\,\omega_3\\ \dot{\psi}=\dfrac{S\phi}{C\theta}\omega_2+\dfrac{C\phi}{C\theta}\omega_3\end{cases}$$

where T denotes the $\tan(\bullet)$ operator. If $r_{3,1}=\pm1$, then $\theta=\mp\pi/2$, but ϕ and ψ are no longer well-defined. Therefore, the Euler angles constitute a parameterization of the rotation group, except at points corresponding to $\theta=\pm\pi/2$. Furthermore, when $\theta=\pm\pi/2,\dot{\phi}$ and $\dot{\psi}$ are not well-defined either. The problem of singularities is a weakness of the Euler angle parameterization and, as a matter of fact, of all three-dimensional parameterization techniques.

Unit quaternion parameterization

Compared to three-dimensional parameterizations, four-dimensional parameterizations allow singularities to be avoided. The earliest formulation of the four-dimensional parameterization, as pointed out in [39], was given by Euler in 1776. Earlier in 1775, he stated that in three dimensions, every rotation has an axis. This statement can be reformulated as follows (see e.g., [39], [37] for the proof).

Euler's theorem: For any $R\in\mathrm{SO}(3)$, there is a non-zero vector $\mathrm{v}\in\mathbb{R}^3$ satisfying $R\mathrm{v}=\mathrm{v}$.

This theorem implies that the attitude of a body can be specified in terms of a rotation by some angle about some fixed axis. It also

indicates that any rotation matrix has an eigenvalue equal to one. A number of four-dimensional parameterizations can be found in the literature (see e.g., [39], [7]), such as the Euler parameters, the quaternionparameters, the Rodrigues parameters and the Cayley-Klein parameters. Here, only the quaternion parameters are presented.

The quaternions were first proposed by Hamilton in 1843 [14] and further studied by Cayley and Klein. A unit
quaternion has the form
$q=s+iv_1+jv_2+kv_3$
where s, v_1, v_2, v_3 are real numbers satisfying $s^2+v_1^2+v_2^2+v_3^2=1$, called constituents of the quaternion q, and i, j, k are imaginary units that satisfy
$i^2=j^2=k^2=-1,\ ij=-ji=k,\ jk=-kj=i,\ ki=-ik=j$

In the literature, the unit quaternion q can be represented in a more concise way as $q=(s,v)\in\mathbb{R}\times\mathbb{R}^3$, where $s\in\mathbb{R}$ is the real part of the quaternion q and v and $v=[v_1,v_2,v_3]^T\in\mathbb{R}^3$ is its pure part or imaginary part. The quaternions are not commutative, but associative, and they form a group known as the quaternion group where the unit element is $\mathbf{1}\triangleq(1,0)$ and the quaternion product $\star$ associated with this group is defined by

$$\begin{bmatrix} s \\ v \end{bmatrix} \star \begin{bmatrix} \overline{s} \\ \overline{v} \end{bmatrix} = \begin{bmatrix} s\overline{s}-v^T\overline{v} \\ s\overline{v}+\overline{s}v+v\times\overline{v} \end{bmatrix}$$

The transformed rotation matrix R is uniquely defined from the unit quaternion q, using Rodrigues' rotation formula $R=I_3+2sS(v)+2S(v)^2$.

On the other hand, converting a rotation matrix to a quaternion is less direct. In fact, there always exists at least one component of the unit quaternion q different from zero. Once this component is identified, the quaternion can be deduced. Note that only two values of the unit quaternion q correspond to the rotation matrix R and that they have opposed signs. For example, if $\mathrm{tr}(R)\neq -1$, then

$$s=\pm\frac{1}{2}\sqrt{1+tr(R)},\quad S(v)=\frac{R-R^T}{4}$$

Finally, the quaternion kinematics are given by $\dot{q}=\frac{1}{2}q\star\begin{bmatrix}0\\ \omega\end{bmatrix}$

The quaternion parameterization involves four parameters (i.e., only one redundant parameter) and is free of singularities. The associated differential equation is linear in q. Furthermore, the structure of the quaternion group is, by itself, of great interest.

Overview on attitude estimation based on vector observations

Algebraic attitude determination

The attitude is often reconstructed from the observation of at least two non-collinear vectors. The first solution is the TRIAD algorithm, proposed by Black in 1964 [44], which algebraically computes the attitude matrix from the information in both the body frame and the inertial frame of two non-collinear unit vectors. More precisely, by denoting $v_1^{\mathcal{I}}, v_1^{\mathcal{B}}, v_2^{\mathcal{I}}, v_2^{\mathcal{B}}$ as the vectors of coordinates, expressed in the inertial frame and the body frame respectively, of two unitary Euclidean vectors $\vec{v}_1$ and $\vec{v}_2$, one has $v_1^{\mathcal{I}}=Rv_1^{\mathcal{B}}, v_2^{\mathcal{I}}=Rv_2^{\mathcal{B}}$, and the TRIAD algorithm provides the attitude matrix R as

$$R=\sum_{i=1}^{3}s_ir_i^T=\begin{bmatrix}s_1 & s_2 & s_3\end{bmatrix}\begin{bmatrix}r_1 & r_2 & r_3\end{bmatrix}^T,$$

with two orthonormal triads

$$\begin{cases} s_1=v_1^{\mathcal{I}}, s_2=\dfrac{v_1^{\mathcal{I}}\times v_2^{\mathcal{I}}}{|v_1^{\mathcal{I}}\times v_2^{\mathcal{I}}|}, s_3=s_1\times s_2, \\ r_1=v_1^{\mathcal{B}}, r_2=\dfrac{v_1^{\mathcal{B}}\times v_2^{\mathcal{B}}}{|v_1^{\mathcal{B}}\times v_2^{\mathcal{B}}|}, r_3=r_1\times r_2. \end{cases}$$

Although this algorithm is simple to implement, the resulting estimated attitude matrix, in the presence of measurement noises, is not guaranteed to remain in the rotation group $SO(3)$ and, thus, additional projection of the computed attitude into the group $SO(3)$ is often required (using, for example, the Gram-Schmidt orthonormalization). Moreover, the TRIAD algorithm can only accommodate two vector observations, which may lead to difficulty in treating information when the observation of more than two vectors is available. For instance, in this case, the observation of which pair of vectors provides the best attitude estimate using the TRIAD algorithm may not be known a priori. Additionally, it does not take the relative reliability of the vector observations into account, even in the case of two vector observations. These drawbacks of the TRIAD algorithm disappear in optimal algorithms, which compute the best attitude estimate based on a cost function for which all vector observations are taken into account simultaneously. Optimal algorithms are, however, computationally more expensive than the TRIAD algorithm. The first and also the best-known optimal attitude estimation problem is the least-square Wahba problem [52]. It consists in finding a rotation matrix $\hat{A}\in SO(3)$ which minimizes the cost function

$$J(A)\triangleq\frac{1}{2}\sum_{i=1}^{n\geq 2}k_i\,|\,v_i^{\mathcal{B}}-Av_i^{\mathcal{I}}\,|^2 \qquad (2)$$

where A corresponds to the transpose of the estimated attitude $\hat{R}\in SO(3)$ $(ie., A=\hat{R}^T)$; $\{v_i^{\mathcal{B}}\}$ is a set of measurements of $n(\geq 2)$ unit vectors, expressed in the body frame; $\{v_i^{\mathcal{B}}\}$ are the corresponding unit vectors, expressed in the inertial frame; and $\{k_i\}$ is a set of non-negative weights, which can be designed based on the reliability of the corresponding measurements. Wahba's problem allows arbitrary weighting of vector observations. In [42], the author proposes the particular choice $k_i=\sigma_i^{-2}$, the inverse variance of the measurement $v_i^{\mathcal{B}}$, in order to relate Wahba's problem to Maximum Likelihood Estimation of the attitude based on an uncorrelated noise model [42]. In fact, the cost function $J(A)$ defined by Eq. (2) can be rewritten as

$$\begin{aligned} J(A)&=\frac{1}{2}\sum_{i=1}^{n\geq 2}k_i tr\left((v_i^{\mathcal{B}}-Av_i^{\mathcal{I}})(v_i^{\mathcal{B}}-Av_i^{\mathcal{I}})^T\right) \\ &=\frac{1}{2}\sum_{i=1}^{n\geq 2}k_i\left(|\,v_i^{\mathcal{I}}\,|^2+|\,v_i^{\mathcal{B}}\,|^2\right)-tr(AB^T) \end{aligned}$$

with $B\triangleq\sum_{i=1}^{n\geq 2}k_iv_i^{\mathcal{B}}(v_i^{\mathcal{I}})^T$. Therefore, the problem of finding a rotation matrix $\hat{A}$ that minimizes $J(A)$ is equivalent to finding a rotation matrix $\hat{A}$ that maximizes $tr(AB^T)$. The first solutions to Wahba's problem, based on this observation, were proposed in 1966 by Farrell and Stuelpnagel [53] and by Wessner, Velman, Brock in the same paper[1]. However, these solutions, being computationally expensive, are not well suited to real-time applications. For instance, Farrell and

[1] These solutions were sent to Wahba and he presented them in that paper.

Stuelpnagel's method requires a polar decomposition of the matrix B into a product $B = UP$ (with U an orthogonal matrix[2] and P a symmetric and positive semidefinite matrix) and a diagonalization of P into $P = WDWT$ [3] (with W a orthogonal matrix and D a diagonal matrix whose diagonal elements are arranged in decreasing order, i.e., $D = diag(d_1, d_2, d_3)$ with $d_1 \geq d_2 \geq d_3$). The optimal rotation matrix $\hat{A}$ is then given by $\hat{A} = UWdiag(1,1,\det(U))W^T$.

As for Wessner's solution, which is a particular case of Farrell and Stuelpnagel's solution, the optimal rotation matrix $\hat{A}$ is calculated according to

$$\hat{A} = \left(B^T\right)^{-1}\left(B^T B\right)^{1/2}$$

For this solution, due to the inverse of B^T, a minimum of three (non-collinear) vector observations must be available, knowing that two non-collinear vectors are sufficient for attitude reconstruction using the TRIAD algorithm. In addition, the calculation of the square root of the matrix B^TB also requires expensive computation. For example, it is necessary to diagonalize B^TB as $B^TB = W_B D_B W_B^T$ to obtain $(B^TB)^{1/2} = W_B D_B^{1/2} W_B^T$.

No solution to Wahba's problem was able to replace the TRIAD algorithm in practice, until Davenport's q method [10] and the numerical technique QUEST (QUaternion ESTimator) [41] were proposed. By using the quaternion parameterization, Davenport transformed Wahba's problem into the problem of finding the largest eigenvalue λ_{max} of the symmetric Davenport matrix $\in \mathbb{R}^{4\times4}$ defined by

$$K \triangleq \begin{bmatrix} C-\gamma I_3 & z \\ z^\top & \gamma \end{bmatrix}$$

with $C \triangleq B + B^T$, $\gamma \triangleq \text{tr}(B)$, $z \triangleq \sum_{i=1}^{n\geq2} k_i v_i^{\mathcal{B}} \times v_i^{\mathcal{I}}$. The optimal quaternion, corresponding to the optimal rotation matrix $\hat{A}$ of Wahba's problem, is the normalized eigenvector q_{max} of K associated with the eigenvalue λ_{max}. In fact, the largest eigenvalue λ_{max} may be obtained by solving analytically the largest zero of the fourth-degree characteristic polynomial $det(K - \lambda I_4)$ [10]. However, Davenport's q-method is also computationally complex. This leads to the development of the QUEST algorithm by Shuster [41] on the basis of Davenport's q-method. QUEST consists in solving numerically the equation $det\ (K - \lambda I_4)$, or equivalently

$$\lambda^4 - (a+b)\lambda^2 - c\lambda + (ab + c\gamma - d) = 0 \qquad (3)$$

with[4] $a \triangleq \gamma^2 - tr(adj(C))$, $b \triangleq \gamma^2 + |z|^2$, $c \triangleq det(C) + z^\top Cz$, $d \triangleq z^\top C^2 z$. More precisely, based on Shuster's observation that λ_{max} is close to $\lambda_o \triangleq \sum_{i=1}^{n\geq2} k_i$, QUEST makes use of the Newton-Raphson method to solve Eq. (3), with λ_0 as the initial guess. It thus avoids the computation of all eigenvalues of K (i.e., all solutions to Eq. (3)). QUEST is theoretically less robust than Davenport's q-method, but it is clearly faster (normally few iterations are sufficient) and has proven to be reliable in practice (e.g., QUEST was implemented in the Magsat satellite in 1979). Many alternative numerical solutions for QUEST and Davenport's q-method to Wahba's problem have been proposed like, for instance, the Singular Value Decomposition (SVD), the Fast Optimal Attitude Matrix (FOAM), the Estimator of the Optimal Quaternion (ESOQ), ESOQ-1, ESOQ-2 algorithms [29]. These solutions, along with QUEST, for Wahba's problem require a trade-off between computational time and precision; for instance, the number of iterations has to be defined in advance. Additionally, their main shortcoming concerns the memoryless characteristic in the sense that information contained in measurements of past attitudes is not preserved.

Dynamic attitude filtering and estimation

Since a filtering algorithm is usually preferred when measurements are obtained over a range of times and especially when vector measurements are noisy, many alternative solutions to algebraic methods have been proposed. They combine the vector measurements with the kinematic equation of rotation (i.e., Eq. (1)) and the angular velocity measurements. In this manner, the attitude estimation methods such as TRIAD, QUEST, SVD, FOAM, ESOQ, etc., can still be used as a preprocessor (i.e., the role of an attitude sensor) for a certain number of attitude filtering methods, such as in many Kalman filters (KFs), extended Kalman filters (EKFs), or Kalman-like filters (see, e.g., [11], [21], [28], [9], [3] and the references therein), or nonlinear observers (see, e.g., [40], [24], [25], [47], [51]). However, this process is not a prerequisite and is loosened in many attitude filtering methods, including KFs and EKFs, as proposed in [43], [1], [13], [25], [26], [30], [50], [9], etc. This leads to simpler, faster and (probably) more accurate methodologies. For instance, consider the filter QUEST algorithm (a recursive discrete-time Kalman-like estimator) [43]. The author proposes to calculate the estimated attitude using the QUEST algorithm and by propagating and updating the matrix B (which is, itself, involved in the Davenport matrix K) as

$$B(t_k) = \mu\ \Phi_{3\times3}(t_k, t_{k-1})B(t_{k-1}) + \sum_{i=1}^{n_k} k_i v_i^{\mathcal{B}} (v_i^{\mathcal{I}})^T$$

where $\Phi_{3\times3}(t_k, t_{k-1})B(t_{k-1})$ is the state transition matrix of the transpose of the rotation matrix R, μ is a fading memory factor and n_k is the number of vector observations at time t_k. An alternative sequential algorithm for the filter QUEST is the recursive quaternion estimator (REQUEST) [1] which propagates and updates the Davenport matrix K by

$$K(t_k) = \mu\ \Phi_{4\times4}(t_k, t_{k-1})B(t_{k-1}) + \sum_{i=1}^{n_k} k_i K_i$$

where $\Phi_{4\times4}(t_k, t_{k-1})B(t_{k-1})$ is the quaternion state transition matrix and K_i is the Davenport matrix for a single vector observation

$$K_i = \begin{bmatrix} v_i^{\mathcal{B}}(v_i^{\mathcal{I}})^T + v_i^{\mathcal{I}}(v_i^{\mathcal{B}})^T - ((v_i^{\mathcal{B}})^T v_i^{\mathcal{I}})I_3 & (v_i^{\mathcal{B}} \times v_i^{\mathcal{I}}) \\ (v_i^{\mathcal{B}} \times v_i^{\mathcal{I}})^T & (v_i^{\mathcal{B}})^T v_i^{\mathcal{I}} \end{bmatrix}$$

The main shortcoming of the filter QUEST and REQUEST algorithms concerns the fading memory factor μ which, being arbitrarily chosen, makes these solutions suboptimal filters. This leads to the development of the Optimal-REQUEST algorithm [8][Ch.3] which, being essentially based on the REQUEST algorithm, further optimally calculates the fading memory factor μ in the update stage of REQUEST according to a covariance optimization argument. Note that the filter QUEST, REQUEST and Optimal-REQUEST algorithm, being based on

[2] $det(U)$ can be either 1 or -1.

[3] Note that any symmetric matrix is diagonalizable.

[4] Recall that $adj(A)A = Aadj(A) = det(A)I_n$, for an $n \times n$ matrix A.

QUEST, are numerical gradient methods and computationally more expensive than other modern algorithms (such as nonlinear attitude observers) due to their optimization-based nature.

Other worthy examples are nonlinear attitude observers. The earliest nonlinear observer was proposed by Salcudean [40] using the unit quaternion representation. This work has led to the development of a significant body of nonlinear attitude observers over the last fifteen years, by exploiting either the unit quaternion or the rotation matrix [40], [35], [51], [47], [24], [46], [25], [49], [32], [12], [16]. The performance of recent observers is comparable to modern nonlinear filtering techniques [9]. They generally have much stronger stability and robustness properties and are simple to tune and implement [25]. Most early nonlinear attitude observers were developed on the basis of Lyapunov analysis. However, the attitude estimation problem has also become an intuitive example for the development of recent theories on invariant observers for systems with symmetry [4], [5], [6], [25], [19], [48]. For instance, an illustrative example concerning the nonlinear explicit complementary filter proposed by Mahony et al. [25] is taken. This method is basically inspired by the Luenberger observer [22] in the sense that the dynamics of the estimated attitude $\hat{R}$ contains two parts: a main part copying the dynamics of the real attitude (i.e., Eq. (1)), and an innovation part allowing the correction of the estimated attitude to the real attitude. For instance, if the gyro measurements are not affected by biases, the observer dynamics is given as

$$\dot{\hat{R}} = \hat{R}S\left(\omega + \sum_{i=1}^{n} k_i v_i^{\mathcal{B}} \times \hat{v}_i^{\mathcal{B}}\right) \tag{4}$$

with k_i positive constant gains designed based on the reliability of the corresponding measurements and $\hat{v}_i^{\mathcal{B}} \triangleq \hat{R}^T v_i^{\mathcal{I}}$. An important issue of the attitude filtering concerns the gyro bias, leading to a complementary approach in which the gyroscopes are used to filter the vector measurements, which are in turn used to estimate the gyro bias (e.g., [25], [47], [51]). By making a constant gyro bias assumption (i.e., $\omega_y = \omega + b_\omega$, with ω_y the gyro measurement and b_ω the constant gyro bias vector), a complete version of the explicit complementary filter is given as [25]

$$\begin{cases} \dot{\hat{R}} = \hat{R}S(\omega_y - \hat{b}_\omega + \sigma_R) \\ \dot{\hat{b}}_\omega = \sigma_b \\ \sigma_R \triangleq \sum_{i=1}^{n} k_i v_i^{\mathcal{B}} \times \hat{v}_i^{\mathcal{B}} \\ \sigma_b \triangleq -k_b \sigma_R \end{cases} \tag{5}$$

where k_b, k_i are positive constants. This is a continuous version; however, a discrete version can be easily derived. It can also be conveniently rewritten in quaternion form [25], [16]. Furthermore, a fast attitude estimation method from vector observations like TRIAD or QUEST can be used for a good initial attitude estimate. This observer is derived by considering the Lyapunov function candidate

$$L \triangleq \sum_{i=1}^{n} k_i \, | \, v_i^{\mathcal{B}} - \hat{R}^T v_i^{\mathcal{I}} \, |^2 + \frac{2}{k_b} | \, b_\omega - \hat{b}_\omega \, |^2 \tag{6}$$

whose derivative along any solution to the error system is $\dot{\mathcal{L}} = -2 \, \| \, \mathbf{P}_a(\tilde{R}M) \, \|^2 \leq 0$, with , $\tilde{R} \triangleq \hat{R}^T R$, $M \triangleq \sum_{i=1}^{n} v_i^{\mathcal{B}} (v_i^{\mathcal{B}})^T$, $\|\cdot\|$ the Frobenius norm and $\mathbf{P}_a(\bullet)$ the skew-symmetric projection operator in square matrix space. With at least two non-collinear vector observations the estimated attitude asymptotically converges to the real one, for almost all initial conditions, even in the case of time-varying vectors (see [25], [48] for the detailed proofs). This is the strongest possible result knowing that no smooth globally asymptotically stable observers exist on $SO(3)$ due to the topology obstruction of the rotation group [2]. In the case of a single vector observation, the solutions to this filter are still well-posed, whereas analytically reconstructing the attitude from a single vector observation is not possible. In particular, almost-global asymptotical stability is still ensured in the case of a single vector observation under the persistent excitation assumption (i.e., if the vector's direction or the vehicle's attitude is permanently varying) [26].

Remark

Interestingly, the first term on the right-hand side of the Lyapunov function L defined by (6) corresponds to the cost function $J(\hat{R}^T)$ related to Wahba's problem. Therefore, in the absence of gyro bias, the (numerical) solution to the bias-free explicit complementary filter (4) converges asymptotically/exponentially to the optimal solution to Wahba's problem for almost all initial conditions under observability conditions.

A generalized version of the explicit complementary filter (5) recently proposed by Jensen [18] is given as follows

$$\begin{cases} \dot{\hat{R}} = \hat{R}S(\omega_y - \hat{b}_\omega + K_P \sigma) \\ \dot{\hat{b}}_\omega = \hat{R}S(\omega_y - \hat{b}_\omega + K_P \sigma) \\ \sigma \triangleq \sum_{i=1}^{n} k_i v_i^{\mathcal{B}} \times \hat{v}_i^{\mathcal{B}} \end{cases} \tag{7}$$

where k_i are positive constant scalar gains; K_P and K_I are symmetric positive definite matrices (not necessarily constant). Clearly, this generalized observer offers a larger tuning space than the explicit complementary filter (5) while ensuring similar stability and convergence properties; in addition, it is shown that bias-free multiplicative extended Kalman filter (MEKF) and constant-gain MEKF correspond to its particular cases [18].

Nonlinear attitude observers for aerial robotic systems

Sensor measurements

The most basic instrumentation embarked on an aerial robot for attitude estimation is a MEMS IMU, which can be complemented with a GPS or airspeed sensors. Assume that the IMU consists of a 3-axis gyroscope, a 3-axis accelerometer and a 3-axis magnetometer.

- The 3-axis gyroscope measures the angular velocity ω

$$\omega_y = \omega + b_\omega + \eta_\omega$$

where $\in \mathbb{R}$ is the measurement noise and $b_\omega \in \mathbb{R}^3$ denotes a constant (or slowly time-varying) gyro bias. Generally, gyroscopes are rather robust to noise and quite reliable for aerial robotics applications.

- The specific acceleration $a_{\mathcal{B}} \in \mathbb{R}^3$ expressed in the body-fixed frame $\mathcal{B}$ is defined as $a_{\mathcal{B}} = R^T(\dot{v} - ge_3)$, where the vehicle's

acceleration expressed in the inertial frame $\mathcal{I}$ is $\dot{v}$ and the gravitational acceleration expressed in the frame $\mathcal{I}$ is ge_3, with $e_3 = (0,0,1)^T$. The 3-axis accelerometer measures this specific acceleration

$$a_y = a_{\mathcal{B}} + b_a + \eta_a$$

where $\eta_a \in \mathbb{R}^3$ is the measurement noise and $b_\omega \in \mathbb{R}^3$ denotes a bias term. Accelerometers are highly sensitive to vibration and, thus, often require significant low-pass mechanical and/or digital filtering to be usable.

• The 3-axis magnetometer measures the ambient magnetic field

$$m_y = m_{\mathcal{B}} + B_m + \eta_m$$

where $\eta_m \in \mathbb{R}^3$ is the measurement noise, $m_B = R^T m_I \in \mathbb{R}^3$ is the Earth's magnetic field expressed in the body frame B and $B_m \in \mathbb{R}^3$ represents the sum of all local magnetic disturbances generated by motors and currents, as well as external magnetic fields experienced by vehicles that maneuver in built environments. While the magnetometer noise η_m is usually low, the local magnetic disturbance B_m can be significant, especially if the sensor is near the power wires of the motors.

Application of the explicit complementary filter for IMUs

In view of the attitude estimation survey in Section 3, most existing methods of attitude estimation make use of the measurement of at least two known non-collinear inertial vectors. Obtaining measurements of two known vectors is, however, difficult in practice and in particular for small aerial robots. In fact, while the magnetometer provides a vector measurement related to the Earth's magnetic field, the accelerometer does not directly measure the gravitational direction. This is due to the vehicle's linear acceleration involved in the specific acceleration that is measured by the accelerometer. However, most robotic vertical take-off and landing (VTOL) vehicles (such as multicopters, or ducted fans) spend a significant amount of time in near hovering or slow forward flight, with $v \approx 0$; thus, using the accelerometer as an inclinometer has been shown to be efficient in practice during this flight regime. In fact, it is known that for an ideal thrust controlled aerial vehicle, the measurement of the gravitational direction cannot be directly extracted from accelerometer measurement data [27], [33], [36]. However, VTOL robotic vehicles are subject to secondary aerodynamic forces (e.g., blade flapping and induced drag) that inject low frequency information on the gravitational direction into the accelerometer measurements [27], [33]. It follows that the model $a_{\mathcal{B}} \approx -gR^T e_3$ is an effective model for vector attitude measurement in a wide range of practical systems [13], [24], [25], [32].

Standard implementation and associated coupling issues

Once the approximation $a_{\mathcal{B}} \approx -gR^T e_3$ is made, the standard implementation of the explicit complementary filter (5) consists in defining the innovation term σ_R as [25]

$$\sigma_R \triangleq k_1 u_{\mathcal{B}} \times \hat{u}_{\mathcal{B}} + k_2 \bar{m}_{\mathcal{B}} \times \hat{\bar{m}}_{\mathcal{B}}, \qquad (8)$$

with $k_{1,2}$ positive constant gains, $u_{\mathcal{B}} \triangleq -a_{\mathcal{B}} / g$, $u_{\mathcal{I}} \triangleq e_3$, $\hat{u}_{\mathcal{B}} \triangleq \hat{R}^T u_{\mathcal{I}}$, $\bar{m}_{\mathcal{B}} \triangleq m_{\mathcal{B}} / |m_{\mathcal{I}}|$, $\bar{m}_{\mathcal{I}} \triangleq m_{\mathcal{I}} / |m_{\mathcal{I}}|$, $\hat{\bar{m}}_{\mathcal{B}} \triangleq \hat{R}^T \bar{m}_{\mathcal{I}}$. However, it has been recognized that this standard implementation encounters some coupling issues that are well discussed in [16], [17], [30].

• Magnetic disturbances and bias influence the estimation of roll and pitch angles. In many applications, especially for small-size electric motorized aerial robots, significant magnetic disturbances are almost unavoidable, leading to significant time-varying deterministic error between $m_{\mathcal{B}}$ and $R^T m_{\mathcal{I}}$. This not only leads to large estimation errors of the yaw angle, but also non-negligible errors in the roll and pitch estimation.

• The dynamics of roll, pitch and yaw estimates are highly coupled. This implies that the estimation of the yaw angle strongly affects the estimation of the roll and pitch angles. This issue can be observed by analyzing the linearized system around the system equilibrium. For the sake of simplicity, let us, for instance, neglect the gyro-bias b_ω and the dynamics of the estimated bias $\hat{b}_\omega$ only in this discussion. This supposition in association with Eqs. (5) and (8) yields the following dynamics of the error attitude matrix $\tilde{R} = R\hat{R}^T$

$$\dot{\tilde{R}} = -(k_1 e_3 \times \tilde{R} e_3 + k_2 \bar{m}_{\mathcal{I}} \times \tilde{R} \bar{m}_{\mathcal{I}}) \times \tilde{R} \qquad (9)$$

Consider a first order approximation of $\tilde{R}$ around the equilibrium $\tilde{R} = I_3$ as $\tilde{R} \approx I_3 + x_\times$, with $x \in \mathbb{R}^3$. Note that locally the first, second and third components of x correspond, respectively, to the roll, pitch and yaw error estimates. From Eq. (9) it can easily be verified that

$$\dot{x} \approx \begin{bmatrix} -k_1 - k_2(1-\bar{m}_{\mathcal{I},1}^2) & k_2 \bar{m}_{\mathcal{I},1} \bar{m}_{\mathcal{I},2} & k_2 \bar{m}_{\mathcal{I},1} \bar{m}_{\mathcal{I},3} \\ k_2 \bar{m}_{\mathcal{I},1} \bar{m}_{\mathcal{I},2} & -k_1 - k_2(1-\bar{m}_{\mathcal{I},2}^2) & k_2 \bar{m}_{\mathcal{I},2} \bar{m}_{\mathcal{I},3} \\ k_2 \bar{m}_{\mathcal{I},1} \bar{m}_{\mathcal{I},3} & k_2 \bar{m}_{\mathcal{I},2} \bar{m}_{\mathcal{I},3} & -k_2(1-\bar{m}_{\mathcal{I},3}^2) \end{bmatrix} x \qquad (10)$$

In practice, the gravity vector and the Earth's magnetic field vector (i.e., e_3 and $\bar{m}_{\mathcal{I}}$) can be "ill-conditioned" in the sense that they are very close to each other. In such a case, the third component of $\bar{m}_{\mathcal{I}}$ is dominant in relation to its first and second ones. For example, in France $\bar{m}_{\mathcal{I},3} \approx 0.9$. As a consequence, in view of Eq. (10) the dynamics of the roll and pitch errors (i.e., x_1 and x_2) are strongly coupled with the yaw error dynamics (i.e., x_3).

• On the other hand, the ill-conditioning of the two vectors e_3 and $\bar{m}_{\mathcal{I}}$ may also lead to the impossibility of finding a set of "non-high" gains $\{k_1, k_2\}$ so as to provide the system with fast time response, bearing in mind that high gains may excessively amplify measurement noises. For discussion purposes and without loss of generality, let us, for instance, assume that $\bar{m}_{\mathcal{I},2} \approx 0$ (i.e., $\bar{m}_{\mathcal{I},1}^2 + \bar{m}_{\mathcal{I},3}^2 \approx 1$) and $\bar{m}_{\mathcal{I},3}^2 \gg \bar{m}_{\mathcal{I},1}^2$. Under this approximation, it is straightforward to verify that three poles of System (10) are given by:

$$\begin{cases} \lambda_1 = -(k_1 + k_2) \\ \lambda_2 = -\frac{1}{2}(k_1 + k_2)\left(1 + \sqrt{1 - \frac{4k_1 k_2 \bar{m}_{\mathcal{I},1}^2}{(k_1+k_2)^2}}\right) \\ \lambda_3 = -\frac{1}{2}(k_1 + k_2)\left(1 - \sqrt{1 - \frac{4k_1 k_2 \bar{m}_{\mathcal{I},1}^2}{(k_1+k_2)^2}}\right) \approx -\frac{k_1 k_2 \bar{m}_{\mathcal{I},1}^2}{k_1 + k_2} \end{cases}$$

The pole λ_1 is associated with the pitch dynamics and the poles λ_2 and λ_3 are associated with the coupled roll and yaw dynamics. The less negative pole λ_3, approximated by $-k_1 k_2 \bar{m}_{\mathcal{I},1}^2 / (k_1 + k_2)$, will be very close to zero if k_1 and k_2 are not chosen sufficiently high, since $\bar{m}_{\mathcal{I},1}^2 \ll 1$. This leads to slow time response of the coupled roll and yaw dynamics.

Modified solutions for overcoming coupling issues

Decoupling of input signals to ensure that the roll and pitch estimates are not influenced by magnetic disturbances was considered in [32], [30], [17], [16] and represents an important modification of the standard implementation of the explicit complementary filter (i.e., Observer (5) with σ_R given by (8)) in order to improve the overall quality of the attitude estimate. Let us discuss these strategies.

The solution proposed by Martin and Salaun [30], [32] consists in creating another inertial vector as the cross product of the gravitational direction and the Earth's magnetic field direction, and its associated "artificial" measurement based on the cross product of accelerometer and magnetometer measurements. More precisely, the following vectors are computed

$$v_I \triangleq \frac{u_{\mathcal{I}} \times m_{\mathcal{I}}}{|u_{\mathcal{I}} \times m_{\mathcal{I}}|} \qquad v_B \triangleq \frac{u_{\mathcal{B}} \times m_{\mathcal{B}}}{|u_{\mathcal{B}} \times m_{\mathcal{I}}|} \tag{11}$$

with $u_{\mathcal{B}} \triangleq -a_{\mathcal{B}} / g$, $u_{\mathcal{I}} \triangleq e_3$. Then, the explicit complementary filter (5) can be applied with the innovation term σ_R now defined as (compare to (8))

$$\sigma_R \triangleq k_1 u_{\mathcal{B}} \times \hat{u}_{\mathcal{B}} + k_2 v_{\mathcal{B}} \times \hat{v}_{\mathcal{B}} \tag{12}$$

with $k_{1,2}$ positive constant gains, $\hat{u}_B \triangleq \hat{R}^T u_{\mathcal{I}}$, $\hat{v}_B \triangleq \hat{R}^T v_{\mathcal{I}}$. This solution ensures the decoupling of the roll and pitch estimates from magnetic disturbances locally in the general case [30] and globally in the presence of a constant magnetic bias in the inertial frame (as additionally proved in [17]).

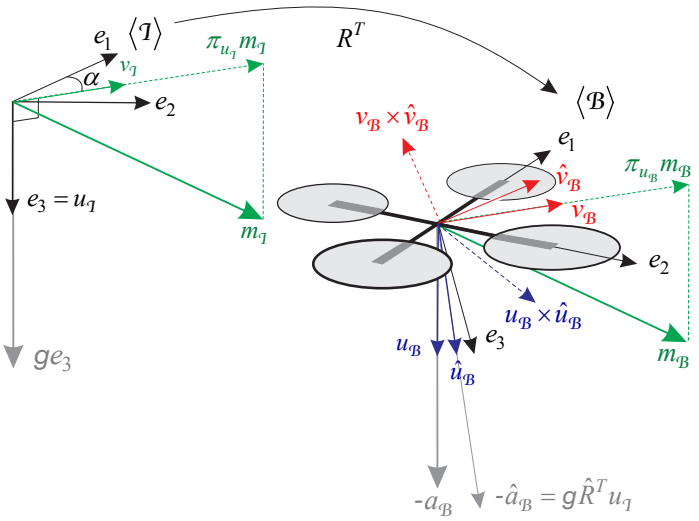


Figure 2 - Vectors involved in the conditioned observer

Inspired by the work of Martin and Salaun, we have proposed another observer termed conditioned observer [16], which still takes the same form as the explicit complementary filter (5) but with the modified innovation terms σ_R and σ_b given by

$$\begin{cases} \sigma_R \triangleq k_1 u_{\mathcal{B}} \times \hat{u}_{\mathcal{B}} + k_2 \left((v_{\mathcal{B}} \times \hat{v}_{\mathcal{B}})^T \hat{u}_{\mathcal{B}} \right) \hat{u}_{\mathcal{B}} \\ \sigma_b \triangleq k_3 u_{\mathcal{B}} \times \hat{u}_{\mathcal{B}} + k_4 v_{\mathcal{B}} \times \hat{v}_{\mathcal{B}} \end{cases} \tag{13}$$

with $k_{1,2,3,4}$ positive constant gains satisfying $k_3 > k_4$, $u_{\mathcal{B}} \triangleq -a_{\mathcal{B}} / g$, $u_{\mathcal{I}} \triangleq e_3$, and (compare to (11) and see figure 2)

$$v_{\mathcal{I}} \triangleq \frac{\pi_{u_{\mathcal{I}}} m_{\mathcal{I}}}{|\pi_{u_{\mathcal{I}}} m_{\mathcal{I}}|} \qquad v_B \triangleq \frac{\pi_{u_{\mathcal{B}}} m_{\mathcal{B}}}{|\pi_{u_{\mathcal{I}}} m_{\mathcal{I}}|} \tag{14}$$

with $\pi_x \triangleq |x|^2 I_3 - xx^T, \forall x \in \mathbb{R}^3$, denoting the projection on the plane orthogonal to x. The conditioned observer ensures the global decoupling of the roll and pitch estimates from magnetic disturbances and also from the dynamics of yaw estimate in the general case. This decoupling property is clearly stronger than that of the previous solution. Moreover, in contrast with the standard implementation of the explicit complementary filter (5), fast convergence rate can still be achieved with non-high gains, even in the case of ill-conditioning of the gravity and Earth's magnetic field directions [16].

GPS–aided attitude observers

Most existing ("classical") attitude observers/filters rely on the small acceleration assumption (i.e., $\dot{v} \ll g$) so that the gravitational direction measurement can be approximated by the accelerometer measurement, as discussed in the previous subsection. For many VTOL vehicles in aggressive motion, however, the vehicle's linear accelerations can be important and can induce large errors in the attitude estimate. This is also the case for fixed-wing aircraft maneuvering in a limited space and making some rapid turns. To deal with strong linear accelerations, a complementary GPS measurement of the linear velocity can be combined with the accelerometer measurement to estimate the vehicle's acceleration and, subsequently, improve the precision of the attitude estimate. In this way, some GPS–aided attitude observers have been proposed recently [31], [15], [38] on the basis of the following differential equations

$$\begin{cases} \dot{v} = ge_3 + Ra_{\mathcal{B}} \\ \dot{R} = RS(\omega) \end{cases} \tag{15}$$

For instance, the cascade attitude observer proposed by Hua [15] consists in, first, estimating the specific acceleration expressed in the inertial frame $\alpha_{\mathcal{I}} \triangleq \dot{v} - ge_3$ and, then, in using this estimated value along with magnetometer measurements to recover the whole attitude estimate on the basis of the explicit complementary filter [25]. More precisely, in order to estimate the specific acceleration $\alpha_{\mathcal{I}}$, the following observer was proposed

$$\begin{cases} \dot{\hat{v}} = k_1 (v - \hat{v}) + ge_3 + Qa_{\mathcal{B}} \\ \dot{Q} = QS(\omega) + k_v (v - \hat{v}) {a_{\mathcal{B}}}^T - k_q \max(0, \|Q\| - \sqrt{3}) Q \end{cases} \tag{16}$$

with k_1, k_v, k_q positive constant gains and $Q \in \mathbb{R}^{3\times 3}$ an auxiliary matrix that is not a rotation matrix. The last term in the expression of $\dot{Q}$ creates a dissipative effect when the Frobenius norm of Q becomes larger than $\sqrt{3}$, allowing it to be driven back to this threshold and thus avoiding numerical drifts of Q. It is shown that the errors $(a_I - Qa_{\mathcal{B}}, v - \hat{v})$ converge to zero [15]. Consequently, one can view either $Qa_{\mathcal{B}} + k_1 (v - \hat{v})$ as the estimate of $\alpha_{\mathcal{I}}$. From here, the author proposed the following attitude observer on the basis of the explicit complementary filter [25]

$$\begin{cases} \dot{\hat{R}} = \hat{R} S(\omega + \sigma_R) \\ \sigma_R \triangleq k_2 m_{\mathcal{B}} \times \hat{R}^T m_{\mathcal{I}} + k_3 a_{\mathcal{B}} \times \hat{R}^T (Qa_{\mathcal{B}} + k_1 (v - \hat{v})) \end{cases} \tag{17}$$

with $k_{2,3}$ positive constant gains. Almost global convergence of the observer is proved. Furthermore, in the special case of constant

accelerations of the vehicle, almost-global asymptotic stability of the observer is achieved.

Invariant Attitude Observers

When the objective consists in combining the estimation of the attitude and the filtering of the linear velocity (and eventually the position), some invariant attitude observers have been proposed recently [31], [15], [38]. The earliest nonlinear invariant GPS–aided attitude observer was proposed by Martin and Salaun [31]. When measurements are not affected by bias, this observer has the form

$$\begin{cases} \dot{\hat{v}} &= k_1(v-\hat{v}) + ge_3 + \hat{R}a_{\mathcal{B}} \\ \dot{\hat{R}} &= \hat{R}S(\omega+\sigma_R) \\ \sigma_R &\triangleq k_2((m_{\mathcal{B}} \times \hat{R}^T m_{\mathcal{I}})^T a_{\mathcal{B}})a_{\mathcal{B}} + k_3 a_{\mathcal{B}} \times \hat{R}^T(v-\hat{v}) \end{cases} \quad (18)$$

with $k_{1,2,3}$ positive constant gains. This defines an invariant observer [5], [20] in the sense that it preserves the (Lie group) invariance properties of System (15) w.r.t. constant velocity translation $v \mapsto v + v_0$ and constant rotation of the body frame $R \mapsto RR_0$. A practical advantage of this solution is the (local) decoupling of the roll and pitch angle estimation from the measurements of the Earth's magnetic field (which may be rather erroneous due to magnetic disturbances). However, only local exponential stability of the estimation error is proven in [31] (based on the linearized estimation error dynamics), under some assumptions on the reference motion (i.e., "smooth trajectory").

Motivated by this result, other GPS–aided attitude invariant observers have been proposed with associated Lyapunov-based convergence and stability analyses [15], [38]. The invariant observer proposed by Hua [15] is given by

$$\begin{cases} \dot{\hat{v}} &= k_1(v-\hat{v}) + ge_3 + \hat{R}a_{\mathcal{B}} \\ \dot{\hat{R}} &= \hat{R}S(\omega+\sigma_R) \\ \sigma_R &\triangleq k_2 m_{\mathcal{B}} \times \hat{R}^T m_{\mathcal{I}} + k_3 a_{\mathcal{B}} \times \hat{R}^T(v-\hat{v}) \end{cases} \quad (19)$$

with $k_{1,2,3}$ positive constant gains. In fact, observer (19) is slightly different from observer (18), which is a simplified version of the observer proposed in [31] suited to the case without gyro biases. The sole difference between observers (18) and (19) lies in the definition of σ, where the term $k_2((m_{\mathcal{B}} \times \hat{R}^T m_{\mathcal{I}})^T a_{\mathcal{B}})a_{\mathcal{B}}$ in (18) is replaced in (19) by $k_2(m_B \times \hat{R}^T m_I)$. Another invariant observer was proposed by Robert and Tayebi [38], which can be rewritten in the following form

$$\begin{cases} \dot{\hat{v}} &= k_1(v-\hat{v}) + ge_3 + \hat{R}a_{\mathcal{B}} + \frac{1}{k_1}\hat{R}S(\sigma_R)a_{\mathcal{B}} \\ \dot{\hat{R}} &= \hat{R}S(\omega+\sigma_R) \\ \sigma_R &\triangleq k_2 m_{\mathcal{B}} \times Rm_{\mathcal{I}} + k_3 a_{\mathcal{B}} \times \hat{R}^T(v-\hat{v}) \end{cases} \quad (20)$$

with $k_{1,2,3}$ positive constant gains. The additional term $(1/k_1)\hat{R}^T S(\sigma)a_{\mathcal{B}}$ involved in the dynamics of $\hat{v}$ in (20) constitutes the difference between observers (20) and (19), allowing the authors to establish simpler Lyapunov based stability and convergence nalyses. The main interest of both studies [15] and [38] is to yield semi-global exponential convergence proofs. Both observers (19) and (20) guarantee the semi-global stability property under a "high-gain"-like condition on k_1, which indicates that the size of the basin of attraction is proportional to k_1 and tends to be almost-global when k_1 tends to infinity. In fact, the "high-gain" condition is only sufficient and simulation results seem to indicate that the basin of attraction does not depend on the value of k_1 (>0). However, the proof of this property remains an open problem. It is worth noting that, contrary to observer (18), all three observers (19), (20) and (16)–(17) do not ensure the (local) decoupling of the estimation of the roll and pitch (Euler) angles from the magnetic measurements. This suggests –as an open problem– the design of an observer that combines the advantages of these observers. For instance, observers (18) and (16)–(17) can be combined, yielding the following attitude observer (in the replacement of (17))

$$\begin{cases} \dot{\hat{R}} &= \hat{R}S(\omega+\sigma_R) \\ \sigma_R &\triangleq k_2((m_{\mathcal{B}} \times \hat{R}^T m_{\mathcal{I}})^T a_{\mathcal{B}})a_{\mathcal{B}} + k_3 a_{\mathcal{B}} \times \hat{R}^T(Qa_{\mathcal{B}} + k_1(v-\hat{v})) \end{cases} \quad (21)$$

with Q the (numerical) solution to System (16). Specifying the stability domain of this observer, however, remains open.

Airspeed-aided attitude observer for fixed-wing UAVs

For fixed-wing UAVs that maneuver in GPS denied environments (e.g., indoor or near to buildings), an alternative solution of attitude estimation based on IMU and improved with GPS data is the use of pressure sensors, such as pitot tubes that measure the magnitude of the airspeed (i.e., the speed of the vehicle relative to the air) as a replacement of GPS velocity measurements. A nonlinear complementary filter/observer of this nature was proposed [23]. Magnetometer is not used in this study since the authors are only interested in roll and pitch estimation, but the incorporation of magnetometer measurements into the observer for additional yaw estimation can be done as described hereafter.

In [23], Mahony et al. consider the case where an aircraft performs a level turn (i.e., constant altitude) with constant turn radius $\rho > 0$ and zero sideslip angle. In this case, the vehicle experiences the centripetal acceleration $a_c \approx \omega \times (\omega \times \rho r)$, with r the unit vector from the aircraft to the turning center. In order to eliminate the dependence on the unknown turn geometry, the approximation $\omega \times \rho r \approx V_{\text{air}}$ is made, so that the centripetal acceleration can be approximately given by $a_c \approx \omega \times V_{\text{air}}$. The airspeed vector Vair is not directly measured, but it can be recovered from the measurement of the norm $|V_{\text{air}}|$ given by the pitot tubes and from the knowledge of the angle-of-attack α as follows

$$V_{\text{air}} = |V_{\text{air}}| \begin{bmatrix} C\alpha \\ 0 \\ S\alpha \end{bmatrix} \quad (22)$$

The linearized dynamics model of the angle-of-attack approximately satisfy

$$\dot{\alpha} = -\frac{c_0}{|V_{\text{air}}|}\alpha + \dot{\theta} + \alpha_0 \quad (23)$$

with c_0 and α_0 constant parameters and $\dot{\theta} \approx \omega_2$. By numerically integrating Eq. (23), the angle-of-attack α can be obtained, which enables the computation of the airspeed vector V_{air} according to Eq. (22) and, subsequently, of the approximated measurement of the centripetal acceleration a $(\approx \quad \times V_{\text{air}})$.

Once the centripetal acceleration ac is computed, the gravitational direction expressed in the body frame can also be obtained from accelerometer readings as

$$u_{\mathcal{B}} = R^T u_I \approx \frac{-(a_{\mathcal{B}} - a_c)}{|a_{\mathcal{B}} - a_c|}$$

with $u_I \triangleq e_3$. Then, the explicit complementary filter (5) can be applied with the innovation term σ_R defined as $\sigma_R \triangleq k_1 u_{\mathcal{B}} \times \hat{u}_{\mathcal{B}}$ with positive constant gain k_1 and $\hat{u}_{\mathcal{B}} \triangleq \hat{R}^T u_I$. Although, several assumptions and approximations are made, the reported experimental results are quite satisfying [23].

The yaw angle may be recovered under the persistent excitation condition [26]. It can also be estimated when magnetometer measurements are involved by using the conditioned observer [16] (i.e., observer (5) with the innovation terms σ_R and σ_b defined by (13)–(14)).

Conclusions

Several attitude estimation techniques –ranging from algebraic vector observations-based attitude determination algorithms to dynamics attitude filtering and estimation methodologies– have been reviewed and commented upon in relation to practical implementation issues. A particular attention is devoted to the applications of the well known nonlinear explicit complementary filter/observer [25] to aerial robotics, using a low-cost and light-weight inertial measurement unit, which can be complemented with a GPS or airspeed sensors. In the case of "weak" linear accelerations, the vector direction estimate of the gravitational direction can be derived from accelerometer measurements with reasonably good accuracy and, thus, the explicit complementary filter can be directly applied. In this case, decoupling of input signals to ensure that the roll and pitch estimates are not disturbed by deviation in the magnetometer measurements represents an important improvement of the basic algorithm. On the other hand, in the case of "strong" linear accelerations, the combination of IMU with GPS-velocity or airspeed measurements allows the overall quality of the attitude estimate to be effectively improved ■

Acknowledgements

This work has been supported by the French Agence Nationale de la Recherche through the ANR ASTRID SCAR project "Sensory Control of Aerial Robots".

References

[1] I. Y. BAR-ITZHACK - *REQUEST: a Recursive QUEST Algorithm for Sequentiel Attitude Determination.* AIAA Journal of Guidance, Control, and Dynamics, 19(5):1034-1038, 1996.
[2] S. P. BHAT, D. S. BERNSTEIN - *A Topological Obstruction to Continuous Global Stabilization of Rotational Motion and the Unwinding Phenomenon.* Systems & Control Letters, 39(1):63-70, 2000.
[3] S. BONNABEL - *Left-Invariant Extended Kalman Iter and Attitude Estimation.* IEEE Conference on Decision and Control, pages 1027-1032, 2007.
[4] S. BONNABEL, P. MARTIN, P. ROUCHON - *Groupe de Lie et observateur non-lineaire*. In Conference Internationale Francophone d'Automatique 2006.
[5] S. BONNABEL, P. MARTIN, P. ROUCHON - *Symmetry-Preserving Observers*. IEEE Transactions on Automatic Control, 53(11):2514-2526, 2008.
[6] S. BONNABEL, P. MARTIN, P. ROUCHON - *Non-linear Symmetry Preserving Observers on Lie Groups*. IEEE Transactions on Automatic Control, 54(7):1709-1713, 2009.
[7] M. LE BORGNE - *Quaternions et controle sur l'espace de rotation*. Technical Report 751, INRIA, 1987.
[8] D. CHOUKROUN - *Novel Methods for Attitude Determination Using Vector Observations*. PhD thesis, Israel Institute of Technology, 2003.
[9] J. L. CRASSIDIS, F. L. MARKLEY, Y. CHENG - *Survey of Nonlinear Attitude Estimation Methods.* AIAA Journal of Guidance, Control, and Dynamics, 30(1):12-28, 2007.
[10] P. B. DAVENPORT - *A Vector Approach to the Algebra of Rotations with Applications*. Technical Report TN D-4696, NASA, 1968.
[11] J. L. FARRELL - *Attitude Determination by Kalman Itering: Volume I.* Technical Report CR598, NASA, 1964.
[12] H. F. GRIP, T. I. FOSSEN, T. A. JOHANSEN, A. SABERI - *Attitude Estimation Using Biased Gyro and Vector Measurements with Time-Varying Reference Vectors*. IEEE Transactions on Automatic Control, 57(5):1332-1338, 2012.
[13] T. HAMEL, R. MAHONY - *Attitude Estimation on SO(3) Based on Direct Inertial Measurements*. IEEE Conference on Robotics and Automation, pages 2170-2175, 2006.
[14] W. R. HAMILTON - Letter from Sir William R. Hamilton to John T. Graves, Esq. on Quaternions, 1843.
[15] M.-D. HUA - *Attitude Estimation for Accelerated Vehicles Using GPS/INS Measurements.* Control Engineering Practice, 18(7):723-732, 2010.
[16] M.-D. HUA, G. DUCARD, T. HAMEL, R. MAHONY, K. RUDIN - *Implementation of a Nonlinear Attitude Estimator for Aerial Robotic Vehicles.* IEEE Transactions on Control Systems Technology, 2013. DOI:10.1109/TCST.2013.2251635.
[17] M.-D. HUA, K. RUDIN, G. DUCARD, T. HAMEL, R. MAHONY - *Nonlinear Attitude Estimation with Measurement Decoupling and Anti-Windup Gyro-Bias Compensation.* IFAC World Congress, pages 2972-2978, 2011.
[18] K. J. JENSEN - *Generalized Nonlinear Complementary Attitude Iter.* AIAA Journal of Guidance, Control, and Dynamics, 34(5):1588-1592, 2011.
[19] C. LAGEMAN, J. TRUMPF, R. MAHONY - *Gradient-Like Observers for Invariant Dynamics on a Lie Group*. IEEE Trans. on Automatic Control, 55(2):367-377, 2010.
[20] C. LAGEMAN, J. TRUMPF, R. MAHONY - *Gradient-Like Observers for Invariant Dynamics on a Lie Group.* IEEE Transactions on Automatic Control, 55(2):367-377, 2010.

[21] E. LEERTS, F. MARKLEY, M. SHUSTER - *Kalman Itering for Spacecraft Attitude Estimation.* AIAA Journal of Guidance, Control, Navigation, 5:417-429, 1982.
[22] D. LUENBERGER - *An Introduction to Observers.* IEEE Transactions on Automatic Control, 16(6):596-602, 1971.
[23] R. MAHONY, M. EUSTON, J. KIN, P. COOTE, T. HAMEL - *A Non-Linear Observer for Attitude Estimation of Axed-Wing Unmanned Aerial Vehicle without GPS Measurements.* Transactions of the Institute of Measurement and Control, 33(6):699-717, 2011.
[24] R. MAHONY, T. HAMEL, J.-M. PIMLIN - *Complementary Iter Design on the Special Orthogonal Group SO(3).* IEEE Conference on Decision and Control, pages 1477-1484, 2005.
[25] R. MAHONY, T. HAMEL, J.-M. PIMLIN - *Nonlinear Complementary Iters on the Special Orthogonal Group*. IEEE Transactions on Automatic Control, 53(5):1203-1218, 2008.
[26] R. MAHONY, T. HAMEL, J. TRUMPF, C. LAGEMAN - *Nonlinear Attitude Observers on SO(3) for Complementary and Compatible Measurements: A Theoretical Study*. In IEEE Conference on Decision and Control, pages 6407-6412, 2009.
[27] R. MAHONY, V. KUMAR, P. CORKE - *Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor.* IEEE Robotics & Automation Magazine, 19(3):20-32, 2012.
[28] F. L. MARKLEY - *Attitude Error Representations for Kalman Itering*. AIAA Journal of Guidance, Control, and Dynamics, 26(2):311-317, 2003.
[29] F. L. MARKLEY, D. MORTARI - *Quaternion Attitude Estimation Using Vector Observations*. The Journal of the Astronautical Sciences, 48(2, 3):359-380, 2000.
[30] P. MARTIN, E. SALAUN - *Invariant Observers for Attitude and Heading Estimation from Low-Cost Inertial and Magnetic Sensors*. IEEE Conference on Decision and Control, pages 1039-1045, 2007.
[31] P. MARTIN, E. SALAUN - *An Invariant Observer for Earth-Velocity-Aided Attitude Heading Reference Systems.* IFAC World Congress, pages 9857-9864, 2008.
[32] P. MARTIN, E. SALAUN - *Design and Implementation of a Low-Cost Observer-Based Attitude and Heading Reference System*. 18(7):712-722, 2010.
[33] P. MARTIN, E. SALAUN - *The True Role of Accelerometer Feedback in Quadrotor Control.* IEEE Conference on Robotics and Automation, pages 1623-1629, 2010.
[34] A. MAYER - *Rotations and their Algebra.* SIAM Journal on Control and Optimization, 2(2):77-122, 1960.
[35] H. NIJMEIJER, T. I. FOSSEN - *New Directions in Nonlinear Observer Design*, volume 244 of Lecture Notes in Control and Information Sciences. Springer, 1999.
[36] S. OMARI, M.-D. HUA, G. DUCARD, T. HAMEL - *Nonlinear Control of VTOL UAVs Incorporating Flapping Dynamics.* International Conference on Intelligent Robots and Systems (IROS), 2013.
[37] B. PALAIS, R. PALAIS - *Euler's Fixed Point Theorem: The Axis of a Rotation.* Journal of Fixed Point Theory and Applications, 2:215-220, 2007.
[38] A. ROBERTS, A. TAYEBI - *On the Attitude Estimation of Accelerating Rigid-Bodies Using GPS and IMU Measurements.* IEEE Conference on Decision and Control, pages 8088-8093, 2011.
[39] A. C. ROBINSON - *On the use of Quaternions in Simulations of Rigid-Body Motion*. Technical Report 58-17, Wright Air Development Center, 1958.
[40] S. SALCUDEAN - *A Globally Convergent Angular Velocity Observer for Rigid Body Motion.* IEEE Transactions on Automatic Control, 36(12):1493-1497, 1991.
[41] M. D. SHUSTER - *Approximate Algorithms for Fast Optimal Attitude Computation.* AIAA Guidance and Control Conference, pages 88-95, 1978.
[42] M. D. SHUSTER - *Maximum Likelihood Estimation of Spacecraft Attitude*. The Journal of the Astronautical Sciences, 37(1):79-88, 1989.
[43] M. D. SHUSTER - *Kalman Itering of Spacecraft Attitude and the QUEST Model.* The Journal of the Astronautical Sciences, 38(3):377-393, 1990.
[44] M. D. SHUSTER - *The QUEST for Better Attitudes.* The Journal of the Astronautical Sciences, 54(3,4):657-683, 2006.
[45] J. STUELPNAGEL - *On the Parametrization of the Three-Dimensional Rotation Group.* SIAM Journal on Control and Optimization, 6(4):422-430, 1964.
[46] A. TAYEBI, S. MCGILVRAY, A. ROBERTS, M. MOALLEM - *Attitude Estimation and Stabilization of a Rigid Body Using Low-Cost Sensors.* In IEEE Conference on Decision and Control, pages 6424-6429, 2007.
[47] J. THIENEL, R. M. SANNER - *A Coupled Nonlinear Spacecraft Attitude Controller and Observer with an Unknown Constant Gyro Bias and Gyro Noise.* IEEE Transactions on Automatic Control, 48(11):2011-2015, 2003.
[48] J. TRUMPF, R. MAHONY, T. HAMEL, C. LAGEMAN - *Analysis of Non-Linear Attitude Observers for Time-Varying Reference Measurements.* IEEE Transactions on Automatic Control, 57:2789-2800, 2012.
[49] J. F. VASCONCELOS, R. CUNHA, C. SILVESTRE, P. OLIVEIRA - *A Landmark Based Nonlinear Observer for Attitude and Position Estimation with Bias Compensation.* IFAC World Congress, pages 3446-3451, 2008.
[50] J. F. VASCONCELOS, C. SILVESTRE, P. OLIVEIRA - *A Nonlinear GPS/IMU Based Observer for Rigid Body Attitude and Position Estimation*. IEEE Conference on Decision and Control, pages 1255-1260, 2008.
[51] B. VIK, T. FOSSEN - *A Nonlinear Observer for GPS and INS Integration*. IEEE Conference on Decision and Control, pages 2956-2961, 2001.

Acronyms

IMU	(Inertial Measurement Unit)
GPS	(Global Positioning System)
MEMS	(MicroElectroMechanical systems)
VTOL	(Vertical Take-Off and Landing)
UAV	(Unmanned Aerial Vehicle)
QUEST	(QUaternion ESTimator)
SVD	(Singular Value Decomposition)
FOAM	(Fast Optimal Attitude Matrix)
ESOQ	(Estimator of the Optimal Quaternion)

AUTHORS

Minh-Duc Hua graduated from Ecole Polytechnique, France, in 2006, and received his Ph.D. from the University of Nice-Sophia Antipolis, France, in 2009. He spent two years as a postdoctoral researcher at I3S UNS-CNRS, France. He is currently researcher of the French National Centre for Scientific Research (CNRS) at the ISIR laboratory of the University Pierre and Marie Curie (UPMC), France. His research interests include nonlinear control theory, estimation and teleoperation with applications to autonomous mobile robots such as UAVs and AUVs.

Guillaume Ducard received his Master degree in Electrical Engineering from ETH Zurich in 2004. He completed his doctoral work in 2007 and his two-year post doc in 2009 both at the ETH Zurich in the field of unmanned aerial vehicles. He is currently researching at I3S UNS-CNRS Sophia Antipolis, France, in nonlinear control and estimation, fault-tolerant flight control and guidance systems for UAVs, and formation flight control.

Tarek Hamel is Professor at the University of Nice Sophia Antipolis since 2003. He conducted his Ph.D. research at the University of Technologie of Compiègne (UTC), France, and received his doctorate degree in Robotics from the UTC in 1996. After two years as a research assistant at the (UTC), he joined the Centre d'Etudes de Mécanique d'Ile de France in 1997 as an associate professor. His research interests include nonlinear control theory, estimation and vision-based control with applications to Unmanned Aerial Vehicles. He is currently Associate Editor for IEEE Transactions on Robotics and for Control Engineering Practice.

Robert Mahony received the B.Sc. degree in applied mathematics and geology and the Ph.D. degree in systems engineering from the Australian National University (ANU), Canberra, in 1989 and 1995, respectively. He is currently a Professor in the Department of Engineering, ANU. He worked as a Marine Seismic Geophysicist and an Industrial Research Scientist before completing a Postdoctoral Fellowship in France and a Logan Fellowship at Monash University, Victoria, Australia. He has held his post at ANU since 2001. His research interests are in non-linear control theory with applications in robotics, geometric optimization techniques, and systems theory.

N. R. J. Lawrance
(Australian Centre for Field Robotics, University of Sydney)
J. J. Acevedo
(University of Sevilla)
J. J. Chung, J. L. Nguyen, D. Wilson, S. Sukkarieh
(Australian Centre for Field Robotics, University of Sydney)

E-mail : n.lawrance@acfr.usyd.edu.au

DOI : 10.12762/2014.AL08-05

Long Endurance Autonomous Flight for Unmanned Aerial Vehicles

This paper presents a summary of research performed at the University of Sydney towards extending the flight duration of fixed-wing unmanned aerial vehicles. A historical context to extended flight is provided and particular attention is paid to research in autonomous soaring and aerial refueling. Autonomous soaring presents a unique set of challenges whereby an aircraft must autonomously identify sources of energy in the wind field and generate trajectories to exploit these conditions to collect energy. The basic mechanisms of soaring flight are examined and methods for generating energy gaining trajectories for exploration, information gathering and patrolling missions with multiple aircraft are detailed. Aerial refueling represents a complementary approach for extending flight duration, and the challenges and current efforts towards autonomous refueling between small aircraft are also detailed.

Introduction

Whilst Unmanned Aerial Vehicles (UAVs) have become increasingly capable platforms used in a wide variety of applications, most are still limited in their endurance by the necessity for on-board energy storage for propulsion. However, recent research has aimed to address this issue by examining methods for extending flight duration by collecting energy during flight. There are two distinct approaches to this problem. The first is the direct and ongoing capture of energy from the vehicle's surrounding environment, such as soaring in wind or collecting solar energy using solar panels. The second is the deliberate resupply of energy from other vehicles using aerial refueling or recharging.

Soaring

Soaring is the process of exploiting wind to collect energy. Soaring was discovered when some birds were noted to be capable of flying for extended periods of time without flapping their wings and seemingly without losing airspeed or altitude. Early aerodynamic research had shown that energy must be lost to drag by any object moving through a fluid. Thus, it was determined that since the birds were not directly expending energy they must be capturing energy from the wind [49, 52, 2]. This process is known as soaring and there are two primary methods for energy capture in wind, static soaring and dynamic soaring. Static soaring is the process of flying through air that is rising relative to the ground. This method is utilized by both birds and manned gliders where there are naturally occurring sources of rising air (thermals) [60, 50]. Thermals occur when an area of the ground is heated (usually by the sun) to a warmer temperature than surrounding areas. The warm air is less dense and rises with respect to the cooler surrounding air, and an aircraft that flies in the rising air will collect energy (illustrated in figure 1). Thermals are favoured by both birds and human glider pilots because they are relatively common and easy to utilise for energy gain [51, 57].

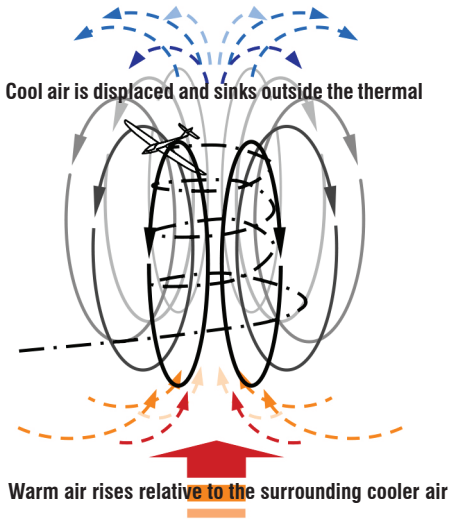


Figure 1 - Static soaring in a thermal

Dynamic soaring utilises trajectories through distributions of wind speed (wind shear) to obtain increased kinetic energy (see figure 2). This is often performed cyclically, with the energy gained in each cycle being used to travel before starting the next cycle. This method was originally discovered being used by birds such as albatrosses over the ocean [24, 62, 69]. Dynamic soaring generally requires good knowledge of the wind field to calculate trajectories which result in energy gain. Naturally occurring sources of wind shear are boundary layers which occur over surfaces (such as the ground or ocean), shear generated by flow around geographic obstacles, and meteorological shear.

Early research in soaring focused on how birds identified and used sources of energy, the amount of energy that could be obtained, and how they integrated soaring behaviours with their needs for travelling and foraging flight [63, 55]. Parallel research for manned aircraft has focused mainly on static soaring which is the primary energy capture method for manned gliders. This led to the development of relatively simple algorithms such as the speed-to-fly rules [39, 40] for cross country gliding. These rules are used to determine when a vehicle should utilise a thermal and when to travel to maximise overall average speed based on an estimate of thermal strengths.

Whilst the bird and manned glider problems have received significant attention, it is only in recent years that soaring with UAVs has been addressed as a research problem. The earliest research attempted to imitate the behavior of manned glider pilots by utilising simple gliding rules (such as estimating the best orbit radius for a thermal) for cross-country flights in autonomous UAVs [4, 5, 3]. Extensions to this work culminated in the demonstration of a fully autonomous glider which flew for over 4 hours on a 97 km round trip flight unofficially setting a new soaring record [18]. Dynamic soaring has also been considered but due to the difficulty of testing has mainly been limited to simulation. Previous work used off-line numerical optimization techniques to calculate the wind strengths required for feasible dynamic soaring and showed that shear layers over the ocean should contain sufficient energy to provide continuous or assisted flight for small (< 10 kg) UAVs [70, 71, 37, 17]. Further work examined on-line reactive strategies for soaring in shear [8, 7, 32] and turbulent fields [29, 15]. Of further interest is the application of machine learning to the soaring problem. Reinforcement learning seems like a natural choice for this type of problem due to the inherent goal of balancing exploration and exploitation, but previous implementations suffered from issues of state space complexity and slow learning rates [64, 65, 28].

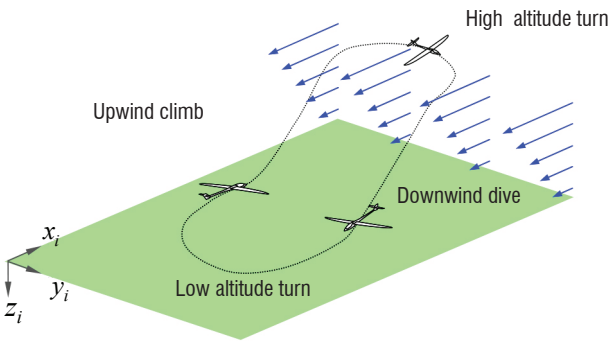


Figure 2 - Dynamic soaring in wind shear

Autonomous aerial refueling

An alternative to capturing energy from the environment is the deliberate resupply of an aerial vehicle during flight. Aerial refueling has been used to extend the mission duration of manned aircraft for decades. This has been almost exclusively within the military space where two strategies are employed. In probe and drogue refueling, a drogue is unwound behind a tanker and a pilot navigates a receiver's probe to contact. Boom and receptacle refueling requires a pilot to station-keep relative to a tanker aircraft while a human operator manually navigates an aerodynamically controlled rigid boom to contact with the receiver's receptacle. Due to the close proximity operations required by these methods, the procedure is inherently dangerous and requires significant pilot training, practice and skill.

With the recent surge in the number of operational UAVs, automating this procedure is the natural next step and is not limited to the military space, or even large UAVs. To date, completely autonomous aerial refueling (AAR) between two UAVs has not been demonstrated. Although an AAR procedure comprises many phases, the barrier to success is generally accepted as being the sensing and navigation challenges surrounding tight formation flight. Here, separation is defined as being less than one wingspan [56] and can translate to only a few meters in some cases.

Work on tight formation relative navigation has focused on utilising air-to-air relative observations to obtain an accurate and timely relative state estimate. Vision-only techniques have included active visual contours [16], silhouette based techniques [27], template matching [44] and feature extraction [41, 61]. Other approaches have fused the vision measurements with inertial and GPS sensors from one or more aircraft in loosely-coupled [43, 21] and tightly coupled arrangements [47, 20, 66]. Experimental validation has mostly occurred in simulations of varying fidelity with notable exceptions being rendezvous experiments [48], GPS-based loose formation flights [22] and the first closed-loop, vision-only loose formation flight [25].

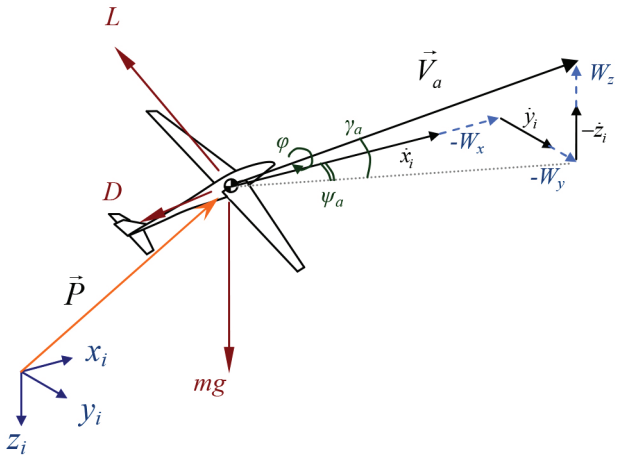


Figure 3 - Air-relative velocity and applied forces for a gliding aircraft. The air relative quantities represent the motion of the vehicle with respect to a stationary air frame, which is actually moving through inertial space due to the wind.

Soaring

Dynamics of soaring flight

The conditions required for energy-gain flight can be identified by analysing the equations of motion of an aircraft in wind. This section briefly describes a dynamic model for a soaring aircraft and examines the mechanisms of soaring through analysis of the resulting energy equations.

Dynamic model

The dynamic model used in the analysis and simulation for this work is an aerodynamic point mass model. The applied forces are the aerodynamic force (decomposed into lift, L, and drag, D) and the weight force (mg). The aerodynamic force is a function of the motion of the vehicle relative to the surrounding air and the physical properties of the aircraft (shape, size, surface properties). Body force due to sideslip is not considered. Weight is the force due to gravity and is directed down (a flat Earth model is assumed due to the relatively small scale of the aircraft and flight paths). Thus there are two important frames of reference: an inertial frame which is fixed with respect to the ground, and the air-relative frame which is the aircraft motion relative to the surrounding air. Figure 3 illustrates the forces acting on a gliding aircraft in wind.

Wind is defined in inertial space and represents the motion of the air relative to the ground-fixed inertial frame. The air-relative velocity vector represents the motion of the vehicle with respect to the surrounding air by treating the wind field as a stationary frame. Thus the airspeed is the magnitude, the air-relative climb γ_a is the vertical angle and ψ_a is the heading angle of this air-relative motion vector. The bank angle ϕ is the rotation of the lift vector around the velocity vector. The air-relative to inertial transformation matrix is denoted C_a^i and is made up of the standard rotation transformation matrices such that $C_a^i = L_z(\psi_a) L_y(\gamma_a) L_x(\phi)$.

The air-relative velocity can be described in terms of the airspeed V_a, heading ψ_a and air-relative climb angle γ_a:

$$\overrightarrow{V_a} = \mathrm{C}_a \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V_a \cos\gamma_a \cos\psi_a \\ V_a \cos\gamma_a \sin\psi_a \\ {}_a \sin\ {}_a \end{bmatrix} \tag{1}$$

The drag coefficient is estimated using the common approximation where the effective drag coefficient (C_D) is the sum of parasitic $(C_{D,0})$ and lift-induced $(C_{D,i})$ drag components [6]. Induced drag is a function of the lift coefficient C_L, aspect ratio $\mathcal{AR}$ and efficiency factor ε.

$$C_D = C_{D,0} + \frac{C_L^2}{\pi \mathcal{AR} \varepsilon} \tag{2}$$

Consider the case of locally spatially-fixed linear wind gradients. Let Jw be the spatial wind gradients at a particular location,

$$\mathrm{J_w} = \begin{bmatrix} \frac{\partial \mathrm{W_x}}{\partial \mathrm{x}} & \frac{\partial \mathrm{W_x}}{\partial \mathrm{y}} & \frac{\partial \mathrm{W_x}}{\partial \mathrm{z}} \\ \frac{\partial \mathrm{W_y}}{\partial \mathrm{x}} & \frac{\partial \mathrm{W_y}}{\partial \mathrm{y}} & \frac{\partial \mathrm{W_y}}{\partial \mathrm{z}} \\ \frac{\partial \mathrm{W_z}}{\partial \mathrm{x}} & \frac{\partial \mathrm{W_z}}{\partial \mathrm{y}} & \frac{\partial \mathrm{W_z}}{\partial \mathrm{z}} \end{bmatrix} \tag{3}$$

By summing the applied forces (lift, drag, and weight) in the inertial frame and differentiating the velocities for acceleration, a set of dynamic equations can be obtained which describe the motion of a gliding aircraft. Assuming that roll rate $(d\varphi / dt)$ is directly controlled, the system can be solved by specifying climb angle rate $(d\gamma_a / dt)$ or specific lift (L/m)as a control input. Physical limitations mean that the maximum specific lift is limited by maximum lift coefficient $(C_{L,max})$ and load factor constraints (n_{min}, n_{max}). In such cases, the lift is specified and (6) returns the climb angle rate. The resulting equations for the system are shown in (5)–(9), where $\dot{\vec{P}}$ is the velocity of the vehicle in the inertial frame.

Further information and full derivation of the equations can be found in [33].

$$\overrightarrow{V_w} = \begin{bmatrix} cos\psi_a \, sin\gamma_a \\ sin\psi_a \, sin\gamma_a \\ cos\gamma_a \end{bmatrix}^T \mathrm{J_w} \dot{\vec{P}} \tag{4}$$

$$\frac{L}{m} = \frac{1}{cos\phi}\left(V_a \frac{d\gamma_a}{dt} + g\, cos\gamma_a - \overrightarrow{V_w} \right) \tag{5}$$

$$\frac{d\gamma_a}{dt} = \frac{1}{V_a}\left(\frac{L}{m} cos\phi - g\, cos\gamma_a + \overrightarrow{V_w} \right) \tag{6}$$

$$\frac{D}{m} = \frac{1}{2}\rho V_a^2 S C_{D,0} + \frac{L^2}{\frac{1}{2}\rho V_a^2 S \pi \quad \varepsilon} \tag{7}$$

$$\frac{dV_a}{dt} = \frac{-D}{m} - g\, sin\gamma_a - \hat{\vec{V}}_a \mathrm{J_w} \dot{\vec{P}}^{\mathcal{AR}} \tag{8}$$

$$\frac{d\psi_a}{dt} = \frac{1}{V_a\, cos\gamma_a}\left(\frac{L}{m} sin\phi + \left[sin\psi_a, -cos\psi_a, 0 \right] \mathrm{J_w} \dot{\vec{P}} \right) \tag{9}$$

These equations can be integrated numerically to simulate gliding flight in wind. This model assumes that a controller is able to track a specified roll rate and either pitch rate or lift coefficient. The simulations used in the following sections are based on a remote control RnR SBXC scale glider. Relevant parameters are listed in table 2.

Soaring energy

The dynamic model can be examined in terms of the energy gained or lost to give an understanding of how wind contributes to the overall energy of the platform. The energy of a point mass can be described as the sum of gravitational potential and kinetic energy. We define the air-relative energy $E_a$as the aerodynamic energy of the vehicle with respect to the surrounding air (treating the air as a local inertial frame with respect to the aircraft). Taking the time derivative of the air-relative energy and substituting the airspeed acceleration (equation (8)) yields the overall specific power.

$$\frac{\dot{E}_a}{m} = -V_a \frac{D}{m} - gW_z - \overrightarrow{V_a}^T \mathrm{J_w} \overrightarrow{V_w} \tag{10}$$

This equation illustrates how a gliding aircraft can gain or lose air-relative energy from a wind field. The first term is the power loss due to drag. This is always an energy loss term since the airspeed must be greater than zero. The second term is the static soaring term representing energy gained or lost from vertical wind. The third term is the dynamic soaring term and represents energy gained or lost due

to wind gradients and is affected by airspeed, climb angle and wind gradients. Equation (10) shows that energy gained or lost from wind shear is proportional to the airspeed. In general terms, increased airspeed increases energy capture or loss rate and allows energy capture from lower magnitude wind gradients.

These equations can be used to identify important conditions for gliding flight, such as the optimal airspeed and climb angle for minimum sink and maximum range. Further, the optimal conditions for energy gain in shear can be calculated when the magnitude of the wind gradient in known. These identities are beyond the scope of this article, but can be found in [31].

Exploration of wind fields

An interesting problem in soaring is that of a vehicle attempting to simultaneously explore and exploit an unknown wind field using soaring. This introduces a requirement for on-line mapping as the vehicle must now also attempt to create a map of the field whilst using that map to generate feasible and energy-gaining paths. This task is complicated by the fact that generally the wind cannot be remotely observed by a small UAV with common on-board sensors. Further, wind fields vary spatially and temporally so observations are only locally valid for a limited period of time.

We consider the case of a UAV with the ability to estimate the local wind vector using an air data sensor and an inertial sensor. We assume that the air data sensor measures the speed and direction of airflow relative to the centre of mass of the aircraft, and the inertial sensor measures the acceleration and speed of the vehicle in a fixed inertial frame. Whilst airspeed and inertial measurement units are common on UAVs, air angle measurements are not. One method for estimating air angles is an alpha-beta vane system which records the wind direction around two perpendicular axes each using a lightweight wind vane and potentiometer [23]. An alternative system is a multi-hole pressure sensor which estimates the air angles by calculating the pressure at different orientations to the wind and solving for the wind direction [59]. It is difficult to determine the accuracy required from these types of sensor for accurate wind reconstruction, and this is part of ongoing work. The collection of truth data is also difficult, though some work had been performed towards quantifying the accuracy available from typical sensors available on small UAVs [30].

Gaussian process mapping and control-sampled planning

Our earliest attempt at solving this problem was to use Gaussian Process (GP) regression, a non-parametric regression technique, for wind mapping to avoid the need for explicit wind feature models. The advantage of using a GP to generate the wind map is that the GP provides both the mean estimate and a variance estimate which can be used to identify the uncertain regions of the map. The exploration/exploitation trade-off is managed by weighting a utility function such that when platform energy is low the utility weights towards known areas (low variance in the GP) with high energy gain (exploitation action) and when platform energy is high the utility weights towards unknown areas (exploration action). An overview of the system is provided in figure 4.

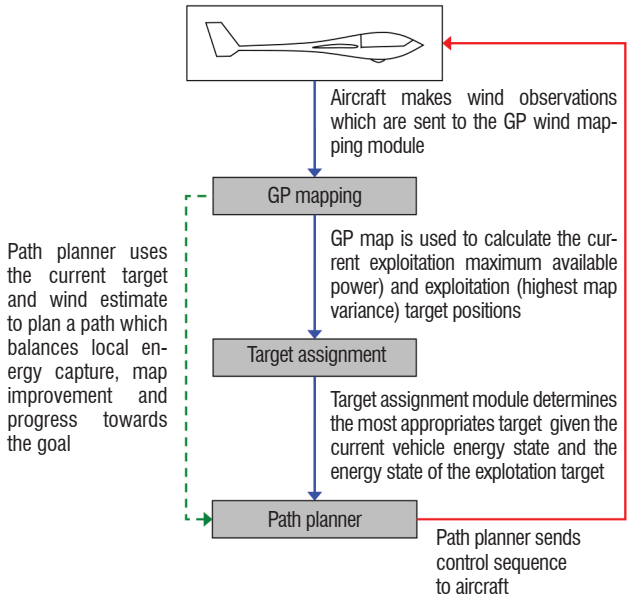


Figure 4 - System overview of simultaneous exploration and exploitation path planning architecture for a gliding UAV [34].

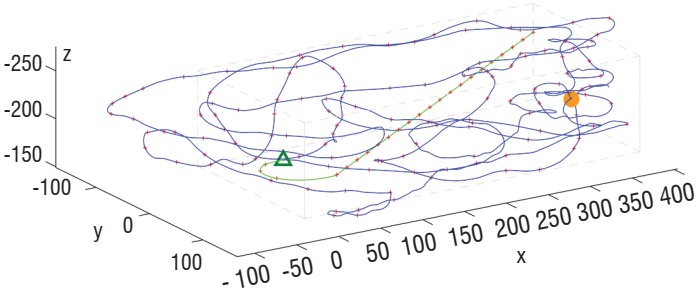


Figure 5 - Single thermal exploration at $t = 500$ s. Autonomous soaring flight starts at the green triangle. There is a single thermal bubble with core vertical wind speed of 3 m/s illustrated by a filled circle [34].

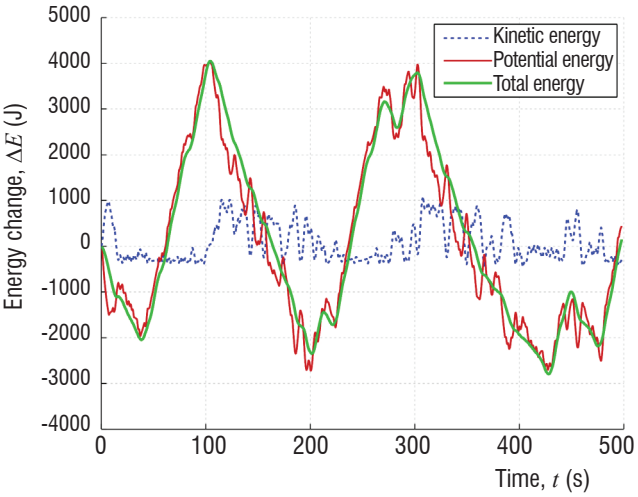


Figure 6 - Energy change during single thermal exploration [34]

Planning

Plans are generated by a control-space sampling planner, that selects from a number of control actions (in this case three pitch and three roll actions), forward simulates the resulting trajectories using the mean estimate from the GP wind map for a given search horizon, and selects the resulting path which yields the highest estimated utility. The utility is a weighted sum of the total estimated energy change over the segment, a navigation reward which estimates progress towards a goal, and an exploration reward which 'optimistically' estimates the amount of energy available at the projected sample locations, where the 'optimism' is a function of the estimated map variance. The current target is selected during each planning cycle based on the energy of the aircraft and is either an exploration target (the map location with maximum estimated variance) or an exploitation target (the position at which maximum power gain was recorded).

Simulation results

The planner showed that it was capable of generating paths that continually improved the map for a specified search region whilst utilising energy sources found during the flight. Figure 5 shows the results of a single simulation where the aircraft is tasked with exploring a three-dimensional rectangular region which contains a single thermal. After approximately 40 s the aircraft finds the thermal and then repeats a process of alternately travelling through the field to reduce variance and using the known thermal to collect energy. The time history of the vehicle energy is shown in figure 6. Further details and additional results are available in [34]. This method showed that a relatively simple control strategy with good knowledge of the vehicle motion model can map and explore a wind field while using energy found during exploration. This was further extended in [35] to account for temporally varying wind fields.

Reinforcement learning for exploration and exploitation management

An alternative or complementary approach to the soaring problem is to pose it as a reinforcement learning (RL) problem where the glider agent must learn the best control action to take given its current state in the wind field. The main advantage of this formulation is that one can elegantly combine the goals of exploring and exploiting the wind field in the RL framework. Further, this does not require explicit identification of soaring behaviours; allowing the system to learn behaviours based on the wind field experienced and resulting energy changes should allow the system to find new soaring behaviours in unknown wind fields.

Reinforcement learning

Standard value-function-based RL control algorithms such as SARSA (λ) [54] learn the value, $Q(s,a)$, of taking a particular action, a, whilst in a particular state, s, by observing the immediate state transition reward, r, and updating the estimated value function according to the following backup equation :

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\delta_t e_t(s,a) \tag{11}$$

where α is the learning step size, the temporal difference iscomputed as,

$$\delta_t = r_{t+1} + \Gamma Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t) \tag{12}$$

and the eligibility trace is given by,

$$e_t(s,a) = \begin{cases} 1 & if\ (s,a) = (s_t,a_t) \\ \Gamma\lambda e_{t-1}(s,a) & otherwise \end{cases} \tag{13}$$

for all (s,a). The two discount factors, Γ and λ, control the contribution of the current reward to the expected return and the value of state-actions previously visited in the trajectory history, respectively. The interested reader is directed to [58] for a full description of SARSA (λ) and other RL algorithms.

The autonomous soaring problem lends itself naturally to this learning framework since it can be considered as a policy learning problem with a well-defined reward (platform energy). Furthermore, the eligibility trace allows credit assignment along the state-action history, promoting the learning of long and potentially complex trajectories, which we expect as a feature of successful soaring policies. However, the state-action space in the soaring problem is continuous, whereas traditional SARSA (λ) deals only with discrete spaces. Thus, value function approximation must be applied to extend SARSA(λ) to handle this. We propose using a GP model to approximate the state-action value function since the GP not only provides an estimate of the mean, it also computes a measure of the uncertainty in the form of a variance. Specifically, the potential reduction in uncertainty (or information gain) from future actions can be used to quantify their exploration utility.

Information measure

The GP variance represents a bounding volume around the estimated function surface and the change in this volume over successive observations can be defined as the information gain of taking those observations [10]. The variance volume can be computed as the integral of the GP covariance function over the state-action space, which has the dimensions $[x_1,...,x_n] = [s_1,...,s_u,a_1,...,a_v]$.

$$V_{bound_t} = \int_{x_{n_a}}^{x_{n_b}} \cdots \int_{x_{1_a}}^{x_{1_b}} cov\left([x_1,...,x_n] \mid X_t\right) dx_1 \ldots dx_n \tag{14}$$

$$I_{gain} = V_{bound_t} - V_{bound_{t+1}} \tag{15}$$

The GP training set X_t consists of the observed $n-$ dimensional state-action pairs, furthermore, $X_{t+1} = X_t \bigcup x_{t+1}$. Given an integrable covariance function, an analytical solution to (14) and subsequently (15) can be found, see [10] for the full solution for the squared exponential covariance function. Drawing inspiration from the eligibility trace for the value function, we define an information value that encapsulates the discounted sum of the information gain from all future state-action observations simulated forward from the next proposed action. Figure 7 gives a graphical representation of the rollout information value calculation :

$$I_{a_{total}} = I_{gain_0} + \Gamma_r I_{gain_1} + \Gamma_r^2 I_{gain_2} + .. + \Gamma_r^p I_{gain_p} \tag{16}$$

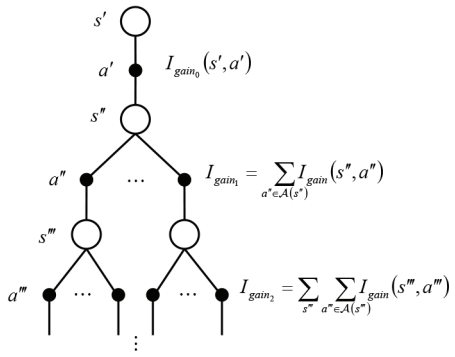


Figure 7 - The rollout method introduced in [10]. The information gain of each rollout level considers all the reachable state-actions at that level. The total information gain is the discounted sum of the information gain of each level.

Action selection

Given the state-action (exploitation) value estimated from the GP model and the information (exploration) value calculated from (16), it is necessary to consolidate the two competing objectives into a single control policy. Prior work [12] has investigated the potential for applying a dynamic scaling of exploration and exploitation metrics according to available platform energy, and this has been adapted in [11] to a 2D simulation of a glider learning to soar in a wind field containing a thermal updraft and a wind shear region. The objective function used normalises both the state-action value and information value and combined the two using a dynamic weighting factor,

$$\hat{Q}_t = \frac{\bar{Q}_t}{max|\bar{Q}_t|} \qquad \hat{I}_t = \frac{I_{t_{total}}}{max|I_{t_{total}}|}$$

$$J_t = \hat{Q}_t + max\left(0, min\left(1, \frac{E_t}{E_{max}}\right)\right)\hat{I}_t \tag{17}$$

where E_t is the current platform energy and E_{max} is the maximum achievable platform energy (corresponding to a maximum speed at a maximum altitude).

Simulation results

The objective function (17) was tested as the sampling policy in the SARSA(λ) learning algorithm with GP value function approximation for a 3D 6DOF glider simulation with a single thermal updraft in the wind field. It was assumed that the thermal centre location was known and so the learning state dimension could be reduced to $\mathbf{s} = [r_{therm}, \psi_{therm}, v_a]$, the relative distance to the thermal centre, the bearing to the thermal centre, and the glider airspeed. The action set dimensions were chosen as $\mathbf{a} = [a_{\dot{\phi}}, a_{\dot{\gamma}}]$, the roll rate and pitch rate. Finally, the rewards were based on the specific energy gain of the platform over each state-action transition, as well as two discrete event costs to cover the stall and crash conditions. The reward function is computed as :

$$r_{stall} = -25\% \times \frac{1}{2} v_{stall}^2 \qquad r_{crash} = -\frac{2E_{max}}{m} \tag{18}$$

$$r_t = \frac{E_t}{m} + stall \times r_{stall} + crash \times r_{crash}$$

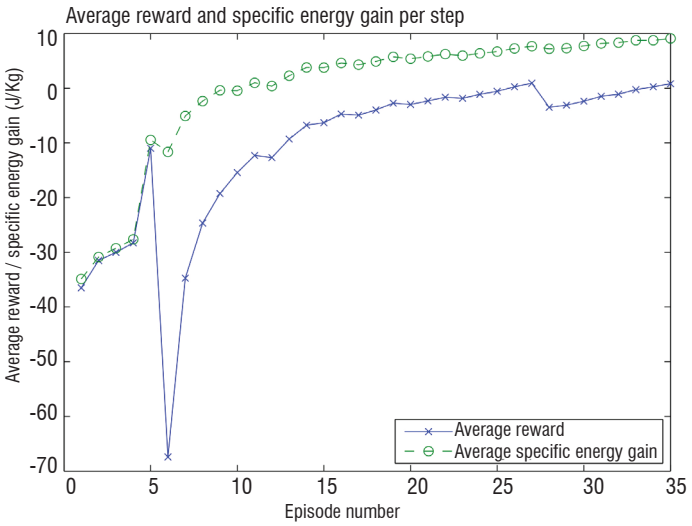


Figure 9 - Progression of the average reward per step and average specific energy gain per step across the learning episodes.

Learning occurred over a period of 35 episodes and a nominal set of flight paths are shown in figure 8. As more observations are taken and the algorithm updates the value function, the flight trajectories rapidly evolve to successfully gain enough energy to exit the field via the upper boundary by episode 14. The energy gaining efficiency of the learnt policy is also seen to increase as shown by the progression of the average reward and average specific energy gain in figure 9.

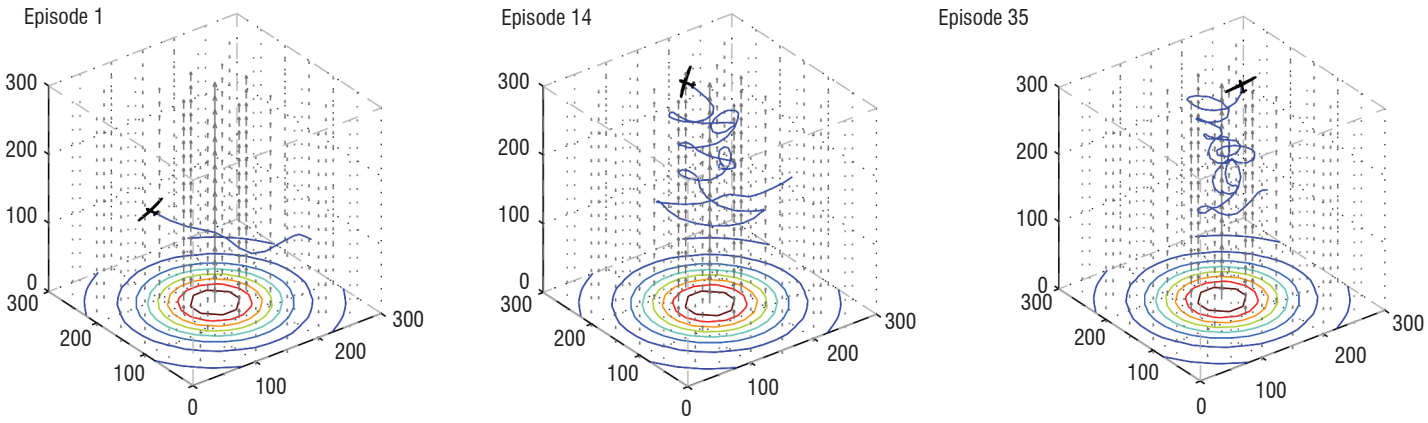


Figure 8 - Evolution of the sampled flight trajectories as learning progresses from episode 1, which terminates when the glider exits the lateral field boundary, to the first instance of the glider exiting via the upper boundary in episode 14, and the final flight trajectory in episode 35.

Soaring for UAV missions

The two previous methods looked at the soaring problem purely as a problem of exploring a wind field and finding the optimal actions to gain energy. However, in most cases there are additional goals for a UAV. The following sections look at methods for how a UAV should manage external goals with the need to capture energy in-flight. Two scenarios are considered : a single glider searching for a lost ground target, and multiple gliders tasked with a long-term surveillance mission over an area of interest.

Long-term information gathering

The first scenario uses thermal static soaring to aid long-term information gathering, as illustrated in figure 10. A gliding UAV equipped with sensors is tasked with searching for a stationary lost ground target. Information of the target state ξ is represented with a probabilistic belief function b^{ξ} on which UAV control actions are planned to increase information. Given a long-duration mission, the UAV is energy-constrained and must also periodically replenish energy at thermals. It is assumed that all thermals are known, stationary, and cover the search area densely enough for it to be explored. The objective is to find an optimal path plan P that maximises information gain over the entire mission time.

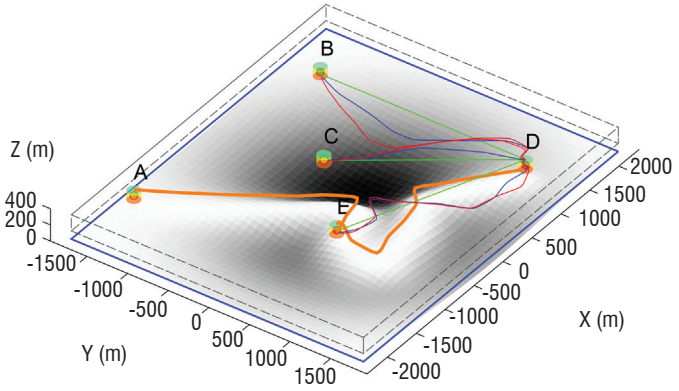


Figure 10 - A long-duration search scenario enabled by thermal soaring. The underlying surface is a probabilistic belief function of a ground target's location; darker regions represent more information. The UAV started at the top of thermal A, and has planned definite path segments to thermal D (orange lines). Green, blue and red lines represent possible next-step trajectories of varying information gain and energy expenditure.

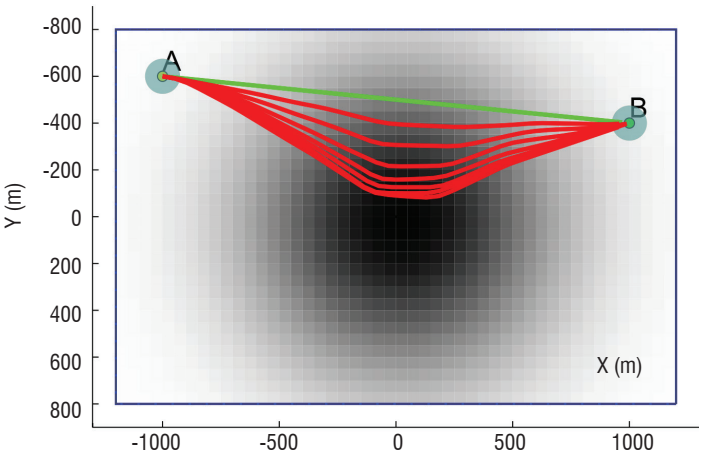


Figure 11 - The set R of inter-thermal paths between thermals A and B, including the initial path r_0 (green). Darker regions represent more information.

Path planning formulation

The target-search problem involves forward propagating UAV actions and applying a sensor model along the resulting paths to capture information. This results in an enormous action search space; to make the problem tractable, we firstly observe that any feasible plan can only consist of (1) information gathering path segments in between thermals, and (2) climbing within thermals to increase energy.

We introduced this as the informative soaring problem in [45] and formulated it as a discrete tree search problem by constructing a tree T of nodes $v \in V$. Starting with an empty root v_0, a child node v_d at depth d is constructed by appending one inter-thermal path segment to its parent v_{d-1}, so that any v represents a sequence of path segments. Each node has an associated utility $J(v)$ and cost $C(v)$. Here, $J(v)$ is the cumulative probability of target detection defined in [68], and $C(v)$ is the traversal time. A feasible plan is a leaf or terminal node with cost $C(v_t)$ within some budget $B > 0$. An optimal plan is one with maximum $J(v_t) \leq 1$.

Between thermals, there exists a continuous spectrum of inter-thermal path segments varying in $J(.)$ and $C(.)$. However, we only consider a subset Q of up to three options : 1) the maximum utility, 2) the minimum cost, and 3) the median utility/cost path segments. $Q \subseteq R$ is selected from the path segments set R generated by deforming an initial path r_0 using gradient descent on the underlying belief function b^{ξ} shown in figure 11. Refer to [45] for equations describing the sensor model and utility function.

Depth-Limited tree Search (DLS)

The optimal plan is ascertained with an exhaustive tree search ; however, this is computationally intractable for large time budgets B. A solution method that trades optimality for reduced computation is depth-limited (or finite-horizon) tree search (DLS). The process begins by building a complete subtree up to a fixed depth. The highest utility leaf node is identified, and a transition to the child node v_1 of the root v_0 along this branch is executed. The child becomes the new root and these steps are repeated until $C(v) \geq B$. DLS ideally allows future information gain to influence local decisions.

Monte Carlo Tree Search (MCTS)

While DLS offers computational practicality, a good search depth is unknown for arbitrary problem instances. DLS complexity also remains exponential in the search depth $\mathcal{O}(|Q|^d)$ such that a better solution at depth $d+1$ may just take too long to compute. Monte Carlo Tree Search (MCTS) is an anytime algorithm that can achieve further computational reductions for similarly high-quality plans to DLS by selectively expanding relevant tree nodes. MCTS achieves this with random rollouts at each node expansion to bias tree growth towards high-yield end states. The algorithm is simple yet powerful, and is well described in [9].

Cluster tree search

Interesting scenarios arise when the a-priori probabilistic belief function b^{ξ}_{init} is partitioned. This could be due to a series of prior uninformed local-area searches, and now the search region has been broadened. For long-term planning, this is problematic for state-of-theart search schemes, which can be too myopic for a given computational-

limit. Results from DLS illustrate that good solutions concentrate tree search effort at clusters of information. We draw on this observation to explicitly perform local cluster searches and optimally combine the resulting local plans with dynamic programming (DP) based on $J(v)$ and $C(v)$ of every node of each cluster tree [46]. Alg. 1 outlines the approach.

First, information clusters are identified using a Gaussian mixture model Gmm. They are ordered in sequence of distance from the UAV start position. Trees for each cluster are stored in the set $\mathcal{T}$. Cluster assigns time budget to the terminal state s_t^i proportionally to a fraction ε greater than the belief proportion b^{ξ^i} in b_{init}^{ξ}. The DP optimal plan may involve taking time from one cluster and allocating it to another ; ε allows for this. Search can be any tree search algorithm such as DLS or MCTS. For DLS, it suffices to use near-greedy search depths (e.g. 2 or 3) because the search effort is already concentrated at clusters. Minimum-cost path segments are used to link cluster plans together. Finally, Dp returns the optimal plan P.

Algorithm 1 *ClusterSearchDp*

$1: [b^{\xi^1}, b^{\xi^2}, \ldots, b^{\xi^n}] \leftarrow Gmm(b_{init}^{\xi})$
$2: \mathbb{T} \leftarrow \Phi$
$3:$ ***for*** $i = 1:n$ ***do***
$4: s_t^i \leftarrow Cluster + (b^{\xi^i}, b_{init}^{\xi}, B, \varepsilon)$
$5: T^i \leftarrow Search + (s_t^i)$
$6: \mathbb{T} \leftarrow \mathbb{T} \cup T^i$
$7:$ ***end for***
$8:$ ***return*** $P \leftarrow + Dp + (\mathbb{T})$

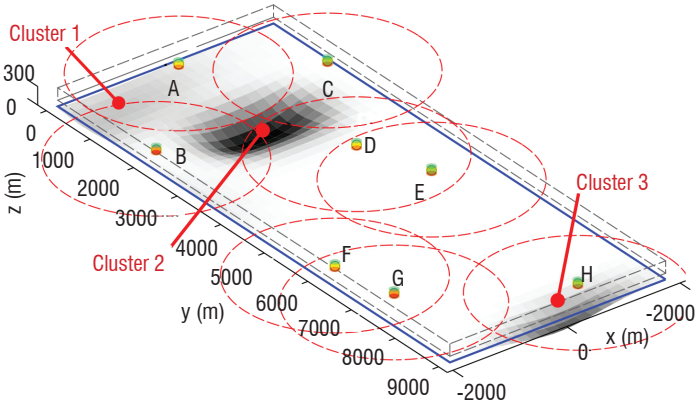


Figure 12 - A complex map scenario with three clusters of belief uncertainty (or information) represented by the dark patches. Thermals are labelled A to H.

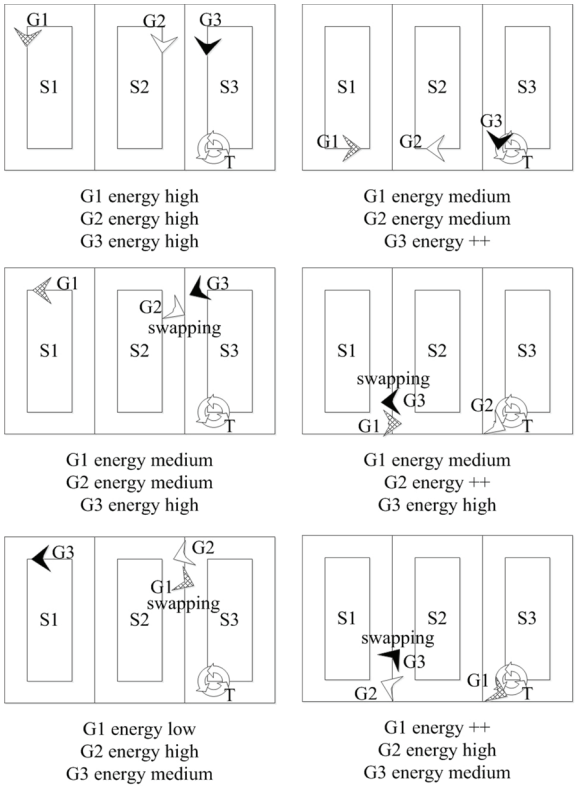


Figure 14 - A team of 3 gliders (G1, G2 and G3) uses the swapping approach to manage the thermal T, extending the mission but keeping the partitioning patrolling strategy.

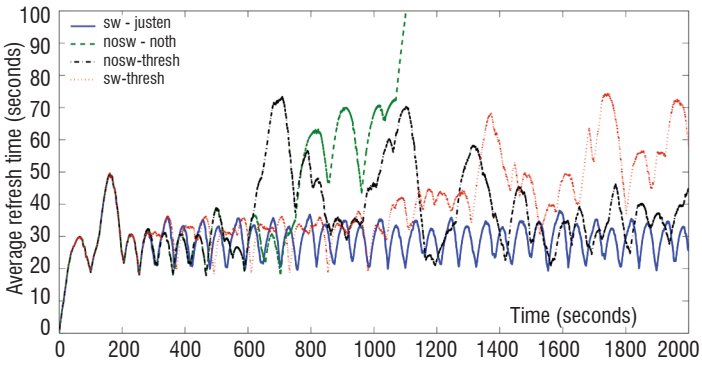


Figure 15 - The average maximum refresh time for a scenario with two thermals and four gliders, tested with different approaches : swapping or non-swapping and just-enough, threshold or no-thermalling approaches.

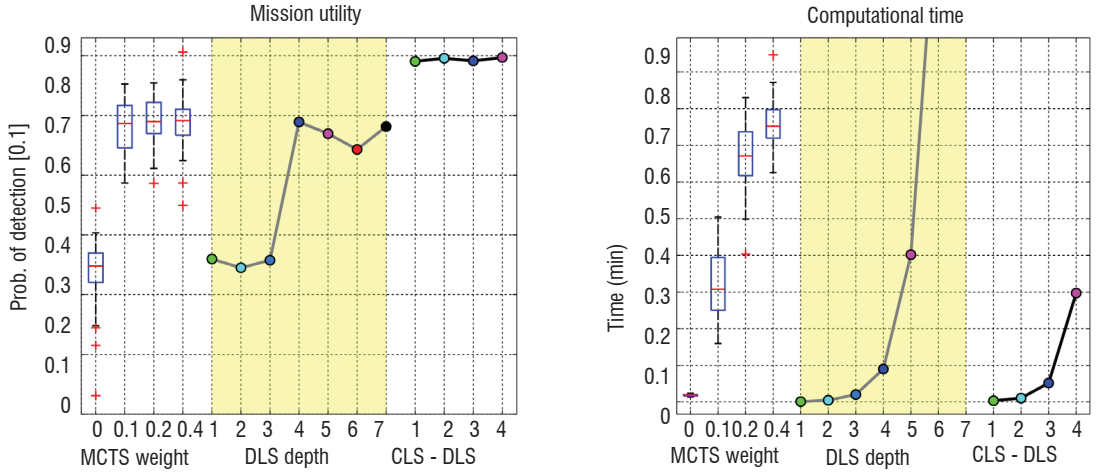


Figure 13 - Comparative results for 1) MCTS, 2) DLS and 3) cluster tree search using DLS on the scenario in figure 12. The trade-off between optimality and complexity is controlled by an exploration weight in MCTS, and the search depth in DLS.

Simulation results

In the interest of space, results are only shown for one map scenario although these methods have been extensively tested on complex target-search scenarios and have been shown to outperform current greedy planners [46]. This map (figure 12) has three information clusters and eight thermals. The UAV starts at thermal A and has a budget of 60 minutes to maximise information gain.

Performance results are illustrated in figure 13. As MCTS is a randomised solver, we conducted 50 trials for each exploration weight {0, 0.1, 0.2, 0.4}. A higher weight corresponds to denser tree search which expands more nodes at the expense of increased computation. Figure 13 suggests that 0.1 works the best here. It's utility is close to that of the best DLS depth of 4. The large step in DLS utility between depths 3 and 4 is a result of a longer horizon directing the plan towards cluster 3 in figure 12. In constrast, cluster tree search with DLS can achieve very high utility for greedy and near-greedy search depths.

Persistent monitoring with multiple gliders

A natural extension of the previous work is to examine how soaring missions could be performed with multiple gliding vehicles. This is particularly applicable to large area monitoring problems where the advantages of long endurance and multiple vehicles are readily apparent. However, multi- UAV systems present some difficult challenges to overcome related to coordination and distributed decision-making. Consider a problem where the objective is for multiple UAVs to cooperate to distribute the surveillance of a target in a persistent manner while minimising the refresh time T_r (period between consecutive visits) of all positions in the area. Previous work [1] considered this problem with powered UAVs, and the solution presented was shown to be robust to the failure of vehicles or the introduction of new vehicles whilst maintaining coverage of the target region and minimising refresh time.

However, extending the problem to consider gliding UAVs that require regular energy resupply is non-trivial. We extend the previous problem to consist of a team of N autonomous gliders tasked with monitoring an area S where there are M known thermal sources [72]. A cooperative patrolling strategy is required to coordinate the gliders such that the objectives can be optimised. A partitioning strategy is proposed to solve this problem because it allows consideration of communication constraints and gliders with different capabilities. The problem is partitioned such that each glider is in charge of patrolling a different non-overlapped region and thermals are associated with the regions in which they are located. In this case, the location and state of thermal sources can be considered as resources to be shared between the UAVs depending on their capabilities. This can be approached as a dynamic resource allocation problem to assign the thermals to the most suitable glider, such that all gliders maintain sufficient energy for continuous flight.

A *swapping* method based on one-to-one coordination can be used to manage the thermals in a distributed manner ensuring that all the regions continue being patrolled by at least one of the gliders. The swapping method implies that neighbouring UAVs share information about the thermal sources (location, state) in their own sub-areas. According to this information and the glider states each pair of neighbouring gliders can decide if they need to swap regions, as shown in figure 14. They decide according to the gliders' current energy and the distance to the nearest thermals. As the gliders converge in their knowledge of the world, they can decide independently and obtain consistent solutions. In this way, each UAV can reach the nearest thermal resource in a known finite time if it needs to gain energy. This approach ensures that the information about the thermals will be shared between all the gliders. Then, the thermals can be dynamically allocated between the gliders whilst maintaining the patrolling strategy.

Finally, another relevant issue is to decide how long a glider has to remain in the thermal. Two general approaches are defined. A threshold approach implies that the gliders go to the nearest thermal when they detect an energy level less than a *threshold* and remain in it until gaining the maximum possible altitude. Alternatively, a just-enough approach implies that when a glider reaches a thermal, it estimates the amount of energy that it will require to reach the next thermal, and remains in the thermal until gaining that estimated energy. This value can be estimated based on the model described in (1) and assuming an *a priori* known path to patrol the whole area.

Early research has shown that a combined just-enough swapping method obtains promising results for cooperative large area monitoring missions with a team of autonomous gliders that exploit the thermal sources. Figure 15 shows the maximum average refresh time computed along the whole path during a mission assuming four gliders and two thermals. A cooperative path partitioning strategy and four different thermal access approaches are considered in the test.

Aerial resupply

In the preceding sections, this paper has described recent work on persistent autonomous flight through the opportunistic exploitation of readily available atmospheric energy. A different, yet complementary approach to persistent flight is the deliberate in-flight resupply of energy. This is a complementary strategy when atmospheric energy is insufficient or unavailable, and an alternative when the platform is not optimised for atmospheric energy collection. In the past this has occurred manually with large, manned aircraft at high altitude, however automating this process allows the pilot to be removed and much smaller aircraft, such as autonomous gliders to utilise the procedure.

It is generally accepted in the literature that the primary barrier to autonomous in-flight resupply is the sensing and navigation challenges surrounding tight formation flight. This challenge is amplified in our work where we consider small, dynamic vehicles that are operating in a turbulent, low altitude environment. These conditions require an accurate and timely relative state estimate that is robust to a dynamic environment. One method to obtain this estimate is to subtract one vehicle's INS/GPS-based state estimate from the other, where the result is known as the raw relative estimate. This approach has merit during high separation formation but the accuracy, particularly in relative position, is in the order of metres and is not sufficiently accurate for close proximity formation flight. The low accuracy is not only attributed to the individual sensor accuracy, but also errors in measurement time synchronisation since absolute measurements are being differenced. This problem also applies to more accurate DGPS systems. Highly dynamic vehicles and sporadic communication dropouts further amplify this effect.

To achieve the necessary accuracy, directly observed relative measurements must be used. Vision is a popular sensor selection in the aerial domain due to its availability, compact size and low weight. The downside to vision is the susceptibility to observation dropouts as a result of occlusion, a constrained field of view (FOV) and uncertain lighting conditions. Further, incorrect observations resulting from clutter and false feature correspondences must be detected and excluded. To negate these shortcomings and create a resilient yet accurate relative state estimator, it is important to incorporate constantly available, albeit absolute, information from sensors such as inertial, magnetic, atmospheric and GPS.

In our scenario, a leader and follower UAV are flying in formation. Visual markers are mounted on the leader at each wingtip, at the top of the tail fin and on the right of the tail plane as shown in figure 16. A forward facing camera is mounted on the follower and provides relative measurements to the leader's visual markers. All onboard sensor data from both aircraft are available on the follower UAV in real-time via wireless communications. The following sections summarise the relative estimation framework, the vision integration and provide preliminary implementation results.

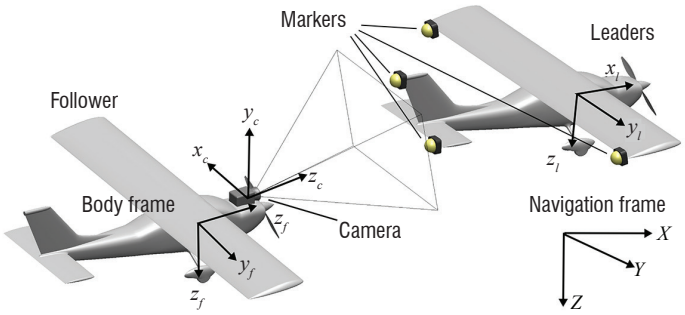


Figure 16 - Leader-follower coordinate frames and the marker based vision system

Multi-vehicle relative navigation

The proposed relative estimator fuses vehicle-to-vehicle visual measurements with information from GPS, inertial, magnetic and atmospheric sensors, located on each UAV, in a tightly coupled fashion. An unscented Kalman filter (UKF) [26] provides the filter framework to estimate the position P, velocity V and attitude quaternion q for each aircraft. P and V are expressed in the local tangential frame, relative to a ground station.

$$x_{l|f} = [\, P_l \; V_l \; q_l \; P_f \; V_f \; q_f \,]^T \tag{19}$$

$$\begin{bmatrix} P_{l/f} \\ V_{l/f} \\ q_{l/f} \end{bmatrix} = \begin{bmatrix} P_l - P_f \\ V_l - V_f \\ q_f^{-1} \otimes q_l \end{bmatrix} \tag{20}$$

A UKF has several advantages over the traditional extended Kalman filter (EKF). It provides at least second-order nonlinear approximation as opposed to the first-order EKF ; derivation of Jacobians is not necessary , the filter is more robust to initial errors and computation can occur in parallel. Resilience to initial error is particularly important because of the large difference in accuracy between the GPS and vision-based measurements. A downside of the UKF is that a quaternion parametrisation of the attitude results in a non-unit quaternion when the mean is computed. A brute force normalisation can be made to work, but is undesirable. Instead, we use generalised Rodrigues parameters to represent the attitude error, as proposed in [13].

Each vehicle state is propagated using the bias corrected onboard inertial measurements and the mechanization equations found in [53]. Details pertaining to the vehicle atmospheric, magnetic and GPS sensor updates are also omitted but can be found in [19]. UKF prediction and update equations are well known and are provided in [26, 47, 13].

Vision integration

Relative pose estimation using vision sensors has been well researched and many valid approaches exist. Our work employs a feature based method where visual markers of a known configuration are mounted on a leader vehicle and observed by a follower. Using the set of n correspondences between the 3D marker positions, ζ_j^l and the 2D observations $\tilde{\delta}_j$, as well as the camera intrinsic parameters, the relative pose can be calculated directly. This requires $n \geq 3$ for a solution and $n \geq 4$ for a unique solution. A number of algorithms are available to solve this $\mathrm{P}n\mathrm{P}$ problem, including POSIT [14] which is used as a benchmark in Section 3.3, the Lu-Hager-Mjolsness algorithm [38] and an efficient approach called $\mathrm{EP}n\mathrm{P}$ [36].

The downside to this vision only approach is that it fails with incorrect point matching, occlusion or a target outside the FOV. These brief or prolonged measurement dropouts are highly undesirable, particularly during close proximity operations. Alternatively, one could fuse the pose estimate from one of the aforementioned algorithms with the onboard sensor data in a loosely-coupled arrangement, however preliminary results with a fixed measurement covariance displayed inferior performance to the tightly-coupled equivalent. One reason may be that the measurement covariance is dynamic and a function of the relative pose, in addition to the pixel noise. Deriving an expression for this is difficult.

Instead, we propose a tightly-coupled approach which uses n raw 2D marker observations, $\tilde{\delta}_j = [u_j v_j]^T$, $j=1,\ldots,n$. In our case $n=4$ and $n \geq 3$ is required for observability within the UKF [66, 20]. The expected observations $\bar{\delta}_j$, $j=1,\ldots,n$ are calculated by first transforming ζ_j^l from the leader's body frame to the world frame, ζ_j^f using (21). In this case the world frame is the follower's body frame.

$$\zeta_j^f = C_n^f \left(C_l^n \zeta_j^l + P_{l|f} \right) \tag{21}$$

Next, the vision sensor extrinsic parameters transform ζ_j^f to the camera frame using (22). $P_{f|c}$ and C_f^c are the translation and rotation from the follower's body frame to the camera frame. C_f^c includes both the camera mounting orientation and the axes transformation.

$$\zeta_j^c = \begin{bmatrix} C_f^c P_{f|c} \end{bmatrix} \begin{bmatrix} \zeta_j^f \\ 1 \end{bmatrix} \tag{22}$$

$\bar{\delta}_j$ is calculated using K, the camera intrinsic matrix which encapsulates the camera focal length, aspect ratio, principal point and distortion. The final vision measurement model is provided in (24) and the correction occurs sequentially.

$$\begin{bmatrix} \bar{\delta}_j \\ 1 \end{bmatrix} = K \begin{bmatrix} \zeta_{xj}^c / \zeta_{zj}^c \\ \zeta_{yj}^c / \zeta_{zj}^c \\ 1 \end{bmatrix} \tag{23}$$

$$h^{vision}[x,\,k] = \left[\bar{\delta}_1\ \bar{\delta}_2\ \ldots\ \bar{\delta}_n\right]^T \tag{24}$$

The vision-based observation model presented in this section updates both the relative position and orientation by assuming correct point matching. This is not always possible, particularly when the target is far and the points are difficult to distinguish from one another. Rather than neglecting such a measurement, a simpler observation model can be utilised to extract $P_{l|f}$ information. Here, the vision observation becomes the average or centroid of $\tilde{\delta}_j$, $j = 1, \ldots, n$ as an approximation for the target's centre of gravity and (21) is replaced with (25) where $m = 1$. Although relative orientation and range become unobservable, $P_{l|f}$ and $V_{l|f}$ accuracy is improved.

$$\zeta_j^f = C_j^f P_{l/f} \tag{25}$$

Before $y_{l|f}^{vision}$ can be used, correspondences between the observed points $\tilde{\delta}_i$ and the projected model points $\bar{\delta}_j$ must be determined. To do this, unique marker characteristics could be used, which may include colour, size, intensity and frequency. However, in our application we have chosen to use homogeneous visual markers to simplify the MV task and instead use the marker model to match the points. To do this, we use a computationally efficient, deterministic mutual nearest point procedure [42]. Before this is implemented, we eliminate the linear translation between the point sets by subtracting the vector $(\bar{\delta}_\mu - \tilde{\delta}_\mu)$ from $\bar{\delta}_j$. This eliminates errors in relative azimuth, elevation as well as follower attitude and simplifies the matching process. The matrix Θ is then populated with the pixel distances between $\tilde{\delta}_i$ and $\bar{\delta}_j$.

$$\Theta = \begin{bmatrix} d(\bar{\delta}_1,\tilde{\delta}_1) & \cdots & d(\bar{\delta}_1,\tilde{\delta}_n) \\ \cdots & \ddots & \cdots \\ d(\bar{\delta}_m,\tilde{\delta}_1) & \cdots & d(\bar{\delta}_m,\tilde{\delta}_n) \end{bmatrix} \tag{26}$$

Where $d(.,.)$ is the linear pixel distance between points. $\Theta_{col}^{\min}$ and $\Theta_{row}^{\min}$ are the minimum value of each column and row of Θ, respectively and Θ_{col}^{index} contains the index of the minimum value in each column.

$$\begin{aligned} \Theta_{col}^{min} &= \left[min(d(\bar{\delta}_i,\tilde{\delta}_1)) \quad \cdots \quad min(d(\bar{\delta}_i,\tilde{\delta}_m))\right] \\ \Theta_{row}^{min} &= \left[min(d(\bar{\delta}_1,\tilde{\delta}_j)) \quad \cdots \quad min(d(\bar{\delta}_n,\tilde{\delta}_j))\right] \\ \Theta_{col}^{index} &= \left[(\Theta_{col_1}^{min}) \quad \cdots \quad index(\Theta_{col_n}^{min})\right] \end{aligned} \tag{27}$$

For a point to be valid, it must satisfy (28), that is to say a valid point in Θ must be the minimum of both its column and row. A threshold of validity can also be implemented to reject outliers and noise.

$$\Theta_{col}^{min}[i] = \Theta_{col}^{min}[\Theta_{col}^{index}[i]] \tag{28}$$

	Raw (1σ)	POSIT (1σ)	$x_{l\|f}(1\sigma)$	Raw impr.
$P_{l\|f}$				
N	2.069 (0.86)	0.353 (0.15)	0.205 (0.23)	90.1 %
E	1.901 (0.76)	1.017 (0.34)	0.427 (0.24)	77.5 %
D	0.152 (0.11)	0.123 (0.03)	0.129 (0.07)	15.1 %
$V_{l\|f}$				
N	0.203 (0.09)	-	0.216 (0.25)	-6.54 %
E	0.227 (0.18)	-	0.183 (0.18)	19.4 %
D	0.068 (0.09)	-	0.052 (0.05)	23.5 %
$Q_{l\|f}$				
φ	1.148 (0.81)	0.295 (0.01)	0.173 (0.24)	85.0 %
θ	1.095 (1.12)	0.660 (0.02)	0.208 (0.17)	81.0 %
ψ	14.743 (8.90)	0.577 (0.01)	0.443 (0.85)	97.0 %

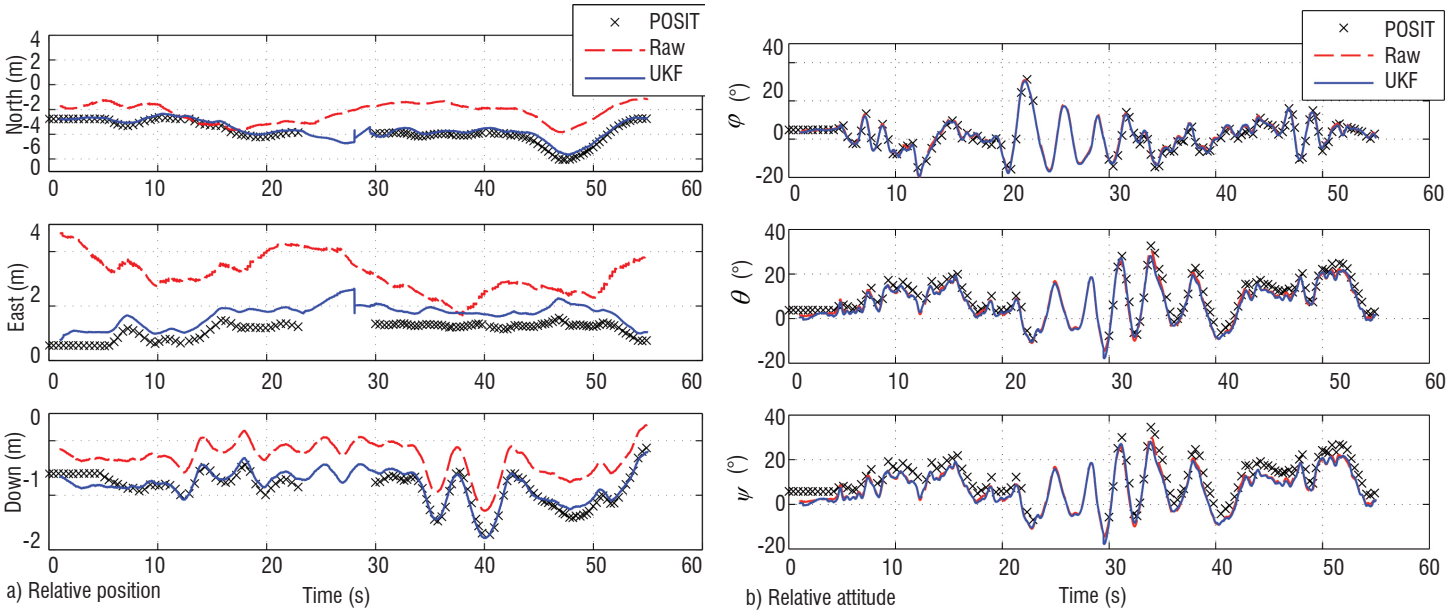
Table 1 - Raw relative, POSIT and relative UKF estimate RMSE comparison from 100 simulations. Axes are N(orth), E(east) and D(own). Positions are measured in m, velocities in m/s and angles in degrees.

Implementation

The estimator was tested in a high fidelity simulated environment [67] where conditions are repeatable and the ground truth is known. The simulation was run 100 times and the results are summarised numerically in table 1. Compelling performance improvements were observed when compared to both the raw relative estimate and the benchmark vision-only pose estimation algorithm, POSIT. Particularly large improvements in horizontal position and ψ can be attributed to the relative inaccuracy of the GPS and magnetometers. As expected, the gains over POSIT are less but remain notable which is likely due to a smoothing effect of the vehicle inertial measurements.

The algorithm was also testing in ground based experiments on a dual-UAV system to isolate the relative navigation problem, and demonstrate the estimation framework functioning in real-time on an embedded system. This system consists of two fixed-wing UAVs, an autopilot and formation flight computer onboard each aircraft, LED markers on the leader, and a camera on the follower.

Relative position and attitude estimates from one such experiment are shown in figures 17a and 17b. Here, we can see good agreement between the vision-only POSIT algorithm and the output of the relative estimator. A slight bias can be observed in the east and ψ components which indicate a slight error in the camera extrinsic calibration. Additionally, we see that POSIT fails between 23-29 seconds because less than four points are available. Here, the relative estimator is able to utilise information from even a single visual marker and only slowly degrades to the raw relative estimate when no visual measurements are available.



a) Relative position b) Relative attitude

Figure 17 - Vision based relative estimation results from the dual-UAV system

Conclusion

The work presented here is aimed at extending the flight duration of fixed-wing UAVs. This paper highlighted some of the work performed at the University of Sydney towards reaching this goal. We examined the basic mechanisms of soaring flight and used these equations to derive utility functions for planning. This was extended to planning in unknown wind fields using GP regression for wind map building, and for use in an RL framework to develop energy gaining trajectories without specifying control strategies. Further, soaring was integrated into target search problems multiple UAVs. The paper also looked at coordination for aerial refueling with small UAVs.

Future work will aim to bring some of these methods together for long-duration UAV missions. The focus will be on autonomy and decision-making, ideally for a system with multiple heterogeneous vehicles to perform an externally driven mission. The system should make decisions about the energy available for soaring and the requirement for inflight refueling to allocate vehicles based on their demands for energy and mission utility ■

Acknowledgements

This work is supported by the Australian Centre for Field Robotics at the University of Sydney. J.J. Acevedo visited the University of Sydney through the framework of the MUAC-IREN (FP7-PEOPLE-295300) project of the European Commission. The author J.J. Acevedo acknowledges too the CLEAR (DPI2011-28937-C02-01) Spanish National Research project and the project of excellence of the Junta de Andalucía WSAN-UAV (P09-TEP-5120).

Acronyms

DLS	(Depth-Limited (tree) Search)
DP	(Dynamic Programming)
EKF	(Extended Kalman Filter)
MCTS	(Monte Carlo Tree Search)
RL	(Reinforcement learning)
RMSE	(Root Mean Square Error)
TD	(Temporal difference)
UAV	(Unmanned Aerial Vehicle)
UKF	(Unscented Kalman Filter)

SBXC Glider Parameters

Parameter	Value	Units	Explanation
$C_{D,O}$	0.012		Parasitic drag coefficient
S	0.957	m²	Wing reference area
AR	19..54		Wing aspect ratio
e	0.85		Oswald's efficiency factor
m	5.44	kg	Vehicle mass
$N_{\max}$	2.0		Maximum load factor
$N_{\min}$	0		Minimum load factor
$C_{L,\max}$	1.2		Maximum lift coefficient
$\frac{d\phi}{dt_{\max}}$	30	°/s	Maximum roll rate
$\gamma_{\alpha,\max}$	50	°	Max. air relative climb
$\left(\frac{L}{D}\right)_{est}$	50		Approximate glide ratio

Table 2 - The aerodynamic and geometric properties of the SB-XC glider model

References

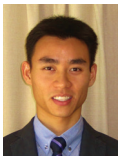
[1] J. J. ACEVEDO, B. N. C. ARRUE, I. MAZA, and A. OLLERO - *Cooperative Large Area Surveillance with a Team of Aerial Mobile Robots for Long Endurance Missions*. Journal of Intelligent and Robotic Systems 70 (2013), 329–345.
[2] H. AIRY - *The Soaring of Birds*. Nature 27, 703 (19 April 1883), 590–592.
[3] Z. ÁKOS, M. NAGY, S. LEVEN and T. VICSEK - *Thermal Soaring Flight of Birds and Unmanned Aerial Vehicles.* Bioinspiration & Biomimetics 5, 4 (2010), 045003.
[4]M. J. ALLEN - *Autonomous Soaring for Improved Endurance of a Small Uninhabited Air Vehicle*. 43rd AIAA Aerospace Sciences Meeting and Exhibit (Reno, Nevada, 2005). AIAA Paper 2005-1025.
[5] M. J. ALLEN, and V. LIN - *Guidance and Control of an Autonomous Soaring UAV*. Technical Memorandum NASA/TM-2007-214611, NASA Dryden Flight Research Center, February 2007.
[6] Jr. J. D. ANDERSON - *Fundamentals of Aerodynamics.* Fourth ed. McGraw-Hill Series in Aeronautical and Aerospace Engineering. McGraw-Hill, New York, 2007.
[7] R. BARATE, S. DONCIEUX and J.-A. MEYER - *Design of a Bio-Inspired Controller for Dynamic Soaring in a Simulated Unmanned Aerial Vehicle*. Bioinspiration & Biomimetics 1, 3 (2006), 76–88.
[8] M. B. BOSLOUGH - *Autonomous Dynamic Soaring Platform for Distributed Mobile Sensor Arrays*. Tech. Rep. SAND2002-1896, Sandia National Laboratories, June 2002.
[9] C. B. BROWNE, E. POWLEY, D. WHITEHOUSE, S. M. LUCAS, P. I. COWLING, P. ROHLFSHAGEN, S. TAVENER, D. PEREZ, S. SAMOTHRAKIS and S. COLTON - *A Survey of Monte Carlo Tree Search Methods*. IEEE Trans. Comput. Intell. AI in Games 4, 1 (2012), 1–43.
[10] J. J. CHUNG, N. R. J. LAWRANCE, and S. SUKKARIEH - *Gaussian Processes for Informative Exploration in Reinforcement Learning*. 2013 IEEE International Conference on Robotics and Automation (2013), IEEE, pp. 2633–2639.
[11] J. J. CHUNG, N. R. J. LAWRANCE and S. SUKKARIEH - *Resource Constrained Exploration in Reinforcement Learning*. Robotics: Science and Systems Workshop on Robotic Exploration, Monitoring, and Information Collection: Nonparametric Modeling, Information-based Control, and Planning under Uncertainty (Berlin, Germany, 2013).
[12] J. J. CHUNG, M. A. TRUJILLO and S. SUKKARIEH - *A New Utility Function for Smooth Transition Between Exploration and Exploitation of a Wind Energy Field*. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems Conference Proceedings (2012), pp. 4999– 5005.
[13] J. CRASSIDIS AND F. MARKLEY - *Unscented Filtering for Spacecraft Attitude Estimation*. AIAA Guidance, Navigation, and Control Conference (2003), AIAA.
[14] D. F. DEMENTHON and L. S. DAVIS - *Model-Based Object Pose in 25 Lines of Code*. International Journal of Computer Vision 15, 1-2 (1995), 123–141.
[15] N. T. DEPENBUSCH and J. W. LANGELAAN - *Receding Horizon Control for Atmospheric Energy Harvesting by Small UAVs.* AIAA Guidance, Navigation and Control Conference (Toronto, Ontario, Canada, 2010). AIAA Paper 2010-8180.
[16] J. DOEBBLER, T. SPAETH, J. VALASEK, M. J. MONDA and H. SCHAUD - *Boom and Receptacle Autonomous Air Refueling Using a Visual Pressure Snake Optical Sensor.* AIAA Atmospheric Flight Mechanics Conference and Exhibit (2006), American Institute of Aeronautics and Astronautics.
[17] S. DONCIEUX, J.-B. MOURET and J.-A. MEYER - *Soaring Behaviors in UAVs : 'Animat Design Methodology and Current Results*. 7th European Micro Air Vehicle Conference (MAV07) (Toulouse, France, 2007), pp. 1–10.
[18] D. J. EDWARDS and L. M. SILVERBERG - *Autonomous Soaring : the Montague Cross-Country Challenge*. Journal of Aircraft 47, 5 (2010), 1763 – 1769.
[19] T. FIORENZANI, C. MANES, G. ORIOLO and P. PELITI - *Comparative Study of Unscented Kalman Filter and Extended Kalman Filter for Position/Attitude Estimation in Unmanned Aerial Vehicles.* Tech. rep., Istituto Di Analisi Dei Sistemi ed Informatica, 2008.
[20] A. FOSBURY and J. CRASSIDIS - *Relative Navigation of Air Vehicles*. Journal of Guidance, Control, and Dynamics 31, 4 (2008), 824–834.
[21] C. GIAMPIERO, F. MARIO LUCA, F. ANTONIO, N. MARCELLO, S. BRAD and P. MARIO - *Autonomous Aerial Refueling for UAVs Using a Combined GPS-Machine Vision Guidance.* AIAA Guidance, Navigation, and Control Conference and Exbit (2004), Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics.
[22] Y. GU, B. SEANOR, G. CAMPA, M. NAPOLITANO, L. ROWE, S. GURURAJAN and S. WAN - *Design and Flight Testing Evaluation of Formation Control Laws*. Control Systems Technology, IEEE Transactions on 14, 6 (2006), 1105–1112.
[23] E. A. HAERING - *Airdata Calibration of a High-Performance Aircraft for Measuring Atmospheric Wind Profiles*. Technical memorandum NASA-TM-101714, NASA, January 1990.
[24] P. IDRAC - *Experimental Study of the "Soaring" of Albatrosses*. Nature 115, 2893 (1925), 532–532.
[25] E. N. JOHNSON, A. J. CALISE, Y. WATANABE, J. HA and J. C. NEIDHOEFER - *Real-Time Vision-Based Relative Aircraft Navigation*. Journal of Aerospace Computing, Information, and Communication 4, 4 (2007), 707–738.
[26] S. J. JULIER, J. K. UHLMANN and H. F. DURRANT - WHYTE - *A New Approach for Filtering Nonlinear Systems.* In American Control Conference. Proceedings of the 1995 (1995), vol. 3, pp. 1628–1632.
[27] S. M. KHANSARI-ZADEH and F. SAGHAFI - *Vision Based Navigation in Autonomous Close Proximity Operations Using Neural Networks*. Aerospace and Electronic Systems, IEEE Transactions on 47, 2 (2011), 864–883.
[28] J. A. KYLE - *Optimal Soaring by a Small Autonomous Glider*. PhD thesis, Oregon State University, 2006.
[29] J. W. LANGELAAN - *Gust Energy Extraction for Mini and Micro Uninhabited Aerial Vehicles*. Journal of Guidance, Control, and Dynamics 32, 2 (2009), 464–473.
[30] J. W. LANGELAAN, J. SPLETZER, C. MONTELLA and J. GRENESTEDT - *Wind Field Estimation for Autonomous Dynamic Soaring*. Robotics and Automation(ICRA), 2012 IEEE International Conference on (May 2012), pp. 16 –22.
[31] N. R. J. LAWRANCE - *Autonomous Soaring Flight for Unmanned Aerial Vehicles*. PhD thesis, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, 2011.
[32] N. R. J. LAWRANCE and S. SUKKARIEH - *A Guidance and Control Strategy for Dynamic Soaring with a Gliding UAV*. "Robotics and Automation (ICRA), 2009 IEEE International Conference on" (Kobe, Japan, 2009), pp. 3632–3637.
[33] N. R. J. LAWRANCE and S. SUKKARIEH - *Wind Energy Based Path Planning for a Small Gliding Unmanned Aerial Vehicle*. AIAA Guidance, Navigation and Control Conference (Chicago, Illinois, 2009). AIAA Paper 2009-6112.
[34] N. R. J. LAWRANCE and S. SUKKARIEH - *Autonomous Exploration of a Wind Field with a Gliding Aircraft*. Journal of Guidance, Control, and Dynamics 34, 3 (2011), 719–733.

[35] N. R. J. LAWRANCE and S. SUKKARIEH - *Path Planning for Autonomous Soaring Flight in Dynamic Wind Fields*. IEEE International Conference on Robotics and Automation (Shanghai, China, 2011), pp. 2499 – 2505.
[36] V. LEPETIT, F. MORENO-NOGUER and P. FUA - *Epnp: An Accurate o(n) Solution to the PNP problem*. International Journal of Computer Vision 81, 2 (2009), 155–166.
[37] P. LISSAMAN - *Wind Energy Extraction by Birds and Flight Vehicles*. 43rd AIAA Aerospace Sciences Meeting and Exhibit (Reno, Nevada, 2005). AIAA Paper 2005-241.
[38] C. P. LU, G. D. HAGER and E. MJOLSNESS - *Fast and Globally Convergent Pose Estimation from Video Images*. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, 6 (2000), 610–622.
[39] P. B. MACCREADY - *Optimum Airspeed Selector.* Soaring 9 (March 1954), 8.
[40] P. B. MACCREADY - *Optimum Airspeed Selector.* Soaring January-February (1958), 10–11.
[41] Z. MAHBOUBI, Z. KOLTER, T. WANG and G. BOWER - *Camera Based Localization for Autonomous UAV Formation Flight.* Proceedings of the AIAA@ Infotech Conference (2011).
[42] M. MAMMARELLA, G. CAMPA, M. R. NAPOLITANO and M. L. FRAVOLINI - *Comparison of Point Matching Algorithms for the UAV Aerial Refueling Problem*. Machine Vision and Applications 21, 3 (2010), 241–251.
[43] M. MAMMARELLA, G. CAMPA, M. R. NAPOLITANO, M. R. FRAVOLINI, M. L., Y. GU and M. G. Perhinschi - *Machine Vision/GPS Integration Using EKF for the UAV Aerial Refueling Problem*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 38, 6 (2008), 791–801.
[44] C. MARTINEZ, T. RICHARDSON and P. CAMPOY - *Towards Autonomous Air-to-Air Efuelling for UAVs Using Visual Information.* Robotics and Automation (ICRA), 2013 IEEE International Conference on (2013), pp. 5756–5762.
[45] J. L. NGUYEN, N. R. J. LAWRANCE, N. R. J. FITCH and S. SUKKARIEH - *Energy-Constrained Motion Planning for Information gathering with autonomous aerial soaring*. Robotics and Automation (ICRA), 2013 IEEE International Conference on (2013), IEEE, pp. 3825–3831.
[46] J. L. NGUYEN, N. R. J. LAWRANCE and S. SUKKARIEH - *Nonmyopic Planning for long-Term Information Gathering With an Aerial Glider*. Robotics and Automation (ICRA), 2014 IEEE International Conference on (2014), 6573-6578.
[47] S. OH and E. N. JOHNSON - *Relative Motion Estimation for Vision-Based Formation Flight Using Unscented Kalman Filter.* AIAA Guidance, Navigation and Control Conference and Exhibit (2007), AIAA 2007-6866.
[48] S. PARK, J. DEYST and J. P. HOW - *A New Nonlinear Guidance Logic for Trajectory Tracking*. Proceedings of the AIAA Guidance, Navigation and Control Conference (2004), Citeseer.
[49] S. PEAL - *Soaring of birds*. Nature 23, 575 (4 November 1880), 10–11.
[50] C. PENNYCUICK - *Field Observations of Thermals and Thermal Streets, and the Theory of Cross-Country Soaring Flight*. Journal of Avian Biology 29, 1 (1998), 33–43.
[51] C. PENNYCUICK, T. ALERSTAM and B. LARSSON - *Soaring Migration of the Common Crane Grus Crus Observed by Radar and from an Aircraft*. Ornis Scandinavica 10, 2 (1979), 241–251.
[52] L. RAYLEIGH - *The Soaring of Birds.* Nature 27, 701 (5 April 1883), 534–535.
[53] R. M. ROGERS - *Applied Mathematics in Integrated Navigation Systems*, 3rd ed. AIAA Education Series. AIAA, 2007.
[54] G. A. RUMMERY and M. NIRANJAN - *On-line Qlearning Using Connectionist Systems*. Tech. rep., Cambridge University Engineering Department, 1994.
[55] G. SACHS - *Minimum Shear Wind Strength Required for Dynamic Soaring of Albatrosses*. Ibis 147, 1 (2005), 1–10.
[56] R. J. SATTIGERI - *Adaptive Estimation and Control with Application to Vision-Based Autonomous Formation Flight.* PhD thesis, School of Aerospace Engineering, Georgia Institute of Technology, 2007.
[57] R. SPAAR and B. BRUDERER - *Optimal Flight Behavior of Soaring Migrants: a Case Study of Migrating Stepp Buzzards, Buteo Buteo Vulpinus*. Behavioral Ecology 8, 3 (1997), 288–297.
[58] R. SUTTON and A. BARTO - *Reinforcement Learning: An Introduction.* The MIT press, 1998.
[59] A. L. TREASTER and A. M. YOCUM - *The Calibration and Application of Five-Hole Probes*. Tech. Rep. TM-78-10, Pennsylvania State University Applied Research Laboratory, 1978.
[60] V. TUCKER - *Flight Energetics in Birds*. American Zoologist 11, 1 (1971), 115–124.
[61] J. VALASEK, K. GUNNAM, J. KIMMETT, J. L. JUNKINS, D. HUGHES and M. D. TANDALE - *Vision-Based Sensor and Navigation System for Autonomous Air Refueling*. Journal of Guidance, Control, and Dynamics 28, 5 (2005), 979–989.
[62] S. Walkden - *Experimental Study of the "Soaring" of Albatrosses.* Nature 116, 2908 (25 July 1925), 132–134.
[63] H. WEIMERSKIRCH, T. GUIONNET, J. MARTIN, S. A. SHAFFER and D. P. COSTA - *Fast and Fuel Efficient? Optimal use of Wind by Flying Albatrosses*. Proceedings of the Royal Society of London - Biological Sciences 267, 1455 (2000), 1869 – 1874.
[64] J. WHARINGTON - Autonomous Control of Soaring Aircraft by Reinforcement Learning. PhD thesis, Royal Melbourne Institute of Technology, 1998.
[65] J. WHARINGTON and I. HERSZBERG - *Control of a High Endurance Unmanned Aerial Vehicle*. 21st Congress of International Council of the Aeronautical Sciences (Melbourne, Australia, September 1998). ICAS-98-3.7.1.
[66] W. R. WILLIAMSON, G. J. GLENN, V. T. DANG, J. L. SPEYER, S. M. STECKO and J. M. TAKACS - *Sensor Fusion Applied to Autonomous Aerial Refueling*. Journal of Guidance, Control, and Dynamics 32, 1 (2009), 262– 275.
[67] D. B. WILSON, A. H. GÖKTOĞAN and S. SUKKARIEH - *UAV Rendezvous: From Concept to Flight Test.* Australasian Conference on Robotics and Automation (ACRA) (2012).
[68] E. WONG, F. BOURGAULT and T. FURUKAWA - *Multivehicle Bayesian Search for Multiple Lost Targets*. Proc. of "Robotics and Automation (ICRA), 2005 IEEE International Conference on".
[69] C. J. WOOD - *The Flight of Albatrosses (a computer simulation)*. Ibis 115, 2 (1972), 244–256.
[70] Y. J. ZHAO - *Optimal Patterns of Glider Dynamic Soaring.* Optimal Control Applications and Methods 25, 2 (2004), 67–89.
[71] Y. J. ZHAO and Y. C. QI - *Minimum Fuel Powered Dynamic Soaring of Unmanned Aerial Vehicles Utilizing Wind Gradients.* Optimal Control Applications and Methods 25, 5 (2004), 211–233.
[72] J.J. ACEVEDO, N.R.J. LAWRANCE, B.C. ARRUE, S. SUKKARIEH, and A. OLLERO - *Persistent monitoring with a team of autonomous gliders using static soaring.* 2014 IEEE/RS J. International Conference on Intelligent Robots and Systems Conference Proceedings (2014), to appear.

Nicholas R. J. Lawrance completed his B.E. in Aeronautical Space Engeering in 2007 and PhD in Robotics in 2011, both at the University of Sydney. He is currently a research associate at the Australian Centre for Field Robotics at the University of Sydney. His research interests include autonomous soaring for unmanned aerial vehicles, aircraft performance, reinforcement learning and informative path planning.

Jose Joaquin Acevedo received the Telecommunication Engineer Degree in 2007 and the Electrical Energy Systems M.Sc Degree in 2012, both from the University of Seville (Spain). From 2007 to 2010, he worked for different ICT companies. He is currently a PhD Student at the Systems Engineering and Automation Department of the University of Seville, at the Robotic, Vision and Control Group. He is working in different national and European research projects and has published several papers in International journals and conferences. His research interests include coordination and cooperation in distributed multi-robot and multi-UAV systems and task and resource allocation.

Jen Jen Chung is currently completing her Ph.D. at the Australian Centre for Field Robotics at the University of Sydney where she also received her B.E. in Aeronautical Space Engineering. Her research focuses on the development of information-based exploration strategies that can be applied within reinforcement learning frameworks to improve learning performance.



Joseph Nguyen is a Ph.D candidate at the Australian Centre for Field Robotics (ACFR) at The University of Sydney. He received a BE degree in Aeronautical (Space) Engineering from The University of Sydney in 2010. His research interests include aerial robotics and path planning for long-endurance autonomous flight.

Daniel B. Wilson received the B.E. degree in mechanical (space) engineering from The University of Sydney, in 2009. He is currently working towards the Ph.D. degree in the aerospace group at the Australian Centre for Field Robotics (ACFR), at The University of Sydney. In 2010, he worked at the Martin Aircraft Company, developing a flight control system for the Martin jetpack. His research interests include relative navigation for autonomous formation flight and guidance and control systems for aerial vehicles.

Salah Sukkarieh is the Professor of Robotics and Intelligent Systems at the University of Sydney, and the Director of Research and Innovation at the Australian Centre for Field Robotics. He has supervised over 10 research fellows, and graduated over 30 PhDs, 5 Masters and 75 honours students. Salah is on the editorial board for the Journal of Field Robotics, Journal of Autonomous Robots, and Transactions of Aerospace Systems, and has over 300 academic and industry publications in robotics and intelligent systems.

P. Martin
(Mines ParisTech)

E-mail : philippe.martin@mines-paristech.fr

DOI : 10.12762/2014.AL08-06

Accelerometers on Quadrotors : What do they Really Measure?

A revisited quadrotor model is proposed, including the so-called rotor drag. It differs from the model usually considered, even at first order, and much better explains the role of accelerometer feedback in control algorithms. The theoretical derivation is supported by experimental data.

Introduction

Quadrotor control has been an active area of investigation for several years. On the one hand, the quadrotor has several qualities, among them its very simple mechanical design, and qualifies as a viable concept of mini Unmanned Aerial Vehicle (UAV) for real-life missions ; on the other hand, it is perceived in the control community as a very rich case study in theoretical and applied control. The first control objective is to ensure a stable flight at moderate velocities and, in particular, in hovering; this fundamental building block is then used to develop higher-level tasks.

However, for experiments designed to work only in the lab with an off-board measuring device, e.g. [1], quadrotors all rely at the heart on strapdown MEMS inertial sensors (gyroscopes and accelerometers). These inertial sensors may be used alone (as far as horizontal stabilization is concerned) [2], or supplemented by other sensors, which usually provide some position-related information. Representative designs are: ultrasonic rangers [3]; (simple) GPS module when outdoors and infrared rangers when indoors [4] ; carrier phase differential GPS [5]; laser rangefinder [6]; vision system [7], [8], [9] ; laser rangefinder and vision system [10], [11]. Unfortunately those extra sensors have inherent drawbacks (low bandwidth, possible temporary unavailability, etc.), hence inertial sensors remain essential for basic stabilization.

Nearly all of the papers in the literature rely on the same physical model: only aerodynamic forces and moments proportional to the square of the propeller angular velocities are explicitly taken into account. Other aerodynamic effects are omitted and considered as small non-modeled disturbances to be rejected by the control law. The reason put forward is that these effects are proportional to the square of the quadrotor linear velocity, hence very small near hovering. Few authors explicitly consider other aerodynamic effects : [12] notes the importance of flapping stability derivatives; [13] and [14] consider aerodynamic effects without physical motivation linear with respect to the quadrotor linear and angular velocities, but propose very small numerical values; [5] judges them to be negligible at low velocities, and focuses on nonlinear aspects at moderate velocities; [15] physically motivates the presence of effects that are nearly linear with respect to the quadrotor linear and angular velocities, but provides no experimental data and is concerned only with the open-loop system.

Figure 1 - Our home-built quadrotor : the “Quadricopter”

On the other hand, the accelerometer measurement vector $\vec{a}$ can be used in two different ways (gyros are used in both cases ; see page 4 for more details about inertial sensors):

1) as an input, directly in the equation $\dot{\vec{V}} = \vec{g} + \vec{a}$ if extra sensors providing position or velocity information are available, using a sensor fusion algorithm that estimates the velocity and the pitch and roll angles

2) as an output, through the approximation $\vec{a} \approx -\vec{g}$. Accordingly, the pitch and roll angles are estimated by a sensor fusion algorithm. Commercial "attitude sensors", such as the 3DM-GX[1] or the MTi[2], run exactly on this principle.

In both cases, the sensor fusion algorithm can be an Extended Kalman Filter (EKF), a complementary filter, linear or nonlinear, or a nonlinear observer; see for example [16], [17] for an account of the two cases. Recall that MEMS inertial sensors are not accurate enough for "true" Schuler-based inertial navigation, see for example [18, Chap. 5] for details.

Now, a puzzling issue arises: the "conventional" physical model implies that the longitudinal and lateral (in body axes) accelerometers should always measure zero, which clearly contradicts 2) ; as for 1), even if no particular form of the accelerometers measurements is assumed, one may wonder about the interest of using measurements known to be zero (in addition corrupted by noise and biases). Nevertheless, many successful quadrotor flights have been reported, with control laws based on 1) or 2), or even both, and there is no question that using accelerometers is beneficial.

This paper, which largely draws on [19], proposes a "revisited" model containing extra aerodynamic terms proportional to the propeller angular velocity times the quadrotor linear or angular velocity. In particular, the so-called rotor drag, though rather small, appears at first order and is essential to correctly account for the accelerometer measurements.

The paper is structured as follows: the revisited model is derived in next section ; its main features are experimentally validated then ; finally, its implications for control schemes are discussed.

A revisited quadrotor model

Model of a single propeller "near" hovering

We first consider a single propeller rotating with angular velocity $\varepsilon_i \omega_i$ around its axis k_b; ω_i is positive, with $\varepsilon_1 = 1$ (resp. -1) for counterclockwise (resp. clockwise) rotation. Due to the motion of the quadrotor, the geometric center A_i of the propeller moves with linear velocity $\vec{V}_{A_i}$, while the rotor plane (by definition perpendicular to $\vec{k}_b$) undergoes angular velocity $\vec{\Omega}$; the total angular velocity of the propeller is thus $\vec{\Omega} + \varepsilon_1 \omega_i \vec{k}_b$. A lengthy derivation in the spirit of for example [20, in particular Chap. 5] shows that the aerodynamic efforts on the propeller resolve into the force $\vec{F}_i$ and moment $\vec{M}_i$ at A_i ,

$$\vec{F}_i = -a\omega_i^2 \vec{k}_b - \omega_i \left(\lambda_1 \vec{V}_{A_i}^{\perp} + \lambda_2 \vec{\Omega} \times \vec{k}_b \right) - \varepsilon_i \omega_i \left(\lambda_3 \vec{V}_{A_i} \times \vec{k}_b + \lambda_4 \vec{\Omega}^{\perp} \right) \quad (1)$$

$$\vec{M}_i = -b\varepsilon_i \omega_i^2 \vec{k}_b - \varepsilon_i S \omega_i \left(\mu_1 \vec{V}_{A_i}^{\perp} + \mu_2 \vec{\Omega} \times \vec{k}_b \right) - \omega_1 \left(\mu_3 \vec{V}_{A_i} \times \vec{k}_b + \mu_4 \vec{\Omega}^{\perp} \right) \quad (2)$$

where a, b, the λ_i's and μ's are positive constants ; the projection of a vector $\vec{U}$ on the rotor plane is denoted by

$$\vec{U}^{\perp} = \vec{k}_b \times \left(\vec{U} \times \vec{k}_b \right) = \vec{U} - \left(\vec{U} \cdot \vec{k}_b \right) \vec{k}_b$$

Moreover λ_2 and μ_2 are very small (they would be exactly zero if the blade axis were orthogonal to $\vec{k}_b$). Notice that all of the force and moment terms orthogonal to $\vec{k}_b$ arise from the velocity imbalance of the blade on a complete turn (because of the translational motion, the blade moves faster with respect to free air when it is advancing than when it is retreating).

The above relations rely on classical blade element theory, with two extra assumptions:

- the propeller is considered to be perfectly rigid, which is approximately true for most quadrotor propellers. The flapping due to the slight flexibility of a real propeller has only a marginal effect ;
- the components of the linear velocity $\vec{V}_{A_i}$ are considered small with respect to the propeller tip speed ; similarly the components of the angular velocity $\vec{\Omega}$ are considered to be small with respect to ω_i . This is valid "near" hovering, i.e., for "small" $\vec{V}_{A_i}$ and $\vec{\Omega}$: typically, the tip speed is of about $50\, m.s^{-1}$, so that $10\, m.s^{-1}$ can still be seen as a "small" velocity.

The velocities in the previous equations are of course velocities with respect to the air stream, not with respect to the ground. They coincide when there is no wind, which we assume in the sequel.

The term $\omega_i \lambda_1 \vec{V}_{A_i}^{\perp}$ in (1) is often called H-force or rotor drag in the helicopter literature. Also notice that the simplified expressions (1)-(2), though directly based on textbook helicopter aerodynamics, do not seem to appear in the literature under this compact form, very handy for control purposes. The reason for this is probably that helicopter literature is primarily concerned with articulated and/or rather flexible propellers, operating moreover at much higher ratios of linear velocity to propeller tip speed.

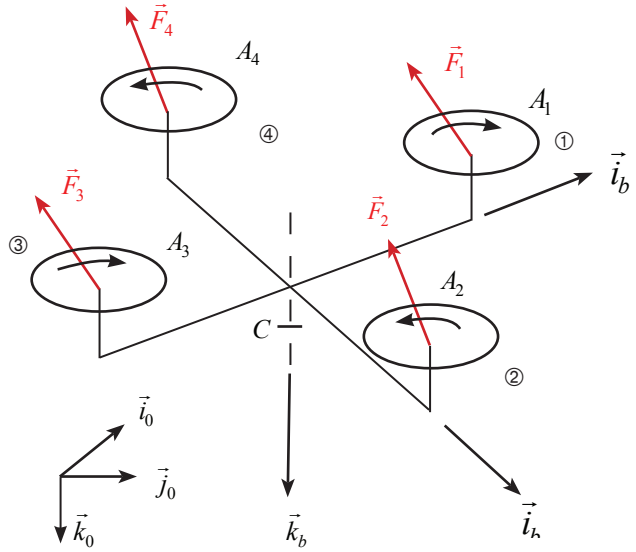


Figure 2 - Sketch of the complete quadrotor

Complete quadrotor model

The quadrotor consists of a rigid frame with four propellers, (directly) driven by electric motors, see figure 2. The structure is symmetrically

[1] www.microstrain.com
[2] www.xsens.com

arranged, with one pair of facing propellers rotating clockwise and the other pair rotating counterclockwise. The four propellers have the same axis $\vec{k}_b$.

$\vec{i}_b = \frac{\overrightarrow{A_3A_1}}{\|\overrightarrow{A_3A_1}\|}$, $\vec{j}_b = \frac{\overrightarrow{A_4A_2}}{\|\overrightarrow{A_4A_2}\|}$ and $\vec{k}_b$ thus form a direct coordinate frame.
Let A be the geometric center of the $A_1's$ and $l = \frac{1}{2}\|\overrightarrow{A_3A_1}\| = \frac{1}{2}\|\overrightarrow{A_2A_4}\|$;

clearly, $\sum_{i=1}^{4} \overrightarrow{AA_i} = 0$.

The whole system $\mathcal{B}$, with mass m and center of mass C, thus involves five rigid bodies : the frame/stator assembly $\mathcal{B}_0$ and the four propeller/motor assemblies $\mathcal{B}_i$. Clearly $\overrightarrow{CA} = h\vec{k}_b$ for some (signed) length h; notice that for most quadrotor designs h is very small. Resolved in the $(\vec{i}_b, \vec{j}_b, \vec{k}_b)$ frame, the velocity of C is written as $\vec{V}_C = u\vec{i}_b + v\vec{j}_b + w\vec{k}_b$ and the angular velocity of $\mathcal{B}_0$ is written as $\vec{\Omega} = p\vec{i}_b + q\vec{j}_b + r\vec{k}_b$

We assume that the only efforts acting on $\mathcal{B}$ are the weight and the aerodynamic efforts created by the propellers, as described in the previous section. In particular, we neglect the drag created by the frame, which is quadratic with respect to the velocity, hence small at low velocities with respect to the rotor drag. Newton's laws for the entire system are thus written as

$$m\,\dot{\vec{V}}_C = m\,\vec{g} + \sum_{i=1}^{4} \vec{F}_i \tag{3}$$

$$\dot{\vec{\sigma}}_C^{\mathcal{B}} = \sum_{i=1}^{4} \overrightarrow{CA_i} \times \vec{F}_i + \vec{M}_i \tag{4}$$

where $\vec{\sigma}_C^{\mathcal{B}} = \int_{\mathcal{B}} \overrightarrow{CM} \times \dot{\overrightarrow{CM}}\, d\mu(M)$ is the kinetic momentum of $\mathcal{B}$. For each $\mathcal{B}_i$, we can further write

$$\dot{\vec{\sigma}}_{A_i}^{\mathcal{B}_i} \cdot \vec{k}_b = \vec{M}_i \cdot \vec{k}_b + \varepsilon_i \Gamma_i \tag{5}$$

where $\vec{\sigma}_{A_i}^{\mathcal{B}_i} = \int_{\mathcal{B}_i} \overrightarrow{A_iM} \times \dot{\overrightarrow{A_iM}}\, d\mu(M)$ is the kinetic momentum of $\mathcal{B}_i$ and Γ_i is the (positive) electromagnetic torque of the motor. For simplicity, we have considered A_i as the center of mass of $\mathcal{B}_i$ (in fact the two points are slightly apart). We also consider the $\Gamma_{i'}s$ as the control inputs (it is nevertheless easy to include the behavior of the electric motors, both for modeling and control).

We now evaluate the right-hand sides of (3)-(4). Since

$$\vec{V}_{A_i} = \vec{V}_C + \overrightarrow{CA} + \overrightarrow{AA_i} = \vec{V}_C + h\vec{\Omega} \times \vec{k}_b + \vec{\Omega} \times \overrightarrow{AA_i}$$

we have

$$\begin{aligned}
&\lambda_3 \vec{V}_{A_i} \times \vec{k}_b + \lambda_4 \vec{\Omega}^{\perp} \\
&\quad = \lambda_3 (\vec{V}_C + h\vec{\Omega} \times \vec{k}_b + \vec{\Omega} \times \overrightarrow{AA_i}) \times \vec{k}_b + \lambda_4 \vec{\Omega}^{\perp} \\
&\quad = \lambda_3 \vec{V}_C \times \vec{k}_b + \lambda'_4 \vec{\Omega}^{\perp} + r\lambda_3 \overrightarrow{AA_i}
\end{aligned}$$

$$\begin{aligned}
&\lambda_1 \vec{V}_{A_i}^{\perp} + \lambda_2 \vec{\Omega} \times \vec{k}_b \\
&\quad = \lambda_1 (\vec{V}_C^{\perp} + (h\vec{\Omega} \times \vec{k}_b)^{\perp} + (\vec{\Omega} \times \overrightarrow{AA_i})^{\perp} + \lambda_2 \vec{\Omega}^{\perp} \quad k_b \\
&\quad = \lambda_1 \vec{V}_C^{\perp} + \lambda'_2 \vec{\Omega} + \vec{k}_b - r\lambda_1 \overrightarrow{AA_i} \times \vec{k}_b
\end{aligned}$$

where we have used the fact that $\overrightarrow{AA_i}$ is collinear to either $\vec{i}_b$ or $\vec{j}_b$, and set $\lambda'_2 = \lambda_1 + h\lambda_1$ and $\lambda'_4 = \lambda_4 + h\lambda_3$. Therefore,

$$\begin{aligned}
\sum_{i=1}^{4} \vec{F}_i &= -a\left(\sum_{i=1}^{4} \omega_i^2 \vec{k}_b\right) - \left(\sum_{i=1}^{4} \omega_i\right)(\lambda_1 \vec{V}_C^{\perp} + \lambda'_2 \vec{\Omega} \times \vec{k}_b) \\
&\quad - \left(\sum_{i=1}^{4} \varepsilon_i \omega_i\right)(\lambda_3 \vec{V}_C \times \vec{k}_b + \lambda'_4 \vec{\Omega}^{\perp}) \\
&\quad + r\lambda_1 \left(\sum_{i=1}^{4} \omega_i \overrightarrow{AA_i}\right) \times \vec{k}_b - r\lambda_3 \left(\sum_{i=1}^{4} \varepsilon_i \overrightarrow{AA_i}\right) \\
&\approx -a\left(\sum_{i=1}^{4} \omega_i^2\right)\vec{k}_b - \left(\sum_{i=1}^{4} \omega_i\right)(\lambda_1 \vec{V}_C^{\perp} + \lambda'_2 \vec{\Omega} \times \vec{k}_b)
\end{aligned}$$

In the last line, we have neglected small terms according to the second extra assumption of the single properller model. Indeed, in hovering $\vec{V}_C$ and $\vec{\Omega}$, hence $\vec{V}_{A_i}$ are zero ; from (1)–(4) this implies that $a(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) = mg$
and $\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 = \omega_1^2 - \omega_3^2 = \omega_2^2 - \omega_4^2 = 0$,
and eventually $\omega_i = \bar{\omega} = \sqrt{\frac{mg}{4a}}$.

As a consequence $\sum_{i=1}^{4} \varepsilon_i \omega_i$, $\frac{1}{l}\sum_{i=1}^{4} \omega_i \overrightarrow{AA_i}$ and $\frac{1}{l}\sum_{i=1}^{4} \varepsilon_i \omega_i, \overrightarrow{AA_i}$ also vanish in hovering ; "near" hovering they are therefore small with respect to $\sum_{i=1}^{4} \omega_i$.

Similar computations yield

$$\begin{aligned}
&\sum_{i=1}^{4} \overrightarrow{CA_i} \times \vec{F}_i + \overrightarrow{AA_i} \times \vec{F}_i + \vec{M}_i \\
&\quad \approx -a\left(\sum_{i=1}^{4} \omega_i^2 \overrightarrow{AA_i}\right) \times \vec{k}_b - b\left(\sum_{i=1}^{4} \varepsilon_i \omega_i^2\right)\vec{k}_b \\
&\quad - r\lambda_1 l^2 \left(\sum_{i=1}^{4} \omega_1\right)\vec{k}_b - \left(\sum_{i=1}^{4} \omega_1\right)(\mu'_3 \vec{V}_C \times \vec{k}_b + \mu''_4 \vec{\Omega}^{\perp})
\end{aligned}$$

where $\mu'_3 = \mu_3 - h\lambda_1$ and $\mu''_4 = \mu_4 + h(\mu_1 + \lambda'_2)$.

Notice that the contributions of λ_3, λ_4 in the forces (1) and of μ_1, μ_2 in the moments (2) (nearly) cancel out in the right-hand sides of (3)-(4), due to the fact there are two clockwise and two counterclockwise-rotating propellers.

We then evaluate the left-hand sides of (3)–(5). The approach is fairly standard.

$$\begin{aligned}
\vec{\sigma}_C^{\mathcal{B}} &= \int_{\mathcal{B}} \overrightarrow{CM} \times \dot{\overrightarrow{CM}}\, d\mu(M) \\
&= \int_{\mathcal{B}_0} \overrightarrow{CM} \times \dot{\overrightarrow{CM}}\, d\mu(M) \\
&\quad + \sum_{i=1}^{4} \int_{\mathcal{B}_1} \overrightarrow{CM} \times (\dot{\overrightarrow{CA_i}} + \dot{\overrightarrow{A_iM}})\, d\mu(M) \\
&= \int_{\mathcal{B}_0} \overrightarrow{CM} \times (\vec{\Omega} \times \overrightarrow{CM})\, d\mu(M) \\
&\quad + \sum_{i+1}^{4} \int_{\mathcal{B}_i} \overrightarrow{CM} \times (\vec{\Omega} \times \overrightarrow{CA_i} + (\vec{\Omega} + \varepsilon_i \omega_i \vec{k}_b) \times \overrightarrow{A_iM})\, d\mu(M) \\
&= \int_{\mathcal{B}} \overrightarrow{CM} \times (\vec{\Omega} \times \overrightarrow{CM})\, d\mu(M) \\
&\quad + \sum_{i=1}^{4} \int_{\mathcal{B}_1} \overrightarrow{A_iM} \times (\varepsilon_i \omega_i \vec{k}_b \times \overrightarrow{A_iM})\, d\mu(M) \\
&= \mathcal{I}_C^{\mathcal{B}} \cdot \vec{\Omega} + \sum_{i=1}^{4} \varepsilon_i \omega_i (\mathcal{I}_{A_i}^{\mathcal{B}_i} \cdot \vec{k}_b) \\
&= Ip\vec{i}_b + Iq\vec{j}_b + (Jr + J_r \sum_{i=1}^{4} \varepsilon_i \omega_i)\vec{k}_b
\end{aligned}$$

where $I; J; J_r$ are strictly positive constants. In the last equation, in the computation of the inertia tensors $\mathcal{I}^{\mathcal{B}}$, $\mathcal{I}_{A_i}^{\mathcal{B}_i}$ we have replaced

the actual propellers by disks with the same masses and radii, and taken advantage of the various symmetries ; this "averaging" approximation is justified by the fact that the propeller angles vary much faster than all of the other kinematic variables (besides, this approximation is already heavily used in the blade element theory used to derive (1)-(2)). Using the same approximation,

$$\begin{aligned}\vec{\sigma}_{A_i}^{\mathcal{B}_i} &= \int_{\mathcal{B}_i} \overrightarrow{A_iM} \times \overrightarrow{A_iM}\, d\mu(M) \\ &= \int_{\mathcal{B}_i} \overrightarrow{A_iM} \times (\vec{\Omega} + \varepsilon_i \omega_i \vec{k}_b) \times \overrightarrow{A_iM})\, d\mu(M) \\ &= \mathcal{I}_{A_i}^{\mathcal{B}_i} . (\vec{\Omega} + \varepsilon_i \omega_i \vec{k}_b) \\ &= I_r p \vec{i}_b + I_r q \vec{j}_b + J_r (r + \varepsilon_i \omega_i) \vec{k}_b\end{aligned}$$

where I_r is a strictly positive constant. Eventually,

$$\begin{pmatrix} \dot{\vec{V}}_C \cdot \vec{i}_b \\ \dot{\vec{V}}_C \cdot \vec{j}_b \\ \dot{\vec{V}}_C \cdot k_b \end{pmatrix} = \begin{pmatrix} \dot{u} + qw - ru \\ \dot{v} + ru - pw \\ \dot{w} + pu - qu \end{pmatrix}$$

$$\begin{pmatrix} \dot{\vec{\sigma}}_C^{\mathcal{B}} . \vec{i}_b \\ \dot{\vec{\sigma}}_C^{\mathcal{B}} . \vec{j}_b \\ \dot{\vec{\sigma}}_C^{\mathcal{B}} . \vec{k}_b \end{pmatrix} = \begin{pmatrix} I\dot{p} + (J - I) qr + J_r q \sum_{i=1}^4 \varepsilon_i \omega_i \\ I\dot{q} + (J - I) pr + J_r p \sum_{i=1}^4 \varepsilon_i \omega_i \\ J\dot{r} + J_r \sum_{i=1}^4 \varepsilon_i \dot{\omega}_i \end{pmatrix}$$

$$\dot{\vec{\sigma}}_{C_i}^{\mathcal{B}_i} . \vec{k}_b = J_r (\dot{r} + \varepsilon_i \dot{\omega}_i) \qquad i = 1,2,3,4$$

To describe the orientation of the quadrotor, we use the classical ϕ, θ, Ψ Euler angles (quaternions could of course be used). The direction cosine matrix $R_{\phi,\theta,\Psi}$ to convert from Earth coordinates to aircraft coordinates is then

$$\begin{pmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ S\phi S\theta C\psi - C\phi S\psi & S\phi S\theta S\psi + C\phi C\psi & S\phi C\theta \\ C\phi S\theta C\psi + S\phi S\psi & C\phi S\theta S\psi - S\phi S\psi & C\phi C\theta \end{pmatrix}$$

so that $g = g(-i \sin\theta + j \sin\phi\cos\theta + k \cos\phi\cos\theta)$.

Collecting the previous findings (3)–(5), we eventually have

$$\dot{u} + qw - rv = -g \sin\theta - u \frac{\lambda_1}{m} \sum_{i=1}^4 \omega_i \tag{6}$$

$$\dot{v} + ru - pw = g \sin\phi \cos\theta - v \frac{\lambda_1}{m} \sum_{i=1}^4 \omega_i \tag{7}$$

$$\dot{w} + pv - qu = g - \cos\phi\cos\theta - \frac{a}{m} \sum_{i=1}^4 \omega_i^2 \tag{8}$$

$$\begin{aligned} I\dot{p} + (J - I) qr + J_r q \sum_{i=1}^4 \varepsilon_i \omega_i \\ = a(\omega_1^2 - \omega_3^2) + (\mu_3' v + \mu_4'' p) \sum_{i=1}^4 \omega_i \end{aligned} \tag{9}$$

$$\begin{aligned} I\dot{q} - (J - I) pr - J_r p \sum_{i=1}^4 \varepsilon_i \omega_i \\ = a(\omega_1^2 - \omega_3^2) + (\mu_3' u - \mu_4'' q) \sum_{i=1}^4 \omega_i \end{aligned} \tag{10}$$

$$(J - 4J_r)\dot{r} = -\lambda_1 l^2 \sum_{i=1}^4 \omega_i - \sum_{i=1}^4 \varepsilon_i \Gamma_i \tag{11}$$

$$J_r (\varepsilon_i \dot{r} + \omega_i) = \Gamma_i - b\omega_i^2 \qquad i = 1,2,3,4 \tag{12}$$

In (6)–(7) we have assumed $\lambda'_2 = \lambda_2 + h\lambda_1 \approx 0$, which is sensible since λ_2 and h are nearly 0 (notice that $\lambda'_2 = 0$ can always be enforced by slightly shifting the center of mass). Finally, the angles and angular velocities are linked by

$$\dot{\phi} = p + (q \sin\phi + r \cos\phi) \tan\phi \tag{13}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{14}$$

$$\dot{\psi} = \frac{q \sin\phi + r \cos\phi}{\cos\theta} \tag{15}$$

Equations (6)–(15) form the complete 13-dimensional nonlinear model of the quadrotor.

A further simplification is to replace $\sum_{i=1}^4 \omega_i$ by $4\bar{\omega}$ in (6) – (12) since ω_i remains close to $\bar{w}$ in normal flight and moreover use the fact that the propeller moment of inertia J_r is very small with respect to $J - I$; this yields

$$\dot{u} + qw - rv = -g \sin\theta - \frac{4\bar{\omega}\lambda_1}{m} u \tag{16}$$

$$\dot{v} + ru - pw = -g \sin\phi\cos\theta - \frac{4\bar{\omega}\lambda_1}{m} v \tag{17}$$

$$\dot{w} + pv - qu = g \cos\phi\cos\theta - \frac{a}{m} \sum_{i=1}^4 \omega_i^2 \tag{18}$$

$$I\dot{p} + (J - I) qr = a(\omega_4^2 - \omega_2^2) 4\bar{\omega}(\mu'_3 v + \mu''_4 p) \tag{19}$$

$$I\dot{q} + (J - I) qr = a(\omega_4^2 - \omega_2^2) 4\bar{\omega}(\mu'_3 u + \mu''_4 p) \tag{20}$$

$$J\dot{r} = -4\bar{\omega}\lambda_1 l^2 r - \sum_{i=1}^4 \varepsilon_i \Gamma_i \tag{21}$$

$$J_r \dot{\omega}_i - \varepsilon_i \frac{4\bar{\omega}\lambda_1 l^2 J}{I} r = \Gamma_i - b\omega_1^2, \quad i = 1,2,3,4 \tag{22}$$

Equations (16)–(22) can be used instead of (6)–(12) with no noticeable loss of accuracy.

Model of the inertial sensors

The quadrotor is equipped with strapdown triaxial gyroscope and accelerometer. Without restriction, we assume that the sensing axes coincide with $\vec{i}_b, \vec{j}_b, \vec{k}_b$. The gyroscope measures the angular velocity $\vec{\Omega}$, projected on its sensing axes, i.e., $(g_x, g_y, g_z) = (p, q, r)$, the accelerometer measures the specific acceleration $\vec{a} = \dot{\vec{V}}_P - \vec{g}$ of the point P where it is located, projected on its sensing axes; see for example [18, Chap. 4] for details on inertial sensors. Hence, by (3) if the accelerometer is located at the center of mass C, which is the case for most quadrotors, it measures

$$\vec{a} = \dot{\vec{V}}_C - \vec{g} = \frac{1}{m} \sum_{i=1}^4 \vec{F}_i$$

by (3), the accelerometer thus measures

$$a_x = \vec{a} . \vec{i}_b = -\frac{\lambda_1}{m} (\omega_1 + \omega_2 + \omega_3 + \omega_4) u \tag{23}$$

$$a_y = \vec{a} . \vec{j}_b = -\frac{\lambda_1}{m} (\omega_1 + \omega_2 + \omega_3 + \omega_4) v \tag{24}$$

$$a_z = \vec{a} . \vec{k}_b = -\frac{a}{m} (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{25}$$

As in the previous section we can replace (23)–(24), without a noticeable loss of accuracy, by

$$a_x = -\frac{4\bar{\omega}\lambda_1}{m}u \tag{26}$$

$$a_y = -\frac{4\bar{\omega}\lambda_1}{m}v \tag{27}$$

This shows that a_x, a_y actually measure the quadrotor longitudinal and lateral velocities (while a_z measures the total thrust).

Linearized model

To highlight the salient features of the revisited model (6)–(15) and its measurements, it is enough to consider its first order approximation. Suitably putting together variables, this linearized model can be divided into four independent subsystems :

- longitudinal subsystem (states $u,\ \theta,\ q, \omega_1 - \omega_3$; input $\Gamma_1 - \Gamma_3$ measurements $a_x \approx -\frac{4\bar{\omega}\lambda_1}{m}u$ and $g_y \approx q$)

$$\dot{u} \approx -g\theta - \frac{4\bar{\omega}\lambda_1}{m}u$$
$$\dot{\theta} \approx q$$
$$I\dot{q} \approx 4\bar{\omega}\mu'_3\, u - 4\bar{\omega}\mu''_4\, q + 2al\bar{\omega}(\omega_1 - \omega_3)$$
$$J_r(\dot{\omega}_1 - \dot{\omega}_3) \approx \Gamma_1 - \Gamma_3 - 2b\bar{\omega}(\omega_1 - \omega_3)$$

- lateral subsystem (states $v,\ \phi,\ p,\ \omega_4 - \omega_2$; input $\Gamma_4 - \Gamma_2$ measurements $a_y \approx -\frac{4\bar{\omega}\lambda_1}{m}\ v$ and $g_x \approx p$)

$$\dot{v} \approx -g\phi - \frac{4\bar{\omega}\lambda_1}{m}v$$
$$\dot{\phi} \approx p$$
$$I\dot{p} \approx -4\bar{\omega}\mu'_1\,\bar{\omega}v - 4\mu''_4\,\bar{\omega}p + 2al\bar{\omega}(\omega_4 - \omega_2)$$
$$J_r(\dot{\omega}_4 - \dot{\omega}_2) \approx \Gamma_4 - \Gamma_2 - 2b\bar{\omega}(\omega_4 - \omega_2)$$

- vertical subsystem (states $w, \sum_{i=1}^4 \omega_i$ input $\sum_{i=1}^4 \Gamma_i$ measurement $a_z \approx -g - \frac{2a\bar{\omega}}{m}\sum_{i=1}^4 \omega_i)$

$$\dot{w} \approx \frac{2a\bar{\omega}}{m}\left(4\bar{\omega} - \sum_{i=1}^4 \omega_i\right)$$
$$J_r \sum_{i=1}^4 \dot{\omega}_i \approx \sum_{i=1}^4 \Gamma_i - 2b\bar{\omega}\sum_{i=1}^4 \Gamma_i$$

- heading subsystem (states $\psi;\ r; \sum_{i=1}^4 \varepsilon_i\omega_i$; input $\sum_{i=1}^4 \varepsilon_i\Gamma_i$ measurement $g_z \approx r$)

$$\dot{\psi} \approx r$$
$$J\dot{r} \approx -4\bar{\omega}\lambda_1 l^2 r - \sum_{i=1}^4 \varepsilon_i\Gamma_i$$
$$J_r \sum_{i=1}^4 \varepsilon_i\dot{\omega}_i \approx \frac{16\bar{\omega}\lambda_1 l^2 J_r}{J}r + \sum_{i=1}^4 \varepsilon_i\Gamma_i - 2b\bar{\omega}\sum_{i=1}^4 \varepsilon_i w_i$$

In the sequel we concentrate on the longitudinal system, where accelerometer feedback is of paramount importance (the lateral subsystem is the same up to a sign-reversing coordinate change). Setting $\omega_q = \omega_1 - \omega_3$, $\Gamma_q = \frac{\Gamma_1 - \Gamma_3}{J_r}$ and

$$(f_1, f_2, f_3, f_4, f_5) = \left(\frac{4\bar{\omega}\lambda_1}{m}, \frac{4\bar{\omega}\mu'_3}{I}, \frac{4\bar{\omega}\mu''_4}{I}, \frac{2al\bar{\omega}}{I}, \frac{2b\bar{\omega}}{J_r}\right)$$

the longitudinal subsystem is thus written as

$$\dot{u} = -f_1 u - g\theta \tag{28}$$
$$\dot{\theta} = q \tag{29}$$
$$\dot{q} = f_2 u - f_3 q + f_4\omega_q \tag{30}$$
$$\dot{\omega}_q = \Gamma_q - f_5\omega_q \tag{31}$$

with measurements $a_x = -f_1 u$ and $g_y = q$

Departure from the "conventional" model in the literature

Most authors consider a propeller model with only the $\vec{k}_b$ terms in (1)-(2), i.e., with all λ_i's and μ_i's equal to zero. Hence, the "conventional" model is the same as the revisited one but with the λ_i's and μ_i's equal to zero in (16)–(22) and (26)– (27).

However, there is obviously a problem with such a model : indeed $\vec{a} = \dot{\vec{V}}_C - \vec{g}$ is collinear with $\vec{k}_b$, hence $a_x = a_y = 0$, which is certainly not very useful for feedback. This paradox is usually not acknowledged and the approximation $\vec{a} \approx -\vec{g}$ is used instead, i.e.,

$$(a_x, a_y, a_z) \approx (g\sin\theta, -g\sin\phi\cos\theta, -g\cos\phi\cos\theta) \tag{32}$$

The reason proposed is that $\dot{\vec{V}}_C$ is small near hovering, at least in average. This is indeed true if the aircraft is stabilized by some extraneous means (such as a human pilot), but is a questionable assumption to use from a closed loop perspective. Nevertheless, many successful flights with controllers based on this approximation have been reported. In IV-C, we suggest an explanation reconciling all of those facts in the light of the revisited quadrotor model.

The resulting "conventional" longitudinal subsystem is then

$$\dot{u} = -g\theta \tag{33}$$
$$\dot{\theta} = q \tag{34}$$
$$\dot{q} = f_4\omega_q \tag{35}$$
$$\dot{\omega}_q = \Gamma_q - f_5\omega_q \tag{36}$$

with measurements $a_x = g\theta$ and $g_y = q$, to be compared with (28)–(31) with measurements $a_x = -f_1 u$ and $g = q$

Experimental validation

Experimental setup

To validate the model, we recorded flight data with our home-built "Quadricopter", see figure 1. Due to limitations of our experimental setup, we could collect data to validate only the force model (28), but not the moment model (30); this is nevertheless the most important part of the model, since it accounts for the accelerometer measurements. The quadrotor was fitted with a MIDG2 "GPS-aided Inertial Navigation System"3 and a radio data link to the ground station. The MIDG2 consists of a triaxial accelerometer, a triaxial gyroscope, a triaxial magnetometer, a GPS engine and an on-board computer. The raw measurements are fused by an EKF on the onboard computer to

provide estimates of the orientation and of the velocity vector with respect to the Earth axes. The MIDG2 is an "independent" device with no knowledge of the specific system that it is fitted on; it heavily relies on the GPS engine for good dynamic estimates, without using assumption (32). All of the data can be issued at a pace of up to 20 ms. Due to the low throughput of the radio data link, only the accelerometer raw measurements a_{xm}, a_{ym} and the MIDG2-computed quantities ϕ_m, θ_m, ψ_m and V_x, V_y, V_z were transmitted to the ground station, at the reduced pace of 40 ms.

We flew the quadrotor performing repeated back and forth translations at a (nearly) constant altitude and recorded one minute of flight data. Since a GPS module is used, the test was conducted outdoors, on a very calm day to respect the no-wind assumption.

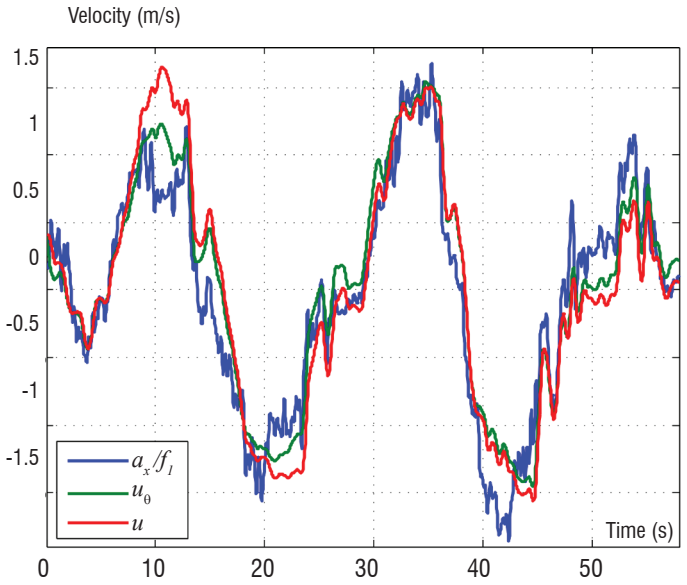


Figure 3 - Comparison between $\frac{a_x}{f_1}, u_\theta$ and u

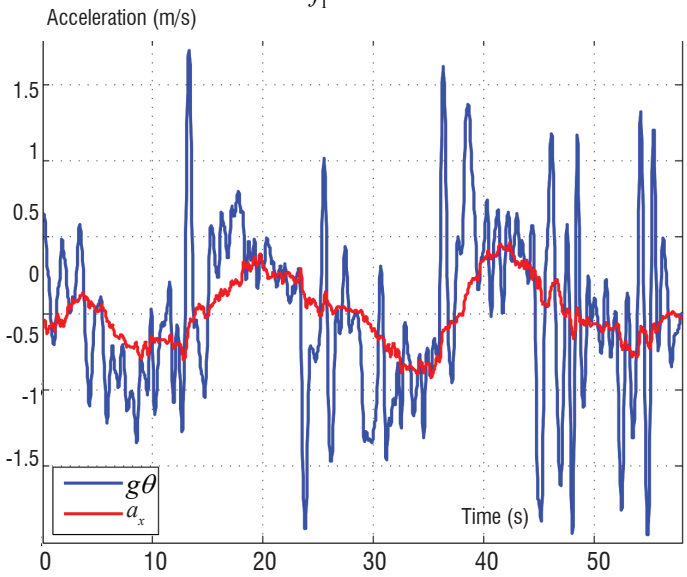


Figure 4 - Comparison between a_x and $g\theta$

Validation of the force model

Due to an imperfect mechanical design of our quadrotor, the MIDG2 case is not exactly aligned with the quadrotor frame, but rather tilted by the unknown (small) angles ϕ_0, θ_0, ψ_0 . The angle and acceleration data must be rotated accordingly to be expressed with respect to the quadrotor axes (the velocity data requires no correction, since it is expressed relative to the Earth axes), that is

$$(\phi, \theta, \psi) = (\phi_m - \phi_o, \theta_m - \theta_0, \psi_m - \psi_0)$$

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = R^T_{\phi_0,\theta_0,\psi_0} \begin{pmatrix} a_{xm} \\ a_{ym} \\ a_{zm} \end{pmatrix}$$

Dropping higher-order terms, this yields

$$a_x \approx a_{xm} - \psi_0 a_{ym} + \theta_0 a_{zm} \approx a_{xm} - \theta_0 g$$

$$a_y \approx \psi_0 a_{xm} + a_{ym} - \theta_0 a_{zm} \approx a_{ym} - \phi_0 g$$

The velocity vector in relation to the body axes is obtained by

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = R_{\phi,\theta,\psi} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

and is considered as the "true" reference velocity to validate our modeling assumptions.

We also wanted to compute the velocities u_θ and u_ϕ predicted by the integration of the linearized force model (28)

$$\dot{u}_\theta = -f_1 u_\theta - g\theta$$

$$\dot{v}_\phi = -f_1 v_\phi + g_\phi$$

with initial conditions $u(0)$ $u(0)$ and $v_\phi(0) = v(0)$.

The task was then to adjust $f_1, \phi_0, \theta_0, \psi_0$ to get a good fit between $-\frac{a_x}{f_1}, u$ and u_θ on the one hand, and between $-\frac{a_y}{f_1}, v$ v and v_ϕ on the other hand. Since the accelerometer data are quite noisy and requires some filtering, the same filter (5th order Bessel filter with 2 Hz cutoff frequency) was applied to all of the data, in order to preserve the transfer functions among them.

With $(f_1, \phi_0, \theta_0, \psi_0) = (0.25s^{-1}, 1.2°, -2.4°, 2°)$ the agreement is good between the "true" (i.e., MIDG2-given) velocity u, the "accelerometer-based" velocity $-\frac{a_x}{f_1}$ and the velocity u_θ "predicted" by the model from the "true" (MIDG2-given) pitch angle, see figure 3, which reasonably validates our force model. The agreement between $v, -a_y / f_1$ and v_ϕ not shown for lack of space, is equally good.

To test the conventional approximation (32) we also plotted $(a_x, g\theta)$, see figure 4. Though the trend is roughly correct, the fit is much worse ; the result is similar for $(a_y, -g\phi)$.

Other validations of the force model in the literature

Since the publication of [19] several authors have experimentally confirmed (using a motion tracking system) the proposed force model [21], [22], [23].

Implications for control schemes

We now investigate the relevance of the revisited model in the presence of a feedback controller, with (section IV-A) and without (sections IV-B and IV-C) velocity measurements. We use the numerical values

$(f_1, f_2, f_3, f_4, f_5) = (0.25, 0.76, -9.8, 0.34, 12.74)$

f_1 was determined from flight tests and f_4, f_5 were determined from static tests on the motor-propeller subsystems. The aerodynamic coefficients f_2, f_3 were derived analytically ; their values are plausible but nevertheless questionable.

Two-time-scale "full-state" feedback

We first assume that the entire state is known, or which turns out to be equivalent, that u and q are measured without noise so that they can be used in ideal Proportional-Derivative (PD) controllers. It is customary to design a two-time-scale control law, with a fast inner loop to control q, ω_q and a slow outer loop to control u, θ.

The fast inner loop is the ideal PD controller

$$\Gamma_q = -\frac{k_p}{\varepsilon^2} q - \frac{k_d}{\varepsilon} \dot{q} + \frac{k_p}{\varepsilon^2} q_r$$

where q_r is the desired pitch rate ; k_p, k_d are the PD gains and $\varepsilon > 0$ is a "small" parameter. Applying this feedback to (28)-(31) yields

$$\dot{u} = -f_1 u - g\theta$$
$$\dot{\theta} = q$$
$$\varepsilon \dot{q} = f_4 \tilde{\omega}_q + \mathcal{O}(\varepsilon)$$
$$\varepsilon \dot{\tilde{\omega}}_q = -k_p q - f_4 k_d \tilde{\omega}_q + k_p q_r + \mathcal{O}(\varepsilon)$$

where $\tilde{\omega}_q = \varepsilon \omega_q$. From standard arguments of singular perturbation theory [24], the convergence of the fast variables is up to order ε ruled by the well-known coefficient f_4 and the PD gains ; and the behavior of the slow variables u, θ is up to order ε ruled by the slow approximation

$$\dot{u} = -f_1 u - g\theta \tag{37}$$

$$\dot{\theta} = q_r \tag{38}$$

Hence, the role of the aerodynamic coefficients f_2, f_3 is marginal if the inner loop is fast enough.

The slow outer loop is the ideal PD controller

$$q_r = k_1 u + k_2 \dot{u} - k_1 u_r$$

where u_r is the desired velocity and k_1, k_2 the PD gains. Applying this feedback to (37)-(38) yields

$$\dot{u} = -f_1 u - g\theta$$

$$\dot{\theta} = (k_1 - f_1 k_2) u - g k_2 \theta - k_1 u_r$$

with characteristic polynomial $s^2 + (f_1 + g k_2) s + g k_1$. A reasonable closed-loop settling time is of about 1s, which requires $g k_1 = 6^2$ and $f_1 + g k_2 = 6\sqrt{2}$. This means that $f_1 = 0.25$ is negligible with respect to the effect of the controller.

We thus see that the revisited moment equation (30) does not really matter if the gyroscope measurements are good enough for a fast loop, which is usually the case in practice; nevertheless, taking into account f_2 and especially f_3 may help to design a better inner loop. As for the force model (28), it does not really matter either, provided that a velocity measurement is available, which agrees with [5]. The importance of f_1 is nevertheless paramount to account for the accelerometer measurements, as will be seen in the following sections.

Conventional interpretation of accelerometer feedback

Once the inner loop is closed, the usual slow model is

$$\dot{u} = -g\theta$$

$$\dot{\theta} = q_r$$

with measurement $a_x = g\theta$. Since the velocity u is clearly not observable, the role of the outer loop is simply to control the measured angle θ. In theory, the simple proportional feedback

$$q_r = k(\theta_r - \frac{a_x}{g})$$

does the trick, but in practice the accelerometer measurements are too noisy to be used directly (not only because of the intrinsic sensor noise, but also because of mechanical vibrations). Instead, an "angle estimator" is often used, based on the model $\dot{\theta} = q$ with measurements $a_x = g\theta$ and $g_y = q$. A more elaborate estimator, for example an EKF or a nonlinear observer, can also be used, see the references in the introduction; it is then based on the nonlinear kinematic equations (13)–(15) and relies on the approximation (32). Whatever the filter, the first-order approximation is essentially the linear observer

$$\dot{\hat{\theta}} = g_y + l(\frac{a_x}{g} - \hat{\theta})$$

it can also be seen as a complementary filter, since its transfer function is $\hat{\theta} = \frac{s}{s+l}\theta_q + \frac{l}{s+l}\theta_{a_x}$ where $\theta_q = \frac{q}{s}$ is the pitch angle obtained from gyro integration and $\theta_{a_x} = \frac{a_x}{g}$ is the pitch angle given by the accelero.

The outer loop is thus the controller-observer

$$q_r = k(\theta_r - \hat{\theta}) \tag{39}$$

$$\dot{\hat{\theta}} = q + l\left(\frac{a_x}{g} - \hat{\theta}\right) \tag{40}$$

Applied to the usual model and defining the observation error $e_\theta = \hat{\theta} - \theta$ it yields the closed-loop system

$$\dot{u} = -g\theta$$

$$\dot{\theta} = k(\theta_r - \theta - e_\theta)$$

$$\dot{e}_\theta = -l e_\theta$$

For θ_r constant, the last two equations have the unique steady state $(\theta, e_\theta) = (\theta_r, 0)$. The characteristic polynomial is $\Delta_0 = (s+k)(s+l)$, and the closed-loop transfer functions are

$$\theta = \frac{k}{s+k}\theta_r \tag{41}$$

$$u = \frac{-gk}{s(s+k)}\theta_r \tag{42}$$

Provided that $k, l > 0$ we have as desired $(\theta, e_\theta) \to (\theta_r, 0)$, while u grows linearly unbounded. For robustness, a good tuning of (39)-(40) requires that the controller and observer act in distinct time

scales (Loop Transfer Recovery), i.e., $l >> k$ or $k>>l$. We consider in the sequel a "slow" observer, which is representative of commercial "angle sensors" such as the 3DM-GX, and a "fast" controller ; for a settling time of about $1s$, we choose for example $k = 1/0.3$ and $l = 1/12$

We tested this control scheme experimentally, with a rather satisfying result : the angle θ reaches the desired θ_r, though the dynamics are somewhat more sluggish than expected. The usual analysis could thus be considered as reasonably justified. Nevertheless, it does not account for the following experimental observations already visible to the naked eye :

- when pushed away from hovering, the quadrotor returns to hovering (of course at a different position)
- when flying at a constant velocity u , the angle θ is not zero, but rather approximately proportional to u
- in response to a constant θ_r, u does not grow unbounded, but rather reaches a value approximately proportional to θ_r.

Though these experimental facts are well-known to people in the field, they do not seem to be reported in the literature. The discrepancy is usually attributed to the neglected second order aerodynamical drag and the inevitably imperfect experimental conditions. Another more subtle discrepancy is that the observer gain l must be smaller than that predicted by the theory, in order to avoid a badly damped transient (e.g., $l=1/3$ does not work well in practice).

As will be seen in the following section, these experimental facts can be explained by the revisited model.

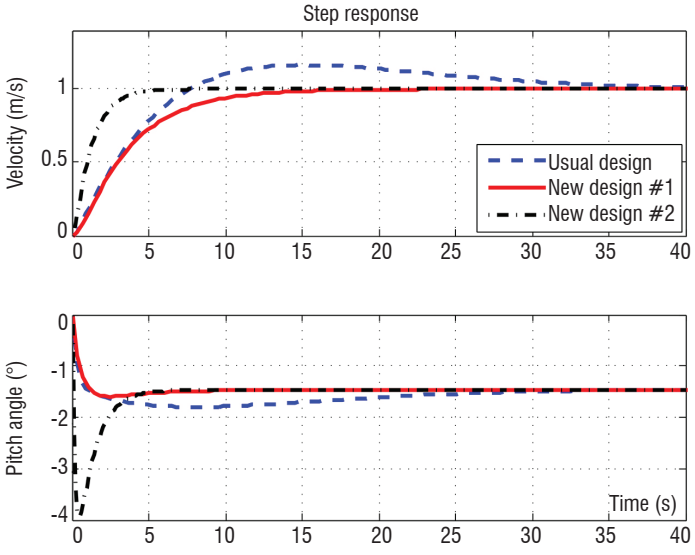


Figure 5 - Comparison between control schemes (simulation)

Revisited interpretation of accelerometer feedback

We now apply the controller-observer (39)-(40) to the revisited longitudinal model. The closed-loop system is now

$$\dot{u} = -f_1 u - g\theta$$

$$\dot{\theta} = k(\theta_r - \theta - e_\theta)$$

$$\dot{e}_\theta = -l\left(\frac{f_1}{g}u + \theta + e_\theta\right)$$

with $e_\theta = \hat{\theta} - \theta$. For θ_r constant, the only steady state is $(u, \theta, e_\theta) = (-\frac{g}{f_1}\theta_r, \theta_r, 0)$; the characteristic polynomial is

$$\Delta = s^3 + (k + l + f_1)s^2 + f_1(k + l)s + f_1 kl$$

If $k >> l$, $\Delta \simeq (s + k)(s^2 + f_1 s + f_1 l)$ so that the closed-loop system is stable as soon as $k,l>0$. Hence $\theta \to \theta_r$ as desired, and $e_\theta \to 0$ as expected from the observer ; u now tends to the finite value $-\frac{g}{f_1}\theta_r$ which is more consistent with experimental tests. If moreover $l << f_1$, $\Delta \approx (s + f_1)(s + k)(s + l) = (s + f_1)\Delta_0$

As a consequence, the closed-loop transfer functions are

$$\theta = \frac{k(s + f_1)(s + l)}{\Delta}\theta_r \approx \frac{k}{s + k}\theta_r$$

$$u = \frac{-gk(s + l)}{\Delta}\theta_r \qquad \approx \frac{-gk}{(s + f_1)(s + k)}\theta_r$$

to be compared with (41)-(42) : the angle dynamics is nearly the same as that given by the usual interpretation, while the velocity dynamics is dominated by the rotor drag time constant $1/f_1$. Defining the reference velocity $u_r = -g/f_1\theta_r$ we see that the usual control scheme, designed as an angle controller, is in fact a velocity controller!

The behavior experienced in practice is qualitatively and quantitatively well predicted by the revisited model, see figure 5 ("usual design"), the time response to a $-1.5°$ step in θ_r (i.e. a $1\ m/s$ step in u_r).

From this analysis, we see that the importance of the coefficient f_1 is paramount : the usual scheme works reasonably well only because f_1 is positive and not too small.

A better control law

The performance of the usual control scheme is limited by the rotor drag time constant $1/f_1$. Better performance can be achieved by

considering a controller-observer based on the revisited model,

$$q_r = -k_1\hat{u} - k_2\hat{\theta} + \left(k_1 - \frac{f_1 k_2}{g}\right)u_r$$

$$\dot{\hat{u}} = -f_1\hat{u} - g\hat{\theta} + l_1(a_x + f_1\hat{u})$$

$$\dot{\hat{\theta}} = g_y + l_2(a_x + f_1\hat{u})$$

where u_r is the velocity reference, k_1, k_2 are the controller gains and l_1, l_2 are the observer gains. Figure 5 shows simulation results for the same scenario as before (1 m/s reference step in velocity). Two different tunings were used: in the first case ("New Design #1") the controller is tuned for a settling time of about 12 s and the observer is tuned for about 48 s, so that the angle and velocity have initial transients similar to the tuning used previously for the usual design

(and with a similar control effort); in the second case ("New Design #2") the controller is made four times faster.

Both designs were successfully implemented, resulting in a quadrotor that is much easier to fly than with the usual scheme. In practice, it was difficult to accelerate the time responses much further, probably mainly due to accelerometer noise.

Conclusion

We have proposed and experimentally verified a revisited model of the quadrotor. It is different, even at first-order, from the model usually considered in the literature. It gives a different interpretation of accelerometer measurements and explains why control schemes based on the conventional model nevertheless behave reasonably well ■

Acronyms

EKF (Extended Kalman Filter)
GPS (Global positioning system)
MEMS (Micro Electro Mechanical Systems)
UAV (Unmannned Aerial Vehicle)

References

[1] D. MELLINGER, N. MICHAEL and V. KUMAR - *Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors*. International Journal of Robotics Research, vol. 31, n°. 5, pp. 664–674, 2012.
[2] N. GUENARD, T. HAMEL and V. MOREAU - *Dynamic Modeling and Intuitive Control Strategy for an "X4-Flyer*. Int. Conf. on Control and Automation, 2005, pp. 141–146.
[3] F. KENDOUL, D. LARA, I. FANTONI-COICHOT and R. LOZANO - *Real-Time Nonlinear Embedded Control for an Autonomous Quadrotor Helicopter*. Journal of Guidance, Control, and Dynamics, vol. 30, n°. 4, pp. 1049– 1061, 2007.
[4] S. SALAZAR-CRUZ, J. ESCAREÑO, D. LARA and R. LOZANO - *Embedded Control System for a Four-Rotor UAV*. International Journal of Adaptive Control and Signal Processing, vol. 21, n°. 2-3, pp. 189–204, 2007.
[5] G. HOFFMANN, H. HUANG, S. WASLANDER and C. TOMLIN - *Quadrotor Helicopter Flight Dynamics and Control : Theory and Experiment*. In Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2007, vol. 2, 2007, pp. 1670–1689.
[6] S. GRZONKA, G. GRISETTI and W. BURGARD - *Towards a Navigation System for Autonomous Indoor Flying*. IEEE Int. Conf. on Robotics and Automation, 2009, pp. 2878–2883.
[7] S. BOUABDALLAH and R. SIEGWART - *Full Control of a Quadrotor.* IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2007, pp. 153– 158.
[8] N. GUENARD, T. HAMEL and R. MAHONY - *A Practical Visual Servo Control for an Unmanned Aerial Vehicle.* IEEE Transactions on Robotics, vol. 24, n°. 2, pp. 331–340, 2008.
[9] F. KENDOUL, I. FANTONI and K. NONAMI - *Optic Flow-Based Vision System for Autonomous 3D Localization and Control of Small Aerial Vehicles.* Robotics and Autonomous Systems, vol. 57, n°. 6-7, pp. 591–602, 2009.
[10] M. ACHTELIK, A. BACHRACH, R. HE, S. PRENTICE and N. ROY - *Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-Denied Indoor Environments.* Proc. SPIE, vol. 7332, n°. 1, 2009, pp. 733 219–10.
[11] A. BACHRACH, S. PRENTICE, R. HE and N. ROY - *Range-Robust Autonomous Navigation in GPS-Denied Environments*. Journal of Field Robotics, vol. 28, n°. 5, pp. 644–666, 2011, cited By (since 1996)19.
[12] P. POUNDS, R. MAHONY and P. CORKE - *Modelling and Control of a Quad-Rotor Robot. Australasian Conf*. on Robotics and Automation, 2006.
[13] L. DERAFA, T. MADANI and A. BENALLEGUE - *Dynamic Modelling and Experimental Identification of Four Rotors Helicopter Parameters.* IEEE Int. Conf. on Industrial Technology, 2006, pp. 1834–1839.
[14] T. MADANI and A. BENALLEGUE - *Backstepping Control with Exact 2- Sliding Mode Estimation for a Quadrotor Unmanned Aerial Vehicle.* IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2007, pp. 141–146.
[15] P.-J. BRISTEAU, P. MARTIN, E. SALAÜN and N. PETIT - *The Role of Propeller Aerodynamics in the Model of a Quadrotor UAV*. European Control Conf., 2009, pp. 683–688.
[16] P. MARTIN and E. SALAÜN - *Design and Implementation of a Low-Cost Observer-Based Attitude and Heading Reference System*. Control Engineering Practice, vol. 18, no. 7, pp. 712 – 722, 2010.
[17] P. MARTIN and E. SALAÜN - *An Invariant Observer for Earth-Velocity Aided Attitude Heading Reference Systems.* IFAC World Congress, 2008, paper identifier 10.3182/20080706-5-KR-1001.3577.
[18] P. D. GROVES - *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.
[19] P. MARTIN and E. SALAÜN - *The True Role of Accelerometer Feedback in Quadrotor Control.* IEEE International Conference on Robotics and Automation, 2010, pp. 1623–1629.
[20] W. JOHNSON - *Helicopter Theory*. Princeton University Press, 1980.
[21] P.-J. BRISTEAU, F. CALLOU, D. VISSIÈRE and N. PETIT - *The Navigation and Control Technology Inside the ar.Drone micro UAV*. IFAC World Congress, 2011, pp. 1477–1484, paper identifier 10.3182/20080706- 5-KR-1001.3577.
[22] J. MACDONALD, R. LEISHMAN, R. BEARD and T. MCLAIN - *Analysis of an Improved Imu-Based Observer for Multirotor Helicopters*. Journal of Intelligent & Robotic Systems, pp. 1–13, 2013.
[23] D. ABEYWARDENA, S. KODAGODA, G. DISSANAYAKE and R. MUNASINGHE - *Improved State Estimation in Quadrotor MAVs: A Novel Drift-Free Velocity Estimator*. IEEE Robotics Automation Magazine, vol. 20, n°. 4, pp. 32–39, 2013.
[24] P. KOKOTOVIC, H. K. KHALIL and J. O'REILLY - *Singular Perturbation Methods in Control : Analysis and Design*. SIAM, 1999.

AUTHOR

Philippe Martin received the PhD degree in Mathematics and Control from the Ecole des Mines de Paris in 1992, where he is currently a Senior Researcher. In 1993-1994 he visited the Center for Control Engineering and Computation of the University of California at Santa Barbara and the Department of Mathematics of the University of North Carolina at Chapel Hill. Since 2000 he has also been a part-time Professor at Ecole Centrale Paris. His interests include theoretical aspects of nonlinear control and their applications to industrial control problems.

H. de Plinval
(Onera)
A. Eudes, P. Morin
(Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie)

E-mail : Henry.de_Plinval@onera.fr

DOI : 10.12762/2014.AL08-07

Control and Estimation Algorithms for the Stabilization of VTOL UAVs from Mono-Camera Measurements

This paper concerns the control of Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicles (UAVs) based on exteroceptive measurements obtained from a mono-camera vision system. By assuming the existence of a locally planar structure in the field of view of the UAV's videocamera, the so-called homography matrix can be used to represent the vehicle's motion between two views of the structure. In this paper we report recent results on both the problem of homography estimation from the fusion of visual and inertial data and the problem of VTOL UAV feedback stabilization from homography measurements

Introduction

Obtaining a precise estimation of the vehicle's position is a major issue in aerial robotics. The GPS is a very popular sensor in this context and it has been used extensively with VTOL UAVs, especially for navigation via waypoints. Despite recent progress of this technology, especially in terms of precision, many applications cannot be addressed with the GPS as the only position sensor. First, GPS is not available indoors and it can also be masked in some outdoor environments. Then, most inspection applications require a relative localization with respect to the environment, rather than an absolute localization as provided by the GPS. Finally, evolving in dynamic environments also requires relative localization capabilities. For all of these reasons, it is important to develop control strategies based on exteroceptive sensors that can provide relative position information with respect to the local environment. Examples of such sensors are provided by cameras, lasers, radars, etc. Cameras are interesting sensors to use with small UAVs, because they are light, low cost and provide rich information about the environment at a relatively high frequency. Precise 3D relative position information is best obtained from a stereo vision system with a "long" baseline (i.e., interdistance between the optical centers of the cameras). In this case, available feedback controllers that require position errors as inputs can be used. Using a mono-camera system is more challenging, because the depth-information cannot be recovered instantaneously (i.e., based on a single measurement). Nevertheless, a mono-camera system may be preferred in some applications, due to its compacity, or because the distance between the camera and the environment is large so that even a stereo-system would provide poor depth-information.

This paper concerns the control of VTOL UAVs from mono-camera measurements. We assume the existence of a locally planar structure in the environment. This assumption is restrictive, but it is relevant in practice because i) many man-made buildings are locally planar and ii) when the distance between the camera and the environment is large, the planarity assumption can be satisfied locally as a first approximation, despite the environment not being perfectly planar (e.g., as in the case of ground observation at a relatively high altitude). Based on two camera views of this planar structure, it is well known in computer vision that one can compute the so-called homography matrix, which embeds all of the displacement information between these two views [15]. This matrix can be estimated without any specific knowledge regarding the planar structure (such as its size or orientation). Therefore, it is suitable for the control of UAVs operating in unknown environments. Homography-based stabilization of VTOL UAVs raises two important issues. The first is the estimation of the homography matrix itself. Several algorithms have been developed within the computer vision community to obtain such an estimation (see, for example, [15, 1]). Recently, IMU-aided fusion algorithms have been proposed to cope with noise and robustness limitations associated with homography estimation algorithms based on vision data only [16, 9]. The second issue concerns the design of stabilizing feedback laws. The homography associated with two views of a planar scene is directly related to the Cartesian displacement (in both position and orientation) between these two views, but this relation depends on unknown parameters (normal and distance to the scene). Such uncertainties significantly complicate the design and stability analysis of feedback controllers. This is all the more true since VTOL UAVs are usually underactuated systems, with high-order dynamic relations between the vehicle's position and the control input. For example, horizontal displacement is related to the roll and pitch control torque through fourth-order systems. For this reason, most existing control strategies based on homography measurements make additional assumptions regarding the environment, i.e., the knowledge of the normal to the planar scene [20, 21, 18, 14]. This simplifies the control design and stability analysis since, in this case, the vehicle's Cartesian displacement (rotation and position up to an unknown scale factor) can be extracted from the homography measurement.

This paper reports recent results by the authors and co-authors on both the problem of homography estimation via the fusion of inertial and vision data [16, 9] and the design of feedback controllers based on homography measurements [5, 7]. The paper is organized as follows : Preliminary background and notation are given in § "Background". Feedback control algorithms are presented in § “Feedback Control Design” and homography estimation algorithms in § “Homography estimation”. Finally, some implementation issues are discussed in § “Computational aspects”.

Background

In this section, we review background on both the dynamics of VTOL UAVs and the homography matrix associated with two camera images of a planar scene. Let us start by defining the control problem addressed in this paper.

Control problem

Figure 1 illustrates the visual servoing problem addressed in this paper. A VTOL UAV is equipped with a mono-camera. A reference image of a planar scene $\mathcal{T}$, which was obtained with the UAV located in a reference frame $\mathcal{R}^*$, is available. From this reference image and the current image, obtained from the current UAV location (frame $\mathcal{R}$), the objective is to design a control law that can asymptotically stabilize $\mathcal{R}$ to $\mathcal{R}^*$. Note that asymptotic stabilization is possible only if $\mathcal{R}^*$ corresponds to a possible equilibrium, i.e., in the absence of wind the thrust direction associated with $\mathcal{R}^*$ must be vertical.

Dynamics of VTOL UAVs

We consider the class of thrust-propelled underactuated vehicles consisting of rigid bodies moving in 3D-space under the action of one body-fixed force control and full torque actuation [13]. This class contains most VTOL UAVs (quadrotors, ducted fans, helicopters, etc.). Being essentially interested here in hovering stabilization, throughout the paper we neglect aerodynamic forces acting on the vehicle’s main body. Assuming that $\mathcal{R}^*$ is a NED (North-East- Down) frame (see figure 1), the dynamics of these systems is described by the following well-known equations :

$$\begin{cases} m\ddot{p} = -TRb_3 + mgb_3 \\ \dot{R} = RS(\omega) \\ J\dot{\omega} = J\omega\times\omega + \Gamma \end{cases} \quad (1)$$

where p is the position vector of the vehicle’s center of mass, expressed in $\mathcal{R}^*$, $\mathcal{R}$ is the rotation matrix from $\mathcal{R}$ to $\mathcal{R}^*$, ω is the angular velocity vector of $\mathcal{R}$ with respect to $\mathcal{R}^*$ expressed in $\mathcal{R}$, $S(.)$ is the matrix-valued function associated with the cross product, i.e., $S(x)y = x\times y, \forall x,y\in\mathbb{R}^3$, m is the mass, T is the thrust control input, $b_3 = (0,0,1)^T$, J is the inertia matrix, Γ the torque control input and g is the gravity constant.

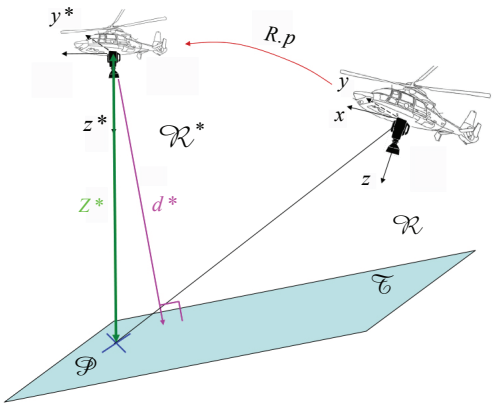


Figure 1 - Problem scheme

Homography matrix and monocular vision

With the notation of figure 1, consider a point $\mathcal{P}\in\mathcal{T}$ and denote by X^* the coordinates of this point in $\mathcal{R}^*$. In $\mathcal{R}^*$, the plane T is defined as $\{X^*\in\mathbb{R}^3; n^{*T}X^* = d^*\}$ where n^* are the coordinates in $\mathcal{R}^*$ of the unit vector normal to $\mathcal{T}$ and d^* is the distance between the origin of $\mathcal{R}^*$ and the plane. Let us now denote as X the coordinates of $\mathcal{P}$ in the current frame . One has $X^* = RX + p$ and therefore,

$$\begin{aligned} X &= R^TX^* - R^Tp \\ X &= R^TX^* - R^Tp[\frac{1}{d^*}n^{*T}X^*] \\ &= (R^T - \frac{1}{d^*}R^Tpn^{*T})X^* \\ &= \bar{H}X^* \end{aligned} \quad (2)$$

where

$$\bar{H} = R^T - \frac{1}{d^*}R^Tpn^{*T} \quad (3)$$

The matrix $\bar{H}$ could be determined by matching 3D-coordinates in the reference and current camera planes of points of the planar scene. The cameras do not provide these 3D-coordinates, however, since only the 2D-projective coordinates of P on the respective image planes are available. More precisely, the 2D-projective coordinates of $\mathcal{P}$ in the reference and current camera planes are respectively given by

$$\mu^* = K\frac{X^*}{z^*} \qquad \mu = K\frac{X}{z}$$

where z^* and z denote the third coordinate of X^* and X respectively (i.e., the coordinate along the camera optical axis), and K is the calibration matrix of the camera. It follows from (2) and (4) that

$$\mu = G\mu^* \quad (4)$$

with $$G \propto K\bar{H}K^{-1}$$

where $\propto$ denotes equality up to a positive scalar factor. The matrix $G\in\mathbb{R}^{3\times3}$, defined up to a scale factor, is called the *uncalibrated homography matrix*. It can be computed by matching projections onto

the reference and current camera planes of points of the planar scene. If the camera calibration matrix K is known, then the matrix $\bar{H}$ can be deduced from G, up to a scale factor, i.e., $K^{-1}GK = \alpha\bar{H}$. As a matter of fact, the scale factor α corresponds to the mean singular value of the matrix $K^{-1}GK : \alpha = \sigma_2(K^{-1}GK)$ (see, for example, [15, page 135]). Therefore, α can be computed together with the matrix $\bar{H}$. Another interesting matrix is

$$H = det(\bar{H})^{-\frac{1}{3}}\ \bar{H} = \eta\bar{H} \tag{5}$$

Indeed, det $(H) = 1$ so that H belongs to the Special Linear Group $SL(3)$. As we will see further on, this property can be used for homography filtering and estimation purposes. Let us finally remark that

$$\eta^3 = \frac{d^*}{d}$$

Feedback Control Design

In this section, we present two classes of feedback control laws for the asymptotic stabilization of VTOL UAVs based on homography measurements of the form $\bar{H}$ defined by (3). The first class consists of control laws that are affine with respect to the homography matrix components. These control laws ensure local asymptotic stabilization under very mild assumptions regarding the observed scene. The second class consists of nonlinear control laws that ensure large stability domains under stronger assumptions regarding the scene.

Linear control

The main difficulty in homography-based stabilization comes from the mixing of position and orientation information in the homography matrix components, as shown by relation (3). If the normal vector n^* is known, then one can easily extract from $\bar{H}$ the rotation matrix and the position vector up to the scale factor $1/d^*$. When n^* is unknown, however, this extraction is no longer possible and this mixing of information must be dealt with. The control laws presented here rely on the possibility of extracting partially decoupled position and rotation information from $\bar{H}$. This is shown by the following result, first proposed in [6].

Proposition 1
Let $\bar{e} = Me$ *with*

$$M = \begin{pmatrix} 2I_3 & S(m^*) \\ -S(m^*) & I_3 \end{pmatrix},\ e = \begin{pmatrix} e_p \\ e_\Theta \end{pmatrix} \tag{6}$$

and

$$\begin{aligned} e_p &= (I - \bar{H})m^*, e_\Theta = vex(\bar{H}^T - \bar{H}) \\ m^* &= b_3 = (0,0,1)^T \end{aligned} \tag{7}$$

where $vex(.)$ *is the inverse of the* $S(.)$ *operator : vex* $(S(x)) = x\ ;\ \forall x \in \mathbb{R}^3$. *Let* $\Theta = (\phi;\ \theta;\ \psi)^T$ *denote any parameterization of the rotation matrix* R *such that* $R \approx I_3 + S(\Theta)$ *around* $R \approx I_3$ *(e.g., Euler angles). Then,*

1. $(p,R) \to e$ *defines a local diffeomorphism around* $(p,R) = (0, I_3)$. *In particular,* $\bar{e} = 0$ *if and only if* $(p,R) = (0, I_3)$.

2. In a neighborhood of $(p,R) = (0, I_3)$,

$$\bar{e} = L\begin{pmatrix} p \\ \Theta \end{pmatrix} + O^2(p,\Theta) \qquad L = \begin{pmatrix} L_p & 0 \\ L_{p\Theta} & L_\Theta \end{pmatrix} \tag{8}$$

with $L_{p\Theta} = S((\alpha^*;\ \beta\ ;\ 0)^T)$,

$$L_p = \begin{pmatrix} c^* & 0 & \alpha^* \\ 0 & c^* & \beta^* \\ 0 & 0 & 2c^* \end{pmatrix} \quad L_\Theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

where α^* *and* β^* *are the (unknown) constant scalars defined by* $n^* = d^*(\alpha^*, \beta^*, c^*)^T, c^* = \frac{1}{\| X^* \|}$ *and* O^2 *terms of order two at least.*

Eq. (8) shows the rationale behind the definition of $\bar{e}$ *:* at first order, components $\bar{e}_1, \bar{e}_2, \bar{e}_3$ contain information on the translation vector p only, while components $\bar{e}_4, \bar{e}_5, \bar{e}_6$ contain decoupled information on the orientation (i.e., L_Θ is diagonal), corrupted by components of the translation vector. Although the decoupling of position and orientation information in the components of e is not complete, it is sufficient to define asymptotically stabilizing control laws, as shown below.

Let $\bar{e}_p \in \mathbb{R}^3$ (respectively $\bar{e}_\Theta \in \mathbb{R}^3$) denote the first (respectively last) three components of e, i.e., $\bar{e} = (\bar{e}_p^T\ ,\ \bar{e}_\Theta^T)^T$. The control design relies on a dynamic extension of the state vector defined as follows :

$$\dot{\xi} = -K_7\xi - \bar{e}_p \tag{9}$$

where K_7 is a diagonal gain matrix. The variable ν copes with the lack of measurements of $\dot{\bar{e}}$. The control design is presented through the following theorem.

Theorem 1
Assume that the target is not vertical and that the camera frame is identical to $\mathcal{R}$ *(as shown in figure 1). Let*

$$\begin{cases} T = m\left(g + k_1\bar{e}_3 + k_2\xi_3\right) \\ \Gamma = -JK_3\left(\omega - \omega^d\right) \end{cases} \tag{10}$$

with

$$\begin{cases} \omega^d &= -\dfrac{K_4}{g}\left(g\bar{e}_\Theta + b_3 \times \gamma^d\right) \\ \gamma^d &= -K_5\bar{e}_p - K_6\xi \end{cases} \tag{11}$$

Then,

1. Given any upper-bound $c_M^* > 0$, *there exist diagonal gain matrices* $K_i = Diag(k_j^i)$ $i = 3,..., 7$; $j = 1, 2, 3$ *and scalar gains* k_1, k_2, *such that the control law (10) makes the equilibrium* $(p, R, v, \omega, \xi) = (0, I_3, 0, 0, 0)$ *of the closed-loop System (1)-(9)-(10)-(11) locally exponentially stable for any value of* $c^* \in (0, c_M^*)$.

2. If the diagonal gain matrices K_i *and scalar gains make the closed-loop system locally exponentially stable for* $c^* = c_M^*$, *then local exponential stability is guaranteed for any value of* $c^* \in (0, c_M^*)$.

This result calls for several remarks.

1) The control calculation only requires the knowledge of $\bar{H}$ (via $\bar{e}$) and ω. Thus, it can be implemented with a very minimal sensor suite consisting of a mono-camera and gyrometers only.

2) This result does not address the case of a vertical target. This case can also be addressed with the same kind of technique and stability result. Such an extension can be found in [7], together with several other generalizations of Theorem 1.

3) Since $c^* = 1/\|X^*\|$ and $\|X^*\| \geq d^*$, a sufficient condition for $c^* \in (0, c_M^*)$ is that $d^* \geq 1/c_M^*$. Thus,
Property 1) ensures that stabilizing control gains can be found given any lower bound on the distance between the reference pose and the observed planar target. This is a very weak requirement from an application point of view.
Property 2) is also a very strong result, since it implies that in order to find stabilizing control gains for any $c^* \in (0, c_M^*)$, it is sufficient to find stabilizing control gains for $c^* = c_M^*$. This is a much easier task, which can be achieved with classical linear control tools. In particular, by using the Routh-Hurwitz criterion, explicit stability conditions on the control gains can be derived (see [7] for more details).

Nonlinear control laws

Theorem 1 shows that homography-based stabilizing control laws can be designed from very limited a priori information (essentially, a lower bound on the distance to the scene at the desired configuration and the scene planarity property). A weakness of this stability result, however, is the lack of knowledge regarding the size of the stability domain. Under some assumptions regarding the scene orientation, it is possible to derive stabilizing control laws with explicit (and large) stability domains. A first case of interest in practice is when the target is horizontal. In this case, the normal vector to the scene is known and the extraction of the orientation and position up to a scale factor, from $\bar{H}$, allows available nonlinear control laws with large stability domains to be used. Another interesting scenario for applications is when the target is vertical. This case is more challenging, since knowing that the scene is vertical does not completely specify its orientation. We present below a nonlinear feedback control to address this case.

First, let us remark that $n_3^* = 0$ when the scene is vertical. Indeed, the normal vector to the scene is horizontal and the reference frame $\mathcal{R}^*$ is associated with an equilibrium configuration so that its third basis vector is vertical (pointing downward). Then, it follows from (3) that

$$\begin{cases} \sigma = \bar{H}b_2 \times \bar{H}b_3 - \bar{H}b_1 = R^T M(\frac{n^*}{d^*})p \\ \gamma = g\bar{H}b_3 = gR^T b_3 \end{cases} \quad (12)$$

with $M(\tau) = \tau_1 I_3 + S(\tau_2 b_3)$. These relations show that decoupled information can be extracted from H in terms of position and orientation. Compared to the result given in proposition 1, this result is stronger, since the decoupling is complete and it holds without any approximation. On the other hand, it is limited to a vertical scene. Note that γ corresponds to the components of the gravity vector in the body frame. This vector, which is used in conventional control schemes based on Cartesian measurements, is typically estimated from accelerometer and gyrometer measurements of an IMU, assuming small accelerations of the UAV [17].

Eq. (12) leads us to address the asymptotic stabilization of UAVs from pose measurements of the form $\sigma = R^T Mp, \gamma = gR^T b_3$ where M is an unknown positive definite matrix. We further assume that the velocity measurements ω and $\upsilon = R^T \dot{p}$ are also available. The variable υ can be estimated, for example, via optical flow algorithms [10, 11, 9]. In most studies on feedback control of underactuated UAVs, M is assumed to be the identity matrix, so that the relation between the measurement function and the cartesian coordinates is perfectly known. Several control design methods ensuring semi-global stability of the origin of system (1) have been proposed in this case (see, for example, [19, 13]). We show below that similar stability properties can be guaranteed in the case of uncertainties regarding the matrix M. To this end, let us introduce some notations.

For any square matrix M, $M_s = (M+M^T)/2$ and $M_a = (M-M^T)/2$ respectively denote the symmetric and antisymmetric part of M. Given a smooth function f defined on an open set of $\mathbb{R}$, its derivative is denoted as f'. Given with $\delta = [\delta_m, \delta_M]$ with $0 < \delta_m < \delta_M$, we introduce the saturating function

$$sat_\delta(\tau) = \begin{cases} 1 & if\ \tau \leq \delta_m^2 \\ \frac{\delta_M}{\sqrt{\tau}} - \frac{(\delta_M - \delta_m)^2}{\sqrt{\tau}(\sqrt{\tau} + \delta_M - 2\delta_m)} & if\ \tau > \delta_m^2 \end{cases} \quad (13)$$

Note that $\tau \to \tau sat_\delta(\tau^2)$ defines a classical saturation function, in the sense that it is the identity function on $[0, \delta_m]$ and it is upper-bounded by δ_M.

We can now state the main result of this section (See [5] for more details, generalizations and proof). By a standard time separation argument commonly used for VTOL UAVs, we assume that the orientation control variable is the angular velocity ω instead of the torque Γ (i.e., once a desired angular velocity ω^d has been defined, a torque control input Γ that ensures convergence of ω to ω^d is typically computed through a high gain controller).

Theorem 2
Let sat_δ and $sat_{\bar{\delta}}$ denote two saturating functions. Assume that M is positive definite and consider any gain values $k_1, k_2 > 0$ such that

$$\begin{cases} k_2^2 \lambda_{min}(M_s) & > & k_1 \| M_a \| \| M \| C \\ C & \triangleq & sup_\tau \left(sat_\delta(\tau) + 2\tau \left| sat'_\delta(\tau) \right| \right) \\ k_2 \delta_m & > & k_1 \\ k_1 + k_2 \delta_M & < & g \end{cases} \quad (14)$$

Define a dynamic augmentation :

$$\dot{\xi} = \xi \times \omega - k_3(\xi - \sigma),\ k_3 > 0 \quad (15)$$

together with the control (T,ω) such that:

$$\begin{cases} \omega_1 & = & -\frac{k_4 |\bar{\mu}| \bar{\mu}_2}{(|\bar{\mu}| + \bar{\mu}_3)^2} - \frac{1}{|\bar{\mu}|^2} \bar{\mu}^T S(b_1) R^T \dot{\mu} \\ \omega_2 & = & \frac{k_4 |\bar{\mu}| \bar{\mu}_1}{(|\bar{\mu}| + \bar{\mu}_3)^2} - \frac{1}{|\bar{\mu}|^2} \bar{\mu}^T S(b_2) R^T \dot{\mu} \\ T & = & m\bar{\mu}_3 \end{cases} \quad (16)$$

where $\bar{\mu}, \mu$ *and the feedforward term* $R^T \dot{\mu}$ *are given by*

$$\bar{\mu} = \gamma + k_1 sat_\delta\left(|\xi|^2\right)\xi + k_2 sat_{\bar{\delta}}\left(|v|^2\right)v$$

$$\mu = R\bar{\mu}$$

$$R^T\dot{\mu} = -k_1 k_3\left[sat_\delta(|\xi|^2)I_3 + 2sat_{\delta'}(|\xi|^2)\xi\xi^T\right](\xi - \sigma)$$
$$+ k_2\left[sat_{\bar{\delta}}(|v|^2)I_3 + 2sat_{\bar{\delta}'}(|v|^2)vv^T\right](\gamma - ub_3)$$

Then,

i) there exists $k_{3,m} > 0$ *such that, for any* $k_3 > k_{3m}$, *the equilibrium* $(\xi, p, \dot{p}, \gamma) = (0, 0, 0, gb_3)$ *of the closed-loop system (1)-(15)-(16) is asymptotically stable and locally exponentially stable with convergence domain given by* $\left\{(\xi, p, \dot{p}, \gamma)(0); \bar{\mu}(0) \neq -\left|\bar{\mu}(0)\right| b_3\right\}$.

ii) if M_s *and* M_a *commute, the same conclusion holds for the first inequality in (14) replaced by :*

$$k_2^2\lambda_{min}(M_s) > k_1 \| M_a \| \left(\| M_a \| sup_\tau sat_\delta(\tau) + \| M_s \| sup_\tau\, 2\tau\, |sat'_\delta(\tau)|\right) \quad (17)$$

Let us comment on the above result. It follows from (14) that

$$|k_1 sat_\delta\left(|\xi|^2\right)\xi + k_2 sat_{\bar{\delta}}\left(|v|^2\right)v| \leq k_1 + k_2\delta_M < g = |\gamma|$$

This guarantees that $\bar{\mu}(0) \neq -|\bar{\mu}(0)|\, b_3$ whenever

$$gb_3^T R(0)b_3 > -(k_1 + k_2\delta_M)$$

Consequently, the only limitation on the convergence domain concerns the initial orientation error and there is no limitation on the initial position/velocity errors. Note also that the limitation on the initial orientation error is not very strong. Note that ω_3, which controls the yaw dynamics, is not involved in this objective. Thus, it can be freely chosen. In practice, however, some choices are better than others (see below for more details).

Application to the visual servoing problem

From (12), Theorem 2 applies directly with

$$M = M\left(\frac{n^*}{d^*}\right) = \frac{n_1^*}{d^*}I_3 + S\left(\frac{n_2^*}{d^*}b_3\right)$$

In this case, one verifies that the stability conditions (14)-(17) are equivalent to the following :

$$\begin{cases} n_1^* > 0 \\ k_1, k_2 > 0 \\ k_2\delta m > k_1 \\ k_1 + k_2\delta_M < g \\ n_1^* d^* k_2^2 > k_1\left|n_2^*\right|\left(\left|n_2^*\right| + \dfrac{2n_1^*}{3\sqrt{3}}\right) \end{cases} \quad (18)$$

Note that the first condition, which ensures that M is positive definite, essentially means that the camera is "facing" the target at the reference pose. This is a very natural assumption from an application point of view. When (loose) bounds are known for $d^* : d_{\min} \leq d^* \leq d_{\max}$ *and* $n_1^* \geq n_{1\min}$, and recalling that $|n^*| = 1$, the last condition of equation (18) can be replaced by :

$$n_{1\min} d_{\min} k_2^2 > k_1\left(1 + \frac{2}{3\sqrt{3}}\right) \quad (19)$$

The yaw degree of freedom is not involved in the stabilization objective. On the other hand, it matters to keep the target inside the field of view of the camera. We propose to use the following control law :

$$\omega_3 = k_5 H_{21} \quad (20)$$

Upon convergence of the position, velocity, roll and pitch angles due to the other controls, the yaw dynamics will be close to $\dot{\psi} \approx -k_5 sin(\psi)$, thus ensuring the convergence of ψ to zero unless is initially equal to π (case contradictory to the visibility assumption). Another nice feature of this yaw control is that it vanishes when $H_{21} = 0$, i.e., when the target is seen, from the yaw prospective, as it should be at the end of the control task. This means that the controller tries to reduce the yaw angle only when the position/velocity errors have been significantly reduced.

Homography estimation

Obtaining a good estimate of the homography matrix in real-time is a key issue for the implementation of the stabilization algorithms presented earlier. In this section, we first briefly review existing computer vision algorithms to obtain an estimate of the homography matrix. Then, we focus on the use of inertial measurements to improve and speed-up the estimation process.

Computer vision methods

There are two main classes of vision algorithms for computing the homography matrix between two images of the same planar scene:

1. Interest point based methods

2. Intensity based methods

In the first case, the homography matrix is recovered from point correspondence between the two images in a purely geometrical way. A first step consists in the detection of interest points. These correspondences can be estimated by matching (with interest point detection and descriptor) or KLT tracking (based on intensity). The homography matrix is recovered from this correspondence with algorithms such as DLT [12], which are most of the time coupled with robust estimation techniques like RANSAC or M-estimator, in order to avoid false matching. For more details on interest point based methods, the reader is also referred to [12].

In the second case, the homography matrix is estimated by striving to align two images (the reference image or "template" T and the current image I). This is done, for example, by defining a transformation (usually called "warping") from the reference image to the current image $w_\rho : q^* \to q = w_\rho(q^*)$, where q^* denotes a pixel in the reference image, q denotes a pixel in the current image and ρ is a parameterization of the homography matrix, for example a parameterization of the Lie algebra of SL(3). This definition leads to an optimization problem that is solved numerically. The problem consists in minimizing with respect to ρ a measurement of the distance between the reference image $T = \{T(q^*)\}$ and the transform of the image I by the warping : $\{I(w_\rho(q^*))\}$. The cost function of the optimization problem varies with the proposed method, but most of the time it essentially boil downs to a sum over the image's pixels of the distance

between the pixel intensities in the two images. Usually, the optimization process only provides the optimal solution locally, i.e., provided that the distance between the two images is small enough. One way to improve the convergence of this type of method is to rely on Gaussian pyramids [4]. In this case, the template image is smoothed by a Gaussian and recursively down-sampled by a factor two to form a pyramid of images, with the template image at the bottom and the smallest image at the top. The visual method is then successively applied at each level of the pyramid, from top to bottom. Thus, large movements are kept small in pixel space and the convergence domain of the method is improved.

In this paper we focus on two estimation algorithms of this second class of methods : the ESM algorithm (Efficient Second order Minimization) [3], and the IC algorithm (Inverse Compositional) [2]. Table 5.2 summarizes the main features of both methods. The main interest of the IC method is that it allows a great amount of pre-computation to be performed based on the reference image. Indeed, the Jacobian matrix J of the cost function is computed from the template image, i.e., it depends neither on the current image nor on the homography parameterization ρ. Thus, the inverse of J^TJ can also be pre-computed. Only the computation of the intensity error and matrix multiplication are needed for each iteration. By contrast, the ESM is a second order method that uses both the current image gradient and template image to find the best quadratic estimation of the cost function. Therefore, each iteration of the optimization algorithm is longer than for the IC method. As a counterpart, the convergence rate of the method is faster.

IMU-aided homography estimation

Cameras and IMUs are complementary sensors. In particular, the camera frame rate is relatively low (around 30Hz) and, in addition, vision data processing can take a significant amount of time, especially on small UAVs with limited computation power. By contrast, IMUs provide data at a high frequency and this information can be processed quickly. Since IMUs are always present on UAVs for control purposes, it is thus natural to make use of them to improve the homography estimation process. In this section we present nonlinear observers recently proposed in [16] to fuse a vision-based homography estimate with IMU data. This fusion process is carried out on the Special Linear Lie Group $SL(3)$ associated with the homography representation (5), i.e., $\det(H) = 1$. This allows the Lie group invariance properties to be made use of in the observer design. We focus on two specific observers.

The first observer considered is based on the general form of the kinematics on $SL(3)$:

$$\dot{H} = -X H \tag{21}$$

where $H \in SL(3)$ and $X \in sl(3)$. The observer is given by

$$\begin{cases} \dot{\hat{H}} = -Ad_{\hat{H}}\left(\hat{X} - k_1 \mathbb{P}\left(\tilde{H}(I_3 - \tilde{H})\right)\right)\hat{H} \\ \dot{\hat{X}} = -k_2 \mathbb{P}\left(\tilde{H}(I_3 - \tilde{H})\right) \end{cases} \tag{22}$$

where $\hat{H} \in SL(3)$, $X \in sl(3)$ and $\tilde{H} = \hat{H}H^{-1}$. It is shown in [16] that this observer ensures almost global asymptotic stability of $(I_3, 0)$ for the estimation error $(\tilde{H}; \tilde{X}) = (\hat{H}H^{-1}; X - \hat{X})$ (i.e., asymptotic convergence of the estimates to the original variables) provided that X is constant (see [16, Th. 3.2] for details). Although this condition is seldom satisfied in practice, this observer provides a simple solution to the problem of filtering homography measurements. Finally, note that this observer uses homography measurements only.

A second observer, which explicitly takes into account the kinematics of the camera motion, is proposed in [16]. With the notation of Section 3, recall that the kinematics of the camera frame is given by

$$\begin{cases} \dot{R} = RS(\omega) \\ \dot{p} = Rv \end{cases} \tag{23}$$

With this notation, the group velocity X in (21) can be shown to be given by

$$\begin{aligned} X &= S(\omega) + \frac{vn^T}{d} - \frac{vn^T}{3d} I_3 \\ &= S(\omega) + \eta^3 \mathbb{P}(M) \end{aligned}$$

with $Y = \dfrac{vn^T}{d^*}$ (24)

The following observer of H and Y is proposed in [16] :

$$\begin{cases} \dot{\hat{H}} = -Ad_{\tilde{H}}\left(S(\omega) + \eta^3 \mathbb{P}(\hat{Y}) - k_1 \mathbb{P}\left(\tilde{H}(I_3 - \tilde{H})\right)\right)\hat{H} \\ \dot{\hat{Y}} = \hat{Y}S(\omega) - k_2 \eta^3 \mathbb{P}\left(\tilde{H}(I_3 - \tilde{H})\right) \end{cases} \tag{25}$$

where $\hat{H} \in SL(3)$; $\hat{Y} \in \mathbb{R}^{3\times 3}$ and $\tilde{H} = \hat{H}H^{-}$.

Conditions under which the estimates $(\hat{H}, \hat{Y})$ almost globally converge to (H, Y) are given in [16, Cor. 5.5]. These conditions are essentially reduced to the following: i) ω is persistently exciting, and ii) v is constant. The hypothesis of persistent excitation on the angular velocity is used to demonstrate the convergence of $\hat{Y}$ to Y. In the case of lack of persistent excitation, $\hat{Y}$ converges only to $Y + a(t)I_3$ where $a(t) \in \mathbb{R}$, but the convergence of $\hat{H}$ to H still holds. The hypothesis of v constant is a strong assumption. Asymptotic stability of the observer for v constant, however, guarantees that the observer can provide accurate estimates when v is slowly time varying with respect to the filter dynamics. This will be illustrated later in the paper and verified experimentally.

Architecture and data synchronization

Implementation of the above observers from IMU and camera data is done via a classical prediction/ correction estimation scheme. The quality of this implementation requires careful handling of data acquisition and communication. Synchronization and/or time-stamping of the two sensor data are instrumental in obtaining high-quality estimates. If the two sensors are synchronized, time-stamping may be ignored provided that the communication delay is short enough and that no data loss occurs. Discrete-time implementation of the observers can then be done with a fixed sampling rate. If the sensors are not synchronized, it is necessary to timestamp the data as close to the sensor output as possible and deal with possibly variable sampling rates.

Figure 2 gives a possible architecture of the interactions between estimator and sensors (Vision and IMU). Homography prediction obtained from IMU data is used to initialize the vision algorithm. Once a new image has been processed, the vision estimate obtained, considered as a measurement, is used to correct the filter's homography estimate. Due to the significant duration of the vision processing with

respect to the IMU sampling rate, this usually requires the prediction process to be reapplied via IMU data from the moment of the image acquisition. This leads us to maintain two states of the same estimator (see figure 2) : the real-time estimator, obtained from the last homography measurement and IMU data, and a post-processed estimator that is able to correct a posteriori the homography estimates from the time of the last vision data acquisition to the time when this data was processed.

Experimental setup

We make use of a sensor consisting of an xSens MTiG IMU working at a frequency of 200 Hz and an AVT Stingray 125B camera that provides 40 images with a resolution of 800 x 600 pixels per second. The camera and the IMU are synchronized. The camera uses wide-angle lenses (focal 1.28 mm). The target is placed over a surface parallel to the ground and is printed out on a 376 x 282 mm sheet of paper to serve as a reference for the visual system. The reference image has a resolution of 320 x 240 pixels. Thus, the distance d^* can be determined as 0.527 m. The processed video sequence presented in the accompanying video is 1321 frames long and presents high velocity motion (rotations of up to 5 rad/s, translations, scaling change) and occlusions. In particular, a complete occlusion of the pattern occurs slightly after $t = 10\ (s)$.

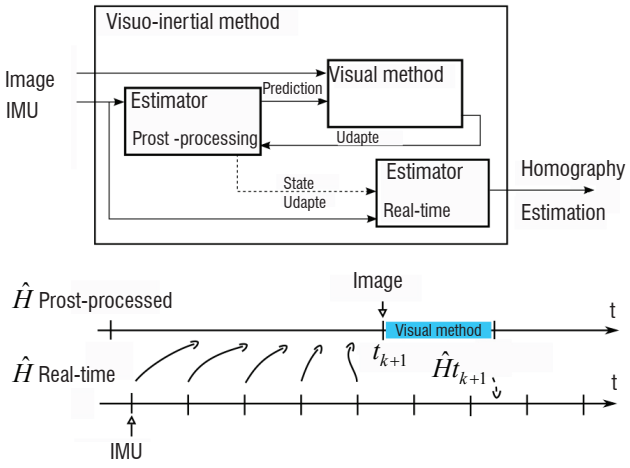


Figure 2 - Visuo-Inertial method scheme and sensor measurement processing timeline

Four images of the sequence are presented in figure 3. A "ground truth" of the correct homography for each frame of the sequence has been computed using a global estimation of the homography by SIFT, followed by the ESM algorithm. If the pattern is lost, we reset the algorithm with the ground-truth homography. The sequence is used at different sampling rates to obtain more challenging sequences and evaluate the performance of the proposed filters.

For both filters (22) and (25), the estimation gains have been chosen as $k_1 = 25$ and $k_2 = 250$. Following the notation of the description available at *http://esm.gforge.inria.fr/ESM.html*, the ESM algorithm is used with the following parameter values : $prec = 2$, $iter = 50$.

Tracking quality

In this section we measure the quantitative performance of the different estimators. This performance is reflected by the number of frames for which the homography is correctly estimated. We use the correlation score computed by the visual method to discriminate between well and badly estimated frames. A first tracking quality indicator is the percentage of well-estimated frames. This indicator will be labeled as "%track". Another related criterion concerns the number of time-sequences for which the estimation is successful. For this, we define a track as a continuous time-sequence during which the pattern is correctly tracked. We provide the number of tracks in the sequence (label "nb track") and also the mean and maximum track length. Table 1 presents the results obtained for the full sequence at various sampling rates (40 Hz, 20 Hz and 10 Hz).

The ESMonly estimator works well at 40 Hz since 95% of the sequence is correctly tracked, but performance rapidly decreases as the distance between images increases (72% at 20 Hz and only 35% at 10 Hz). It must be noted that the ESM estimator parameters are tuned for speed and not for performance, with real-time applications in mind.

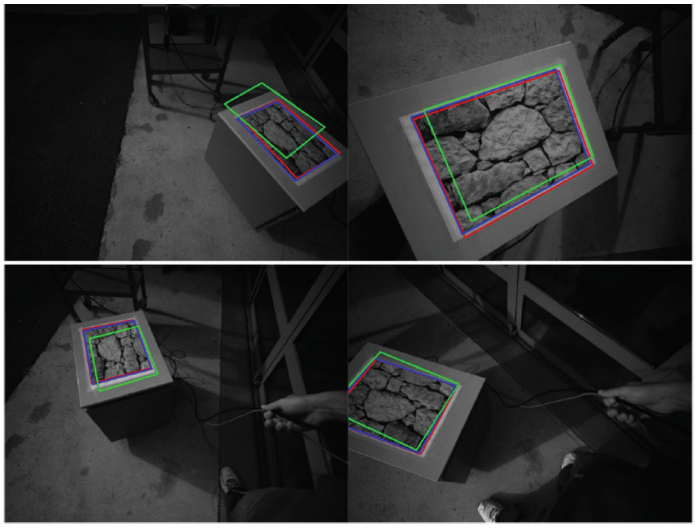


Figure 3 - Four images of the sequence at 20 Hz : pattern position at previous frame (green), vision estimate (blue) and prediction of the filterIMU (red)

The filternoIMU estimator outperforms the ESMOnly filter on the sequence at 40 Hz. Tracks are on average twice as long and many pattern losses are avoided (11 tracks versus 19 for ESMonly). At 20 Hz, the performance is even better, but the difference between these two solutions becomes smaller. At 10 Hz, the filter degrades performance.

The filterIMU tracks almost all of the sequence at both 40 Hz and 20 Hz. There is just one tracking failure, which occurs around time t = 10 s due to the occlusion of the visual target. The improvement provided by the IMU is clearly shown. At 10 Hz, the performance significantly deteriorates, but this filter still outperforms the other ones. Let us finally remark that these performances are obtained despite the fact that the assumption of constant velocity in the body frame (upon which the filter stability was established) is violated.

Frame rate	Method	% track	nb track	Track length	
				mean	max
40 Hz 1321 img	ESMonly FilternoIMU FilterIMU	94.31 97.74 98.78	19 11 2	65.36 114.27 646.5	463 607 915
20 Hz 660 img	ESMonly FilternoIMU FilterIMU	72.38 80.5 97.42	59 52 2	8.0 10.17 321.5	89 94 456
10 Hz 330 img	ESMonly FilternoIMU FilterIMU	38.79 32.36 58.66	46 58 59	2.78 1.72 3.27	27 4 27

Table 1 - Good track rate for various frame-rates and methods : percentage of well estimated frames, number of tracks, mean and maximum track length on the sequence

Computational aspects

Implementing vision algorithms on small UAVs is still a challenge today. Computational optimization is often necessary, in order to achieve real-time implementation (e.g., vision processing at about 10 - 20 Hz). In this section, we discuss some possible approaches to speed up the vision processing for the homography estimation problem considered here.

Computational optimization

Two types of optimization can be considered. The first one concerns the optimal use of the computing power. It consists, for example, in computation parallelization (SIMD instructions, GPU, multiprocessor/core), fix-point computation, or cache optimization. This type of optimization does not affect the vision algorithm accuracy. Another type of optimization concerns the vision algorithm itself and the possibilities of lowering its computational cost. This may affect the accuracy of the vision algorithm output. These two types of optimization have been utilized here: SIMD (Single Instruction Multiple Data) for computing power optimization and pixel selection for vision algorithm optimization.

SIMD instructions allow the data to be processed by packets. In SSE (x86 processor) and NEON (arm processor), it is possible to process four items of floating point data with one instruction. Thus, using this instruction with careful data alignment can theoretically improve performance by a factor of four. This theoretical figure is limited by load/store operation and memory (cache) transfer issues. This optimization is only done on computation intensive parts of the program, such as intensity gradient computation, image warping, or Jacobian estimation.

One approach to speed up dense vision algorithms is to use only the pixels that provide effective information for the minimization process. Indeed, the lower the number of pixels, the lower the computation cost. There are many ways to select good pixels for the pixel intensity minimization between two images ([8]). One approach consists in using only pixels with a strong gradient, since intensity errors provide

		Machine	ESM		IC	
			Without SIMD	With SIMD	Without SIMD	With SIMD
Pixel Selection	No	PC	60.0 (94)	20.0 (94)	73.0 (81)	29.5 (81)
	Yes	PC	27.0 (86)	15.0 (86)	7.5 (72)	4.4 (72)
	No	Odroid	347 (94)	202 (94)	409 (81)	314 (81)
	Yes	Odroid	165 (85)	140 (86)	53 (72)	45 (73)

Table 2 - Visual method performance : time (in ms) and accuracy (in %) for the different combination of optimization and platform

Method	ESM	IC
Minimization objective	$\min_{\rho} \sum_{q^*} \left[T(q) - I(w_{\rho}(q^*)) \right]^2$	
Step minimization objective	$\min_{\delta_\rho} \sum_{q^*} \left(T(q^*) - I(w_{(\rho+\delta_\rho)}(q^*)) \right)^2$	$\min_{\delta_\rho} \sum_{q^*} (T(w_{\delta_\rho}(q^*)) - I(w_{\rho}(q^*)))^2$
Effective computation	$\delta_\rho = \left(J^T J \right)^{-1} J^T (T(q^*) - I(w_{\rho}(q^*)))$	
Jocobian J	$\frac{1}{2}(\Delta T + \Delta I) \frac{\partial w}{\partial \rho}\Big\vert_{\rho}$	$\Delta T \frac{\partial w}{\partial \rho}\Big\vert_{0}$
Use current image gradient (ΔI)	Yes	No
Use template gradient (ΔT)	Yes	Yes

Table 3 - Visual method summary

position/orientation information contrary to image parts with no intensity gradient. In the experimental results reported below, we used the best 2500 pixels.

Evaluation

In this section, we report experimental results obtained with both the ESM and IC methods. For each method, we used the same stop criteria for the optimization: the maximal number of steps per scale is 30 and the stop error is 1e-3. The number of scales in the pyramid is four.

Table 2 provides the mean frame time (in ms) and mean performance (percentage of correctly estimated homographies) of the various combinations of optimization and methods on the sequence at 40 Hz (see experimental setup). The computation is performed on a desktop PC (Intel(R) Core(TM) i7-2600K CPU @ 3.40 GHz) and the same result is provided for an embedded platform (Odroid U2) based on an Exynos4412 Prime 1.7 Ghz ARM Cortex-A9 Quad processor.

With SIMD, the performance gain is from 3.0 x to 1.7 x on x 86 and 1.7 x to 1.17 x on the arm. With pixel selection the gain is better, from 1.3 to 2.1 for ESM and from 1.3 x to 9 x for IC.

Finally, the ratio between the fastest and the slowest is 13.6 x with a loss of 22% of correctly tracked frames.

Conclusion

We have presented recent stabilization and estimation algorithms for the stabilization of VTOL UAVs based on mono-camera and IMU measurements. The main objective is to rely on a minimal sensor suite, while requiring the least amount of information on the environment possible. Estimation algorithms have already been evaluated experimentally. The next step is to conduct full experiments on a UAV with both stabilization and estimation algorithms running on-board. This work is currently in progress. Possible extensions of the current work are multiple, such as for example the use of accelerometers to improve the homography estimation and/or the stabilization, or the extension of this work to possibly non-planar scenes ■

Acknowledgement

A. Eudes and P. Morin have been supported by “Chaire d’excellence en Robotique RTE-UPMC “.

Acronyms

VTOL	(Vertical Take-Off and Landing)
UAV	(Unmanned Aerial Vehicle)
GPS	(Global Positioning System)
3D	(three-dimensionnal)
IMU	(Inertial Measurement Unit)
NED	(North-East-Down)
KLT	(Kanade-Lucas-Tomasi (feature tracker))
DLT	(Direct Linear Transformation)
RANSAC	(RANdom SAmple Consensus)
ESM	(Efficient Second-order Minimization)
IC	(Inverse Compositional)
SIFT	(Scale-Invariant Feature Transform)
SIMD	(Single Instruction Multiple Data)
GPU	(Graphics Processing Unit)

References

[1] A. AGARWAL, C. V. JAWAHAR, and P. J. NARAYANAN - *A Survey of Planar Homography Estimation Techniques*. Technical Report Technical Report IIIT/TR/2005/12, IIIT, 2005.
[2] S. BAKER and I. MATTHEWS - *Lucas-Kanade 20 Years on : A Unifying Framework*. International Journal of Computer Vision, 56(3) : 221-255, 2004.
[3] S. BENHIMANE and E. MALIS - *Homographybased 2D Visual Tracking and Servoing*. International Journal of Robotic Research, 26(7) : 661-676, 2007.
[4] J.R. BERGEN, P. ANANDAN, K.J. HANNA, and R. HINGORANI - *Hierarchical Model-Based Motion Estimation*. Computer VisionECCV’92, pp. 237-252. Springer, 1992.
[5] H. DE PLINVAL, P. MORIN, and P. MOUYON - *Nonlinear Control of Underactuated Vehicles With Uncertain Position Measurements and Application to Visual Servoing*. American Control Conference (ACC), pp. 3253-3259, 2012.
[6] H. DE PLINVAL, P. MORIN, P. MOUYON, and T. HAMEL - *Visual Servoing for Underactuated VTOL UAVs: a Linear Homography-Based Approach*. IEEE Conference on Robotics and Automation (ICRA), pages 3004-3010, 2011.
[7] H. DE PLINVAL, P. MORIN, P. MOUYON, and T. HAMEL - *Visual Servoing for Underactuated VTOL UAVs: a Linear Homography-Based Framework*. International Journal of Robust and Non-linear Control, 2013.
[8] F. DELLAERT AND R. COLLINS - *Fast Image-Based Tracking by Selective Pixel Integration*. Proceedings of the ICCV Workshop on Frame-Rate Vision, pp. 1-22, 1999.
[9] A. EUDES, P. MORIN, R. MAHONY, and T. HAMEL - *Visuo-Inertial Fusion for Homography-Based Itering and Estimation*. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS), pages 5186-5192, 2013.
[10] V. GRABE, H.H. BÜLTHO, and P. ROBUO GIORDANO - *On-Board Velocity Estimation and Closedloop Control of a Quadrotor UAV Based on Opticalow*. IEEE Conf. on Robotics and Automation (ICRA), 2012.
[11] V. GRABE, H.H. BÜLTHO, and P. ROBUO GIORDANO - *A Comparison of Scale Estimation Schemes for a Quadrotor UAV Based on Opticalow and IMU Measurements*. IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS), pp. 5193-5200, 2013.
[12] R. HARTLEY and A. ZISSERMAN - *Multiple View Geometry in Computer Vision*. Volume 2. Cambridge Univ Press, 2000.
[13] M.D. HUA, T. HAMEL, P. MORIN, and C. SAMSON - *A Control Approach for Thrust-Propelled Underactuated Vehicles and its Application to VTOL Drones*. IEEE Trans. on Automatic Control, 54 : 1837-1853, 2009.

[14] F. LE BRAS, T. HAMEL, R. MAHONY, and A. TREIL - *Output Teedback Observation and Control for Visual Servoing of VTOL UAVs*. International Journal of Robust and Nonlinear Control, 21 : 1-23, 2010.
[15] Y. MA, S. SOATTO, J. KOSECKA, and S.S. SASTRY - *An Invitation to 3-D Vision : From Images to Geometric Models*. SpringerVerlag, 2003.
[16] R. MAHONY, T. HAMEL, P. MORIN, and E. MALIS - *Nonlinear Complementary Iters on the Special Linear Group*. International Journal of Control, 85 : 1557-1573, 2012.
[17] P. MARTIN and E. SALAUN - *The True Role of Accelerometer Feedback in Quadrotor Control*. IEEE Conf. on Robotics and Automation, pp. 1623-1629, 2010.
[18] N. METNI, T. HAMEL, and F. DERKX - *A UAV for Bridges Inspection : Visual Servoing Control Law with Orientation Limits*. 5th Symposium on Intelligent Autonomous Vehicles (IAV 04), 2004.
[19] J.-M. P IMLIN, P. SOUERES, and T. HAMEL - *Position Control of a Ducted fan VTOL UAV in Crosswind*. 80:666-683, 2007.
[20] O. SHAKERNIA, Y. MA, T. KOO, and S. SASTRY - *Landing an Unmanned Air Vehicle : Vision Based Motion Estimation and Nonlinear Control*. Asian Journal of Control, 1(3) : 128-145, 1999.
[21] D. SUTER, T. HAMEL, and R. MAHONY - *Visual Servo Control Using Homography Estimation for the Stabilization of an x4-flyer*. 2002.

AUTHORS

Henry de Plinval graduated from Ecole Polytechnique, France, in 2006 and received his MSc in Aeronautics and Astronautics in 2006 from MIT (USA), and his PhD in automatic control in 2014 from ISAE-Supaéro. Since 2008, he is with the Systems Control and Flight Dynamics Department within Onera - the French Aerospace Lab. His areas of interest include guidance, navigation and control, mostly for UAVs, with a particular focus on visual servoing for VTOL UAVs.

Alexandre Eudes received the Phd degree in robotics and computer vision from university Blaise Pascal, under the supervision of M. Lhuillier (Pascal institute) and S. Naudet (CEA). During his Phd, he worked on visual SLAM for real-time car localization applications with strong emphasis on uncertainty propagation and vision/odometry fusion. He is currently a post-doc fellow in Pascal Morin's team (ISIR), where he works on feedback control design for UAVs visual stabilization and visuo-inertial fusion for state estimation.

Pascal Morin received the Maîtrise degree from Université Paris-Dauphine in 1990, and the Diplôme d'Ingénieur and Ph.D. degrees from Ecole des Mines de Paris in 1992 and 1996 respectively. He spent one year as a post-doctoral fellow in the Control and Dynamical Systems Department at the California Institute of Technology. He was Chargé de Recherche at INRIA, France, from 1997 to 2011. He is currently in charge of the "Chaire RTE-UPMC Mini-drones autonomes" at the ISIR lab of University Pierre et Marie Curie (UPMC) in Paris. His research interests include stabilization and estimation problems for nonlinear systems, and applications to mechanical sytems such as nonholonomic vehicles or UAVs.

Nonlinear Feedback Control of VTOL UAVs

D.Pucci
(Istituto Italiano di Tecnologia)
M.-D. Hua, P. Morin
(CNRS, UMR 7222)
T. Hamel
(I3S/UNSA)
C. Samson
(I3S/UNSA, INRIA)

E-mail: daniele.pucci@iit.it

DOI : 10.12762/2014.AL08-08

This paper addresses the nonlinear feedback control of Unmanned Aerial Vehicles (UAVs) with Vertical Take-Off and Landing (VTOL) capacities, such as multi-copters, ducted fans, helicopters, convertible UAVs, etc. First, dynamic models of these systems are recalled and discussed. Then, a nonlinear feedback control approach is presented. It applies to a large class of VTOL UAVs and aims at ensuring large stability domains and robustness with respect to unmodeled dynamics. This approach addresses most control objectives encountered in practice, for both remotely operated and fully autonomous flight.

Introduction

Like other engineering fields, flight control makes extensive use of linear control techniques [43]. One reason for this is the existence of numerous tools to assess the robustness properties of a linear feedback controller [38] (gain margin, phase margin, H_2, H_∞ or LMI techniques, etc.). Another reason is that flight control techniques have been developed primarily for full-size commercial airplanes, which are designed and optimized to fly along very specific trajectories (trim trajectories with a very narrow range of angles of attack). Control design is then typically achieved from the linearized equations of motion along desired trajectories and this makes linear control especially suitable. Some aerial vehicles are required to fly in very diverse conditions, however, with large and rapid variations of the angle of attack. Examples are given by fighter aircraft, convertible aircraft, or small UAVs operating in windy environments. In such cases, ensuring large stability domains matters, and nonlinear feedback designs can be useful for this purpose.

Nonlinear feedback control of aircraft can be traced back to the early eighties. Following [41], control laws based on the dynamic inversion technique have been proposed to extend the flight envelope of military aircraft (see, *e.g.*, [45] and the references therein). The control design strongly relies on tabulated models of aerodynamic forces and moments, like the High-Incidence Research Model (HIRM) of the Group for Aeronautical Research and Technology in Europe (GARTEUR) [26]. Compared to linear techniques, this type of approach allows the flight domain to be extended without involving gain scheduling strategies. The angle of attack is assumed to remain away from the stall zone, however, and should this assumption be violated the behavior of the system is unpredictable. Comparatively, nonlinear feedback control of VTOLs is more recent, but it has been addressed with a larger variety of techniques. Dynamic inversion has been used as well [10], but many other techniques have also been investigated, such as the Lyapunov-based design [25, 16], Backstepping [4], Sliding modes [4, 46], or Predictive control [20, 3]. A more complete bibliography on this topic can be found in [13]. Most of these studies address the stabilization of hover flight or low-velocity trajectories and therefore little attention is paid to aerodynamic effects. These are typically either ignored or modeled as a simple additive perturbation, the effect of which has to be compensated for by the feedback action. In highly dynamic flight conditions or harsh wind conditions, however, aerodynamic effects become important. This raises several questions, which are little addressed in the control or robotics communities, such as, for example, *which models of aerodynamic effects should be considered for the control design? Or which feedback control solutions can be inferred from these models so as to ensure large stability domains and robustness?*

This paper presents a nonlinear feedback control approach for VTOL UAVs, which aims at ensuring large stability domains together with good robustness properties with respect to additive perturbations. The control design covers several control objectives associated with different autonomy levels (teleoperation with thrust direction and thrust intensity reference signals, teleoperation with linear velocity reference signals, fully autonomous flight with position reference signals). The approach, which explicitly takes into account aerodynamic forces in the control design, is particularly well suited to aerial vehicles submitted to small lift forces (*e.g.*, classical multi-copters, or helicopters) or to vehicles with shape symmetry properties with respect to the thrust axis (rockets, missiles, or airplanes with annular wings). The control methodology has been developed by the authors for several years [14, 12, 36, 37] and this paper provides a summary of these developments together with perspectives.

The paper is organized as follows. In § "Dynamics of aircraft motions", dynamical equations of VTOL UAVs are recalled and the various forces affecting the flight dynamics are discussed. § "Preliminaries on control design" provides some preliminaries on the feedback control design and a discussion of the merits of nonlinear feedback control. In § "Symmetric bodies and spherical equivalence", we show that for a class of symmetric bodies, the dynamical equations can be transformed into a simpler form (the so-called "spherical case"). This transformation is then used in § "Control design" to propose a feedback control design method applicable to several vehicles of interest.

Dynamics of aircraft motion

Aircraft dynamics are described by a set of differential equations that characterize the state of the aircraft in terms of the vehicle's orientation, position, and angular and linear velocities. These variables are measured with respect to some reference frames.

Let $\mathcal{I} = \left\{O; \vec{i}_0, \vec{j}_0, \vec{k}_0\right\}$ denote a fixed inertial frame with respect to (w.r.t.) which the vehicle's absolute pose is measured. This frame is chosen as the NED frame (North-East-Down) with $\vec{i}_0$ pointing to the North, $\vec{j}_0$ pointing to the East, and $\vec{k}_0$ pointing to the center of the earth. Let $\mathcal{B} = \left\{G; \vec{i}, \vec{j}, \vec{k}\right\}$ be a frame attached to the body, with G the body's center of mass. The linear and angular velocities $\vec{v}$ and $\vec{\omega}$ of the body frame $\mathcal{B}$ are then defined by

$$\vec{v} = \frac{d}{dt}\overrightarrow{OG}, \quad \frac{d}{dt}(\vec{i}, \vec{j}, \vec{k}) = \vec{\omega} \times (\vec{i}, \vec{j}, \vec{k}) \tag{1}$$

where, here and throughout the paper, the time-derivative is taken w.r.t. the inertial frame $\mathcal{I}$.

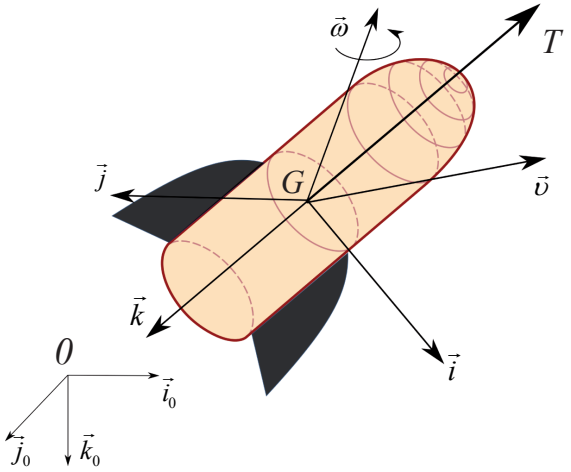


Figure 1 – Notation

Equations of motion for a flat earth

Let $\vec{F}$ and $\vec{M}$ denote respectively the resultant of the external forces acting on a rigid body of mass m and the moment of these forces about the body's center of mass G. Newton's and Euler's theorems of mechanics state that

$$\frac{d}{dt}\vec{p} = \vec{F} \qquad \frac{d}{dt}\vec{h} = \vec{M} \tag{2}$$

with

$$\vec{p} = m\vec{v} \qquad \vec{h} = -\int_{P' \in body} \overrightarrow{GP'} \times (\overrightarrow{GP'} \times \vec{\omega})\, dm = \vec{J}.\vec{\omega} \tag{3}$$

where $\vec{\mathcal{J}}$ denotes the inertia operator at G. Throughout this paper aircraft are modeled as rigid bodies of constant mass m and we focus on the class of vehicles controlled via four control inputs: the thrust intensity $T \in \mathbb{R}$ of a body-fixed thrust force $\vec{T} = -T\vec{k}$ and the three components (in body-frame) of a control torque vector $\vec{\Gamma}_G$. This class of systems contains (modulo an adequate choice of control inputs) most aerial vehicles of interest, like multicopters, helicopters, convertibles UAVs, or even conventional airplanes. The torque actuation can be obtained in different ways, for example, control surfaces (fixed-wing aircraft), propellers (multi-copters), swash-plate mechanism and tail-rotor (helicopters). By neglecting round-earth effects and buoyancy forces[1], the external forces and moments on the aircraft are commonly modeled as follows [8, Ch. 2], [12], [42], [43]:

$$\begin{aligned} \vec{F} &= m\vec{g} + \vec{F}_a - T\vec{k} + \vec{F}_b \\ \vec{M} &= \overrightarrow{GP} \times \vec{F}_a + T\vec{k} \times \overrightarrow{G\Theta} + \vec{G}_G \end{aligned} \tag{4}$$

where $\vec{g} = g\vec{k}_0$ is the gravity acceleration vector with g the gravity constant, $(\vec{F}_a, P)$ is the resultant of the aerodynamic forces and its application point[2], and Θ is the application point of the thrust force. In eq. (4) we assume that the gyroscopic torque (usually associated with rotor craft) is negligible or that it has already been compensated via a preliminary torque control action. The force $\vec{F}_b$ is referred to as a body force. It is induced by the control torque vector $\vec{\Gamma}_G$ and thus represents the effect of the control torque actuation on the position dynamics. Conversely, the term $T\vec{k} \times \overrightarrow{G\Theta}$ in (4) represents the effect of the control force actuation on the orientation dynamics.

Besides the gravity force, eq. (4) allows three types of forces (and torques) to be identified:

- body forces, which represent couplings between thrust and torque actuations;
- control forces;
- aerodynamic forces.

This decomposition is based on a separation principle that is only valid in first approximation (this issue will be detailed later on). Nevertheless, identifying the dominant effects of dynamics is useful from a control point of view, since it allows generic control strategies to be worked out, which can be refined case by case for specific classes of vehicles. We now discuss the modeling of these three types of forces in more detail.

Body forces

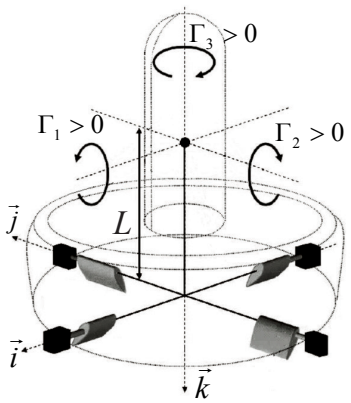


Figure 2 – Ducted-fan tail-sitter HoverEye of Bertin Technologies

[1] The aircraft is assumed to be much heavier than air.

[2] The point P is the so called body's center of pressure. This point depends on several variables such as the vehicle's velocity and environmental conditions. As a consequence, its determination is as complex as that of the aerodynamic forces $\vec{F}_a$, and is beyond the scope of this paper.

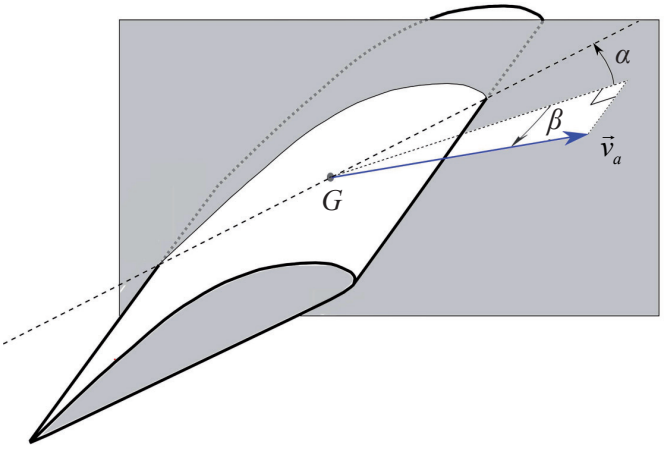


Figure 3 – Classic definition of (α, β) angles for a flat wing

The influence of the torque control inputs on the translational dynamics via the body force $\vec{F}_b$ depends on the torque generation mechanism. More specifically, this coupling term is negligible for quadrotors [9], [32], [6], but it can be significant for helicopters because of the swashplate mechanism [11, Ch.1], [7], [22], [24], [28, Ch. 5] and for ducted-fan tail-sitters because of the rudder system [29, Ch. 3], [31]. Thus, the relevance of this body force must be discussed in relation to the specific application [31] [29, Ch. 3] [13]. Let us remark, however, that the body force $\vec{F}_b$ is typically small compared to either the gravity, the aerodynamic force, or the thrust force. Similarly, the term $T\vec{k}\times\overrightarrow{G\Theta}$ in (4), which represents the influence of the thrust control input on the rotational dynamics, is usually small because Θ is close to the axis $(G,\vec{k})$. These body forces will be omitted from now on, since they can be either neglected, or compensated by the control action.

Control forces

The model (4) should be complemented by a modeling of the actuators that generate the inputs T and $\vec{\Gamma}_G$. By assuming that the dynamics of these actuators are (sufficiently) faster than the vehicle's dynamics, they can be neglected in the first approximation. The effects of the vehicle's motion and/or wind on the actuation efficiency are another aspect that cannot be neglected if a precise modeling is required. For example, blade flapping is a well-known phenomenon that highlights the difficulty in making the control force and torque completely independent of external aerodynamic conditions for aerial vehicles actuated by propellers. For the sake of simplicity and genericity, we will assume in the paper that it is possible to completely decouple the control action from the vehicle's motion and wind. We are aware, however, that this can be an important issue in practice.

Aerodynamic forces

The modeling of aerodynamic forces and torques $\vec{F}_a$ and $\vec{M}_a=\overrightarrow{GP}\times\vec{F}_a$ acting on the vehicle remains one of the major problems in the modeling process. Results on this topic can be found in [1] [42, Ch. 2] [43, Ch. 2] for fixed-wing aircraft, in [32] [15] [6] for quadrotors, in [17] [21] [29, Ch. 3] [30] for ducted-fan tail-sitters and in [27], [33], [44] for helicopters. As explained above, we assume that the actuators (*e.g.*, propellers) are not affected by environmental conditions and, therefore, we focus hereafter on the modeling of aerodynamic forces acting on the vehicle's main body.

A well-accepted general expression of aerodynamic forces and moments can be deduced by applying the so-called Buckingham π theorem [1, p. 34] [5]. More precisely, we denote with $\vec{v}_a$ the air velocity, which is defined as the difference between $\vec{v}$ and the wind velocity $\vec{v}_w$, *i.e.*, $\vec{v}_a=\vec{v}-\vec{v}_w$. The lift force $\vec{F}_L$ is the aerodynamic force component along a perpendicular to the air velocity and the drag force $\vec{F}_D$ is the aerodynamic force component in the direction of the air velocity. Now, consider a (any) pair of angles (α, β) characterizing the orientation of $\vec{v}_a$ with respect to the body frame (*e.g.*, figure 3). Combining the Buckingham π–theorem [1, p. 34] with the knowledge that the intensity of the steady aerodynamic force varies approximately as the square of the air speed $|\vec{v}_a|$ yields the existence of two dimensionless functions $C_L(\cdot)$ and $C_D(\cdot)$ depending on the Reynolds number Re, the Mach number M, and (α, β), and such that

$$\begin{aligned}
&\vec{F}_a=\vec{F}_L+\vec{F}_D\\
&\vec{F}_L=k_a|\vec{v}_a|C_L\left(R_e,M,\alpha,\beta\right)\vec{r}\left(\alpha,\beta,|\vec{v}_a|\right)\times\vec{v}_a\\
&\vec{F}_D=-k_a|\vec{v}_a|C_D(R_e,M,\alpha,\beta)\vec{v}_a\\
&\vec{r}.\vec{v}_a=0\qquad|\vec{r}|=1\\
&k_a=\rho\Sigma/2
\end{aligned}\tag{5}$$

where ρ is the free stream air density, Σ is an area germane to the given body shape, $\vec{r}(\cdot)$ is a unit vector-valued function, and $C_D(\in\mathbb{R}^+)$ and $C_L(\in\mathbb{R})$ are the aerodynamic characteristics of the body, *i.e.*, the drag coefficient and lift coefficient, respectively. By using the above representation of the aerodynamic force – first introduced in [37] – the lift direction is independent from the aerodynamic coefficients, which in turn characterize the aerodynamic force intensity $(|\vec{F}_a|=k_a|\vec{v}_a|^2\sqrt{C_L^2+C_D^2})$, while the lift direction is fully characterized by the unit vector $\vec{r}(\cdot)$, which only depends on (α, β) and the air velocity magnitude $|\vec{v}_a|$. We will see that geometric symmetries of the vehicle's shape imply precise expressions of the vector $\vec{r}(\cdot)$.

The main assumption under which the model (5) holds, is that the effects of the vehicle's rotational and unsteady motions on its surrounding airflow pattern are not preponderant [42, p. 199]. For instance, a constant angular velocity flight generates a different airflow pattern from that in steady cruise, which means that the aerodynamic forces and moments in general depend also on the vehicle's angular velocity. In addition, the aircraft translational and rotational accelerations also perturb the airflow pattern, which in turn causes transient effects that should be taken into account for precise aerodynamic modeling. These effects will be neglected here, which leads us to assume (5) as the model of the aerodynamic forces.

Preliminaries on control design

From the assumptions and simplifications made in § "Dynamics of aircraft motions", the control model reduces to

$$\begin{aligned}
&m\vec{a}=m\vec{g}+\vec{F}_a-T\vec{k}\\
&\frac{d}{dt}(\vec{i},\vec{j},\vec{k})=\vec{\omega}\times(\vec{i},\vec{j},\vec{k})\\
&\frac{d}{dt}(\vec{J}.\vec{\omega})=\overrightarrow{GP}\times\vec{F}_a+\vec{\Gamma}_G
\end{aligned}\tag{6}$$

where $\vec{a}=\frac{d\vec{v}}{dt}$ is the linear acceleration of the vehicle. To develop general control principles applicable to a large number of aerial vehicles, it is necessary to become free of actuation specificities and

concentrate on the vehicle's governing dynamics. In agreement with a large number of works on VTOL control (see [13] for a survey) we assume that the torque control $\vec{\Gamma}_G$ allows us to modify the body's instantaneous angular velocity $\vec{\omega}$ at will. Consequently, the angular velocity $\vec{\omega}$ can be considered as an intermediate control input. The above consideration implicitly means that the torque calculation and the ways of producing this torque can be decoupled from high-level control objectives, at least in the first design stage. The corresponding physical assumption is that "almost" any desired angular velocity can be obtained within a short amount of time. In the language of Automatic Control, this is a typical "backstepping" assumption. Once it is made, the vehicle's actuation consists of four input variables, namely, the thrust intensity and the three components of $\vec{\omega}$. The control model (6) then reduces to

$$\begin{aligned} m\vec{a} &= m\vec{g} + \vec{F}_a - T\vec{k} \\ \frac{d}{dt}(\vec{i},\vec{j},\vec{k}) &= \vec{\omega}\times(\vec{i},\vec{j},\vec{k}) \end{aligned} \qquad (7)$$

where T and $\vec{\omega}$ are the system's control inputs.

Basics of control design

The control model (7) highlights the role of the gravity force $m\vec{g}$ and aerodynamic force $\vec{F}_a$ in obtaining the body's linear acceleration vector $\vec{a}$. It shows, for instance, that to move with a constant reference velocity the controlled thrust vector $T\vec{k}$ must be equal to the resultant external force

$$\vec{F}_{ext} = m\vec{g} + \vec{F}_a$$

When $\vec{F}_a$ does not depend on the vehicle's orientation, as in the case of spherical bodies subjected to orientation-independent drag forces only, the resultant external force does not depend on this orientation either (see figure 4 for an illustration). The control strategy then basically consists in aligning the thrust direction $\vec{k}$ with the direction of $\vec{F}_{ext}$ (orientation control with $\vec{\omega}$) and in opposing the thrust magnitude to the intensity of $\vec{F}_{ext}$ (thrust control with T). In other words, the desired thrust direction and magnitude are defined by

$$\vec{k} = \pm\frac{\vec{F}}{|\vec{F}|} \qquad T = \pm|\vec{F}| \qquad (8)$$

where

$$\vec{F} = \vec{F}_{ext}$$

Now, to ensure asymptotic stabilization of the reference velocity, it is necessary to incorporate feedback terms in the velocity dynamics. This can easily be done by changing the definition of $\vec{F}$ in (8). More precisely, the first equation in (7) can also be written as

$$m\vec{a} = m\frac{d\vec{v}}{dt} = \vec{F} - T\vec{k} + m\xi(\vec{v},t)$$

with

$$\vec{F} = m\vec{g} + \vec{F}_a - m\xi(\vec{v},t) \qquad (9)$$

and where $\xi(\vec{v},t)$ is some stabilizing control, which contains typically both feedback and feedforward terms. It then follows from (8) that

$$\frac{d\vec{v}}{dt} = \xi(\vec{v},t)$$

Figure 4 – A physical illustration of the spherical model

When aerodynamic forces depend on the vehicle's orientation, as is the case of most aerial vehicles, the above control strategy raises important issues. In particular, the resultant force $\vec{F}_{ext}$ being now orientation-dependent, the existence and uniqueness of the equilibrium in terms of the vehicle's orientation is no longer systematic, since the right-hand side of the first equality in (8) may also depend on $\vec{k}$. Even when such an equilibrium solution is well defined and locally unique, its stabilization can be very sensitive to thrust orientation variations. As a matter of fact, the capacity of calculating the direction and intensity of $\vec{F}_a$ at every time-instant – already a quite demanding requirement – is not sufficient to design a control law capable of performing well in (almost) all situations. Knowing how this force changes when the vehicle's orientation varies is necessary, but is still not sufficient. In the following section we point out the existence of a set of aerodynamic models that allow the control problem to be recast into that of controlling a spherical body. Of course, the underlying assumptions are that these models reflect the physical reality sufficiently well and that the corresponding aerodynamic forces can be either measured or estimated on-line with sufficient accuracy.

Nonlinear versus linear feedback control

Good stability properties can be obtained with linear feedback control for some operating conditions, such as, for example, hover flight with moderate wind, cruising flight at constant or slowly varying linear velocity, etc. In very windy environments or for very aggressive flight, however, several reasons advocate for the use of nonlinear feedback. Let us mention some of them.

- As explained above, the basic principle of aerial vehicle control is to align the thrust direction with the direction of external forces. This orientation control problem can be solved locally, via a local parameterization of the orientation error (*e.g.*, Euler angles). It is well known that this kind of parameterization introduces singularities and artificially limits the stability domain. This is a problem in the case of strong perturbations that can temporarily destabilize the vehicle's attitude. In order to ensure large stability domains, it is necessary to design the feedback law directly on the underlying manifold (unit sphere for thrust direction control, or special orthogonal group for the control of the full orientation). Linear feedback is not best suited to the control on such compact manifolds.

- From (8), the thrust direction control is well defined only if $\vec{F}$ does not vanish. This is not a problem around the hover flight configuration, since $\vec{F} \approx \vec{F}_{ext} \approx m\vec{g}$. For large initial errors or demanding reference trajectories, however, $\vec{F}$ may vanish due to the control ξ (see (9)). In this case again, instead of a linear feedback it is better to use a bounded

nonlinear one (with norm smaller than the gravity constant), so as to limit the risk of $\vec{F}$ vanishing.

• Although this problem is not specifically addressed in this paper, control limitations in both magnitude and rate can be problematic in practice. For example, linearizing the model (6) around the hover flight configuration yields two second-order linear systems (yaw and vertical dynamics) and two fourth-order linear systems (horizontal dynamics). While saturating the input of a Hurwitz-stable second-order linear system does not destroy its global asymptotic stability property, this is no longer true for linear systems of higher order (three or more). This also advocates for the use of nonlinear feedback solutions to address these control limitation issues (see for example [2], [18] for results on this topic).

Symmetric bodies and spherical equivalence

We have briefly explained in the previous section the basics of feedback control design for a spherical body (*i.e.*, subjected to aerodynamic forces independent of the vehicle's orientation). For most vehicles encountered in practice, however, aerodynamic forces depend on the vehicle's orientation. We show in this section that for a class of such systems, a preliminary feedback transformation on the input allows the dynamics to be rewritten in the same form as in the case of a spherical body. This will be instrumental for the control design methodology described further on.

The expression (5) of the aerodynamic forces holds independently of the body's shape. As has already been shown in [37], [35], in the case of shape symmetries, aerodynamic properties that simplify the associated control problem can be pointed out.

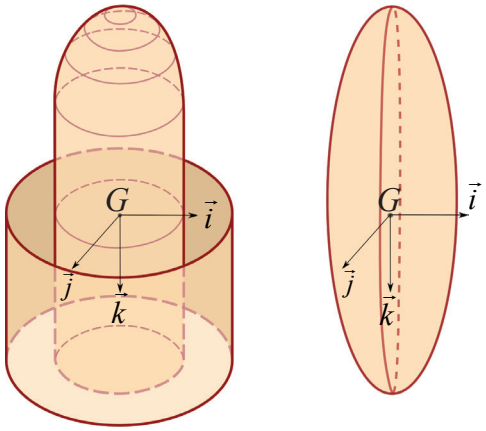


Figure 5 – Symmetric and bisymmetric shapes

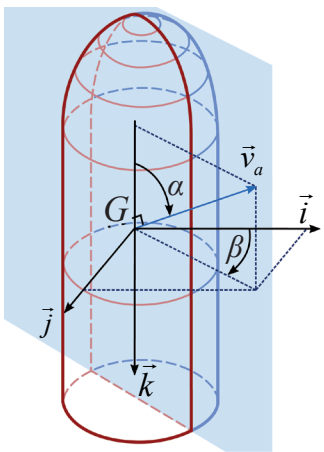


Figure 6 – The (α, β) angles

More specifically, if the body's shape is symmetric[3] around the thrust axis $\vec{k}$, then the unit vector $\vec{r}(\cdot)$ in (5) is given by

$$\vec{r} = \cos(\beta)\vec{j} - \sin(\beta)\vec{i}. \tag{10}$$

This allows us to decompose the aerodynamic force $\vec{F}_a$ as follows [37] [35]:

$$\vec{F}_a = -k_a |\vec{v}_a| \left[\left(C_D(R_e, M, \alpha) + C_L(R_e, M, \alpha)\cot(\alpha) \right) \vec{v}_a + \frac{C_L(R_e, M, \alpha)}{\sin(\alpha)} |\vec{v}_a| \vec{k} \right] \tag{11}$$

where $\alpha \in [0, \pi]$ is defined as the angle of attack between $-\vec{k}$ and $\vec{v}_a$, and $\beta \in (-\pi, \pi]$ as the angle between the unit frame vector $\vec{i}$ and the projection of $\vec{v}_a$ on the plane $\{G; \vec{i}, \vec{j}\}$ (see figure 6), *i.e.*,

$$\alpha = \cos^{-1}\left(-\frac{v_{a_3}}{|\vec{v}_a|} \right) \qquad \beta = \operatorname{atan2}(v_{a_2}, v_{a_1}) \tag{12}$$

Note that

$$\begin{aligned} v_{a_1} &= |\vec{v}_a| \sin(\alpha)\cos(\beta) \\ v_{a_2} &= |\vec{v}_a| \sin(\alpha)\sin(\beta) \\ v_{a_3} &= -|\vec{v}_a| \cos(\alpha) \end{aligned} \tag{13}$$

where $v_{a_i}\ (i = 1, 2, 3)$ denote the coordinates of $\vec{v}_a$ in the body-fixed frame. Note also that the above choice for the angles (α, β) renders the aerodynamic coefficients in (11) independent of the angle β.

For constant Reynolds and Mach numbers, the aerodynamic coefficients depend only on α. By using the relation (11), it is a simple matter to establish the following result.

Proposition 1 ([37], [35]) *Consider a symmetric thrust-propelled vehicle. Assume that the aerodynamic forces are given by (5) - (10) and that the aerodynamic coefficients satisfy the following relation*

$$C_D(\alpha) + C_L(\alpha)\cot(\alpha) = C_{D_0} \tag{14}$$

where C_{D_0} denotes a constant number. Then, the body's dynamic equation (7) can also be written as

$$m\vec{a} = m\vec{g} + \vec{F}_p - T_p\vec{k} \tag{15}$$

with

$$\begin{aligned} T_p &= T + k_a |\vec{v}_a|^2 \frac{C_L(\alpha)}{\sin(\alpha)} \\ \vec{F}_p &= -k_a C_{D_0} |\vec{v}_a| \vec{v}_a \end{aligned} \tag{16}$$

The interest of this proposition is to point out the possibility of viewing a symmetric body subjected to both drag and lift forces as a sphere subjected to the drag force $\vec{F}_p$ and powered by the thrust force $\vec{T}_p = -T_p\vec{k}$. The main condition is that the relation (14) must be satisfied. Obviously, this condition is compatible with an infinite number of functions C_D and C_L. Let us point out a particular set of simple functions that also satisfy the π-periodicity property with respect to the angle of attack α associated with bisymmetric bodies.

[3] See [37], [35] for a precise definition of shape symmetry and bisymmetry.

Proposition 2 *The functions C_D and C_L defined by*

$$C_D(\alpha) = c_0 + 2c_1 \sin^2(\alpha),$$
$$C_L(\alpha) = c_1 \sin(2\alpha) \quad (17)$$

where c_0 and c_1 are two real numbers, satisfy the condition (14) with $C_{D_0} = c_0 + 2c_1$. The equivalent drag force and thrust intensity are then given by

$$\vec{F}_p = -k_a C_{D_0} |\vec{v}_a| \vec{v}_a$$
$$T_p = T + 2c_1 k_a |\vec{v}_a|^2 \cos(\alpha) \quad (18)$$

A particular bisymmetric body is the sphere whose aerodynamic characteristics (zero lift and constant drag coefficient) are obtained by setting $c_1 = 0$ in (17). Elliptic-shaped bodies are also symmetric but, in contrast with the sphere, they do generate lift in addition to drag. The process of approximating measured aerodynamic characteristics with functions given by (17) is illustrated by the figure 7a, where we have used experimental data borrowed from [19, p.19] for an elliptic-shaped body with Mach and Reynolds numbers equal to $M = 6$ and $Re = 7.96.10^6$ respectively. For this example, the identified coefficients are $c_0 = 0.43$ and $c_1 = 0.462$. Since missile-like devices are "almost" bisymmetric, approximating their aerodynamic coefficients with such functions can also be attempted. For instance, the approximation shown in figure 7b has been obtained by using experimental data taken from [40, p.54] for a missile moving at $M = 0.7$. In this case, the identified coefficients are $c_0 = 0.1$ and $c_1 = 11.55$. In both cases, the match between experimental data and the approximating functions, although far from perfect, should be sufficient for feedback control purposes.

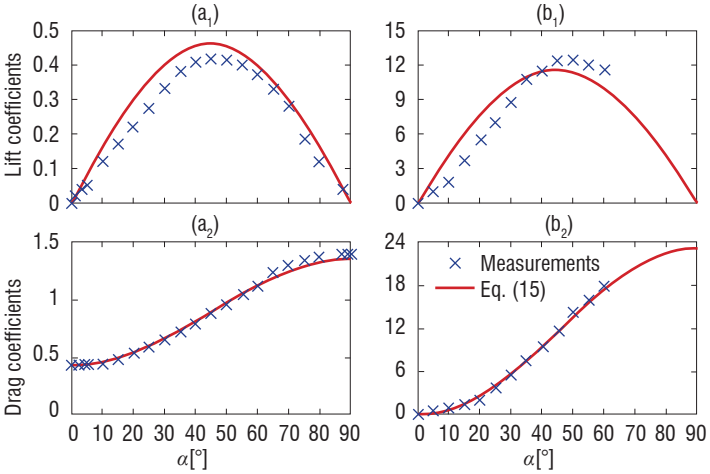


Figure 7 – Aerodynamic coefficients of: ($a_{1,2}$) elliptic bodies; ($b_{1,2}$) missile-like bodies

Control design

In this section, we propose nonlinear feedback laws for various control objectives. The first objective is the thrust direction control, which is essential for the control of VTOL UAVs. It is useful by itself, since the basic teleoperation mode for a VTOL UAV relies on thrust direction and thrust intensity reference inputs. Thrust direction control is also the cornerstone for higher-level (semi-)autonomous flight modes, such as, for example, velocity control, position control, or vision-based control. The second objective considered in this section is velocity control. After thrust direction control, velocity control is the next step in increasing the system's autonomy. Since the velocity dynamics is involved, the role of aerodynamic forces becomes predominant. This will be an opportunity to show the interest of the transformation proposed in § "Symmetric bodies and spherical equivalence". Once the velocity control level has been defined, the control design can be developed further to address, for example, disturbance rejection and/or position control. These topics are also briefly discussed in this section.

Thrust direction control

Consider a time-varying reference thrust (unit) direction $\vec{k}_r$. It is assumed that $\vec{k}_r$ varies smoothly with time, so that $\frac{d\vec{k}_r}{dt}(t)$ is well defined for any time t. The following result provides control expressions for the angular velocity control input $\vec{\omega}$ yielding a large stability domain.

Proposition 3 *The feedback law*

$$\vec{\omega} = \frac{k_0}{(1+\vec{k}\cdot\vec{k}_r)^2}\vec{k}\times\vec{k}_r + \vec{\omega}_r - (\vec{k}\cdot\vec{\omega}_r)\vec{k} + \lambda\vec{k} \quad (19)$$

with $\vec{\omega}_r = \vec{k}_r \times \frac{d\vec{k}_r}{dt}$, k_0 a positive real number, and λ any real number (not necessarily constant), ensures exponential stability of the equilibrium $\vec{k} = \vec{k}_r$ with domain of attraction $\{\vec{k}(0) : \vec{k}(0)\cdot\vec{k}_r(0) \neq -1\}$.

The limitation on the stability domain is related to the topology of the unit sphere, which prevents the existence of smooth autonomous feedback controllers yielding global asymptotic stability. The first term on the right-hand side of (19) is a nonlinear feedback term on the error between $\vec{k}$ and $\vec{k}_r$ (here defined from the cross product). The second and third terms are feedforward terms. In practice, these terms are often neglected because the vector $\frac{d\vec{k}_r}{dt}$ (and thus $\vec{\omega}_r$) is unknown. For example, if $\vec{k}_r$ corresponds to a reference thrust direction provided by a pilot via a joystick, $\frac{d\vec{k}_r}{dt}$ is not available. Omitting these feedforward terms does not prevent good results from being obtained, provided that k_0 is chosen sufficiently large and/or $\vec{k}_r$ does not vary too rapidly. Finally, the last term on the right-hand side of (19) is associated with the rotation about the axis $\vec{k}$ (yaw degree of freedom). It does not affect the thrust direction dynamics, since $\frac{d\vec{k}}{dt} = \vec{\omega}\times\vec{k}$.

Velocity control for vehicles with symmetric body shapes

Let us thus consider a symmetric vehicle and its velocity dynamics given by (15). The problem is to asymptotically stabilize a reference velocity $\vec{v}_r$. We follow the control strategy briefly sketched in § "Preliminaries on control design". Let us define the velocity error $\tilde{\vec{v}} = \vec{v} - \vec{v}_r$ and the reference acceleration $\vec{a}_r = \frac{d}{dt}\upsilon_r$. It follows from (15) that

$$m\frac{d\tilde{\vec{v}}}{dt} = \vec{F}_p + m(\vec{g} - \vec{a}_r) - T_p\vec{k}$$

The above equation can be written as

$$m\frac{d\tilde{\vec{v}}}{dt} = \vec{F} - T_p\vec{k} + m\xi(\tilde{\vec{v}}) \quad (20)$$

with

$$\vec{F} = \vec{F}_p + m(\vec{g} - \vec{a}_r - \xi(\tilde{\vec{v}})) \tag{21}$$

and where $\xi(\tilde{\vec{v}})$ is some feedback term. If $\xi(\tilde{\vec{v}})$ is chosen as a stabilizing feedback law for the dynamics $\frac{d\tilde{\vec{v}}}{dt} = \xi$, Eq. (20) suggests setting $T_p = |\vec{F}|$ and then applying the angular velocity control law of Proposition 3 with $\vec{k}_r$ defined as the unit vector characterizing the direction of $\vec{F}$, *i.e.*,

$$\vec{k}_r = \frac{\vec{F}}{|\vec{F}|}$$

The conditions under which this strategy ensures the asymptotic stability of $\tilde{\vec{v}} = 0$ are specified in the following proposition.

Proposition 4 *Assume that $\vec{F}$ does not vanish along the reference trajectory $\vec{v}_r$. Then, the feedback law defined by $T_p = |\vec{F}|$ and $\vec{\omega}$ given by (19) with*

$$\xi(\tilde{\vec{v}}) = -k_1 \frac{\tilde{\vec{v}}}{\sqrt{1+|\tilde{\vec{v}}|^2}}, \quad k_0 = k_2 |\vec{F}|^2$$

$k_{1,2}$ two positive real numbers and λ any real number (not necessarily constant), ensures local exponential stability of the equilibrium $(\vec{v}, \vec{k}) = (\vec{v}_r, \vec{k}_r)$.

This proposition is established by showing that the candidate Lyapunov function

$$V = \sqrt{|\tilde{\vec{v}}|^2 + 1} - 1 + \alpha(1 - \vec{k}\cdot\vec{k}_r)$$

with $\alpha > \frac{1}{mk_1k_2}$, is strictly decreasing along the solutions of the controlled system. It is important to note at this point that this property holds true as long as $|\vec{F}|$ is not zero (so that the control law is well defined). Thus, the limitation on the stability domain only comes from the possibility of $\vec{F}$ vanishing. Recall from (16) and (21) that

$$\vec{F} = -k_a C_{D_0} |\vec{v}_a| \vec{v}_a + m(\vec{g} - \vec{a}_r - \xi(\tilde{\vec{v}}))$$

Since $\xi(\tilde{\vec{v}})$ is bounded in norm by k_1, it is easy to impose conditions on k_1 and the reference acceleration $\vec{a}_r$ such that the term $m(\vec{g} - \vec{a}_r - \xi(\tilde{\vec{v}}))$ does not vanish whatever the tracking error $\tilde{\vec{v}}$. This is not sufficient to ensure that $\vec{F}$ never vanishes, however, since the term $k_a C_{D_0} |\vec{v}_a| \vec{v}_a$ can take arbitrary values, depending on the value of $\vec{v}$. If $\vec{F}$ does not vanish along the reference trajectory $\vec{v}_r$, then local stability is guaranteed and, using the fact that V is decreasing along the solutions of the controlled system, (possibly conservative) stability domains can be specified.

Note that, in view of (21), the independence of $\vec{F}_p$ with respect to the vehicle's orientation in turn implies that $\vec{F}$, and thus $\vec{k}_r$, are also independent of the vehicle's orientation. Therefore, the time-derivative of $\vec{k}_r$ does not depend on the vehicle's angular velocity $\vec{\omega}$ either and the expression of $\vec{\omega}$ in (19) is well defined. The interest of the invoked transformation, combined with (14), lies precisely there.

In practice, the control law must be complemented with integral correction terms to compensate for almost constant unknown additive perturbations. With $\vec{x}_r$ denoting the reference position of the center of mass in the inertial frame, the solution proposed in [12] involves the position error $\tilde{\vec{x}} = \vec{x} - \vec{x}_r$ expressed in the inertial frame, which is an integral of the velocity error $\tilde{\vec{v}}$. To further impose a bound on the integral correction action, a smooth bounded strictly positive function h defined on $[0, +\infty)$ and that satisfies the following properties ([12, Sec. III.C]) for some positive constant numbers η, μ can be introduced:

$$\forall s \in \mathbb{R}, \quad |h(s^2)s| < \eta \text{ and } 0 < \frac{\partial}{\partial s}(h(s^2)s) < \mu$$

An example of such a function is $h(s) = \frac{\eta}{\sqrt{1+s}}$, with $\eta > 0$. It then suffices to replace the definition of $\vec{F}$ in (21) by

$$\vec{F} = \vec{F}_p + m(\vec{g} - \vec{a}_r - \xi(\tilde{\vec{v}}) + h(|\tilde{\vec{x}}|^2)\tilde{\vec{x}}) \tag{22}$$

with the feedback control law still defined by $T_p = |\vec{F}|$ and $\vec{\omega}$ given by (19), to obtain a control law that includes an integral correction action and yields strong stability and convergence properties.

The above integral correction is, in fact, suited to the case when the control objective of tracking the desired velocity $\vec{v}_r$ is complemented with that of rendering the position error $|\tilde{\vec{x}}|$ small, with the vehicle's absolute position $\vec{x}$ being measured or estimated on-line. Otherwise, it is better to calculate and use a saturated integral of the velocity error. Such an integral $\vec{I}_v$ is, for instance, obtained as the (numerical) solution to the following equation [23] [39]

$$\frac{d}{dt}\vec{I}_v = -k_I \vec{I}_v + k_I \mathrm{sat}^{\delta}\left(\vec{I}_v + \frac{\tilde{\vec{v}}}{k_I}\right) \quad \vec{I}_v(0) = 0 \tag{23}$$

where k_I is a (not necessarily constant) positive number characterizing the desaturation rate, $\delta > 0$ is the upper bound of $|\vec{I}_v|$ and sat^{δ} is the classical saturation function defined by $\mathrm{sat}^{\delta}(\vec{x}) = \min\left(1, \frac{\delta}{|\vec{x}|}\right)\vec{x}$. A discrete-time version of this saturated integral is

$$I(j\Delta) = \begin{cases} \tilde{\vec{I}}_v(j\Delta) & \text{if } |\tilde{\vec{I}}_v(j\Delta)| \leq \\ \dfrac{}{|\bar{I}(j)|}\bar{I}(j) & \text{otherwise} \end{cases} \tag{24}$$

where $j \in \mathbb{N}$, Δ is the sampling time period and $\tilde{\vec{I}}_v(j\Delta) = \vec{I}_v((j-1)\Delta) + \tilde{\vec{v}}(j\Delta)$ for $j \geq 1$. Setting, for instance, $k_I = k_{I,0} + \frac{|\tilde{\vec{v}}|}{\delta}$ where $k_{I,0} > 0$, the definition of $\vec{F}$ only has to be replaced by

$$\vec{F} = \vec{F}_p + m(\vec{g} - \vec{a}_r - \xi(\tilde{\vec{v}}) + k_v \vec{I}_v) \tag{25}$$

where k_v is a positive gain, to obtain control yielding stability results similar to those obtained with the previous controllers.

Remark 1 *As in the case of velocity control, the position controller presented previously can also be modified to include an integral action that will improve its convergence properties when slowly varying unmodeled additive terms act on the system. The reader is referred to [12] for complementary details about this modification.*

Conclusion and perspectives

This paper has reviewed basic principles of the modeling and control of VTOL UAVs and a nonlinear control approach for a class of vehicles with symmetric body shapes has been proposed. Application examples are given, for example, by rockets and aerial vehicles using annular wings for the production of lift. Specific aerodynamic properties associated with these particular shapes allow for the design of nonlinear feedback controllers yielding asymptotic stability in a very large flight envelope.

Exploiting the aerodynamic characteristics for the design of feedback controllers with large flight envelopes remains a very open research domain. For example, extending the present approach to vehicles with non-symmetric body shapes (*e.g.*, conventional airplanes) is an open topic. A better understanding of the control limitations induced by the stall phenomenon is also necessary (see for example [34] for a study on this topic). Finally, it is very important to take into account the effect of magnitude (and rate) input saturations on the system's stability ■

Acknowledgements

P. Morin has been supported by *"Chaire d'excellence en Robotique RTE-UPMC"*

References

[1] J.D. ANDERSON – *Fundamentals of Aerodynamics.* McGraw Hill Series in Aeronautical and Aerospace Engineering, 5th edition, 2010.
[2] J. R. AZINHEIRA and A. MOUTINHO – *Hovering Control of an UAV with Backtepping Design Including Input Saturations*. IEEE Trans. on Control Systems Technology, 16(3):517–526, 2008.
[3] S. BERTRAND, H. PIET-LAHANIER and T. HAMEL – *Contractive Model Predictive Control of an Unmanned Aerial Vehicle Model.* 17th IFAC Symp. on Automatic Control in Aerospace, Volume 17, 2007.
[4] S. BOUABDALLAH and R. SIEGWART – *Backstepping and Sliding-Mode Techniques Applied to an Indoor Micro Quadrotor.* IEEE International Conference on Robotics and Automation, 2005.
[5] P. W. BRIDGMAN – *Dimensional Analysis.* Encyclopedia Britannica (Wm. Haley, Editor-in-Chief), 7:439–449, 1969.
[6] P.J. BRISTEAU, P. MARTIN and E. SALAUN – *The Role of Propeller Aerodynamics in the Model of a Quadrotor UAV.* European Control Conference, pp. 683–688, 2009.
[7] A. DZUL, T. HAMEL and R. LOZANO – *Modeling and Nonlinear Control for a Coaxial Helicopter.* IEEE Conf. on Systems, Man and Cybernetics, Vol. 6, 2002.
[8] T. I. FOSSEN – *Guidance and Control of Ocean Vehicles.* John Wiley and Sons, 1994.
[9] T. HAMEL, R. MAHONY, R. LOZANO and J. OSTROWSKI – *Dynamic Modelling and Configuration Stabilization for an X4-Flyer.* IFAC World Congress, pp. 200–212, 2002.
[10] J. HAUSER, S. SASTRY and G. MEYER – *Nonlinear Control Design for Slightly Non-Minimum Phase Systems: Application to V/Stol.* Automatica, 28:651–670, 1992.
[11] M.-D. HUA – *Contributions to the Automatic Control of Aerial Vehicles.* PhD thesis, http://hal.archives-ouvertes.fr/tel-00460801, Université de Nice-Sophia Antipolis, 2009.
[12] M.-D. HUA, T. HAMEL, P. MORIN and C. SAMSON – *A Control Approach for Thrust-Propelled Underactuated Vehicles and its Application to VTOL Drones.* IEEE Trans. on Automatic Control, 54(8):1837–1853, 2009.
[13] M.-D. HUA, T. HAMEL, P. MORIN and C. SAMSON – *Introduction to Feedback Control of Underactuated VTOL Vehicles*. IEEE Control Systems Magazine, pp. 61–75, 2013.
[14] M.-D. HUA, P. MORIN and C. SAMSON – *Balanced-Force-Control for Underactuated Thrust-Propelled Vehicles.* IEEE Conf. on Decision and Control, pp. 6435–6441, 2007.
[15] H. HUANG, G. M. HOFFMANN, S. L. WASLANDER and C. J. TOMLIN – *Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering.* IEEE Conf. on Robotics and Automation, pp. 3277– 3282, 2009.
[16] A. ISIDORI, L. MARCONI and A. SERRANI – *Robust Autonomous Guidance: an Internal-Model Based Approach.* Springer Verlag, 2003.
[17] E. N. JOHNSON and M. A. TURBE – *Modeling, Control and Flight Testing of a Small Ducted Fan Aircraft.* Journal of Guidance, Control, and Dynamics, 29(4):769–779, 2006.
[18] F. KENDOUL, D. LARA, I. FANTONI and R. LOZANO – *Nonlinear Control for Systems with Bounded Inputs: Real-Time Embedded Control Applied to UAVs.* IEEE Conf. on Decision and Control, pp. 5888–5893, 2006.
[19] J. W. KEYES – *Aerodynamic Characteristics of Lenticular and Elliptic Shaped Configurations at a Mach Number of 6.* Technical Report NASA-TN-D-2606, NASA, 1965.
[20] H. J. KIM, D. H. SHIM and S. SASTRY – *Nonlinear Model Predictive Tracking Control for Rotorcraft-Based Unmanned Aerial Vehicles.* American Control Conference, pp. 3576–3581, 2002.
[21] A. KO, O. J. OHANIAN and P. GELHAUSEN – *Ducted Fan UAV Modeling and Simulation in Preliminary Design.* AIAA Modeling and Simulation Technologies Conference and Exhibit, n° 2007–6375, 2007.
[22] T. J. KOO and S. SASTRY – *Output Tracking Control Design for a Helicopter Model Based on Approximate Linearization.* IEEE Conf. on Decision and Control, pp. 3635–3640, 1998.
[23] C. SAMSON and M.-D. HUA – *Time Sub-Optimal Nonlinear Pi and Pid Controllers Applied to Longitudinal Headway Car Control.* Int. J. of Control, 84-10:1717–1728, 2011.
[24] R. MAHONY, T. HAMEL and A. DZUL – *Hover Control via an Approximate Lyapunov Control for a Model Helicopter.* IEEE Conf. on Decision and Control, pp. 3490–3495, 1999.
[25] L. MARCONI, A. ISIDORI and A. SERRANI – *Autonomous Vertical Landing on an Oscillating Platform: an Internal Model Based Approach*. Automatica, 38:21–32, 2002.
[26] E. MUIR – *Robust Flight Control Design Challenge Problem Formulation and Manual: the High Incidence Research Model (HIRM).* Robust Flight Control, A Design Challenge (GARTEUR), Vol. 224 of Lecture Notes in Control and Information Sciences, pp. 419–443, Springer Verlag, 1997.
[27] R. NALDI – *Prototyping, Modeling and Control of a Class of Vtol Aerial Robots.* PhD thesis, University of Bologna, 2008.
[28] R. OLFATI-SABER – *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles.* PhD thesis, Massachusetts Institute of Technology, 2001.
[29] J.-M. PFLIMLIN – *Commande d'un minidrone à hélice carénée : de la stabilisation dans le vent à la navigation autonome.* PhD thesis, Ecole Doctorale Systèmes de Toulouse, 2006.
[30] J.-M. PFLIMLIN, P. BINETTI, P. SOUÈRES, T. HAMEL and D. TROUCHET – *Modeling and Attitude Control Analysis of a Ducted-Fan Micro Aerial Vehicle.* Control Engineering Practice, pp. 209–218, 2010.

[31] J.-M. PFLIMLIN, P. SOUÈRES and T. HAMEL – *Hovering Flight Stabilization in Wind Gusts for Ducted Fan UAV.* IEEE Conf. on Decision and Control, pp. 3491–3496, 2004.
[32] P. POUNDS, R. MAHONY and P. CORKE – *Modelling and Control of a Large Quadrotor Robot.* Control Engineering Practice, pp. 691–699, 2010.
[33] R.W. PROUTY – *Helicopter Performance, Stability and Control.* Krieger, 2005.
[34] D. PUCCI – *Flight Dynamics and Control in Relation to Stall.* American Control Conf. (ACC), pp. 118–124, 2012.
[35] D. PUCCI – *Towards a Unified Approach for the Control of Aerial Vehicles.* PhD thesis, Université de Nice-Sophia Antipolis and "Sapienza" Universita di Roma, 2013.
[36] D. PUCCI, T. HAMEL, P. MORIN and C. SAMSON – *Nonlinear Control of PVTOL Vehicles Subjected to Drag and Lift.* IEEE Conf. on Decision and Control (CDC), pp. 6177 – 6183, 2011.
[37] D. PUCCI, T. HAMEL, P. MORIN and C. SAMSON – *Modeling for Control of Symmetric Aerial Vehicles Subjected to Aerodynamic Forces.* arXiv, 2012.
[38] C. ROOS, C. DÖLL and J.-M. BIANNIC – *Flight Control Laws: Recent Advances in the Evaluation of their Robustness Properties.* Aerospace Lab, Issue 4, 2012.
[39] H. KHALIL and S. SESHAGIRI – *Robust Output Feedback Regulation of Minimum-Phase Nonlinear Systems Using Conditional Integrators.* Automatica, 41:43–54, 2005.
[40] B. F. SAFFEL, M. L. HOWARD and E. N. BROOKS – *A Method for Predicting the Static Aerodynamic Characteristics of Typical Missile Configurations for Angles of Attack to 180 Degrees.* Technical Report AD0729009, Department of the navy naval ship research and development center, 1971.
[41] S. N. SINGH and A. SCHY – *Output Feedback Nonlinear Decoupled Control Synthesis and Observer Design for Maneuvering Aircraft.* International Journal of Control, 31(31):781–806, 1980.
[42] R. F. STENGEL – *Flight Dynamics.* Princeton University Press, 2004.
[43] B. L. STEVENS and F. L. LEWIS – *Aircraft Control and Simulation.* Wiley-Interscience, 2nd edition, 2003.
[44] J. C. A. VILCHIS, B. BROGLIATO, A. DZUL and R. LOZANO – *Nonlinear Modelling and Control of Helicopters.* Automatica, 39:1583–1596, 2003.
[45] Q. WANG and R.F. STENGEL – *Robust Nonlinear Flight Control of High-Performance Aircraft.* IEEE Transactions on Control Systems Technology, 13(1):15–26, 2005.
[46] R. XU and U. OZGUNER – *Sliding Mode Control of a Class of Underactuated Systems.* Automatica, 44:233–241, 2008.

Acronyms

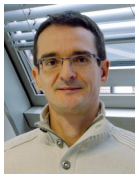
GARTEUR (Group for Aeronautical Research and Technology in Europe)
HIRM (High-Incidence Research Model)
LMI (Linear Matrix Inequality)
NED (North-East-Down)
UAV (Unmanned Aerial Vehicle)
VTOL (Vertical Take-Off and Landing)

AUTHORS

Daniele Pucci received the bachelor and master degrees in Control Engineering with highest honors from "Sapienza", University of Rome, in 2007 and 2009, respectively. He received the PhD title in Information and Communication Technologies from University of Nice Sophia Antipolis, and in Control Engineering, from "Sapienza" University of Rome, in 2013. Since then, he is a post-doctoral fellow at the Italian Institute of Technology. His research interests include control of nonlinear systems and its applications to aerial vehicles and robotics.

Minh-Duc Hua graduated from Ecole Polytechnique, France, in 2006, and received his Ph.D. from the University of Nice-Sophia Antipolis, France, in 2009. He spent two years as a postdoctoral researcher at I3S UNS-CNRS, France. He is currently researcher of the French National Centre for Scientific Research (CNRS) at the ISIR laboratory of the University Pierre and Marie Curie (UPMC), France. His research interests include nonlinear control theory, estimation and teleoperation with applications to autonomous mobile robots such as UAVs and AUVs.

Tarek Hamel is Professor at the University of Nice Sophia Antipolis since 2003. He conducted his Ph.D. research at the University of Technologie of Compiègne (UTC), France, and received his doctorate degree in Robotics from the UTC in 1996. After two years as a research assistant at the (UTC), he joined the Centre d'Etudes de Mécanique d'Ile de France in 1997 as an associate professor. His research interests include nonlinear control theory, estimation and vision-based control with applications to Unmanned Aerial Vehicles. He is currently Associate Editor for IEEE Transactions on Robotics and for Control Engineering Practice.



Pascal Morin received the Maîtrise degree from Université Paris-Dauphine in 1990, and the Diplôme d'Ingénieur and Ph.D. degrees from Ecole des Mines de Paris in 1992 and 1996 respectively. He spent one year as a post-doctoral fellow in the Control and Dynamical Systems Department at the California Institute of Technology. He was Chargé de Recherche at INRIA, France, from 1997 to 2011. He is currently in charge of the ``Chaire RTE-UPMC Mini-drones autonomes" at the ISIR lab of University Pierre et Marie Curie (UPMC) in Paris. His research interests include stabilization and estimation problems for nonlinear systems, and applications to mechanical systems such as nonholonomic vehicles or UAVs.

Claude Samson graduated from the Ecole Supérieure d'Electricité in 1977, and received his Docteur-Ingénieur and Docteur d'Etat degrees from the University of Rennes, in 1980 and 1983, respectively. He joined INRIA in 1981, where he is presently Directeur de Recherche. His research interests are in control theory and its applications to the control of mechanical systems. Dr. Samson is the coauthor, with M. Leborgne and B. Espiau, of the book *Robot Control. The Task-Function Approach* (Oxford University Press, 1991).

M. Sanfourche, A. Plyer,
A. Bernard-Brunel, G. Le Besnerais
(Onera)

E-mail: martial.sanfourche@onera.fr

DOI : 10.12762/2014.AL08-09

3DSCAN: Online Ego-Localization and Environment Mapping for Micro Aerial Vehicles

We present 3DSCAN (3D Scene Characterization for Autonomous Navigation), a software application for state estimation and environment modeling using low-cost 3D sensors, such as a stereorig and RGBD cameras. For state estimation, we describe an original keyframe-based stereoscopic visual odometry technique, which can run at more than 20Hz on a lightweight computer. This so-called 'efficient Visual Odometry' (eVO) has been evaluated on several datasets and provides accurate results and limited drift, even for indoor/outdoor trajectories. Environment modeling aggregates instantaneous depthmaps in a volumetric Octomap [15] representation. Stereoscopic depthmaps are computed by a very fast dense matching algorithm derived from eFolki, an optical flow code implemented on GPU. These developments are combined in the 3DSCAN software, which is successfully demonstrated on our MAV (Micro Aerial Vehicle) system, following indoor, outdoor or mixed trajectories.

Introduction

From an automation point of view, navigation consists in computing a safe and achievable trajectory and in controlling its execution. It requires a precise knowledge of both the dynamic state of the vehicle (position, attitude, speed) and of the 3D structure of the environment.

These prerequisites are very difficult to meet for a Micro Aerial Vehicles (MAV) flying low through a cluttered environment. The first difficulty arises from the partial knowledge of the environment. Despite the spread of available georeferenced information (images and maps), they still cannot be considered to be of sufficient accuracy and density for a mission in an unprepared environment such as flying around a building, avoiding unmapped obstacles, such as trees or parked cars, and entering through an open window. In this context, the vehicle should be able to build online a representation of its environment using embedded exteroceptive sensors.

The second difficulty is, of course, the very limited payload of MAV. Since 2005, demonstrations of on-line mapping have been done using heavy UAV (helicopter with weight > 25 kg) [1, 42] equipped with 3D vision sensors, such as lidar. However, a high resolution (HR) inertial measurement unit (IMU) and high grade differential GPS receiver were used to localize the vehicle. The very limited payload and power of MAV precludes the embedding of lidar and HR IMU. As a result, most current demonstrations of MAV use external resources, in particular external systems, for the estimation of the MAV state, such as multi-camera localization systems.

In this paper, we present a solution for state estimation and environment modeling based on low-cost 3D sensors, compatible with indoor and outdoor environments. There are now several solutions for these 3D sensors, which allow fast mapping of obstacles and lead to a well-behaved ego-localization problem, compared to solutions based on monocular 2D sensors subject to scale ambiguity and its drift. We first describe ego-localization using a stereoscopic visual odometer and then on-line modeling of the environment, the combination of the two functions being called 3DSCAN 3D Scene Characterization for Autonomous Navigation. This 3DSCAN system is demonstrated on publicly available data Kitti [12] and through several experiments performed at Onera using our MAV. In the latter demos, only the first function of 3DSCAN (ego-localization) runs on-board, but the proposed modeling solution is already compatible with recent embedded architectures.

Related Works and Contributions

Localization and mapping are active research fields since more than twenty years and an enormous amount of literature exists. In the fol-

lowing, we review some relevant references on these two topics and discuss our contribution.

Ego-localization

Vision-based ego-localization has reached a high level of maturity over the last decade. For example, NASA's two Mars Exploration Rovers (Spirit and Opportunity) have been successfully using stereo visual odometry since 2003. From a methodological point of view, two approaches are often opposed despite recent convergent trends: Visual Odometry (VO) and Visual Simultaneous Localization and Mapping (V-SLAM). These approaches are briefly summarized below. For a much more detailed review, we advise the reading of the recent two-part tutorial by D. Scaramuzza and F. Fraundorfer [40, 11].

• Basically, VO estimates the relative motion of the camera between t_k and t_k+1 by the camera pose at t_k+1 with regard to 3D reference data, for instance a cloud of 3D points, recorded in the camera frame at tk. VO can be distinguished based on 2D-3D associations [33, 14, 29], 3D-3D associations [20], 2D associations through 3 views using the trifocal constraint [18], or through 4 views using the quadrifocal constraint [6]. From relative motion information, the full trajectory can be estimated by simple dead-reckoning [14, 29] (Dead-Reckoning Visual Odometry, DRVO), or by fusion with inertial measurements; see [22, 3, 38] in the aerial context.

• V-SLAM addresses the problem of self-localization by constructing a globally-consistent map of the environment. This map is usually a sparse representation made of a limited number of landmarks, often 3D points. State variables (i.e., ego-localization and speed) and positions of 3D landmarks are estimated according to a joint criterion based on a 2D projection error. This can be done by filtering techniques like Extended Kalman Filter (EKF) or by multi-view optimization methods like Bundle Adjustment (BA). It is interesting to note that these approaches differ in the frequency of map updating operations. Filtering approaches [7, 2] update the map at each new frame, so as to maintain the consistency of the linearization, which requires the map to be limited to a few hundred landmarks. In contrast, optimization methods [28, 19, 39, 17, 32] wait until the baseline is sufficiently large to allow good localization of 3D landmarks. The latter approach leads to the notion of "keyframes", i.e., the frames used to update the map. Though the selection of keyframes is generally done by considering statistics over the feature tracking, [47] proposes a criterion based on image projection error.

The proposed method, denoted by eVO (efficient Visual Odometer) combines characteristics of both approaches, while being oriented towards low computational cost rather than optimality [31]. Therefore, as in VO, the 3D points of the reference map are estimated at one time instant and are not refined using others views. The notion of keyframe is also used, as in optimization approaches of V-SLAM: the 3D map is computed only at keyframe instants. Cameras are localized by pose estimation with regard to the current map, using a 2D-3D association strategy. We show that this keyframe strategy not only reduces the cost of the algorithm, but also significantly reduces the drift of the estimation error with respect to DRVO. As a result, the proposed eVO is able to run at video frame rates (15 to 25 Hz) on the limited computer available on our MAV.

Note that Nister et al. have proposed in ref. [33] a VO approach with keyframes (they called them reference frames) for ground vehicle applications. More recently, the Pixhawk team [10] have reported scene modeling using a MAV with a stereo sensor and on-board VO; however, there are very few details regarding the odometry in their paper, while we present here a parametric study of the performance of the algorithm on various datasets. In particular, eVO has been successfully tested on the online Kitti benchmark and ranks at the 6th position, and 4th among methods based on stereo data [12][1]

Environment modeling

Let us first discuss 3D sensors. In the proposed solution, we use two concurrent 3D sensors: a stereorig and an active Asus Xtion RGBD camera, a type of sensor made popular by Microsoft's Kinect device. They have complementary characteristics, the RGBD camera being very efficient indoors, but blind in outdoor situations, where the greater amount of natural textures allow a good stereovision performance. For stereo matching, we use a very fast dense algorithm eFolki, a dense Lucas-Kanade (LK) algorithm published in [4] and recently implemented on GPU (Graphic Processing Unit) architectures [34]. It allows dense and reliable 3D maps to be obtained at video rate on a lightweight laptop with a GT650M GPU.

Environment modeling amounts to aggregating noisy 3D point clouds into a consistent and compact 3D model. Note that the model should include not only the occupied areas, but also the free space and the unvisited areas. This data is required for trajectory planning and replanning. Proposed in the seminal work of Elfes [8], occupancy grids have become the standard for 2D and 3D environment models in mobile robotics [30, 1, 15]. In case of 3D modeling, a standard approach consists in subdividing the workspace into cubic volume elements of equal size called voxels [30].

This simplistic representation presents two disadvantages: (1) the maximal extension of the explored area must be known in advance; (2) the memory occupation becomes rapidly intractable for large scale environments. In [1], the authors propose a two-layer model, combining a local representation using standard voxel grid and a global rough polygonal model organized by height slices. This model is successfully used for automatic navigation in urban canyons, but appears not easily scalable. In [16], the model consists in a 2D regular grid, where each cell stores a list of parallelepipedic volumes corresponding to occupied or free areas. Memory efficient, this solution avoids sampling artifacts in the vertical direction but requires the knowledge of the horizontal workspace extension. In [15], occupied and free areas are also explicitly represented, but using a multi-resolution occupancy grid organized as an octree structure (hence its name: 'Octomap'). Such a structure offers a useful flexibility for modeling unknown areas. By adding levels to the tree, the spatial resolution can be adapted to the local 3D structure of the scene, or the global size of the workspace can be expanded easily. For these reasons, we use Octomap, thanks to the freely distributed C++ library[2].

[1]Please refer to the Kitti benchmark at http://www.cvlibs.net/datasets/kitti/eval_odometry.php.

[2]Please refer to the website http://octomap.github.io/

Organization of the paper

This paper is organized as follows. The overall architecture of 3DSCAN is presented in § " System overview". The description and performance evaluation of the ego-localization module eVO are presented in § " Efficient visual odometry (eVO)". § " Environment modeling" presents the 3D modeling module (stereo processing and aggregation). Reconstructions from Kitti data or from data obtained during test flights with our MAV are presented in § " 3DSCAN results". Finally, we conclude and propose some directions for future work.

System overview

Software architecture

Given image and depth data provided by sensors or read from files, the 3DSCAN processing chain builds a 3D environment model as an Octomap grid. Three processing modules, working as an individual thread at different frequencies, are combined: (1) the eVO module computes the camera pose at video rate, (2) the eFolki module computes the depthmap from a rectified and equalized stereo pair provided by eVO and (3) the Octomap module uses the estimated pose and depthmap to aggregate relative 3D data into a global model. These components communicate using ROS (Robot Operating System, www.ros.org). ROS also provides interface modules to obtain images from sensors, or from files and visualization modules.

Figure 1 depicts the implemented software architecture and data exchanges through the module network. In this organization, eVO, the stereo odometry module, plays a central role. In addition to calculating the camera position, the module geometrically and radiometrically rectifies a stereo pair and achieves a temporal sub-sampling of the sequence by automatically selecting keyframes. Note that the camera poses are saved by the Transform-Frame server (ROS/TF server) for further usage. When eVO selects a stereo pair as a new keyframe, the rectified stereo pair is processed by eFolki in order to compute a depthmap, which is converted into a relative-to-sensor 3D point cloud. In parallel to the stereo process, the RGBD sensor node emits depthmaps at 3Hz. The '3D data source selector' selects the most appropriate sensor (stereo or RGBD camera) depending on the density of the RGBD depthmap and transmits a point cloud to the Octomap server for aggregation in the environment model. This involves searching in the pose database stored in the module TF server for the sensor pose at the date indicated in the point cloud message.

Since ROS performs an abstraction of the hardware layer, the 3DSCAN chain has been deployed on various hardware units: PC workstation, laptop, MAV + ground station. The implementation on a MAV and its ground station (e.g., a MAV system) is described in the following section.

Implementation on a MAV system

We have deployed 3DSCAN on a real MAV system composed of a mid-range laptop used as a ground station and the Ascending Technologies Pelican[3] quadrotor depicted in figure 2.

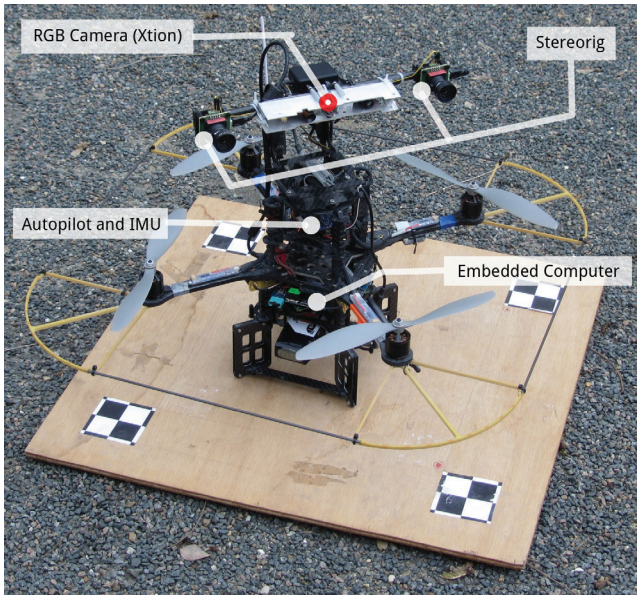


Fig. 2 - Our AscTec Pelican MAV on its landing pad. The visual sensors - stereorig and RGBD camera - are located at the top of the vehicle. The vehicle has a total take-off weight of 2 kg (including the LiPo Battery).

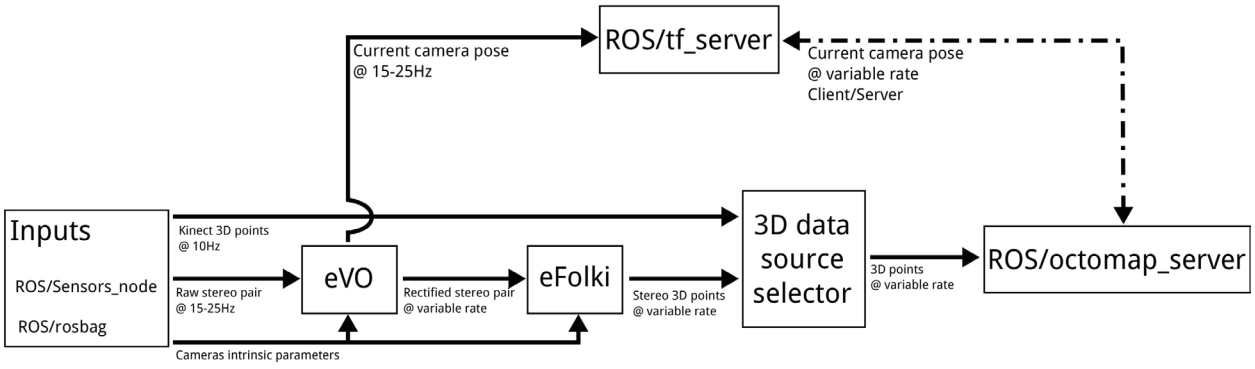


Fig. 1 - The 3DSCAN software architecture. Boxes correspond to ROS nodes. Boxes with a name beginning by 'ROS/' indicate a module provided in the standard ROS library. Solid unidirectional arrows indicate data exchanges in streaming mode, while dot dashed bidirectional arrows indicate exchanges in client/server mode.

[3]Please refer to the manufacturer website for details: http://www.asctec.de/uav-applications/research/products.

The MAV equipment includes a MEMS-based IMU, a low grade GPS receiver and a 3D magnetometer. These sensors are connected to the autopilot providing a standard stabilization mode and waypoint-based navigation. We have added an Asus Xtion RGBD camera and a stereorig composed of two electronically synchronized USB cameras separated by a 28 cm long baseline equipped with a 5.5 mm S-mount lens. This configuration provided a usable range of 10 meters for environment modelling[4]. The cameras are two IDS-Imaging UI-1241LE based on a 1.3 MegaPixel global-shuttered CMOS from e2V. Since the native resolution is too large for onboard processing, the binning mode is activated to capture VGA frames without field of view reduction. These sensors are connected to an embedded PC animated by an Intel dual-core Core2Duo 1.86 GHz.

The limited computational performance of the on-board PC and the requirement of a Cuda Compliant GPU for dense stereo matching by eFolki have led to the full processing chain being dispatched on two computers. The ego-localization by eVO runs onboard, while the entire environment modeling task is done on the ground station: a light macbook laptop equipped with a mid-range Nvidia GT 650M GPU. The datalink between the two computers is provided by Wifi-N. Since the transmission of all video streams is impracticable through the datalink, temporal subsampling is performed. For the RGBD camera, the ROS module permits constant subsampling and we have set the output frequency to 3Hz. For the stereo pairs, the subsampling is done by the mechanism selecting keyframe in eVO. This aspect will be discussed in the following section.

Efficient visual odometry (eVO)

Algorithm overview

As already mentioned in the introduction, eVO builds a map updated in a keyframe scheme as in ref. [28, 19]. In the monocular SLAM case, the keyframe structure is mainly motivated by the need for a minimal baseline to localize new 3D landmarks. With our stereo setting, landmarks are instantaneously localized in 3D. Improving the accuracy of a landmark localization requires the stereorig to get substantially closer to the landmark or to displace the sensor lengthways more than the baseline. Hence, in the case of a smooth motion of the stereorig (with respect to the rate of odometry), updating the map at each frame is useless and the keyframe scheme is a better choice.

In contrast to other keyframe-based SLAM, our system differs by the way in which the map is updated. In standard approaches, the positions of visible landmarks are refined at each keyframe by minimizing a multi-view re-projection criterion with bundle adjustment methods. Here, we skip this step because of the limited computational capacity of the embedded PC. In practice, landmarks are then localized once - the first time they are seen - in the global frame using the current estimated pose.

Direct combination of noisy measurements - camera pose and landmark position - brings eVO closer to DRVO, i.e. dead-reckoning methods. However, using the keyframe approach, this update is done at a lower rate in eVO than in DRVO, with the advantage of a reduced drift. A comparison between these two approaches on real datasets is presented in § "Tuning and advantage of the Keyframe Scheme".

Finally, the other advantage of this structure concerns the computational cost. Indeed, 3D localization by a stereorig is not computationally free. Combining a keyframe scheme with a pose computation algorithm using 2D-3D associations avoids computing the 3D structure at each new stereo frame. More interesting still, this approach allows the global process to be divided into one monocular task, the Tracking and Pose computation, executed for each left image acquired, and one stereo task, (Mapping) executed on demand.

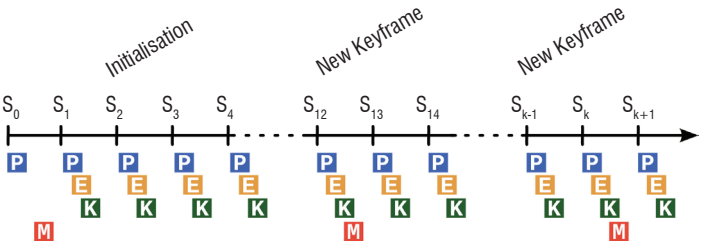


Fig. 3 - Temporal sequencing of the eVO module. The P-box, E-box, K-box and M-box stand respectively for the Image Pre-processing module, the Egomotion module, the Keyframe selector module and the Mapping module.

The eVO process can be described in four modules working sequentially, as depicted in figure 3:

- Image pre-processing: preliminary image warping and equalization;
- Egomotion: estimation of the position and the attitude of the stereorig in the reference frame;
- Keyframe selection: deciding whether a new keyframe is necessary;
- Mapping: stereo pair processing so as to update the landmark map.

Each module is described in the following section; here we briefly describe the eVO process. The algorithm starts by calling the Mapping module, which initializes 3D landmarks. The next available stereo pair is processed by the Egomotion module, which yields the current pose and indicates how many landmarks are still visible. This indicator is used by the keyframe selection module to decide that the current stereo pair is a new keyframe. In figure 3, this loop is repeated until the 13th stereo pair (denoted S12), which is selected as a keyframe. At this point, the mapping module is called to update the map: it adds new landmarks and prunes older ones.

Description of the eVO components

Here we give a detailed description of the eVO components, in the order in which they appear in figure 3.

Image pre-processing Module

The two images are stereo-rectified using the knowledge of the intrinsic parameters. In order to deal with indoor to outdoor (or vice versa) transitions, which lead to locally large illumination changes, two adjustments were necessary. The first one concerns the hardware: the cameras are set to automatically adapt their exposure time, in order to reach a specified intensity average under the constraint that the exposure time cannot exceed a maximal value. The second adjustment consists in equalizing image histograms to avoid dark images.

[4]Under the assumption of a mean disparity error of 0.5 pixels; the usable range is defined as the maximal distance before the precision of 3D localization exceeds the half-width of a voxel (20 to 30 centimeters).

Mapping Module

This module is called when the current stereo-pair is declared as a new keyframe. It uses a stereorectified pair of images to generate an initial map and to update it if necessary by extracting and matching new interest points. The synoptic diagram of this module is shown in figure 4.

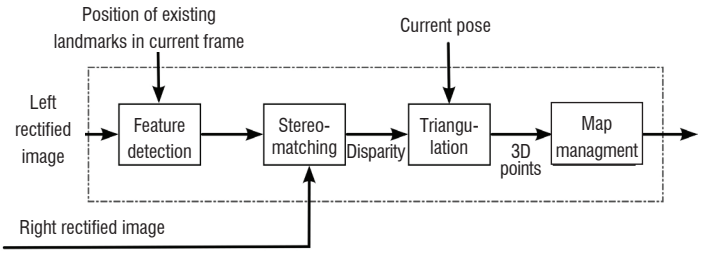


Fig. 4 - Mapping Module structure

The first operation consists in extracting Nf (between 250 and 350 for VGA images) interest points in the left image. This process is done under two geometrical constraints: (i) a minimal separation distance between two features; (ii) a maximal dispersion of the features over the image plane. The former constraint is generally included in feature extractors (like those in OpenCV), while the latter is enforced by a classical bucketing strategy. The image support is subdivided into Nr non-overlapping regions (8X6 regions for VGA images) and the Nf / Nr more relevant features within each region are kept. In order to deal with regions that do not have enough texture, a relaxation technique is used to increment the tolerated amount of features by region. Note that the extraction accounts for mapped landmarks successfully tracked from the previous keyframe, so as to detect only the correct number of new features and maintain Nf features per keyframe.

Two feature detectors have been evaluated: the Harris detector (Shi-Tomasi [43]) and the FAST detector [36]. As expected, the FAST detector is faster than the Harris detector and allows a keyframe (VGA format) to be processed in 55 milliseconds on average (see table 2). On the tested sequence, the choice of the detector has a very limited impact on the estimated trajectories.

In the second step, the features newly detected in the left images are matched in the right image. Based on dense stereovision algorithms, feature stereo matching is done by means of exhaustive searches along the epipolar lines. In practice, the Zero-mean Normalized Cross-Correlation (ZNCC) is used as the image similarity criterion and we test a range of disparities corresponding to 3D points located at least 70 centimeters away from the stereorig. In order to reduce the processing time, we adopt a coarse-to-fine multi-scale approach, with a two-level image pyramid. At the lowest resolution, the image is reduced by a factor of 4 in each direction and the size of the ZNCC window is set to 3X3 pixels. The match is then propagated to the full resolution level and refined by a local search within a region with a radius of 6 pixels, using a 9X9 ZNCC window. In practice, the number of tested disparity hypotheses is largely reduced. In our configuration (focal distance = 5.5 mm and depth greater than 70 cm), this approach allows the number of tested hypotheses to be reduced from 220 to less than 70. Finally, the ZNCC scores are thresholded to prune ambiguous associations

At this point, feature positions, disparities, stereorig parameters and the current pose estimation are used to localize the new landmarks in the reference frame by triangulation. Finally, new landmarks and their image signature are inserted into the map, while the untracked landmarks are removed.

Egomotion Module

As soon as some landmarks have been localized in 3D, the egomotion module estimates the position and the attitude of the left camera frame, by tracking the landmarks in the successive images acquired by the left camera. Figure 5 shows the internal mechanism and the module inputs/outputs.

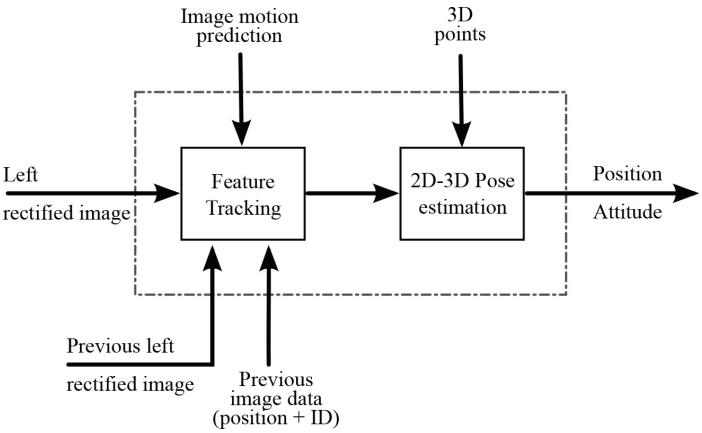


Fig. 5 - Egomotion Module. This module uses the left image only.

As in [23], the features are tracked through the video sequence acquired by the left camera using KLT [43]. In order to pre-emptively prune wrong temporal matchings, the fundamental matrix is robustly estimated using a Least Median of Squares scheme (LMedS) [37]; this operation is henceforth referred to as 'Fcheck'. Since this estimation can be unstable in the case of small relative motion, it is automatically disabled when the motion of features is less than a threshold.

We have also evaluated an active search process, where the search for temporal matches is guided by a prediction of the motion. Without inertial data, as for instance in the KITTI datasets, we use a simplistic motion prediction model based on constant linear and angular speeds. The motion estimated between the two previous frames is then used for motion prediction. If inertial data is available (as for instance in the MAV experiments), we only compensate for a global rotation of the image. Both methods help to reduce the search area for temporal matching.

Features	SHI-TOMASI [43]		FAST [36]	
Frame type	Keyframe	Standard	Keyframe	Standard
Average (ms)	74.2	12.4	56.1	12.4
Std (ms)	4.6	3.6	5.6	3.6
Min (ms)	62.3	5.8	40.8	5.6
Max (ms)	99.9	32.8	72.7	31.5

Table 1 - eVO processing time for one 672_480 stereo pair on a Core2Duo 1.86GHz. Measurements obtained by averaging over 10 Monte-Carlo runs.

From the temporal matchings provided by KLT, associations are established between 3D landmarks stored in the map and current image features. Given these 2D-3D matches, the camera pose (position and attitude) is robustly estimated within a RANSAC procedure [9]. In practice, we have implemented our own RANSAC framework with an

online adaptation of the number of iterations, as proposed by Peter Kovesi [21]. For each random sample, the pose is estimated with the Perspective-3-Point (P3P) algorithm [9, 45]. A bucketing strategy is used to enforce a minimal separation distance between the image features selected in the triplet given to the P3P algorithm. The P3P method often produces multiple solutions (up to 4): in such a case, all of the solutions are considered as random samples in the RANSAC voting process.

The RANSAC procedure returns an initial pose solution and a set of inliers. The pose is refined by minimizing the re-projection error of inlier matches. This non-linear least-squares optimization is solved using the motion-only optimization functions provided in the Lourakis SBA code [24].

Keyframe selection Module

As proposed in [28], a new keyframe is initialized as soon as the ratio between the number of successfully tracked features and the number of 3D points visible on the last keyframe drop under a threshold, denoted by t and set by default to t = 0:8. We discuss the algorithm sensitivity to parameter t in § "Tuning and advantage of the Keyframe Scheme".

Implementation and processing time

The implementation of eVO uses two well-known open-source tools: OpenCV and ROS (Robot Operating System, www.ros.org). Most of the low level image processing — image warping, tracking, feature extraction and template matching — is based on the OpenCV library. This library is optimized for the SIMD instruction set of the on-board CPU (Intel SSE). At the system scale, eVO works on ROS to deal with the physical sensors and share the trajectory estimation with the environment modeling part of 3DSCAN.

In table 1 we present the processing times measured on the embedded computer of our MAV: Ascending Technologies Mastermind Intel Core 2 Duo 1.86 GHz working on Ubuntu 12.04 32bits. Figure 6 shows how the computational time is distributed over the various components of the processing chain. These results demonstrate the great difference between keyframe and standard frame processing time, due to the fact that the 3D landmark generation is bypassed for the latter.

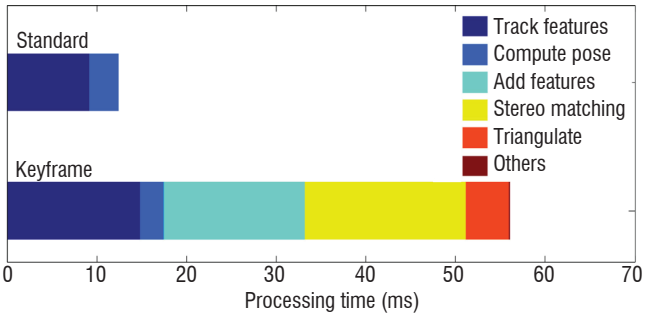


Fig. 6 - Relative computing time of eVO components. Measurements made by averaging over 10 Monte-Carlo runs, using a FAST feature detector [36].

As a consequence, the overall computational performance of eVO depends on the ratio between standard frame and keyframe numbers. In our implementation, this ratio is not fixed but varies with the success rate of the tracking, which itself depends on the vehicle dynamics. However, as discussed later in § "Tuning and advantage of the Keyframe Scheme", the best tuning of the keyframe selector leads to an average keyframe ratio of less than 30%. This means that the average computing time is less than 25 ms/frame. We can also note that the monocular egolocalization process (i.e., processing of a standard frame) could be run at a very high frame rate (up to 80hz) on one core of the embedded PC if the bandwidth of the USB-bus allowed it.

Evaluation

Datasets and performance measurements

Our system has been evaluated on multiple and varied data. Some of it was acquired using our own stereorig, either hand-held or carried by the MAV. No ground-truth state is available for this data, but we have followed loop trajectories in order to use the drift between the first and last frames as a performance indicator. An example of an outdoor experiment with a 60m-long loop is presented in figure 7, showing a drift of approximately 1% of the trajectory length.

We have also used the KITTI odometry dataset [12] composed of 22 video sequences acquired by a car equipped with several sensors (Velodyne R lidar, high resolution IMU and GPS-RTK, stereorig). The video collection covers a large range of environments (highway, suburban or town center) and trajectory profiles (loops, road sections) from one hundred meters to a few kilometers. The first half of the collection is supplied with ground-truth in order to adjust the algorithm parameters. The second half of the collection is used to benchmark algorithms.

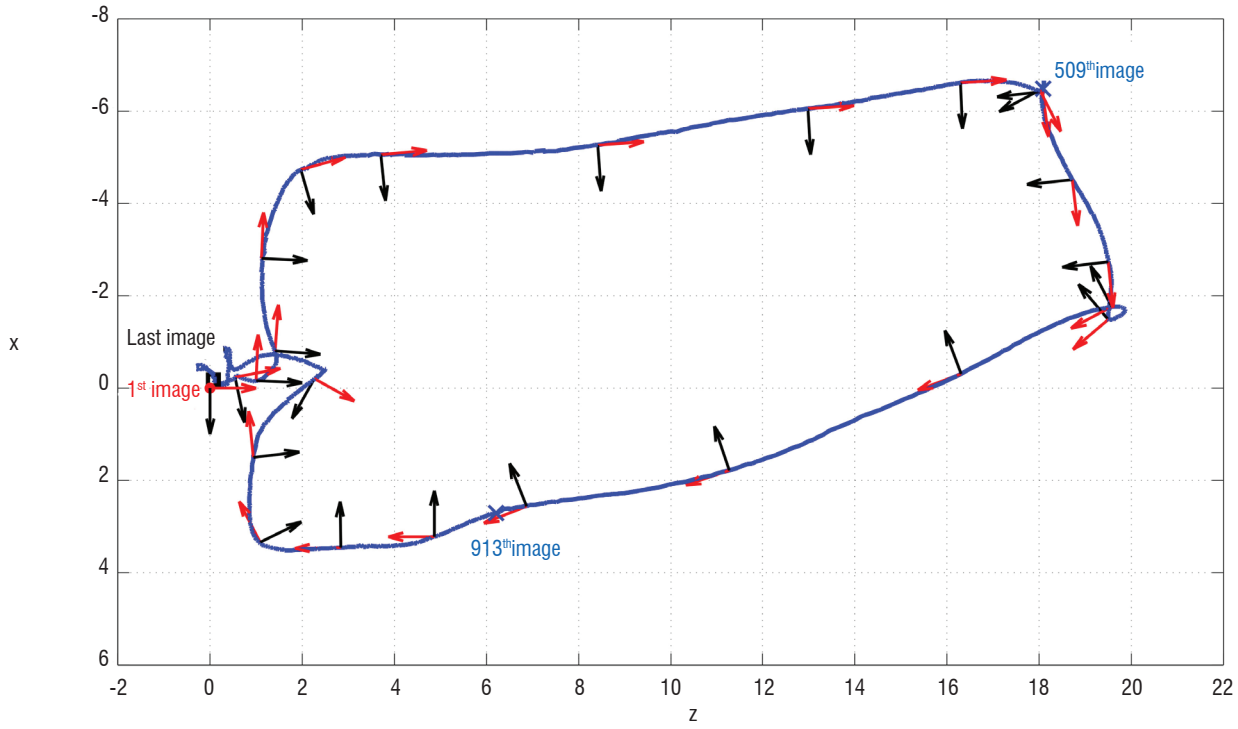
The KITTI Team also provides some performance metrics, together with a tool to compute them on the estimated trajectories. These metrics are: a translational drift expressed as a percentage of the total traveled distance and a rotational drift expressed in degrees by traveled meter. Scores are averaged over all possible sub-sequences of variable lengths, from 100 m to 800 m.

Since our system includes a random sampling scheme (RANSAC), we have performed Monte-Carlo simulations and measured statistical indicators (average performance, standard deviation, median, min-max values).

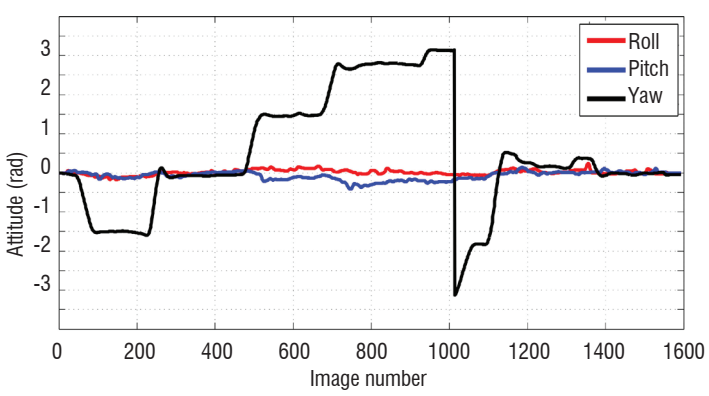
Figure 8 presents the estimated trajectories obtained after 25 Monte-Carlo runs on Sequence 08 of the KITTI odometry dataset. This trajectory in a suburban environment is 2 kilometers long and comprises many moving objects (vehicles, pedestrians and cyclists). On average, the estimated trajectory in the horizontal plane (XZ) is well estimated with a drift of only 4 meters. As usual in odometry, large angular errors occur at each important turn change. The estimation along the third dimension shows a bias at the beginning, which is probably due to an error in the ground truth and a significant variance at the end. We could constrain the eVO estimator to maintain a constant height above the ground, but we choose not to do so, since we intend to use the same algorithm for MAV data.

In the following section, we study the advantage of the keyframe scheme and discuss the tuning of the parameters of the Keyframe selection module, before presenting the global evaluation of eVO on the KITTI benchmark.

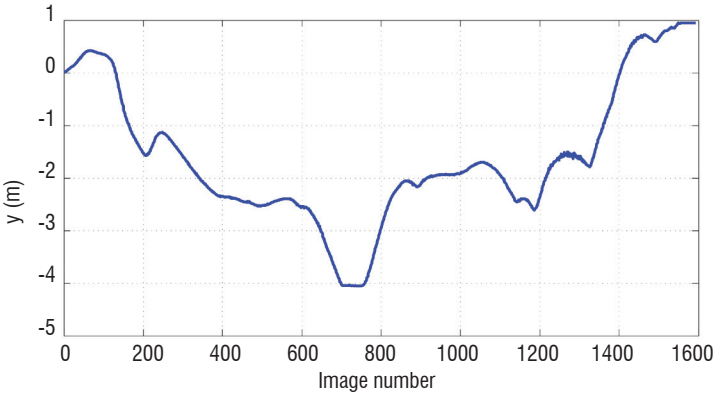
(a)



(b)



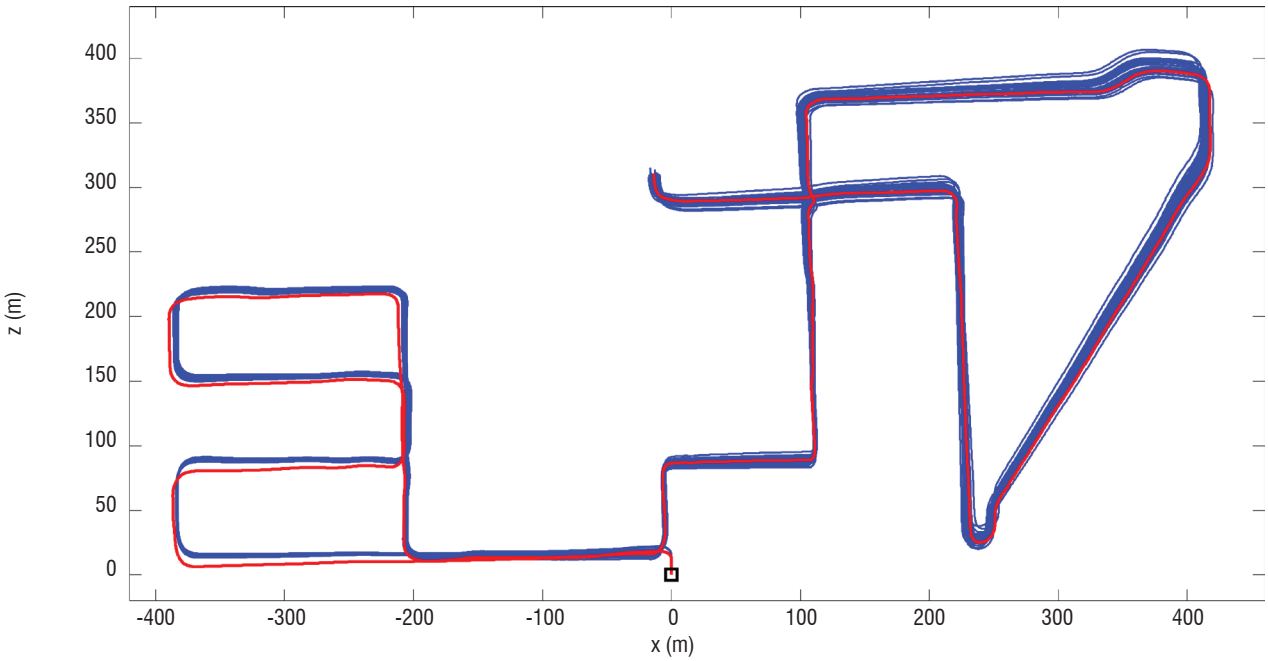
(c)



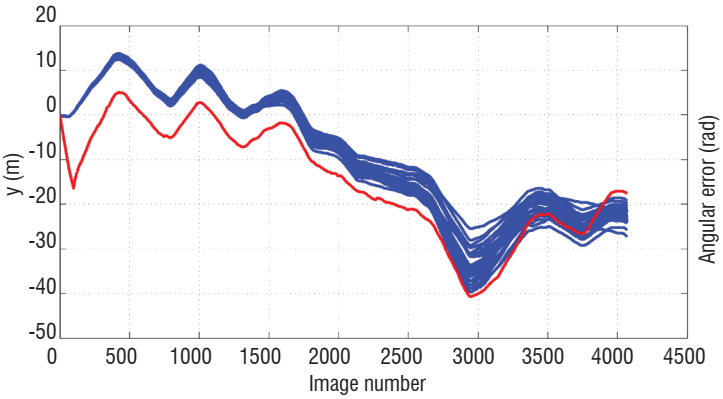
(d)

Fig. 7 - Trajectory estimated by eVO from the sequence 20120727.3 acquired during an outdoor flight of the MAV.
(a) 4 frames of the video sequence (the 1st, 509th, 913th and last image).
(b) Estimated trajectory. The red and black arrows indicate the attitude of our MAV (red: the front of the MAV, black: its right).
(c) Estimated attitude. The measurements provided by the embedded AHRS are not precise enough to serve as ground truth.
(d) Estimated height profile (the Y axis points downward). Note that the actual starting point is approximately 80 cm above the landing pad; hence, the total drift is less than 50 centimeters.

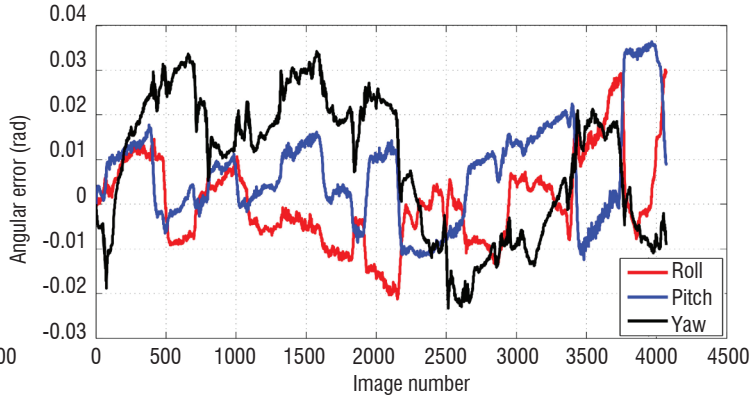
(a)



(b)



(c)



(d)

Fig. 8 - Result of eVO on the"08" sequence of the KITTI odometry dataset.
(a) Four images of the sequence.
(b) Trajectories on the XZ plane (red: ground truth, blue: estimated). Shown in red: the ground truth. Shown in blue: 25 trajectories obtained after as many Monte-Carlo runs.
(c) Average angular errors (in radians).
(d) Trajectories in the 3rd dimension.

Tuning and advantage of the Keyframe Scheme

Here, we discuss the advantages provided by the keyframe scheme in regard to the ego-localization performance, beyond its computational efficiency discussed previously. First, we compare our algorithm with a classical dead-reckoning visual odometer (DRVO) built with the same software components. Then, we investigate the influence of two parameters controlling the keyframe generation, the threshold t defined earlier and the activation of the Fcheck module. All of the results obtained on MAV-representative sequences are summarized in table 2, while results on the KITTI dataset are shown in table 3.

eVO vs. DRVO. As expected, the keyframe scheme allows the localization drift of eVO to be reduced compared to DRVO, even for settings that favor the generation of new keyframes. This is the case when choosing t = 1:0, which means generating a new keyframe as soon as one landmark is lost by the tracking process. The gain is particularly important with MAV data, as shown by comparing the total localization error presented in the two first rows of tables in table 2. On the KITTI dataset, the advantage is less important but significant, with a 10%-reduction of the drift, see table 3. This can be explained by the fact that, due to the car's speed, KITTI sequences exhibit larger inter-frame motion, reducing the interest of the keyframe scheme.

Parameters controlling the key-frame selection. We first study how the drift varies with respect to the ratio t, while the Fcheck module (which checks for the consistency of matches with the epipolar constraint) is activated. On the MAV sequences (table 2) the lower the parameter t, the lower the average localization error, but the higher the dispersion of the results. On the Kitti dataset (table 3) we observe that the choice t = 0.6 leads to larger errors. This is due to a lower frame rate and a higher vehicle speed, which means that the odometry uses tracked features that are farther from the camera and are badly localized. Finally, we choose t = 0.8 as a good trade-off.

The Fcheck procedure also has a significant influence on the number of keyframes. If this validation step is bypassed, the number of keyframes is reduced by half in all processed sequences (for the same ratio t). In the majority of our tests, this entails an error growth, particularly on the KITTI dataset, where the translational drift increases from 1.46 to 1.63. In practice, we choose to enable Fcheck by default.

Result on the KITTI Odometry Benchmark

Table 4 presents the average scores of eVO on the KITTI evaluation dataset, compared to other published methods. eVO obtains a very good performance, with an average translation drift of 1.76% and an angular error of 0.0036°/m. As on the date of its first submission to the IROS conference (March 2013), this performance allowed eVO to rank first. One year later, it is still 4th among methods that use only stereo data - note that methods using lidar data have been recently introduced in the KITTI table and have taken the two first positions.

Environment modeling

In the previous section, we have described how the stereo data is processed in order to estimate the pose of the system during its displacement. These estimated poses are used to fuse 'instantaneous' 3D data into a 3D model of the visited environment. 3D data can be depth measurements provided by an active RGBD sensor or stereo depthmaps. The latter are provided here by a fast and dense stereo-matching code on GPU, which is described in § "Dense stereo-matching". The chosen environment modeling framework is presented in § "Dense stereo-matching".

Dense stereo-matching

Classically, dense stereo-matching algorithms are based on systematic exploration in the disparity space, to evaluate radiometric similarities between pixels of the two images. Here, dense disparity maps are computed using a dense Lucas-Kanade (LK) algorithm [26] derived from an original optical flow algorithm eFolki, described in [35]. The resulting code is remarkably fast on a massively parallel architecture such as GPU. In the following sections, we recall the equations of the algorithm, first published in [4], discuss its implementation on GPU, describe some adaptations made to increase the robustness of the estimated disparity on real stereo images, and finally present a local indicator of the consistency.

All evaluations are performed on data provided in the KITTI benchmark [12].

Efficient dense matching by the LK algorithm

The basic problem of the dense LK algorithm is to register local windows centered around each image pixel $\mathbf{x}$ by minimizing a SSD (Sum of Squared Difference) criterion over a 2D motion vector $\mathbf{u}(\mathbf{x})$:

$$\sum_{\mathbf{x}'} w(\mathbf{x}'-\mathbf{x})\left(I_1(\mathbf{x}')-I_2(\mathbf{x}'+\mathbf{u}(\mathbf{x}))\right)^2 \quad (1)$$

where w is a separable weighting function, uniform or Gaussian, of limited support $\mathcal{W}$, typically a square window parameterized by its radius r. Since we consider here dense matching of rectified stereo data, where epipolar lines are aligned with the horizontal axis of the images, the motion vector is reduced to a scalar disparity: $\mathbf{u}(\mathbf{x}) = [d(\mathbf{x});0]$.

The minimization of criterion 1 is done by an iterative Gauss-Newton coarse-to-fine pyramidal strategy, as in classical implementations of LK. However, using the first order expansion described in [25], an iteration can be completed with only one image interpolation per pixel, while the well-known PyramLK algorithm [5] requires several image interpolations per pixel. An iteration of this convergent dense matching strategy, denoted eFolki, consists in:

$$\begin{aligned} I_2^{(k)}(\mathbf{x}) &= I_2\left(\mathbf{x}+\begin{bmatrix} d^{(k)}(\mathbf{x}) \\ 0 \end{bmatrix}\right), \forall \mathbf{x}\,(\text{interpolation}) \\ \delta I^{(k)} &= I_2^{(k)} - I_1 - \nabla_x I_1 \otimes d^{(k)} \\ c^{(k)} &= w * \left(\nabla_x I_1 \otimes \delta I^{(k)}\right) \\ d^{(k+1)} &= c^{(k)} \oslash \left(\nabla_x I_1^2\right) \end{aligned}$$

where $d^{(k)}$ is the previous disparity guess, ∇_x is the image gradient operator in the x-direction and operator $\otimes$ (respectively $\oslash$) is the component-wise multiplication (respectively division). One can readily observe that eFolki is ideally suited for GPU implementation, because each iteration requires only very regular operations on the images: separable convolutions, pixelwise operations and image interpo-

Sequence	Method	Error X (m)	Error Y (m)	Error Z (m)	Keyframe ratio
Name: **2010727.2** Image Number: 2039 Trajectory length: 150 m hand-held	DRVO	-2.4 ±0.04	-0.5 ±0.05	2.0 ±0.04	100%
	EVO τ=1.0	-1.28 ±0.08	-0.17 ±0.11	1.20 ±0.08	96%
	EVO τ=0.8	-0.95 ±0.18	-0.30 ±0.11	1.01 ±0.14	36%
	EVO τ=0.6	-0.73 ±0.21	-0.06 ±0.25	0.83 ±0.18	19%
Name: **20120724.3** Image Number: 1675 Trajectory length: 70 m Acquired by MAV	DRVO	-6.8 ±0.4	2.05 ±0.2	4.7 ±0.4	100%
	EVO τ=1.0	-0.8 ±0.2	0.23 ±0.24	0.45 ±0.12	91%
	EVO τ=0.8	-0.61 ±0.34	0.23 ±0.35	0.4 ±0.17	27%
	EVO τ=0.6	-0.33 ±0.60	0.35 ±0.47	0.21 ±0.3	13%

Table 2 - Localization error at the end of two closed trajectories acquired with the stereorig of our MAV and ratio of keyframes for different algorithms or algorithm settings.

Sequence	Method	Translational drift (%)	Rotational drift (deg/m)	Keyframe ratio
Name: **Kitti Benchmark** Training SDataset Acquired by a car	DRVO	1.56 ±0.007	0.00166 ±0.00008	100%
	EVO τ=1.0	1.45 ±0.015	0.00145 ±0.0001	99.8%
	EVO τ=0.8	1.46 ±0.014	0.00144 ±0.0002	79.6%
	EVO τ=0.6	1.53 ±0.017	0.00151 ±0.0002	37.8%

Table 3 - Angular and translation drift indicators measured on the KITTI Odometry dataset for various algorithms or algorithm settings.

Rank	Method	Setting	Translation	Rotation	Runtime	Environment
3	MFI	st	1.30%	0.0030 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (C/C++)
4	VoBa	st	1.46%	0.0030 [deg/m]	0.1 s	1 core @ 2.0 Ghz (C/C++)
5	SSLAM	st	1.57%	0.0044 [deg/m]	0.5 s	8 cores @ 3.5 Ghz (C/C++)
6	**eVO**	**st**	**1.76%**	**0.0036 [deg/m]**	**0.05 s**	**2 cores @ 2.0 Ghz (C/C++)**
7	SOVI	st	1.80%	0.0079 [deg/m]	0.1 s	4 cores @ 2.5 Ghz (Matlab)
8	D6DVO	st	2.04%	0.0051 [deg/m]	0.03 s	1 core @ 2.5 Ghz (C/C++)
9	MICP_VO	st	2.13%	0.0065 [deg/m]	0.01 s	1 core @ 2.5 Ghz (C++)
10	SSLAM-HR	st	2.14%	0.0059 [deg/m]	0.5 s	8 cores @ 3.5 Ghz (C/C++)
11	VIS02-S	st	2.44%	0.0114 [deg/m]	0.05 s	1 core @ 2.5 Ghz (C/C++)
12	GT_VO3pt	st	2.54%	0.0078 [deg/m]	1.26 s	1 core @ 2.5 Ghz (C/C++)

Table 4 - Kitti Odometry benchmark chart at 2014-02-03. Please note that only stereo-based algorithms are presented; however, the ranks are those of the published Kitti, where lidar-based methods occupy the two first places. The eVO result is obtained by tracking at the most 500 Shi-Tomasi interest points extracted from 20 x 8 regions. The tracking is initialized with the previous motion. The ransac threshold is set to 1.0, while the parameter τ is equal to 0.8.

Method	out noc	out all	avg noc	avg all	density	GT650M
SSD	32.9%	34.2%	7.2 pix	7.9 pix	100%	27 ms
SSS+Rank+WRA	13.7%	15.4%	2.9 pix	3.3 pix	100%	108 ms

Table 5 - Evaluations of the different variants of the eFolki dense matching technique on Kitti stereo training databases.
'Rank' denotes Rank-n pre-filtering and 'WRA' means Window Radius Adaptation; see text.
Columns 4 and 5 give the percentage of pixels with an error greater than 3 pixels.
The average computing time for a mid-range GPU is shown in the last column.

lations. In 2009, we demonstrated a CUDA implementation of this algorithm able to compute a dense OF estimation on a full HD video (1920 x 1080) in less than 20 ms on a 285 GTX board [34].

Increasing robustness

As is well known, SSD is not a robust criterion and using the previous algorithm on a real-world image leads to inhomogeneous results, as illustrated in the second line of figure 9. However, following the work of Sun et al. for Horn-Schunk OF methods [41], we have found that simple modifications of the algorithm, essentially pre-filtering and adaptation of the coarse-to-fine strategy, can greatly improve the result.

The first problem is that the motion estimation greatly depends on the local image texture and fails in the event of illumination changes. To correct this, we apply a Rank -n transform [46] to the images before SSD minimization. Each pixel $\mathbf{x}$ is replaced by the number of neighboring pixels with an intensity lower than $I_{(\mathbf{x})}$. This transform is fast and has only one parameter: the radius n of the neighborhood. Transformed images have a compressed intensity range, which increases the robustness and homogeneity of the eFolki result.

The second issue is related to convergence: ensuring the convergence of the LK iteration often requires large windows to be chosen, at the cost of a lower resolution of the estimated flow. The solution proposed here is to vary the radius of the window during the iterations: we denote this strategy 'WRA' for Window Radius Adaptation. In practice, our solution consists in adding a loop at each pyramid level and progressively reducing the window size.

The effects of these modifications are illustrated in figure 9 on an image of the Kitti Stereo Dataset. The modifications lead to an estimate (third line of the figure), which appears significantly more accurate and reliable than the previous one. Quantitative comparative measures are given in table 5. The proposed modification leads to a reduction by a factor of 2 in the number of erroneous pixels and the average disparity errors. According to the current Kitti stereo benchmark, our algorithm ranks only at around the 40th position; however, it is among the fastest methods. In addition, its limited accuracy appears sufficient for our 3D modeling task.

A local indicator of reliability

An important issue when using dense stereo-matching for environment modeling and autonomous navigation is to be able to assess locally the reliability (and the accuracy) of the estimated disparity. In particular, it is important to detect regions where disparity estimation has failed, so as to avoid dangerous movements toward undetected obstacles or to plan a revisit to fill up the map.

We propose to compare depth values estimated respectively from the forward disparity $d_{1\leftarrow 2}$ computed using criterion (1) and the backward disparity $d_{1\leftarrow 1}$ computed by exchanging I_1 and I_2 in (1). More precisely, for each pixel $\mathbf{x}$ in I_1, we compute the error ε_Z defined as:

$$\varepsilon_Z(\mathbf{x}) = \frac{fb\left(d_{1\leftarrow 2}(\mathbf{x}) - d_{2\leftarrow 1}\left(\mathbf{x} + d_{1\leftarrow 2}(\mathbf{x})\right)\right)}{d_{1\leftarrow 2}(\mathbf{x})\, d_{2\leftarrow 1}\left(\mathbf{x} + d_{1\leftarrow 2}(\mathbf{x})\right)}$$

where f is the focal distance in pixels and b is the stereo baseline in meters. The threshold (in meters) is typically chosen equal to the voxel resolution of the 3D model.

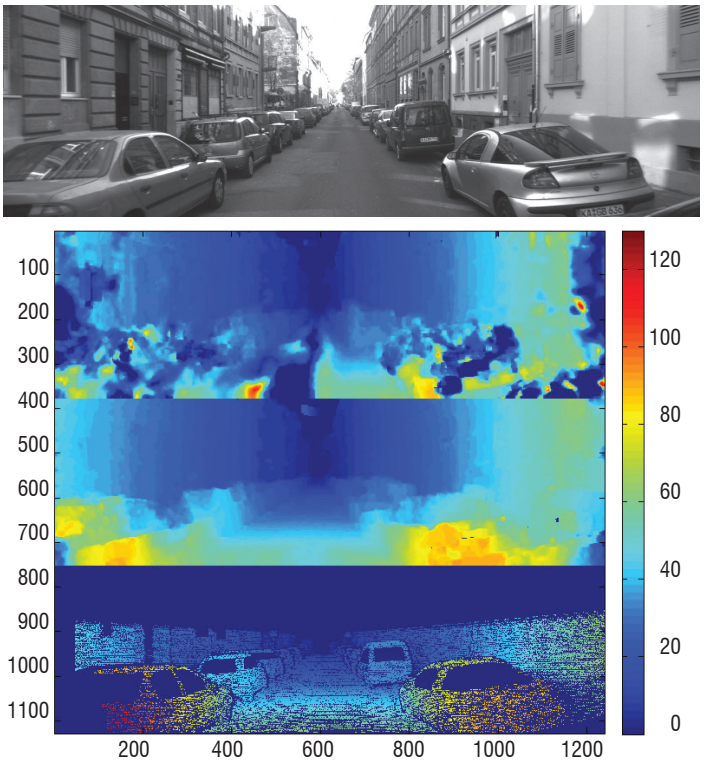


Fig. 9 - Stereovision results on the 172th image of the KITTI Stereo benchmark. From top to bottom: the left image of the stereo pair; disparity map estimated with SSD minimization; disparity map estimated using the modifications (rank transform and window radius adaptation); Ground Truth acquired by a Velodyne sensor.

Building the 3D model of the environment

Octomap model

As discussed in the introduction, a volumetric representation of the 3D environment can be obtained by subdividing the visited space with a regular 3D grid. Each elementary part is called a voxel and stores, for instance, the occupancy probability, as proposed in [27]. Occupancy probabilities are updated by ray-tracing techniques. For a sensor (stereo or active RGBD) delivering a depthmap in some known image geometry, each pixel of the depthmap defines a ray and a 3D point located on this ray approximately at the depth stored in the pixel. All of the voxels that belong to the segment linking the sensor pixel and the 3D point are processed, i.e., their probability of occupancy is updated according to some model of the 3D sensor accuracy.

In the probabilistic 3D mapping framework Octomap of [15] a multi-resolution grid based on an octree data structure replaces the standard regular 3D grid. This solution permits an automatic adaptation of the map resolution to the local 3D geometry, with the advantage of smaller memory requirement and faster data access. Moreover, the octree representation can be defined without a precise prior knowledge of the size of the visited environment. Indeed, when room is needed for new areas, the octree is expanded by a new level. In practice, Octomap is limited to 16 levels, hence to 215 voxels. Note that the Octomap framework provides labels to denote voxels that are in free space and also voxels that have not been explored yet, see figure 10.

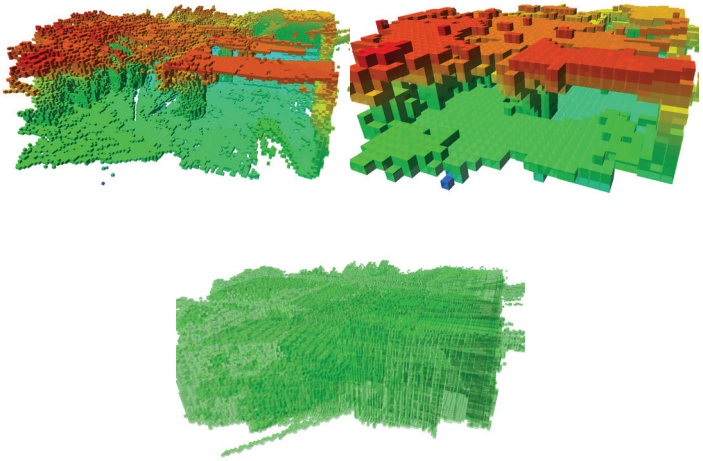


Fig. 10 - Octomap model of a parking area in the basement at Onera.
Top left: 3D occupancy model at the finer scale. Voxels with a probability higher than 80% are colored with a colormap related to their height above the reference plane, which is the horizontal plane at the starting position of the MAV.
Top right: rough 3D model, which can be readily obtained from the octree representation.
Bottom line: freespace voxels (fine scale) colored in transparent green.

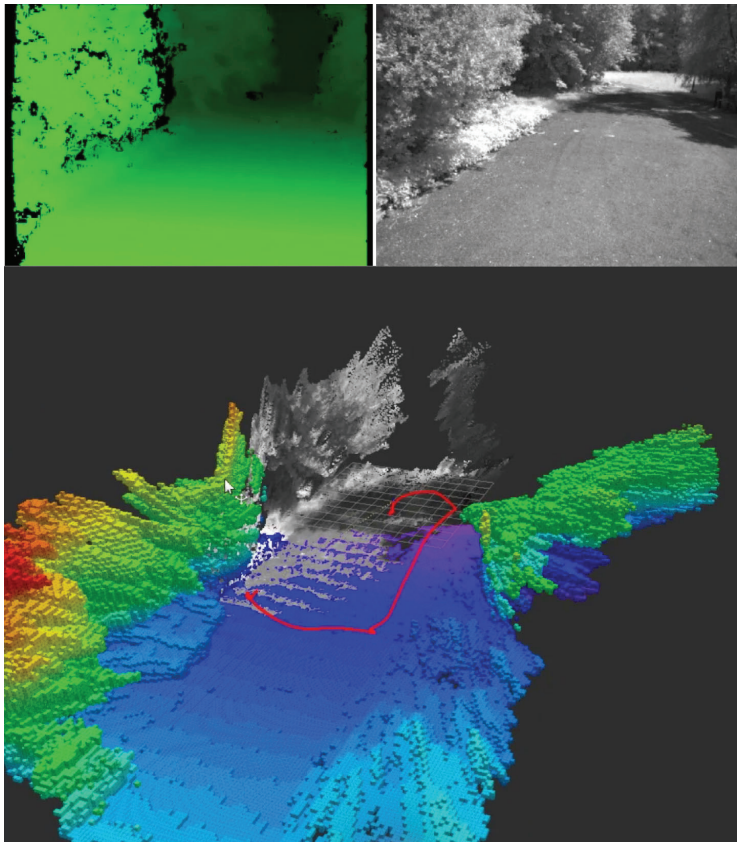


Fig. 11 - Illustration of online 3D scene modeling, outdoor flight.
Voxel resolution: 20 cm.
Top: estimated stereo depthmap and left image recorded by the stereorig.
Bottom: current estimated trajectory of the MAV (red curve), re-projected 3D map (graylevels) and current 3D model (voxel in colors)

3DSCAN results

Outdoor MAV flight

Figure 11 presents a 3D reconstruction obtained on-line from stereo data during an outdoor flight of the MAV (sequence 20120727.3). The estimated trajectory, shown in red in the 3D representation of figure 11, is presented in more detail in figure 7: it is a loop approximately 60 m long. We present, not the final reconstructed model, but images extracted from a screenshot of the ground station during the flight. The current frame taken by the left camera and the corresponding stereo depthmap are presented in the top part of the figure. The forward/backward consistency check described previously has been used to eliminate areas near the edges of the trees that cannot be seen in the two images. The instantaneous 3D map is re-projected in the 3D model with graylevel texture from the current left image. The occupancy model represents obstacles previously detected during the flight. The voxel size is 20 x 20 x 20 cm and the color is related to the height above the initial horizontal plane. Since this reference plane was not aligned with the ground, the color level of the reconstructed ground is variable. Note that the shapes of the scene 3D objects are elongated along the view axis of the onboard stereorig, because of the limited accuracy of 3D triangulation. However, this model provides a good localization of the obstacles that are closest to the MAV during its flight, which is the main objective for this exploration mission. A refined model could be built by getting around the 3D structures, as illustrated in Fig. 13 below.

Indoor MAV flight

Figure 12 presents a 3D reconstruction obtained on-line from stereo data during an indoor flight of the MAV in a parking area located in the basement of a building at Onera. The complete model of the visited part of the parking area was presented in figure10. The estimated trajectory, shown in red in the 3D representation of figure 12 is again a loop approximately 30 m long. As before, we present the left image, the associated depthmap and the current 3D model during the final part of the flight. Details such as the obstacle on the ground and the pipes on the left wall are clearly visible in the reconstructed model. Figure 13 shows how the post in the middle of the parking area is refined as the MAV flies around it: it is at first reconstructed with a large elongation in the viewing direction (left image) then, as the MAV gets around, its shape is refined and fits its actual support more precisely (right image).

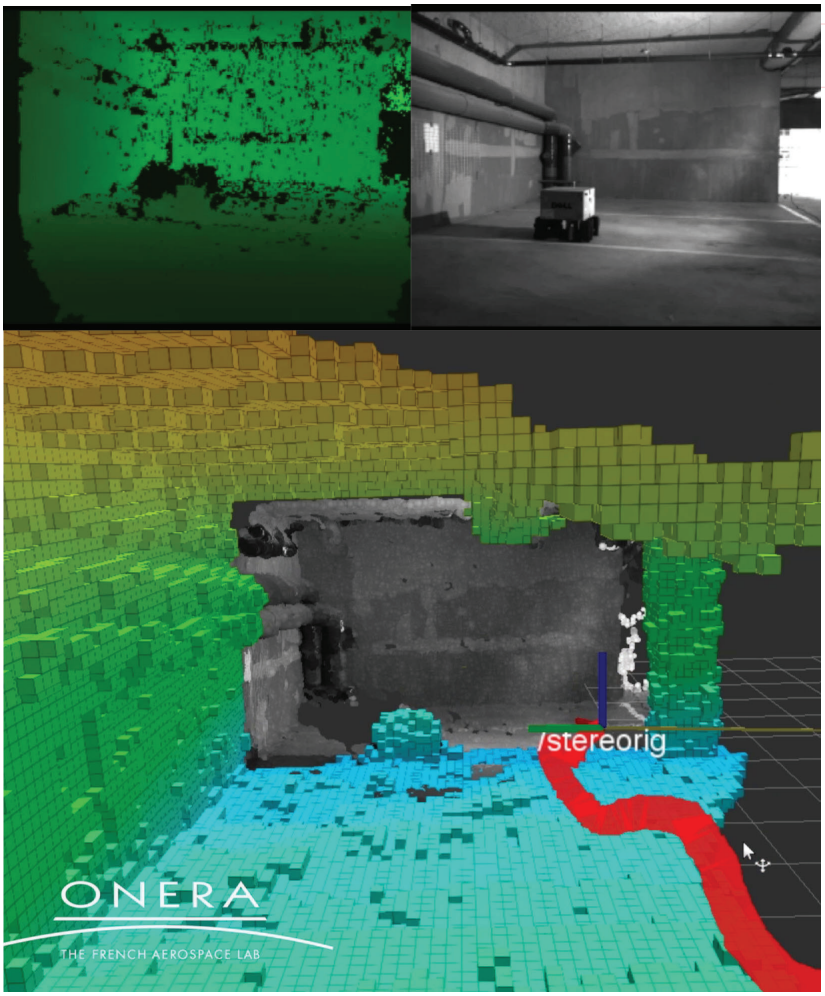


Fig. 12 - Illustration of online 3D scene model, indoor flight. Voxel resolution: 20 cm.
Top: estimated stereo depthmap and left image recorded by the stereorig.
Bottom: current estimated trajectory of the MAV (red curve), re-projected 3D map (graylevels) and current 3D model (voxel in colors)

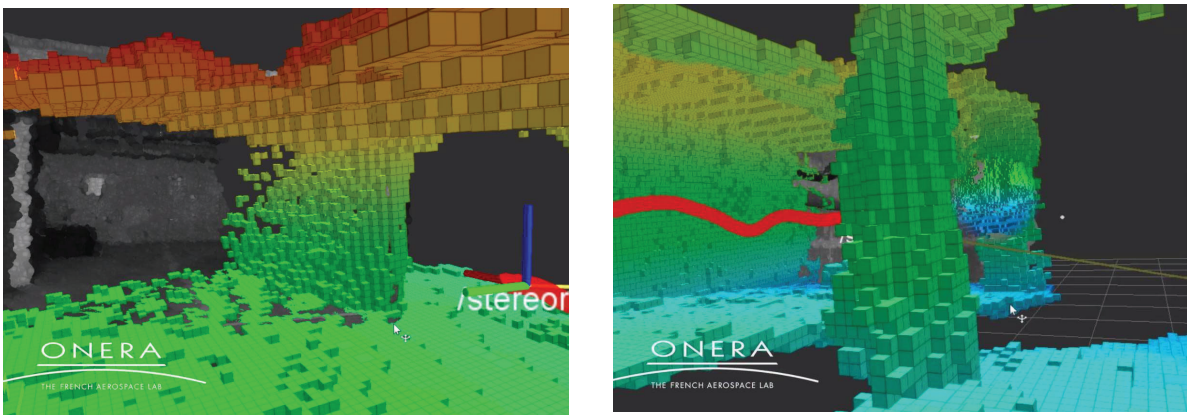


Fig. 13 - Refining the shape of 3D objects by flying around them, see text.

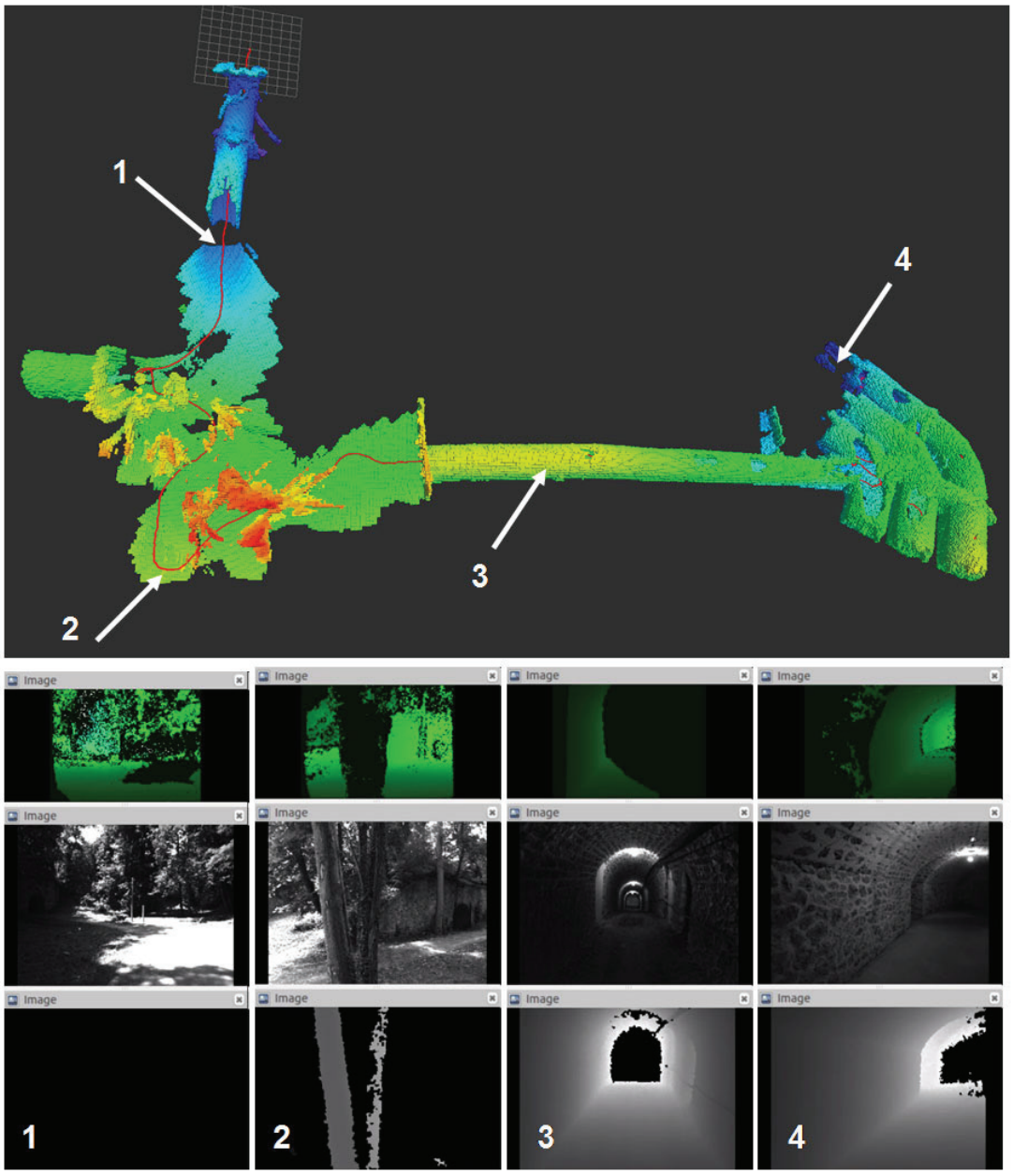


Fig. 14 - 3D Model of the “Caponière”, a historical underground location at Onera.
The thick red line denotes the MAV trajectory. The screenshots on the bottom row present the available data during the experiment, at 4 instants indicated by numbers on the trajectory. For each instant, we present the stereo depthmap (top image in green levels), the left image (B/W image in the middle) and the Xtion depthmap (bottom image in graylevels).

Indoor/outdoor trajectory

Figure 14 shows a large and complex reconstructed model of the "Caponière" area at Onera's Center in Palaiseau (France). The MAV, which is hand-held in this experiment for security reasons, travels along a 200m trajectory passing through a tunnel and a wooded area before going inside a long gallery leading to a centenary underground gunpowder warehouse. The trajectory is computed online and onboard at a 20Hz rate by eVO using stereo data. Note that the estimated trajectory is smooth, despite the transitions between indoor and outdoor areas. The model aggregates stereo or 3D data from the embedded Asus Xtion RGBD camera. The default device is the Xtion, which is, when available, usually more accurate than stereo depths. However, in many situations, especially outdoors, the depthmap delivered by the Xtion is incomplete, or even empty. When less than 80% of the pixels are measured by the Xtion, we use the stereo depthmap. Examples of data delivered by stereo and Xtion, and the switch between them, are presented in the lower part of figure 14. Essentially, stereo is used outside and Xtion inside the tunnels. Note however that, in some situations (see times 2 and 4 in figure 14), both sensors deliver useful information. Designing better fusion rules for both sensors during the modeling and odometry processes is the subject of future studies.

Conclusion

In this paper, 3DSCAN, an efficient framework for egolocalization and 3D modeling of the environment from stereo and RGBD data, has been presented. First, we have demonstrated state estimation from stereo data at 20Hz using one core of the Core2Duo 1.86 GHz onboard the MAV. Higher rates, typically 50Hz, could easily be obtained using multi-threading and with a more recent computer. This visual odometer, denoted eVO, has been evaluated on publicly available stereo data with very good results. Second, a non-supervised 3D modeling software application has been developed using the Octomap framework. It uses stereo data, processed by our fast dense matching code eFolki on GPU and 3D data obtained from an Xtion active RGBD sensor. On our ground station, a light laptop with a mid-range GT 650M GPU, the depthmap computation (limited to the 1-8 m range) and integration into the 3D model runs in 1 to 2 s, which is sufficient for the dynamics of our quadrotor. In the Kitti setup, the 3D data range is greater (up to 30 m), the vehicle is much faster and explores larger areas; hence, the 3D modeling requires a powerful workstation to run with the same rates.

Our current work is aimed at using 3DSCAN for autonomous navigation of MAV in unknown environments, with control and planning issues. Some improvements and adaptations are necessary to improve its robustness and to embed the system on the MAV (using a novel embedded CPU board). We intend to add a multi-view refinement step in eVO for the fusion of eVO with other sensors available onboard (IMU and GPS) to improve the quality, rate and reliability of state estimation. In terms of perception, in the absence of GPU onboard, eFolki will be replaced by an efficient dense stereo-matching algorithm, such as SGBM [13]. We are also working on long-term modeling, including loop closure detection and the associated correction of the 3D model. Finally, we also intend to make use of recent advances in computational photography to obtain 3D data with more compact and lightweight co-designed sensors, such as the 3D chromatic depth-from-defocus camera presented in [44] ■

Acronyms

CPU	(Central Processing Unit)
eVO	(Efficient Visual Odometer)
GPS	(Global Positioning System)
GPS-RTK	(GPS Real-Time Kinematic)
GPU	(Graphics Processing Unit)
IMU	(Inertial Measurement Unit)
KITTI	(Karlsruhe Institute of Technology and Toyota Technological Institute)
KLT	(Kanade-Lucas-Tomasi (feature tracker))
MAV	(Miniature Aerial Vehicle)
RANSAC	(Random Sampling Consensus)
RGBD	(Red, Green, Blue + Depth (4-channel cameras))
ROS	(Robotic Operating System)
SGBM	(Semi-Global Block Matching)
SIMD	(Simple Instruction Multiple Data)
SSD	(Sum of Squared Differences)
UAV	(Unmanned Aerial Vehicle)
ZNCC	(Zero-mean Normalized Cross-Correlation)

References

[1] F. ANDERT, F. ADOLF - *Online World Modeling and Path Planning for an Unmanned Helicopter.* Autonomous Robots, 27:147–164, 2009.

[2] P.F. ALCANTARILLA, L.M. BERGASA, F. DELLAERT - *Visual Odometry Priors for Robust Ekf-slam*. IEEE International Conference on Robotics and Automation (ICRA), pages 3501–3506, 2010.

[3] M. ACHTELIK, A. BACHRACH, R. HE, S. PRENTICE, N. ROY - *Stereo Vision and Laser Odometry for Autonomous Helicopters in gps-denied Indoor Environments*. Proceedings of the SPIE Unmanned Systems Technology XI, volume 7332, Orlando, Florida, 2009.

[4] G. LE BESNERAIS, F. CHAMPAGNAT - *Dense Optical Flow Estimation by Iterative Local Window Registration.* IEEE International Conference on Image Processing (ICIP), pages 137–140, Genova, Italy, September 2005.

[5] J.Y. BOUGUET - *Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker - Description of the Algorithm*. Technical report, Technical report. Intel Corporation, 2001.

[6] A. I COMPORT, E. MALIS, P. RIVES - *Accurate Quadrifocal Tracking for Robust 3d Visual Odometry*. IEEE International Conference on Robotics and Automation (ICRA), pages 40–45, Roma, Italy, April 2007. IEEE.

[7] A.J. DAVISON, I.D. REID, N.D. MOLTON, O. STASSE - *Monoslam: Real-time Single Camera Slam*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 29(6):1052–1067, 2007.

[8] A. ELFES - *Using Occupancy Grids for Mobile Robot Perception and Navigation*. Computer, 22(6):46–57, June 1989.

[9] M.A. FISCHLER, R.C. BOLLES - *Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Communications of the ACM, 24(6):381–395, 1981.

[10] F. FRAUNDORFER, L. HENG, D. HONEGGER, G. HEE LEE, L. MEIER, P. TANSKANEN, M. POLLEFEYS - *Vision-based Autonomous Mapping and Exploration Using a Quadrotor Mav*. IEEE/RSJ International Conference on Intelligent robots and systems (IROS), pages 4557–4564, Algarve, Portugal, October 2012.

[11] F. FRAUNDORFER, D. SCARAMUZZA - *Visual Odometry: Part ii - Matching, Robustness, and Applications*. IEEE Robotics and Automation Magazine, 19(2):78–90, June 2012.

[12] A. GEIGER, P. LENZ, R. URTASUN - *Are we Ready for Autonomous Driving? the Kitti Vision Benchmark Suite*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3354 – 3361, Providence, RI (USA), June 2012.

[13] H. HIRSCHMÜLLER - *Stereo Processing by Semiglobal Matching and Mutual Information.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(2):328–341, 2008.

[14] A. HOWARD - *Real-time Stereo Visual Odometry for Autonomous Ground Vehicles*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3946– 3952, 2008.

[15] A. HORNUNG, K. M.WURM, M. BENNEWITZ, C. STACHNISS, W. BURGARD - *OctoMap: an Efficient Probabilistic 3D Mapping Framework Based on Octrees*. Autonomous Robots, 34(3), 2013.

[16] W. MORRIS, I. DRYANOVSKI, X. JIZHONG - *Multi-volume Occupancy Grids: an Efficient Probabilistic 3d Mapping Model for Micro Aerial Vehicles.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1553–1559, Taipei (Taiwan), October 2010.

[17] K. KONOLIGE, M. AGRAWAL, J. SOLA - *Large-scale Visual Odometry for Rough Terrain.* 13th International Symposium of Robotics Research, Hiroshima, Japan, November 2007.

[18] B. KITT, A. GEIGER, H. LATEGAHN - *Visual Odometry Based on Stereo Image Sequences with Ransac-based Outlier Rejection Scheme.* IEEE Intelligent Vehicles Symposium (IV), pages 486–492, San Diego, CA (USA), June 2010. IEEE.

[19] G. KLEIN, D. MURRAY - *Parallel Tracking and Mapping for Small ar Workspaces*. International Symposium on Mixed and Augmented Reality, Nara, Japan, November 2007.

[20] M. KAESS, K. NI, F. DELLAERT - *Flow Separation for Fast and Robust Stereo Odometry*. IEEE International Conference on Robotics and Automation (ICRA), pages 3539–3544, Kobe, Japan, May 2009.

[21] P. D. KOVESI - *MATLAB and Octave Functions for Computer Vision and Image Processing*. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/_pk/research/matlabfns/>.

[22] J. KELLY, G. S. SUKHATME - *An Experimental Study of Aerial Stereo Visual Odometry.* IFAC Symposium on Intelligent autonomous vehicles, 2007.

[23] J. KELLY, S. SARIPALLI, G. SUKHATME - *Combined Visual and Inertial Navigation for an Unmanned Aerial Vehicle*. Christian Laugier and Roland Siegwart, editors, Field and Service Robotics, volume 42 of Springer Tracts in Advanced Robotics, pages 255–264. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-75404-6 24.

[24] M.I. A. LOURAKIS, A.A. ARGYROS - *SBA: a Software Package for Generic Sparse Bundle Adjustment*. ACM Trans. Math. Software, 36(1):1–30, 2009.

[25] G. LE BESNERAIS, F. CHAMPAGNAT - *Dense Optical Flow by Iterative Local Window Registration*. IEEE International Conference on Image Processing 2005, pages I–137. IEEE, 2005.

[26] B. D. LUCAS, TAKEO KANADE - *An Iterative Image Registration Technique with an Application to Stereo Vision*. IJCAI, volume 81, pages 674–679, 1981.

[27] H.P. MORAVEC, A. ELFES - *High Resolution Maps from Wide Angle Sonar*. International Conference on Robotics and Automation (ICRA), pages 116–121, St Louis, MI (USA), March 1985.

[28] E. MOURAGNON, M. LHUILLIER, M. DHOME, F. DEKEYSER, P. SAYD - *Real Time Localization and 3d Reconstruction*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 363–370, 2006.

[29] A. MALLET, S. LACROIX, L. GALLO - *Position Estimation in Outdoor Environments Using Pixel Tracking and Stereovision*. IEEE ICRA, 2000.

[30] H.P. MORAVEC - *Robot Spatial Perception by Stereoscopic Vision and 3d Evidence Grids*. Technical Report CMU-RI-TR-96-34, Carnegie Mellon University, September 1996.

[31] V. VITTORI, M. SANFOURCHE, G. LE BESNERAIS - *evo: a Realtime Embedded Stereo Odometry for Mav Applications*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2107–2114, Tokyo, Japan, November 2013.

[32] C. MEI, G. SIBLEY, M. CUMMINS, P. NEWMAN, I. REID - *Rslam: a System for Large-scale Mapping in Constant-time Using Stereo*. International Journal of Computer Vision, pages 1–17, 2010. Special issue of BMVC.

[33] D. NISTER, O. NARODITSKY, J. BERGEN - *Visual Odometry for Ground Vehicles Applications*. Journal of Field Robotics, 23(1):3–20, 2006.

[34] A. PLYER, G. LE BESNERAIS, F. CHAMPAGNAT - *Folki-gpu: a Powerful and Versatile Cuda Code for Real-time Optical Flow Computation.* GPU Technology Conference, San Jose, CA (USA), October 2009.

[35] A. PLYER, G. LE BESNERAIS, F. CHAMPAGNAT - *Real-time Lucas-kanade Optical Flow Estimation for Real-world Applications*. 2014.

[36] E. ROSTEN, T. DRUMMOND - *Machine Learning for High-speed Corner Detection*. European Conference on Computer Vision, volume 1, pages 430–443, May 2006.

[37] P. J ROUSSEEUW - *Least Median of Squares Regression*. Journal of the American Statistical Association, 79(388):871–880, 1984.

[38] C. HUERZELER S. WEISS L. KNEIP R. VOIGT, J. NIKOLIC, R. SIEGWART - *Robust Embedded Egomotion Estimation*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2694–2699, San Francisco, Ca (USA), September 2011.

[39] D. DUBE S. A. SCHERER, A. ZELL - *Using Depth in Visual Simultaneous Localisation and Mapping.* IEEE International Conference on Robotics and Automation, St. Paul, Minnesota, USA, May 2012.

[40] D. SCARAMUZZA, F. FRAUNDORFER. *Visual Odometry: Part i - the First 30 Years and Fundamentals*. IEEE Robotics and Automation Magazine, 18(4):80–92, December 2011.

[41] D. SUN, S. ROTH, M. J. BLACK - *Secrets of Optical Flow Estimation and their Principles*. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, pages 2432–2439, 2010.

[42] S. SCHERER, S. SINGH, L. J. CHAMBERLAIN, M. ELGERSMA - *Flying Fast and Low Among Obstacles: Methodology and Experiments*. The International Journal of Robotics Research, 27(5):549–574, May 2008.

[43] J. SHI, C. TOMASI - *Good Features to Track*. 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 593 – 600, 1994.

[44] P. TROUVÉ, F. CHAMPAGNAT, G. LE BESNERAIS, J. SABATER, T. AVIGNON, J. IDIER - *Passive Depth Estimation Using Chromatic Aberration and a Depth from Defocus Approach*. Applied optics, 52(29):7152–7164, 2013.

[45] S. UMEYAMA - *Least-Squares Estimation of Transformation Parameters Between Two Point Patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 13(4):376–380, April 1991.

[46] R. ZABIH, J. WOODFILL - *Non-Parametric Local Transforms for Computing Visual Correspondence*. Computer Vision—ECCV'94, pages 151–158, 1994.

[47] M. MEILLAND, A.I. COMPORT, P. RIVES - *Dense Visual Mapping of Large Scale Environments for Real-Time Localisation*. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francsiso, 2011.

AUTHORS

Martial Sanfourche graduated from Universite de Cergy-Pontoise in computer Sciences (2001) then received the Ph.D. degree in image and signal processing from the Universite de Cergy-Pontoise in 2005. After a postdoctoral position at CNRS-LAAS, he joined Onera/DTIM in 2007 where is now a research engineer in computer vision. His current research interest include online and offline visual localization and mapping for robotic systems.

Aurelien Plyer graduated from Universite Pierre et Marie Curie (Paris 6) in 2008 and received the Ph.D degree in Image Processing from the Universite de Paris 13, in 2013. His research deals with low level video processing and 3D environement perception for robotics, he uses GPU programming in order to implement real-time processing.

Anthelme Bernard-Brunel holds a technological university level diploma in Electrical and Computer Engineering issued by the IUT de Ville d'Avray (2012). Since, Anthelme follows an apprenticeship for being graduated in electrical engineering and computer science. It alternates between courses at UPMC Polytech 'Paris and his job at the Onera/ DTIM.

Guy Le Besnerais graduated from the Ecole Nationale Superieure de Techniques Avancees in 1989 and received the Ph.D. degree in physics from the Universite de Paris-Sud, Orsay, France, in 1993. He joined the Onera in 1994, where he is now a senior scientist in the Information Processing and Modelization Department. His work concerns inversion problems in imagery and computer vision.

Rotary Wing UAV pre-sizing : Past and Present Methodological Approaches at Onera

P.-M. Basset,
A. Tremolet,
T. Lefebvre
(Onera)

E-mail : pierre-marie.basset@onera.fr

DOI : 10.12762/2014.AL08-10

Thanks to their Vertical Take-Off and Landing, hover and low speed capabilities rotorcraft have a wide variety of applications. A very wide range of rotorcraft concepts have been invented and creativity is still abundantly present, especially in the field of Rotary Wing Uninhabited Aerial Vehicles. First, some typical past studies requiring RW-UAV pre-sizing will be described. They were pragmatically dealt with using the means available at that time. The interest as well as the limits of these studies contributed to pushing Onera in the development of a dedicated tool for rotorcraft evaluation and pre-sizing with the most ad hoc models and methods. The second part of the paper will present the main lines of the current methodological approach built in the CREATION project: “Concepts of Rotorcraft Enhanced Assessment Through Integrated Optimization Network”. Then, some evolution perspectives of this numerical platform will be given, to better address especially the RW-UAVs pre-sizing.

Introduction

An Uninhabited or unmanned Aircraft System (UAS) is composed of four main components: the air vehicle (called UAV hereafter), the payload, the control station and the data link. The operators interact with the UAS through the data link and are usually located in the control station. The focus of this paper is on the vehicle itself (UAV) and more precisely on rotorcraft, a widespread category of air vehicles.

Indeed, rotary wing aircraft have a very wide range of applications, thanks to their Vertical Take-Off and Landing (VTOL), hover and low speed capabilities. In addition, since they do not require a runway or any heavy facilities, they are more often used than fixed wing aircraft for research in aerial robotics by universities and research institutes. Therefore, a very wide variety of rotorcraft concepts have been invented. This creativity has been reinforced by the blossoming and rapid expansion of UAS projects, due to their reduced cost and risk of development, compared with inhabited aircraft. This paper is dedicated to Rotary Wing Uninhabited Aerial Vehicles (RW-UAV).

Without claiming to present an exhaustive review of all rotorcraft concepts here, a brief overview of the main categories can however be given.

Five main categories of rotorcraft can be distinguished :

1. "Tilt Blade Tip-Path-Plane": this is the most widespread case, the most well-known rotorcraft being the helicopter, with a main rotor used both for lift, propulsion and the pitch and roll controls. The blade TPP is tilted by using cyclic controls, changing the lift distribution over the rotor disc, causing different blade flapping angles.

2. "Tilt-Body": in this case, different rotors are used and the total aerodynamic force resulting from their thrust can be tilted by inclining the whole aircraft or the part on which the rotors are fixed.

3. "Tilt-Rotor": one or more rotors are tilted entirely, i.e., their shaft is directly oriented in the direction in which the main force must be produced. This can be : one TR like in the Rotoprop case, where the tail rotor is used in hover and low speeds like a classical anti-torque rotor and at higher speeds like a pusher rotor, or two tiltable coaxial contra-rotating rotors like in the Verticopter concept or more TR, etc.

4. Different Lift / Propulsion devices: in these cases, rotors are combined with wings, propellers or other auxiliary propulsion. The rotors are mainly used for producing lift at low speeds; this lifting function is partially or totally completed by wings at higher speeds.

5. Special cases are when the rotor itself becomes fixed wings, for instance by stopping it at high speeds (case of stoppable rotor) or by retracting the blades in a circular wing, in the case of the variable diameter rotor.

Given the large variety of rotorcraft concepts, selecting the best suited concept for a certain type of application is often not straightforward. Indeed, even though some concepts are better suited for high speed flight and others for low speed flight for example, this kind of flight performance criteria must be supplemented by many others for a correct concept selection, including flight safety, cost, maintainability, environmental impact, etc.

Beyond this first difficult step of concept selection, another difficulty is the pre-sizing of the various design parameters typical of a certain rotorcraft concept: number of rotors, number of blades, radius, mean chord, rotational speed, etc. Hence, the design engineer has to cope

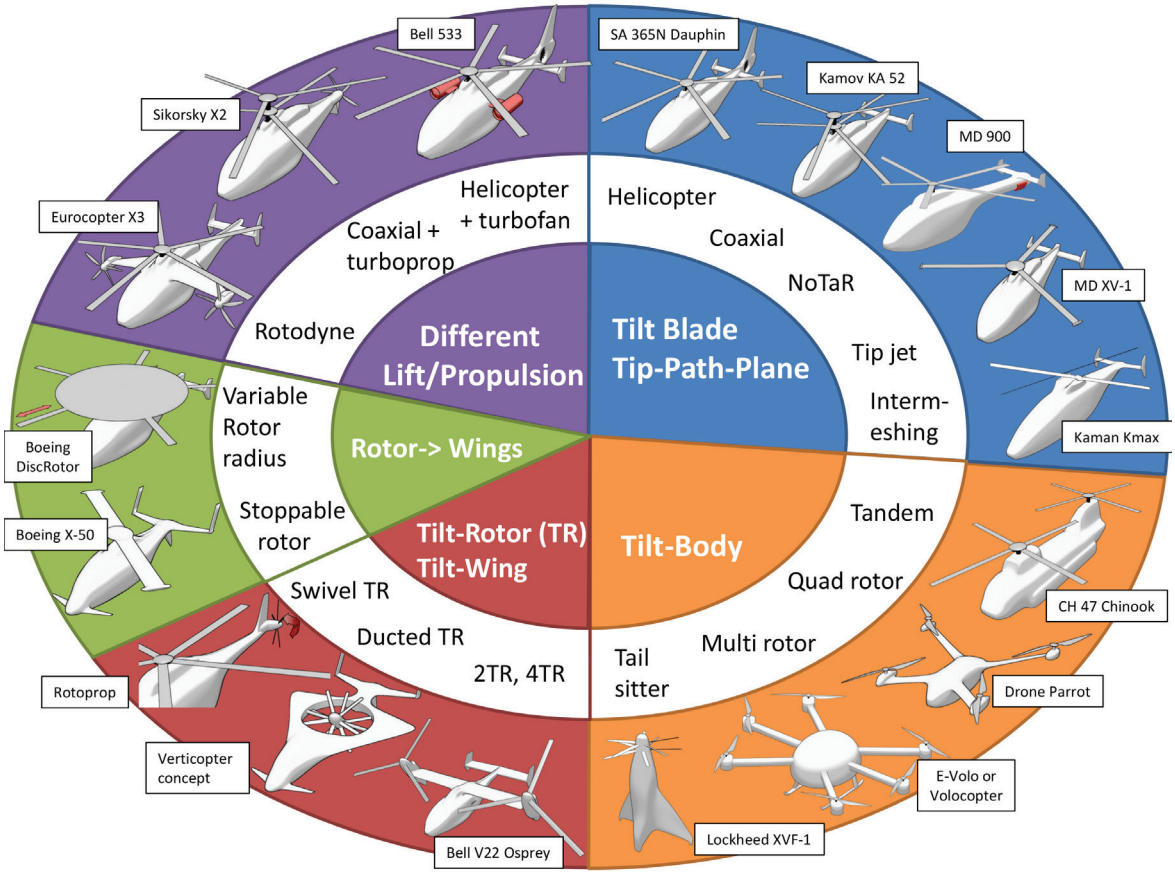


Figure 1 - Overview of the different types of rotorcraft concepts

with a multivariable multi-objective optimization problem under constraints (design rules, operational constraints, airworthiness regulation, etc.).

As indicated from the title, the paper is dedicated to the presentation of the evolution of Onera approaches in this field. Of course, it would have been interesting to include a description of the methods developed elsewhere, but the extent of the paper does not allow such a wide presentation.

First, some typical past studies requiring RW-UAV pre-sizing will be described. They were pragmatically dealt with using the means available at that time. The interest as well as the limits of these studies contributed to pushing Onera in the development of a dedicated tool for rotorcraft evaluation and pre-sizing, with the most ad hoc models and methods. The second part of the paper will present the main lines of this methodological approach built in the CREATION project: "Concepts of Rotorcraft Enhanced Assessment Through Integrated Optimization Network". Then, some perspectives of the evolution of this numerical platform will be given to better address especially the RW-UAV pre-sizing.

Past : earlier studies about RW-UAV pre-sizing at Onera

Three examples of past studies dealing with RW-UAV pre-sizing will be briefly presented hereafter: two concern European projects (CAPECON and MAVDEM); the third is an expert assessment performed by Onera for the French government (ExDro).

CAPECON

CAPECON, which stands for "Civil UAV APplications & Economic Effectivity of Potential CONfiguration solutions", is a European project of the 5th framework, involving 20 organizations (9 industries, 5 aeronautics and space institutions and 6 universities) from eight countries. The project was mainly developed from 2001 to 2005, with three groups working in parallel: two working on fixed wings (one on the High Altitude Long Endurance concept (HALE-UAV) and the other on Medium Altitude Long Endurance (MALE-UAV)) and the third one working on RW-UAVs.

From a survey of potential civil RW-UAV applications, five different application groups that were similar in terms of requirements were defined, taking into account the range, endurance, altitude, payload, speed, safety, all weather capability, etc. An analytical study was performed based on a multi-criteria matrix method resulting in the selection of the two most promising multi-role missions: one for In-Line of Sight missions (local missions) and the other for Out-of-Line of Sight missions (broader range missions). Hence, two operational concepts were derived defining their respective specifications in terms of payload, flight performance and other mission requirements.

At that time, there was no tool (models and methods) for selecting the most suited rotorcraft concepts. Therefore, the choice was done following a rather conservative approach, allowing the use of the available pre-sizing means for a conventional single main rotor / single tail rotor helicopter on the industry side. However another "less classical" configuration was studied : a coaxial rotorcraft with two contra rotating coaxial rotors (see figure 2).

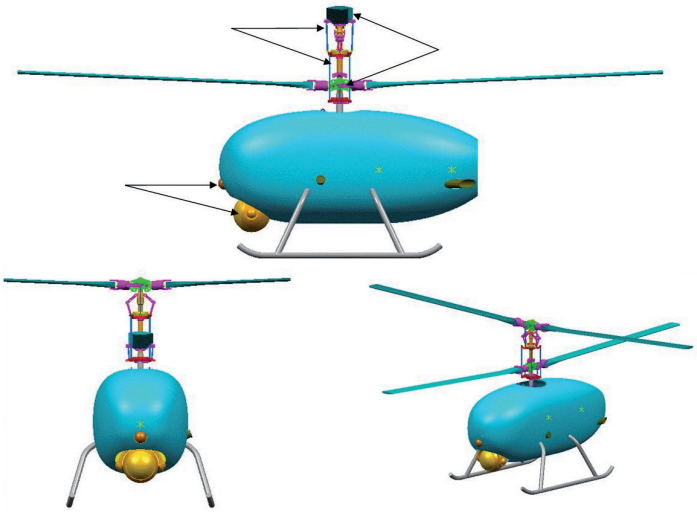


Figure 2 - CAPECON coaxial RW-UAV (from [1])

As described in [1], Eurocopter adapted its empty weight calculation model by adjusting the sizing laws and technological coefficients based on a database collected from existing RW-UAVs available at that time. This adaptation was of course needed to take into account the smaller sizes and thus lighter weights of UAVs compared with inhabited helicopters. The engine plays a significant role in both the weight breakdown and the flight performance. The choice of engine type (electrical, piston or turbine engine) depends primordially on the power demand. In the CAPECON case, the payload weighed 150 kg and the Maximum Take-Off Weight was of about 500 kg. Therefore, a piston engine was selected. A statistical engine model was derived from an aeronautical piston engine database for estimating its weight and performance in terms of available power and fuel consumption versus temperature and altitude.

The flight performance in terms of ceilings, range, endurance and speeds result from the balance between the power required by the rotors in steady flight conditions and the useful power provided by the engine (taking into account the mechanical losses and other power consumption by the equipment). Thus, the required power is a key parameter. For the coaxial configuration, Eurocopter applied its tool built for single main rotor helicopters and then tuned some empirical modeling parameters based on a bibliographical study of coaxial rotorcraft.

Among different contributions to the rotary wing group of the CAPECON project, Onera developed an inflow model for coaxial rotors, in order to better assess the flight performance of this configuration. Indeed, the induced power required by the rotors to produce the lifting force is the most important term in the power demand, especially at low speeds. Moreover, one of the most important specificities of the coaxial configuration (with respect to the single main rotor helicopter) is the aerodynamic interaction between the two rotors. Therefore, a model was created for calculating the mean induced velocity through each rotor in interaction with each other. The model is applicable in hover, vertical climb or descent flight, as well as in forward flight. It takes into account the radial contraction downstream in the rotor wakes, as well as the fact that in forward flight the rotor wakes are skewed backwards, reducing the aerodynamic interaction. Above a certain forward speed, there are no more rotor interferences and the required power is then closed to the case of two isolated rotors. This coaxial rotor inflow model is described in more detail in [2].

This model was later improved and adapted for another RW-UAV, dedicated to the inspection of large structures, such as dams, bridges, dykes, cooling towers, factory chimneys, or cliffs, etc. A flight dynamics model was built by Onera for the Infotron coaxial UAV, within the context of the ADOPIC project *("Aide au Diagnostic d'Ouvrages Par Imagerie Conventionnelle")*.

Figure 3 - IT180-5, coaxial UAV (Infotron)

In this paper [2], three levels of analysis for assessing the RW-UAV steady flight performance have been presented :

- analytical calculation by the energy method ;
- flight mechanics computation, taking into account the comprehensive equilibrium of forces and moments resulting at the rotorcraft center of gravity ;
- overall performance assessment, in terms of the power required and fuel consumption on a complete mission profile, including: hover, climb, cruise, descent, loitering flight, etc.

Here, only the analytical calculation of the required power is recalled and discussed as an illustration of that previous work. At the level of the power balance or energy method, the CAPECON coaxial configuration was compared by Onera with equivalent ones: the helicopter, tandem twin-rotor and tilt-rotor concepts. Here, equivalent means comparable. The equivalent helicopter has a single main rotor with four blades instead of two rotors with two blades. The tandem has two rotors exactly identical to those of the coaxial configuration, but separated. The tilt-rotor concept has two smaller rotors, but with three blades whose dimensions (radius and mean chord) are calculated in such a way that the rotor solidity is the same as in the other configurations. Thus, all the four configurations have the same rotor solidity (ratio between the surface of blades and the rotor disc surface).

An example of a comparison between these four configurations is shown in figure 4, where the power required versus the forward speed is plotted from a hover flight to straight and steady level flights up to 280 km/h. At low speeds, the tandem required less power than the others thanks to its two large separated rotors, the worst case being that of the tilt-rotor due to its two smaller rotors demanding more induced power and producing a downwash on the wings. At higher speeds, the trends are inverted: the tilt-rotor requires less power and can go faster in its airplane mode, whereas the tandem is hindered by the drag penalty, due to its longer fuselage and larger pylons (parts between the fuselage and the rotor heads). The helicopter and coaxial configurations are good compromises. The coaxial configuration is better at low speeds and more significantly at intermediate speeds where the required power becomes very close to that of the tandem

case when the two rotors have negligible aerodynamic interferences, whereas the helicopter is better at higher speeds, because of the drag penalty due to the larger rotor mast of the coaxial configuration.

In these comparisons, the arbitrary choice was made of comparing configurations with the same total weight and same rotor solidity. The goal was mainly to illustrate the effect of different rotor arrangements. However, these configurations have obviously different empty weights. Moreover, even if the considered payload is the same, the fuel weight will be different, not only because the required powers are different, but also because the engine is more likely to be different. Indeed, to take advantage of the tilt-rotor configuration, the useful power should be higher, which means a heavier engine. This preliminary study highlighted the fact that, for a deeper analysis of their flight performance, a more detailed weight breakdown assessment and the modeling of different engines (power, consumption and weight) are needed for a more comprehensive comparison of these configurations.

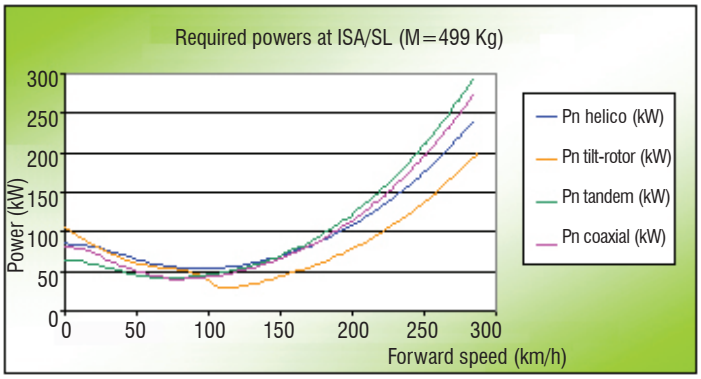


Figure 4 - comparison of the needed powers for the CAPECON coaxial configuration and the equivalent helicopter, tandem and tilt-rotor configurations (extract from [2])

MAVDEM

MAVDEM (Miniature Air Vehicle DEMonstrator) is a project funded by the European Defense Agency. This was a four year project (from September 2005 to September 2009) with a consortium composed of French (Onera and Alcore Technologies), Spanish (SENER), Italian (Oto Melara and Celin Avio) and Norwegian (TellMie) partners.

The objectives of the project are to define, build and flight-test a MAV configuration (less than 50 cm wingspan). This MAV should be capable of hovering and economic fast cruising, in order to perform infantryman support missions. Examples of such missions are open-field observation or city exploration.

In order to perform such missions, this MAV must combine two capabilities :

- Hovering, in order to look inside a building through windows, for example;
- Economic fast cruise, in order to cover the maximal area in a minimum of time, with the maximal endurance.

Those two objectives are conflicting and require the right trade-offs to be made, in order to meet the requirements in terms of endurance and velocity, which are rather challenging :

- Endurance requirement : 15 minutes hovering and 30 minutes of economic cruise ;
- Velocity requirement : 20 m/s as maximum speed.

One important issue for the success of the project was to choose a vehicle concept able to meet the requirements. The methodology proposed for this choice was to look at a wide spectrum of possible solutions, then to detail them and at last to choose the best vehicle to be built. The concepts that were analyzed were either taken from existing state-of-the-art designs or created.

Once potential candidate vehicle configurations had been identified, a three-stage selection process was made. It culminated with the final vehicle built (see figure 5). Each stage enables the design of the candidate configurations to be enhanced, as well as the least adapted ones to be discarded. This is described more in detail in the following paragraphs.

Figure 5 - Illustration of the vehicle configuration selection methodology

The first action of the vehicle configuration selection was to make a survey, as wide as possible, of the existing VTOL concepts. During the preparation of this survey, ideas arose and new original concepts were designed within the consortium. This survey ended with 26 candidate concepts, described by illustrative pictures, and the way to control them on all 3 axes. Then, a multi-criteria analysis was performed, in order to sort the various concepts. The selection was based on a limited number of criteria, divided into 4 main categories: performance, controllability/stability/maneuverability, complexity of design and safety (as shown in table 1). This analysis was not based on calculations, but rather on the expertise of the consortium. First a score was given, once per criterion, for each concept and then each criterion was weighted, in order to account for their relative importance, and a multi-criteria analysis was performed based on these values. The process, checked through alternative multi-attribute decision-making methods, ended in the selection of 5 concepts.

Performance	Hover endurance Lifting power at max. speed Propulsion power at max. speed
Controllability/stability/ maneuverability	Ease of control Degree of inherent stability Maneuverability
Complexity of design	Airframe complexity Aerodynamic shape Propulsion integration
Safety	Operator security Additional safety

Table 1 - Criteria for top-down selection

After this first selection, the level of detail had to be improved (up to the preliminary design), in order to perform a second selection aimed

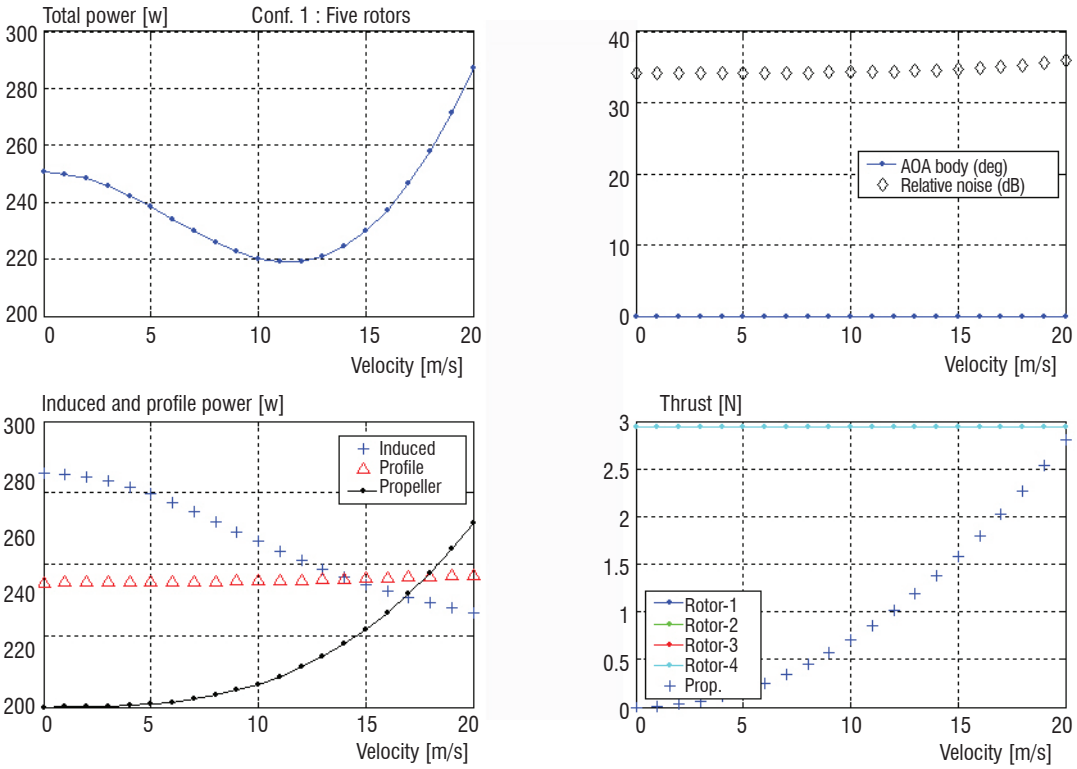


Figure 6 - Performance curves for configuration 1

at keeping only two concepts. This design improvement required several tasks :

- Identification and characterization (weight, dimensions and power consumption) of the required onboard components ;
- Propulsion considerations, especially in regard to battery volume and engine efficiency ;
- Aeroshape design refinement, with the associated estimated lift and drag ;
- Performance estimation.

Concerning this last point, the performance of each concept was calculated using a power balance method, all implemented using the Matlab software application. Several types of curves resulted for each configuration. Figure 6 shows an example of the performance curves.

All of these tasks enabled the design of the 5 retained configurations to be improved. The new designs are presented hereunder (table 2).

This situation led the consortium to decide to combine the best properties of the various configurations into 2 options : a 4-rotor concept (Configuration A) and a double coaxial rotor concept (Configuration B).

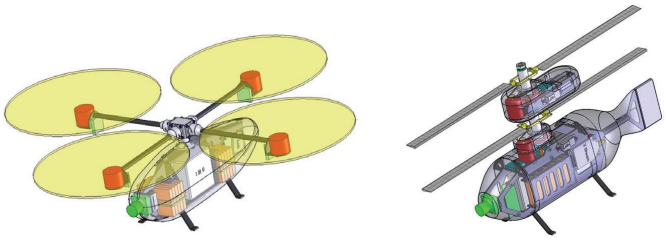
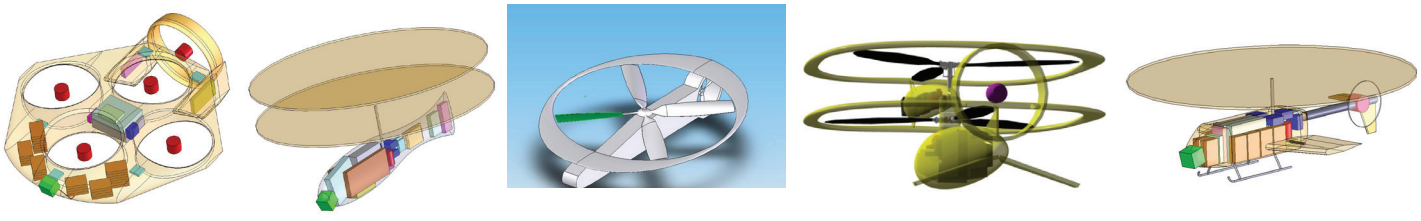


Figure 7 - Illustration of configurations A and B



Configuration 1	Configuration 2	Configuration 3	Configuration 4	Configuration 5

Table 2 - Preliminary design

Before selecting the final configuration to be built and flight tested, another improvement of the configuration designs was necessary. This design improvement included propulsion tests, in order to calibrate the calculation codes with actual values of power, efficiency, torque, etc. Various engines and propellers were tested, in order to choose the best-suited solution for each configuration.
In parallel to these propulsion tests, the structure and internal arrangement were more precisely defined. This design was performed by taking into account operational aspects, such as transport, battery removal and replacement, or manipulation by soldiers equipped with gloves.

Based on this "internal design", external fuselages were optimized to the least possible volume. CFD calculations have been performed to assess their lift and drag characteristics, as a function of the angle of attack (figure 8).

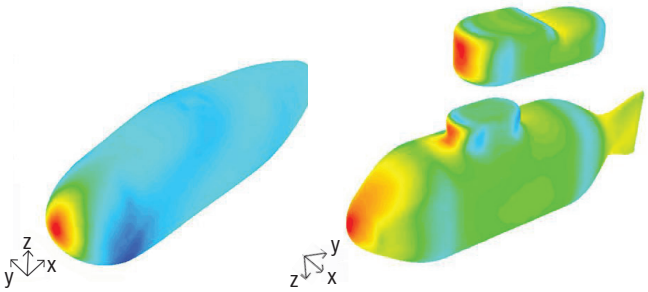


Figure 8 - Example of a CFD result for configurations A (left) and B (right)

A loop calculation of the estimated performance was made for both configurations, based on the experimental propulsion tests, the improved weight budget (from the structure definition) and the aerodynamic analysis. This performance estimation was performed in terms of endurance, which is the most challenging and dimensioning requirement, especially regarding the propulsion. Based on a new multi-criteria analysis and in agreement with the whole consortium, the retained configuration for the MAVDEM was the 4-rotor concept (figure 9).

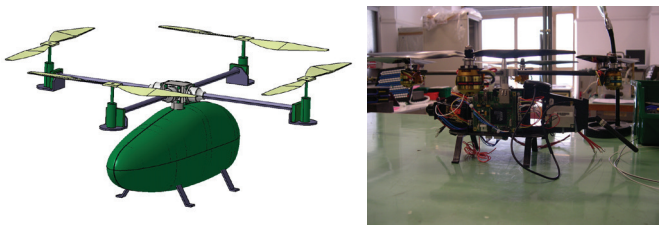


Figure 9 - Selected MAVDEM configuration (left) and partially built concept (right)

The goal of the MAVDEM project was to perform the preliminary conception until the Technology Readiness Level (TRL) 4 (i.e. flight test demonstration). This turned out to be too ambitious with respect to the time frame of the project MAVDEM ended at the software integration validation test stage, even though the previous steps were successfully reached (including individual hardware tests, system integration and guidance, and navigation and flight control software implementation).

Some conclusions can however be drawn from the MAVDEM experience, about the methods used for concept evaluation and pre-sizing. First of all, the MAVDEM process was sequential, i.e., at each level of the selection, new tools had to be selected and updated. Furthermore, specific difficulties arose at each step :

- For the multi-criteria analysis, the criteria table was built using the experts involved in the project and took some times to converge.
- For the second selection loop, specific developments had to be made on the existing tools, in order to take into account the various designs that were selected and an agreement had to be reached between the experts on the values of the tool parameters.
- For the last step, the High Fidelity calculations (C.F.D. and F.E.) were based on the choice of the experts and not driven by the choice of a specific surrogate model.

Therefore, the impact of the experts' judgment was significant in this project. The need is arising for a calculation platform that is less reliant on the direct involvement of discipline experts for the predesign and evaluation of rotorcraft concepts. Moreover, a numerical workshop or prototyping tool would allow these expert assessments to be capitalized in models and methods, as well as allowing design loops to be performed more systematically, with iteration rather than a sequential case-based process.

ExDro

Within the ExDro expert assessment (where ExDro stands for "Expertises Drones", 2008-2009) performed by Onera for the French Ministry of Defense, a work package was dedicated to the RW-UAV for both the Army and Navy. A significant part of the study was aimed at determining the best rotorcraft to respond to the requirements in terms of flight performance. For this purpose, two investigations were carried out :

- the evaluation of the proposals by three industrial consortiums with two helicopter UAVs based on the adaptation of two manned helicopters (the Orka based on the Cabri-G2 and the "Unmanned Mission Enhanced Little Bird" based on the A/MH-6X) and on the Bell Eagle-Eye Tilt-Rotor UAV ;
- the selection and pre-sizing of four different rotorcraft concepts, as well as their evaluation and comparisons regarding the mission specifications.

For evident confidentiality reasons, the expert assessment of the rotary wing air vehicles proposed by industry will not be discussed here. Moreover, the second part of this study is more relevant to the topic of this paper. This work on alternate rotorcraft concepts was done in three steps:

- Step 1 : the review of the different types of rotorcraft bringing out their main strengths and weaknesses, and the pre-selection of four concepts potentially well suited for the missions of both the Army and Navy ;
- Step 2 : the pre-sizing of these four concepts up to a level of description, allowing the use of the available analytical tools for the flight performance computation;
- Step 3 : the evaluation and comparisons of the flight performance of these four rotary wing air vehicles regarding the military mission specification.

The four pre-selected concepts were: a helicopter capable of reducing the revolution speed of the main rotor in flight (as in the case of the Hummingbird A160 RW-UAV), a coaxial rotorcraft, a tilt-rotor tilt-wing concept (as in the case of the Erica concept for reducing the downwash of the rotors on the wings), a compound helicopter with a pair of wings and a vectored thrust device at the rear (as, for example, in the case of the Piasecki Pathfinder and X-49A Speedhawk).

Of course, the purpose here is not to go into the details of the pre-sizing of these four RW-UAVs, but rather to sum up the main common points in terms of the methodological approach.
For every pre-sizing exercise, the starting point and the ending description level must be clearly defined. In this ExDro case related to VTOL UAV, the pre-sizing had to be done from scratch (voluntarily ignoring the air vehicles proposed by industry) and the goal was to obtain a rough draft of the main characteristics of the aircraft allowing the flight performance calculation using the power balance method. For each concept, the input data for that kind of calculation had to be determined: Maximum Take-Off Weight (MTOW), engine data (weight and provided power with respect to the altitude and temperature), rotor characteristics (radius, number of blades, mean chord, rotational speed, etc.), fuselage or airframe data (drag and sizes), wing data (for calculating their lift and drag, in the case of the tilt-rotor and compound configurations).

A common starting point was obviously the payload, since all of these RW-UAVs had to carry the same mission payload. This was completed by an estimation of the weight of all of the other on-board equipment. Taking into account the equipment configuration with the highest demand in terms of weight, the maximum on-board equipment weight (W_{PL} including here the payload and all of the equipment) was set. From that entry, a statistical approach was applied, like the one presented in [4], for example.

After checking that our own rotorcraft database provided results similar to those given in [4], the following logical chain was used for the derivation of the main rotorcraft characteristics.

The first key parameter to be determined is the take-off gross weight W_0. Knowing by statistics the ratio W_{PL}/W_0 for a certain type of rotorcraft concept and W_{PL} from the requirements, W_0 can be estimated. As illustrated in the scheme in figure 10, from this key parameter can be derived: on the one hand, the take-off power, the engine and the main weight breakdown, as well as, on the other hand, the lifting rotor characteristics and the main airframe dimensions.

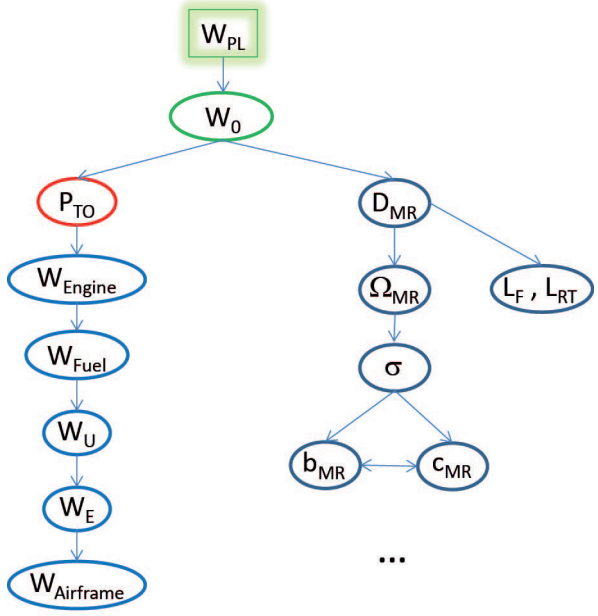


Figure 10 - Example of a logical chain for pre-sizing (MR for Main Rotor)

For the weights, the notation and definition are :

$$W_0 = W_U + W_E$$
$$W_0 = (W_{useful\_load}) + (W_{empty\text{-}weight})$$
$$= (W_{payload} + W_{fuel}) + (W_{engine} + W_{airframe})$$

P_{TO} is the take-off power, from which, by using an engine database, a first choice of engine can be made giving its weight and specific consumption. The fuel weight can then be calculated knowing the mission requirements and typical specific consumption values. The payload, fuel, engine and gross weights being known, the airframe weight can be deduced.

From the design gross weight (W_0), the lifting rotor diameter (D) can be estimated by an analytical expression from the statistics (see figure 10). The fuselage length (L_F) and the rotorcraft over-all length (rotors turning L_{RT}) can then be assessed, also by statistics. The disk loading is defined by : $T/S = W_0.g/(\pi.R^2)$, where R is the rotor radius. By dimensioning the lifting rotor(s) for the most demanding flight case (e.g., hovering at the highest take-off altitude), which sets a certain value of the air density (ρ), and with typical values for helicopters in terms of mean blade lift coefficient $(Czm = 0.6)$ and blade tip Mach number $(M = 0.6)$, the blade rotational tip speed $(\Omega.R \cong 200\ m/s)$ and the rotor solidity can be calculated :

$$\sigma = \frac{6}{\rho(R\Omega)^2}\frac{1}{Czm}\frac{T}{S}$$

By definition, the rotor solidity is the ratio of the surface of blades over the rotor disk surface:

$$= \frac{b.c.R}{.R} = \frac{b.c}{.R}$$

The rotor radius being known $(R=D/2)$, if the number of blades (b) is chosen, for example for a coaxial the minimum number of blades is four (two by rotor), then the mean chord (c) can be calculated. Otherwise, the blade aspect ratio can be first estimated (R/c) and then the number of blades can be deduced.

These main characteristics having been assessed, the power balance method can be applied for a first evaluation of the rotorcraft performance. This second stage may lead to the adjustment of some parameters and/or to another choice for the engine. Therefore, the pre-sizing and the flight performance assessment must be viewed in a loop with iterations.

Of course, this logical chain must be adapted to the rotorcraft concept considered. This simple approach gives a very first rough draft in a conservative way. That is to say, for known rotorcraft concepts with a significant number of examples in the database. However, a strength of this basic method is that the more complex trade-off requiring a wider scope of multidisciplinary models, as well as higher fidelity models, is however implicitly included. Indeed, even at this very early stage of the predesign, the use (in the database) of existing concrete flying rotorcraft, that have reached their full development, allows the constraints and disciplinary inter-dependences involved later in the preliminary conception process to be anticipated.

However, the statistical approach has the obvious drawback of being conservative by nature, i.e., limited to the rotorcraft concepts present in the database. Moreover, the validity ranges of the key parameters, (gross weight, sizes, etc.), are in principle limited to the maximum and minimum values available in the database: applying the design

trends beyond these limits entails extrapolation. Therefore, this method is not suited for the innovation of new concepts or the exploration of existing configurations beyond the currently known limits.

In addition to their own contributions to a specific issue and to the general problem of RW-UAV pre-sizing, these first studies have brought forward the need for a more global multidisciplinary approach and a more comprehensive analysis, including flight performance, safety, environmental impact … They have paved the way toward the definition and construction of a general analysis tool for rotorcraft concepts.

Present : the CREATION project

Rebounding after different studies like the ones previously presented showing recurring needs in this field, a first attempt was made for setting-up methods and tools (see [5]). But definitely for going further, the expertise of several scientific departments must be involved.

CREATION is an Onera multidepartment project launched in January 2011 for a four years period. CREATION means “Concepts of Rotorcraft Enhanced Assessment Through Integrated Optimization Network”. The main goal is to build a numerical platform for the analysis and evaluation of the flight performance and environmental impact (noise and air pollution) of rotorcraft concepts.

The ambition is high, because the evaluation tool must be applicable to any kind of rotorcraft concept, whatever its description level. This last requirement means that the platform must be able to cope with the difficult problem of pre-sizing from scratch, that is to say, when only an idea of a concept has to be explored and/or when only the mission requirements are known.

However, the building of the platform has been scheduled in a pragmatic way, increasing the complexity step by step with realistic milestones:

- Milestone 1 - **evaluation** capability: setting up modules and workflows for the case of an existing helicopter,
- Milestone 2 - **pre-sizing** capability: setting up models and methods for the case of a new helicopter, to be defined from its mission requirements,
- Milestone 3 - **innovation** capability: generalizing the platform to alternate rotorcraft and applying it for an innovative concept.

Framework

The numerical platform CREATION is a computational workshop with models and methods. The models contain the knowledge from the various disciplines and they are the suitable evaluation tools for the available data. The methods correspond to the know-how for using the models together as tools for the evaluation, pre-sizing and innovation purposes.

First, the framework is presented here, i.e., the organization and implementation. Then, the main features of models and methods will be summarized.

From the general specifications, certain important features are derived for the organization of the platform. It must be composed of multidisciplinary modules, as well as multi-modeling levels inside each module.

Seven first main disciplinary modules have been identified as the “seven pillars”. Two are central within the tool; they can be called “goal modules”:

- Flight performance ;
- Environmental impact (acoustics, air pollution, etc.).

Around this bipolar structure of the tool, five “means modules” are present for providing the means for the flight performance and pollution evaluations:

- Missions & Specifications
- Architecture & Geometry
- Weights & Structures (including aeroelasticity)
- Aerodynamics
- Power Generation (engine).

The platform could be enriched later with other modules, depending on specific needs (e.g., regarding mission payload) or on other evaluation criteria or constraints (airworthiness regulation, economic viability, etc.). Safety, and in particular autorotation capability, must be assessed through criteria suited to the level of description of the rotorcraft and to its characteristics ("manned or unmanned" type, gross weight, etc.).

Except for the "Missions & Specifications" module, which provides the flight conditions with respect to the requirements and to the mission profile, several modeling levels have been implemented within each of the other disciplinary modules. This multi-modeling level feature is needed to adapt the "modeling granulometry" to the considered level of detail in the data describing the rotorcraft. Four main modeling levels are currently used :

- Level 0 : Response Surface Models (RSM) based on databases or simulations ;
- Level 1 : simple analytical models based on physics ;
- Level 2 : more comprehensive analytical models ;
- Level 3 : numerical models.

The more the model is complex, the more it is time consuming in terms of computational time, but also the more demanding it is in terms of required data inputs. The Response Surface Models or more generally reduced models are useful, not only for decreasing the computational time, but also for reducing the amount of input data. The modeling complexity and the number of needed inputs vary according to each other.

Therefore, this vertical structure in modeling levels is also fully justified and useful in the pre-sizing process. Indeed, when a predesign must be done from scratch, its definition must be made from the lower models to the upper models, increasing the data describing the aircraft step by step. In the current state of the tool, for the Onera purposes, the starting point is the initialization of the pre-sizing with Level 0 models; then a first pre-sizing loop is performed at Level 1 and the description of the rotorcraft is progressively enriched until Level 3 is reached. Thus, in the current state of development of this tool, the final definition stage of the pre-sizing is actually when all of the data needed for using a numerical rotorcraft flight mechanics code has been defined. More precisely numerical means here that the main rotor model is based on a blade element approach allowing

a fine description of the blade properties (geometry, aerodynamics, inertia, etc.).

In order to give a concrete image of this computational workshop, it can be seen in a 3-Dimensional space as a building (see figure 11).

NFM : Numerical Flight Mechanics
AFM : Analytical Flight Mechanics
BP : Balance of Power

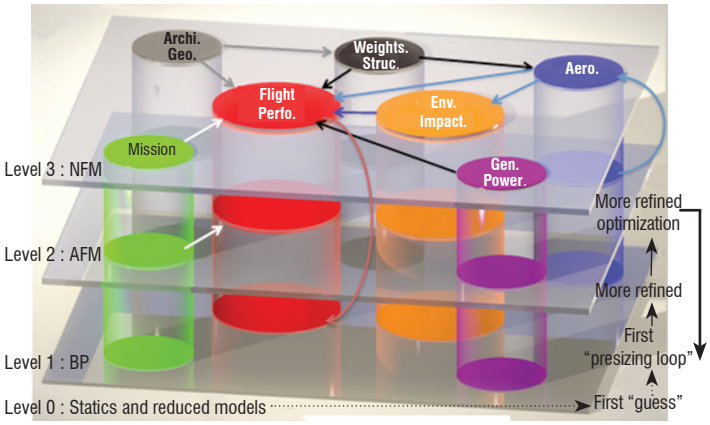


Figure 11 - A 3 dimensional view of the CREATION computational workshop

In its current state, the workshop has four floors corresponding to the modeling levels with Level 0 in its basement and seven pillars corresponding to the disciplinary modules.

The approach is to begin the evaluation of a rotorcraft from the modeling level consistent with the available data describing it. If the rotorcraft exists and all of the required data is known, the evaluation can be performed with the highest modeling level. Otherwise, the evaluation starts from the appropriate level and, if a more detailed evaluation is required, an enrichment of the data can be proceeded to, through a bottom-up process. On the contrary, if no data is available, an "ab initio" pre-sizing must be done. Level 0 provides "first guess" rough estimates of the main data, which are then recalculated and completed with other data in a first pre-sizing loop at Level 1. More refined optimizations are performed at the upper modeling levels, for improving the assessment of this data and/or for the pre-sizing of other parts of the rotorcraft. Macro iterations between these various pre-sizing loops are needed to ensure the consistency of the global optimization.

For example, in the case of the helicopter pre-sizing, Level 0 allows the process to be initialized based on design trends from a database. At Level 1, the main data describing the fuselage and the main rotor is calculated within a multi-objective optimization loop. Then, at Level 2, the complete equilibrium of forces and moments resulting at the rotorcraft center of gravity can be considered with a comprehensive analytical flight mechanics model allowing the pre-sizing of the rear components (horizontal stabilizer, fin and tail rotor). At Level 3, a more refined predesign of the main rotor blade (twist, airfoils, chords, etc.) can be performed, using a rotor blade element model. From this, a new rotor polar (blade mean drag versus mean lift and speed) can be computed, for example, therefore the results of the upper level optimizations can be fed back into Level 1 for a new round of optimization up until the convergence of the whole process has been reached.

The arrows appearing in figure 11 are just to illustrate the connections between the disciplinary modules. The workflows between the models, in terms of data exchanges (inputs/outputs), must be determined by establishing the diagram of the dependencies required for a certain application, depending on the available data and on the modeling level. Data flows between the modeling levels are also used, for example when an upper level model provides a reduced model to a lower level model, or in the bottom-up enrichment of data.

Models

A comprehensive and detailed description of the models is beyond the scope of this paper. First, descriptions have been given in [6] and [7]. Here, only the main common characteristics of these models are highlighted.

In each of the involved disciplines (flight mechanics, aerodynamics, acoustics, structures etc.), Onera has been developing for years expert models based on physics, constantly improving their fidelity by taking into account new results or experimental data. These high fidelity models are very demanding in terms of input data. They are not suited for the evaluation of a rotorcraft described only by few items of data or for pre-sizing studies. Their computational cost is too high for exploring a design space. Moreover, the large amount of inputs required for using these complex models prohibits their application at an early design stage.

Therefore, in the CREATION project, the disciplinary experts were asked to provide simplified models corresponding to the defined modeling levels. This is clearly an added value of the project, to develop simplified models working with few items of data and yet allowing a realistic assessment of the flight performance, as well as the acoustic and air pollution of any rotorcraft. This is a challenging task and the main approaches for developing this kind of models will be summarized hereafter.

Note that the complexity of a model is not a guarantee of its high fidelity. There are of course two kinds of uncertainties: those arising from the model itself (equations, formulations) and those arising from the input data. Using a complex expert model at an early evaluation stage can lead to lower fidelity results than using a simpler model with the few available items of data.

Various approaches have been applied to set these simplified models. They can be based on databases, on simulation results and on physics. For example, a database on existing rotorcraft (about 260) has been gathered, but also a database on aeronautical engines (turbine, piston, electrical) and acoustics based on helicopter certification measurements. Simulation results from upper level models can be used to derive simpler models at lower levels. This is for example the case of the aerodynamic rotor model giving the blade mean drag with respect to the blade mean lift and advance ratio. The formulation is based on physics with three terms: a basic drag, a drag due to airfoil flow separation and stall effects, and a drag due to compressibility effects, completed by a factor for the Reynolds effect. A representative number of simulations with a rotor blade element model are performed, from which the various parameters of the surrogate model are deduced.

Various mathematical techniques are used to generate these reduced models, such as statistical and polynomial regression, kriging, neural network, etc. More than the choice of mathematical method in these meta-modeling tasks, an important point is often to inject physics as far as possible into the model structure. The disciplinary expert must

be able to set the formulation to represent the most important physical effects relevant for the evaluation and consistent with the available data. Therefore, most of the simplified models are based on physics with parameters tuned by using databases obtained from experiments or simulations. Some models result from a hybrid approach, like in the weight module in which some parts are assessed based on statistics (equipment, crew and passengers, etc.) and others with models based on physics (blades, rotor, mechanical transmission, etc.). Some models are Response Surface Models: for the same inputs giving output values as close as possible (with a known precision) to the results given by one or several interacting more complex models.

These reduced models not only decrease the computational time, but also have a reduced set of suited inputs and outputs consistent with the available data at each modeling level.

Methodologies: MDO formulations and optimization techniques

The CREATION platform, like other tools aimed at designing an aerial vehicle, exhibits some multi-disciplinary particularities:

- it incorporates a large number of disciplines that are, sometimes, strongly coupled and must cope with a high number of variables;
- it has several levels of modeling, ranging from statistical tools to high-fidelity ones, thus requiring the use of reduced models;
- it requires a detailed exploration of the design space to be enabled, together with the capability to identify the global optimum of the entire system.

An engineering system method field, aimed at handling several disciplines more efficiently, has been developed to tackle those particularities: the Multidisciplinary Design Optimization (MDO). The objective of the MDO methods is to take advantage of the couplings and the synergies between the various disciplines, in order to achieve the global optimal design. The main targets of the MDO process are the quality of the solution found, the computation time and the robustness of the optimization process (i.e., the ability to converge to an optimum from a large initialization domain). Therefore, by solving the MDO problem early in the design process and taking advantage of advanced computational analysis tools, designers can simultaneously improve the design and reduce the time and cost of the design cycle. Onera has been investigating this field of research since 2004, with a 4 year internal project called DOOM ("Démarche Outillée d'Optimisation Multidisciplinaire" or Multidisciplinary Optimization Tooled Approach) and has made significant studies and achievements in this domain (see [8] and [9]).

When talking about MDO, the first step is to overcome the analysis problem that one wants to solve, that is so say, to identify the disciplinary couplings and the computation of objectives and constraints as a function of the design variables.

Once it has been done, the next step is to formulate the problem, in order to be able to use suitable optimization algorithms, which mean to select the most suitable MDO architecture. The MDO architecture defines both how the different models are coupled and how the overall optimization problem is solved. The architecture can be either monolithic or distributed. In a monolithic approach, a single optimization problem is solved. In a distributed approach, the same problem is partitioned into multiple sub-problems containing small subsets of the variables and constraints. More information on the MDO architectures can be found in [10].

Concerning the CREATION process, some monolithic approaches, such as MDF (MultiDiscipline Feasible) and SAND (Simultaneous Analysis and Design) have been evaluated. These studies have shown that the computational time required to reach an acceptable convergence was far too high for rotorcraft concept exploration. Moreover, the will to use higher fidelity models, in order to introduce more knowledge at early stages of the design process, will increase this computational time. A common way to lower the computational cost is to make smart use of the most advanced modeling tools, using response surface modeling. The meta-models (or surrogate models) the most used in the engineering field aerospace systems are for example : polynomial regression techniques, the kriging statistical model, artificial networks or radial basis functions. All of these meta-modeling techniques differ in terms of degrees of freedom, type of base functions and learning technique, thus leading to various areas of application. Within the CREATION platform, two different kinds of surrogate model were investigated : the kriging statistical model and the MOE techniques (Mixture Of Experts), combining several surrogate models [11]. They both enabled the computational time to be greatly reduced, while achieving a good accuracy in regard to the optimal solutions.

At the time of this paper, the CREATION workshop has been built by dealing with manned – inhabited rotorcraft. An example of an application is presented in [12] for a large civil transport helicopter. For that case, the objectives were to minimize the required fuel weight, the empty weight and the noise produced on the ground during the landing approach. The choice of objectives is case dependent. For a RW-UAV this can be, for example: the required engine power, the empty weight and some crucial performances for the applications considered (endurance, range or a certain speed for best endurance, best range or maximal speed). For rotorcraft pre-sizing, it is often a question of compromise between the hover and forward flight performance. This is why the objectives or criteria are generally : the hover efficiency (Figure of Merit, i.e., ratio between the ideal minimum needed power corresponding to the theoretical induced power according to the momentum theory P_{i0} and the actual required power for hovering, which is the sum of the induced power P_i plus the blade airfoil drag power P_{blade}) and the propulsive efficiency (equivalent lift over drag ratio for a lifting rotor: ratio of the weight W multiplied by the cruise speed V with respect to the total needed power P_{req}).

$$FM = P_{i0} / (P_i + P_{blade}) \text{ and } L/D_e = W.V/P_{req}$$

The design parameters depend of course on the kind of rotorcraft concept. However, the number of rotors, as well as the radius, number of blades, mean chord and rotation speed of each rotor, are the main parameters to be optimized.

Moreover, some adaptations are needed, especially on the aerodynamic and weight models, before coping with the case of the small scale RW-UAVs.

Conclusion and perspectives
Future : RW-UAS presizing

Examples of past studies regarding RW-UAVs evaluation and pre-sizing have shown a clear need for numerical tools combining multi-disciplinary models and multi-objective optimization methods.

An internal Onera project was launched in 2011 to respond to this kind of need and, more generally, to address the evaluation of any kind of rotorcraft concept, first and mainly from the flight performance and environmental impact points of view. A computational platform called CREATION has been developed, integrating multidisciplinary modules within multi-modeling levels, together with methodologies to cope with problems involving multi-design variables (continuous or discrete), multi-objectives and constraints related to multi-missions, etc...

The last milestone of this project is the innovation capability. The differences between uninhabited and inhabited air vehicles have been highlighted in the reference book [13], showing that the field of possible solutions is wider in the UAV case. A high potential for innovation exists in the field of RW-UAVs. Hence, a good candidate as a demonstration exercise will be to deal with the pre-sizing of an innovative RW-UAV. Note that an innovative RW-UAV concept could inspire an innovative manned – inhabited rotorcraft and reciprocally. This is for example the case of the multi-rotor UAV concept (quad-, hexa-, octo-, etc. rotors), which recently gave birth to a version with pilot onboard: the Volocopter and its eighteen electrical rotors.

Of course, the library of models must be enriched and adapted to better take into account, for example, small scale effects like the Reynolds phenomenon. Beyond that, developing a methodology for giving to the tool the ability to produce new concepts not predefined by the engineer is really a challenging task. This assumes the combination of various components (lifting and propulsive devices, airframe, etc.) and the evaluation of the resulting performance taking into account their mutual interaction.

Another aspect of the tool development is the enrichment of the criteria used for the evaluation. Obviously, the evaluation starts with the performance in stabilized flight. However, this will be extended to maneuvers and flight dynamics first related to safety and airworthiness regulations, as well as stability and controllability. This will require more integration, taking into account the Control System design as soon as possible in the overall predesign process, to assess the Uninhabited Aircraft System (UAS) as a whole ■

Acknowledgements

The authors thank the CREATION project team, i.e., all of the colleagues contributing to this multi-department project.

Acronyms

CAPECON	(Civil UAV APplications & Economic Effectivity of Potential CONfiguration solutions)
CFD	(Computational Fluid Dynamics)
CREATION	(Concepts of Rotorcraft Enhanced Assessment Through Integrated Optimization Network)
ExDro	(Expertises Drones)
FE	(Finite Element)
FM	(Figure of Merit (hover efficiency for a rotor))
HALE	(High Altitude Long Endurance)
MALE	(Medium Altitude Long Endurance)
MAVDEM	(Miniature Air Vehicle DEMonstrator)
MDO	(Multidisciplinary Design Optimization)
MOE	(Mixture of Experts)
MTOW	(Maximum Take-Off Weight)
RW	(Rotary Wings)
TPP	(Tip-Path-Plane)
TR	(Tilt-Rotor)
TRL	(Technology Readiness Level)
UAS	(Uninhabited or unmanned Aircraft System)
UAVs	(Uninhabited Aircraft Vehicles)
VTOL	(Vertical Take-Off and Landing)

References

[1] C. SEVIN, P.-M. BASSET, G.-M. SAGGIANI, J-F. BOER - *Operational & Technical Design Aspects of Future Civil Rotary Wing UAVs.* 31st European Rotorcraft Forum, Firenze, Italy, September 13-15, 2005.
[2] P.-M. BASSET, J. DESLOUS - *Performances Comparisons of Different Rotary Wing UAV Configurations.* 31st European Rotorcraft Forum, Florence, Italy, September 13-15, 2005.
[3] C. LE TALLEC, A. JOULIA - *A Miniature Air Vehicle System DEMonstration (MAVDEM project)*. AIAA Infotech@Aeropsace Conference, Seattle, USA, April, 6-9, 2009
[4] V. KHROMOV, O. RAND - *Design Trends for Rotary-Wing UAVs.* Presented at the 25th International Congress of the Aeronautical Sciences, ICAS 2006.
[5] P.-M. BASSET, C. BARTHOMEUF, A. GUILLÉ - *Setting up Methods and Tools for Flight Performance Evaluation of Rotorcraft Concepts.* 3rd European Conference for AeroSpace Sciences, Versailles, 6-9 July, 2009.
[6] P-M. BASSET, A. TREMOLET, F. CUZIEUX, C. SCHULTE, D. TRISTRANT, T. LEFEBVRE, G. REBOUL, M. COSTES, F. RICHEZ, S. BURGUBURU, D. PETOT, B. PALUCH - *The C.R.E.A.T.I.O.N. Project for Rotorcraft Concepts Evaluation : the First Steps.* 37th European Rotorcraft Forum, Gallarate, Italy, 13-15 September 2011.
[7] P.-M. BASSET, A. TREMOLET, F. CUZIEUX, G. REBOUL, M. COSTES, D. TRISTRANT, D. PETOT - C.R.E.A.T.I.O.N. *the Onera Multi-Level Rotorcraft Concepts Evaluation Tool: the Foundations.* Future Vertical Lift Aircraft Design Conference, AHS Specialists meeting, San Francisco, CA, USA, 18-20 January 2012.
[8] S. DEFOORT and al. - *Multidisciplinary Aerospace System design : Principles, Issues and Onera Experience.* Aerospace Lab Journal, issue 4, May 2012.
[9] M. BALESDENT, N. BEREND, P. DEPINCÉ, A. CHRIETE - *A Survey of Multidisciplinary Design Optimization Methods in Launch Vehicle Design.* Struc Multidisc Optim (2012) 45 : 649-642.

[10] JRRA. MARTINS, A.B. LAMBE - *Multidisciplinary Design Optimization : a Survey of Architectures.* AIAA Journal 2013, 51-9:2049-2075.
[11] D. BETTEBGHOR, N. BARTOLI, S. GRIHON, J. MORLIER, M. SAMUELIDES - *Surrogate Modeling Approximation Using a Mixture of Experts Based on EM Joint Estimation*. Structural and Multidisciplinary Optimization, 43 : 243-259, n° 2, 2011.
[12] P.-M. BASSET, A. TREMOLET, N. BARTOLI, T. LEFEBVRE - *Helicopter Pre-Sizing by Multidisciplinary – Multi Objectives Optimization*. OPT-i, International Conference on Engineering and Applied Sciences Optimization, Kos Island, Greece, 4-6 June 2014.
[13] "Les drones aériens: passé, présent et avenir – Approche globale - *The Unmanned Aircraft Systems : past, present and future – Global approach*". Collective book, La Documentation Française, ISBN 978-2-11-009376-9, Paris, 2013.

AUTHORS

Pierre-Marie Basset is a research engineer at Onera, The French Aerospace Lab, working in the field of rotorcraft flight dynamics since 1992. He is a former pilot student of the French Air Force Academy and got his Ph.D in 1995 on the topic of "Helicopter flight dynamics modeling and simulation". His research activities include also Rotary-Wings UAVs, new rotorcraft concepts and the development of preliminary design methodologies. He is the initiator and leader of the CREATION project.

Arnault Tremolet after education in Space Engineering achieved his PhD Thesis at Onera in 2013. Integrated in the Department of Systems Control and Flight Dynamics (DCSD), he contributed to the CREATION project establishing several level 1 models and implementing disciplines in ModelCenter environment. He focused his works on the development of a multidisciplinary design optimization chain able to manage multiple objectives, variables and constraints and the analysis of results with decision aid methods.

Thierry Lefebvre, graduated from the "Ecole Nationale Supérieure d'Arts et Métiers"(ENSAM 2000) and the "Ecole Nationale Supérieure des Pétroles et Moteurs" (ENSPM 2002). He joined Onera in 2002, as a research engineer in the Applied Aerodynamics Department (DAAP), working on the optimisation of tilt-rotor blades within EU projects. In 2008 he moved to the System Design and Performance evaluation department (DCPS), to focuse on advanced aircraft design methods, including Multidisciplinary Optimization.

S. Bertrand, J. Marzat,
H. Piet-Lahanier, A. Kahn,
Y. Rochefort
(Onera)

E-mail: sylvain.bertrand@onera.fr

DOI : 10.12762/2014.AL08-11

MPC Strategies for Cooperative Guidance of Autonomous Vehicles

This paper presents a comprehensive framework for the cooperative guidance of a fleet of autonomous vehicles, relying on Model Predictive Control (MPC). Solutions are provided for many common problems in cooperative control, namely collision and obstacle avoidance, formation flying and area exploration. Cost functions of the MPC strategy are defined to ensure a safe collaboration between the vehicles for these missions. An efficient way to select the optimal cost with limited computation time is also provided. The performance of the proposed approach is illustrated by simulation results.

Introduction

Missions such as large area surveillance or multiple target tracking may often prove tedious, potentially dangerous or cumbersome for a human operator. Using autonomous, or at least partly autonomous, vehicles could greatly contribute to making these missions feasible. However, their complexity may prove very demanding in terms of technological requirements for a single vehicle. Splitting the task into several subtasks makes it easier to fulfill the demands. It is thus necessary to determine how the subtasks are defined and to which vehicles they are attributed. Two approaches can be defended. The first one consists in defining subtasks that do not interfere, but whose collection leads to mission achievement. The second one is aimed at defining imbricated subtasks whose coordinated achievements would be at least equal, but possibly greater, than those of a complex but unique vehicle. This approach, known as cooperative tasking, requires coordination of the entire vehicle set to guarantee the satisfaction of the initial mission needs. The determination of specific control laws and estimation procedures are required to enable vehicles to perform cooperative tasks. This field of research has been very active since the 1980's and encompasses theories from various domains, such as game theory, artificial intelligence or distributed control. The numerous existing approaches vary according to the type of mission and the associated requirements in terms of constraints on formation flying [1], [2], communication exchange [3] or allocation of resources [4]. Many issues must be addressed, from the modeling of the cooperative set of vehicles, the definition and management of the allocated tasks and required information, and the definition of cooperative control strategies enabling the coordination and safety of these vehicles.

In this paper, attention is focused on the design of control laws, assuming that the necessary shared information is available. Various characteristics can be attached to cooperation control from a system-control perspective, e. g., they can be designed as implicit or explicit, based on regulated or reactive control law, obtained by centralized or decentralized control, with equality or hierarchy among the vehicles.

Implicit cooperation describes the behavior of individuals obeying a set of basic rules that results in a cooperative behavior. This constitutes one of the basic features of biomimetic flocking [5]. In explicit cooperation, mission allocation and guidance laws are defined for enhancing cooperative behavior (see e.g. [6]).

Reactive or regulated control laws translate into a long-term versus short-term design of the guidance law. A regulated control law is designed to guide the vehicles along trajectories that have been previously designed, e. g., using path searching procedures, such as A star or Dijkstra [7] or other algorithms [8]. Reactive control is achieved using the current and predicted states of the system, including the vehicles and the environments [9]. It is designed to provide a trade-off between the mission objective and its safety, for the current time or a limited time horizon. Centralized or distributed cooperative control refers to the location where the control takes place. It can be achieved within a unique control unit interacting with all vehicles [10] or computed by each vehicle [11]. The latter implementation presents the advantage of being more robust in case of failure of one of the vehicles of the fleet, at the cost of increasing the amount of vehicle embedded computation load. The selection of centralized or distributed control is also linked to the definition of hierarchy among the vehicles. They can be considered as equal in terms of decision for allocation or objective making, or some vehicles can obtain a superior status providing them with a higher decision weight. As in centralized computation, this hierarchy lightens the computational burden required for each vehicle of the fleet, at the cost of making the fleet more vulnerable to potential failure of the privileged vehicles.

This paper focuses on the design of reactive and distributed cooperative control laws. It can be addressed by designing a common criterion reflecting the mission objectives in terms of aim, safety assessment and so on. This criterion is evaluated according to each vehicle action and result, taking also into account the interaction between vehicles. Control laws can be thus derived by optimization of this criterion, relying on approaches such as model predictive control (MPC). This method has been used at Onera to define cooperative guidance laws for safely performing various cooperative missions with a fleet of autonomous Unmanned Arieal Vehicles (UAVs). The paper is organized as follows. In the next section, basic features of model predictive control are recalled and their extension to cooperative control is presented. Then the proposed MPC method is applied to the main types of cooperative missions for a fleet of autonomous vehicles (formation flight with obstacle avoidance, area exploration).

Model predictive control approach for UAV cooperative guidance

MPC has been widely used for the guidance of UAVs in various contexts. UAV flocking and formation flight has been discussed in [9]. In distributed MPC [12]–[14], each vehicle computes its control inputs at each timestep as a solution of an optimization problem over the future predicted trajectory. For tractability reasons, finite prediction and control horizon lengths, respectively denoted as H_p and H_c, are used.

The future control inputs and the resulting state trajectories of a vehicle i are written as

$$\mathbf{U}_i = ((\mathbf{u}_i(k))^T, (\mathbf{u}_i(k+1))^T, \ldots, (\mathbf{u}_i(k+H_c-1))^T)^T$$
$$\mathbf{X}_i = ((\mathbf{x}_i(k+1))^T, (\mathbf{x}_i(k+2))^T, \ldots, (\mathbf{x}_i(k+H_p))^T)^T$$

When $H_c < H_p$, we assume that the control inputs are 0 after H_c steps. Once the optimal input sequence $\mathbf{U}_i*$ has been computed, each vehicle communicates its predicted trajectory to the rest of the fleet and applies the first sample of the computed optimal control sequence $\mathbf{U}_i*(k)$. The optimization problems at time k take the following form:

$$\begin{aligned} &\text{minimize} && J_i(\mathbf{U_i}, \mathbf{X_i}) \\ &\text{over} && \mathbf{U_i} \in \mathcal{U}_i^{H_c} \\ &\text{subject to} && \forall t \in [k+1; k+H_p], \mathbf{x}_i(t) \in \mathcal{X}_i \end{aligned} \quad (1)$$

J_i is the cost function associated with vehicle i. The constraints coupling the dynamics of the vehicles, such as collision avoidance, are taken into account by means of a penalty factor in the cost function. At the next timestep, each vehicle searches for its solution of problem (1).

The cost function J_i is composed of a sum of terms reflecting the objectives of the mission. These terms are detailed in the following sections.

Vehicle model

The N UAVs are assumed to be identical. For the sake of simplicity, the UAVs are assumed to be pointwise and their trajectories are considered to be two dimensional, in a horizontal plane. Note that extension to 3D motion is straightforward.

The state and control vectors for each vehicle i are defined as:

$$\mathbf{x}_i = \begin{pmatrix} x_i \\ y_i \\ v_i \\ \chi_i \end{pmatrix} \text{ and } \mathbf{u}_i = \begin{pmatrix} u_i^{\omega} \\ u_i^{v} \end{pmatrix} \quad (2)$$

where x,y is the vehicle position, v is its speed amplitude and $\mathcal{X}$ is its direction. The model dynamics are:

$$\begin{cases} x_i(k+1) = x_i(k) + \Delta t.v_i(k)\cos\chi_i(k) \\ y_i(k+1) = y_i(k) + \Delta t.v_i(k)\sin\chi_i(k) \\ v_i(k+1) = v_i(k) + \Delta t.u_i^v(k) \\ \chi_i(k+1) = \chi_i(k) + \Delta t.u_i^{\omega}(k) \end{cases} \quad (3)$$

where Δt is the sampling timestep and (u_i^v, u_i^w) are the longitudinal and rotational accelerations. The constraints on the dynamics (3) and the control inputs are:

$$\begin{aligned} v_{min} \le v_i \le v_{max} \qquad & -\omega_{max} \le \omega_i \le \omega_{max} \\ -\Delta v_{max} \le u_i^v \le \Delta v_{max} \qquad & -\Delta\omega_{max} \le u_i^{\omega} \le \Delta\omega_{max} \end{aligned} \quad (4)$$

We summarize the dynamics and the constraints as $\mathbf{x}_i(k+1) = f(\mathbf{x}_i(k), \mathbf{u}_i(k))$ and $(\mathbf{x}_i, \mathbf{u}_i) \in \mathcal{X}_i \times \mathcal{U}_i$. Communication delays and ranges are not considered here, all of the UAVs are assumed to have access without delay to the exact state of every vehicle at all times.

Costs used for all types of missions

The cost function J_i is composed of a navigation cost J_i^{nav}, a safety cost J_i^{safety} and a control cost J_i^u:

$$J_i(k) = J_i^{nav}(k) + J_i^{safety}(k) + J_i^u(k) \quad (5)$$

The formulation of each cost function is presented in the following subsections.

Navigation cost

The navigation cost J_i^{nav} is aimed at regulating the speed of the vehicles and controlling the way in which they navigate to way-points. It is divided into five cost functions:

$$\begin{aligned} J_i^{nav}(k) = J_i^{nav,horiz}(k) + J_i^{nav,vert}(k) + J_i^{nav,direct}(k) \\ + J_i^{nav,final}(k) + J_i^{nav,fleet}(k) \end{aligned} \quad (6)$$

The first two cost functions, $J_i^{nav,horiz}$ and $J_i^{nav,vert}$, respectively defined by (7) and (8), are aimed at regulating the modulus of the horizontal component $\mathbf{v}^h$ of the velocity around a nominal value v_n and the vertical component v^z of the velocity around a zero value (making the UAV fly at constant altitude in a 3D case).

$$J_i^{nav,horiz}(k) = W^{nav,horiz} \cdot \sum_{n=k+1}^{k+H_c} \left(\left\| \widehat{\mathbf{v}_i^h}(n|k) \right\| - v_n \right)^2 \quad (7)$$

$$J_i^{nav,vert}(k) = W^{nav,vert} \cdot \sum_{n=k+1}^{k+H_c} \left\| \widehat{v_i^z}(n|k) \right\|^2 \quad (8)$$

Weighting coefficients $W^{\bullet}$ are tuned to set relative priorities between each aspect of the mission. A method to tune these coefficients has been proposed in [14].

The following two cost functions, $J_i^{nav,direct}$ and $J_i^{nav,final}$, are used to make the vehicle fly along a straight-line reference trajectory oriented toward the next way-point and to drive it closer to this way-point. The reference trajectory from the current position $\mathbf{p}_i(k)$ of vehicle i at time k to the next way-point p is composed of reference points $\mathbf{p}_{i,p}^{ref}(n|k)\left(n \in [k+1, k+H_p]\right)$ located at positions that vehicle i would reach at timestep n if moving along a straight line to p at nominal velocity v_n, regardless of any constraints. These reference points are defined by (9) and illustrated in figure 1. The resulting definition of cost $J_i^{nav,direct}$ is given by (10).

$$\mathbf{p}_{i,p}^{ref}(n|k) = \mathbf{p}_i(k) + (n-k).\Delta t.v_n.\frac{\mathbf{p}_i(k) - \mathbf{p}_p}{\left\|\mathbf{p}_i(k) - \mathbf{p}_p\right\|} \forall n \in [k+1, k+H_p] \quad (9)$$

$$J_i^{nav,direct}(k) = W^{nav,direct} \cdot \sum_{n=k+1}^{k+H_p} \left\| \widehat{\mathbf{p}_i}(n|k) - \mathbf{p}_{i,p}^{ref}(n|k) \right\|^2 \quad (10)$$

In order to steer the vehicle toward the next way-point by the end of the horizon of prediction, let us define a reference ball $B_{i,p}^{ref}(k)$, illustrated in figure 1, as the smallest ball around way-point p that vehicle i can hypothetically reach from its current position by moving directly toward this way-point at nominal velocity v_n. It is defined as $B_{i,p}^{ref}(k) = \left\{\mathbf{x} \middle\| \left\|\mathbf{x} - \mathbf{p}_p\right\| \le r\left(B_{i,p}^{ref}(k)\right)\right\}$, where the radius $r\left(B_{i,p}^{ref}(k)\right)$ is given by:

$$r\left(B_{i,p}^{ref}(k)\right) = \begin{cases} 0 & \text{if } d_{ip}(k) \le H_p.\Delta t.v_n \\ d_{ip}(k) - H_p.\Delta t.v_n & \text{otherwise} \end{cases} \quad (11)$$

where $d_{ip}(k) = \left\|\mathbf{p}_p - \mathbf{p}_i(k)\right\|$ denotes the current distance between vehicle i and the way-point p. Using these definitions, the cost $J_i^{nav,final}$ is defined by

$$J_i^{nav,final}(k) = W^{nav,final} \cdot \left(\left\| \widehat{\mathbf{p}_j}(k+H_p \mid k) - \mathbf{p}_p \right\| - r\left(B_{i,p}^{ref}(k)\right) \right)^2 \quad (12)$$

Finally, the fifth cost function $J_i^{nav,fleet}$ is aimed at making the vehicles remain together as a fleet. Its definition penalizes the predicted distance $\hat{d}_{ij}(n|k) = \left\|\hat{\mathbf{p}}_j(n|k) - \hat{\mathbf{p}}_i(n|k)\right\|$ between vehicles i and $j(i \ne j)$:

$$J_i^{nav,fleet}(k) = W^{nav,fleet} \times \sum_{\substack{j=1 \\ j \ne 1}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1 + \tanh\left(\left(\widehat{d_{ij}}(n|k) - \beta_{ij}^f\right).\alpha_{ij}^f\right)}{2} \quad (13)$$

where coefficients α_{ij}^f and β_{ij}^f are defined by

$$\alpha_{ij}^f = 6.\left(d_{loss}^v(ij) - d_{des}^v(ij)\right)^{-1} \quad (14)$$

$$\beta_{ij}^f = \frac{1}{2}.\left(d_{loss}^v(ij) + d_{des}^v(ij)\right) \quad (15)$$

The coefficient $d_{des}^v(ij)$ defines a desired distance between the vehicles within the fleet, whereas $d_{loss}^v(ij)$ is the maximum distance allowed between vehicles of the fleet. Vehicles $j(i \ne j)$ beyond this maximum distance are not considered by vehicle i any more. This represents, for example, limited communication and/or sensing ranges.

The definitions of α_{ij}^f and β_{ij}^f have been chosen in such a way as to obtain a nearly constant cost for distances lower than $d_{des}^v(ij)$ or greater than $d_{loss}^v(ij)$ (defined by a derivative lower than 0.05) and a symmetric behavior at borders.

The change in $J_i^{nav,fleet}$ with respect to the distance d_{ij} is plotted in figure 2.

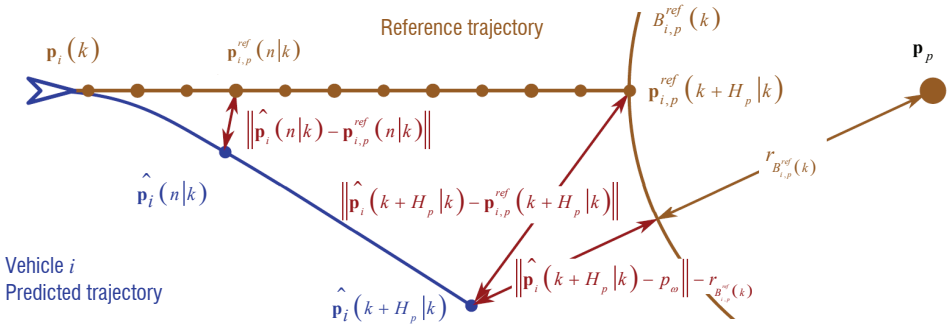


Fig. 1 - Representation of reference points $\mathbf{p}_{i,p}^{ref}$ and reference ball $B_{i,p}^{ref}$

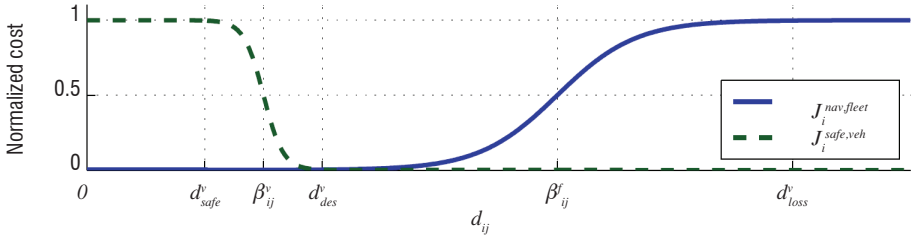


Fig. 2 - Flocking ($J_i^{nav,fleet}$) and avoidance ($J_i^{safe,veh}$) costs

Safety cost

The safety cost J_i^{safety} is aimed at avoiding collisions with obstacles and between vehicles within the fleet. It is composed of three cost functions:

$$J_i^{safety}(k) = J_i^{safe,veh}(k) + J_i^{safe,traj}(k) + J_i^{safe,obs}(k) \quad (16)$$

The first two costs deal with collision avoidance between vehicles, by respectively penalizing the predicted distance d_{ij} between them and ensuring that the new predicted trajectory $\widehat{\mathbf{p}}_i\left(\begin{smallmatrix}k+H_p-1\\k+1\end{smallmatrix}|k\right)$ of vehicle i remains close to the one transmitted to other vehicles at the previous iteration $\widehat{\mathbf{p}}_i\left(\begin{smallmatrix}k+H_p-1\\k+1\end{smallmatrix}|k-1\right)$:

$$J_i^{safe,veh}(k) = W^{safe,veh} \times \sum_{\substack{j=1\\ j\neq 1}}^{N} \sum_{n=k+1}^{k+H_p} \frac{1-\tanh\left(\left(\widehat{d}_{ij}(n|k) - \beta_{ij}^v\right).\alpha_{ij}^v\right)}{2} \quad (17)$$

$$J_i^{safe,traj}(k) = W^{safe,traj} . \sum_{n=k+1}^{k+H_p-1} \left\| \widehat{\mathbf{p}}_i(n|k) - \widehat{\mathbf{p}}_i(n|k-1) \right\|^2 \quad (18)$$

The shape parameters of the hyperbolic tangent function of equation (17), α_{ij}^v and β_{ij}^v, are defined by

$$\alpha_{ij}^v = 6.\left(d_{des}^v(ij) - d_{safe}^v(ij)\right)^{-1} \quad (19)$$

$$\beta_{ij}^v = \frac{1}{2}.\left(d_{des}^v(ij) + d_{safe}^v(ij)\right) \quad (20)$$

where $d_{safe}^v(ij)$ represents a desired safety distance between the vehicles. These shape parameters have been tuned according to the same considerations as previously explained for the definition of the safety cost. The change in $J_i^{safe,veh}$ with respect to the distance d_{ij} is plotted in figure 2.

The third cost $J_i^{safe,obs}$ penalizes the predicted distance $\widehat{d}_{io}$ of vehicle i to any obstacle o. It is defined as:

$$J_i^{safe,obs}(k) = W^{safe,obs} \times \sum_{o=1}^{N^o} \sum_{n=k+1}^{k+H_p} \frac{1-\tanh\left(\left(\widehat{d}_{io}(n|k) - \beta_{io}^o\right).\alpha_{io}^o\right)}{2} \quad (21)$$

where N^o stands for the number of obstacles and the parameters α_{io}^o and β_{io}^o are given by

$$\alpha_{io}^o = 6.\left(d_{des}^o(io) - d_{safe}^o(io)\right)^{-1} \quad (22)$$

$$\beta_{io}^o = \frac{1}{2}.\left(d_{des}^o(io) + d_{safe}^o(io)\right) \quad (23)$$

where $d_{des}^o(io)$ and $d_{safe}^o(io)$ are desired and safe distances to obstacles.

Control cost

As traditionally defined in MPC, the control cost $J_i^u(k)$ is aimed at limiting the control effort and thus the energy consumption of vehicle i. It is defined by the following quadratic form:

$$J_i^u(k) = \sum_{n=k+1}^{k+H_c} \mathbf{u}_i(n|k)^T \begin{bmatrix} W^{u,v} & 0 \\ 0 & W^{u,\omega} \end{bmatrix} \mathbf{u}_i(n|k) \quad (24)$$

Online computation of best cost

The MPC optimization problem (1) is a constrained nonlinear program, the solution of which cannot be found analytically. Numerical optimization must hence be used to approximate the solution.

Global optimization procedures based, for example, on interval analysis [15] or genetic algorithms [16] can be used, but may in practice be computationally prohibitive for real-time implementation. Numerical optimization methods, such as Sequential Quadratic Programming (SQP), Active Set or Interior Point methods, are thus generally preferred [17], [18]. Other methods suitable for MPC problems have also been developed [19]. Nevertheless, a global solution can be hard to find because of potential local minima. The computational time required for a MPC approach strongly depends on the parameterization of the control sequence. Low dimensional parameterizations have, for example, enabled successful applications to control systems with fast dynamics [20], [21]. Another solution consists in considering a finite set of predefined feasible control sequences, from which the one minimizing the cost function will be selected [22]. This last solution is used in this paper for implementation of the MPC strategy, based on [14].
This systematic search strategy has several main advantages over a traditional optimization procedure. Firstly, the computation load necessary to find a control sequence is constant in all situations leading to constant computation delay. The second advantage is that the systematic search strategy can be less sensitive to local minima problems, since the entire control space is explored. Finally, the systematic search requires no initialization of the optimization procedure.

The studied search procedure consists in defining, prior to the mission, a set S of candidate control sequences that satisfy control constraints (4). At each timestep, the control problem (1) is solved using the proposed search procedure, as follows:

- using a model of the vehicle dynamics, predict the effect of each control sequence of the set of candidates S on the state of the vehicle;
- remove from S all of the candidate control sequences that lead to a violation of constraints on the state of the vehicle (4);
- compute the cost J_i corresponding to each remaining candidate control sequence;
- select the control sequence that entails the smallest cost.

Since all of the candidates in the set S will be evaluated, the computation load of associated predictions should be as limited as possible. A simple parameterization of the control sequence is therefore adopted, by considering a control input constant over the control horizon H_c and then null over the remainder of the prediction horizon H_p. In addition, the distribution of the candidate control sequences is chosen so as to limit their number, while providing a good coverage of the control space.

The following three rules have been chosen:

- the set S of candidates includes the extreme control inputs, to exploit the full potential of the vehicles;
- the set S of candidates includes the null control input, to allow the same angular and linear velocities to be continued with;
- candidates are distributed over the entire control space, with an increased density around the null control input.

Constraints on the control inputs (4) can be translated into constraints on the norm of the horizontal component $\left(\left\|\mathbf{a}^h\right\| \le a^h_{\max}\right)$ and on the vertical component $\left(-a^z_{\max} \le a^z \le a^z_{\max}\right)$ of the acceleration of the vehicle. Therefore, it has been chosen to define the set S in terms of accelerations as follows:

$$S = \left\{\left\{S^\omega \times S^v\right\} \cup (0,0)\right\} \times S^z \tag{25}$$

where S^ω, S^v and S^z are respectively the sets of directions, modules and vertical components of the acceleration, defined by:

$$S^\omega = \left\{\frac{2.\pi.p}{\eta^\omega}\right\} \text{ with } p = 1 \text{ to } \eta^\omega \tag{26}$$

$$S^v = \left\{\frac{a^h_{\max}}{\left(\zeta^v\right)^p}\right\} \text{ with } p = 0 \text{ to } \eta^v \tag{27}$$

$$S^z = \left\{\pm\frac{a^z_{\max}}{\left(\zeta^z\right)^p}\right\} \cup \{0\} \quad \text{with } p = 0 \text{ to } \eta^z \tag{28}$$

ζ^v and ζ^z control the interval between two candidates; the number of candidates N^c, N^ω, N^v, and N^z in S, S^ω, S^v, and S^z respectively are deduced from η^ω, η^v, and η^z using (29) to (32).

$$N^c = \left(N^\omega.N^v + 1\right).N^z \tag{29}$$

$$N^\omega = \eta^\omega \tag{30}$$

$$N^v = \eta^v + 1 \tag{31}$$

$$N^z = 2.\eta^z + 1 \tag{32}$$

The resulting complete set S is illustrated in figure 3. Using this set S, a vehicle can be aimed at any arbitrary way-point in some iterations. The minimal distance at which the vehicle can approach the way-point depends on the precision of the control, defined by the values of the ζ• and η• parameters.

Applications

Applications of the MPC strategy for three different types of missions are proposed in this section. The first one concerns the guidance of a fleet of quadrotor UAVs toward given objectives represented by way-points, while avoiding collisions with obstacles and between vehicles [14]. Exploration missions are then addressed by a cooperative grid allocation approach [23] and a cost-oriented approach [24]. The third type of mission is formation flying, for which an adaptable virtual structure is proposed along with the MPC approach [25].

Guidance of a fleet toward predefined objectives

Many missions consist in making a group of several autonomous vehicles successively reach predefined objectives. These objectives may be defined in terms of a sequence of way-points, toward which the group of vehicles must be guided. Cooperation hence consists in sharing information (predicted trajectories) and controlling each vehicle of the group in such a way that the fleet as a whole can safely reach each of the waypoints.

The proposed cooperative MPC scheme is illustrated in this section for such a mission, where a group of $N=7$ vehicles must successively reach three way-points while avoiding collisions with obstacles and within the fleet. The vehicles considered are quadrotor UAVs, for which true dynamics include an inner loop for attitude control. Robustness to model mismatch of the MPC guidance strategy is therefore evaluated, since the prediction model used in the MPC guidance scheme consists in a 3D extension of (3). The quadrotor model and simulation results are presented in the next paragraphs.

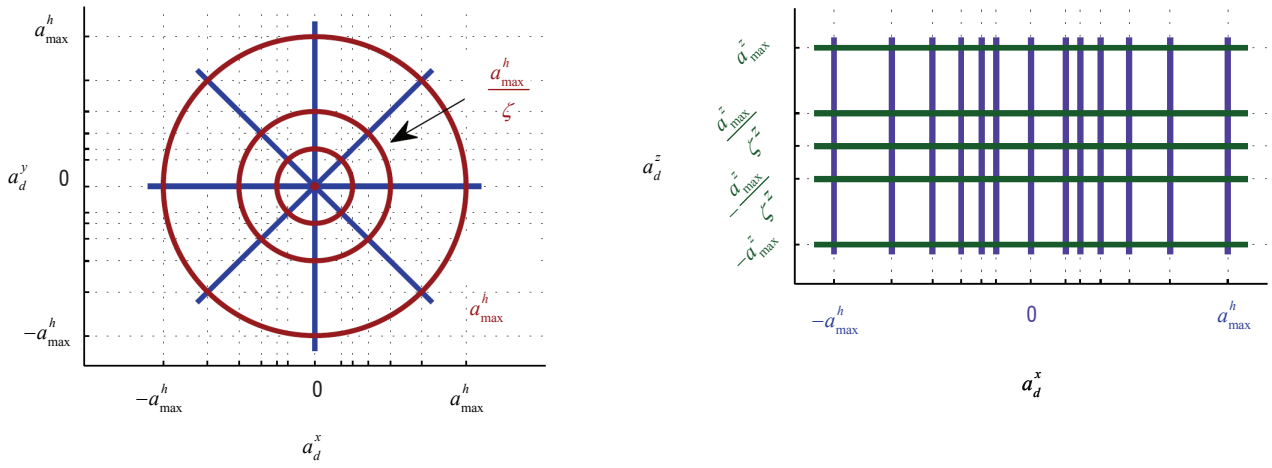


Fig. 3 - Illustration of the set S of candidate control sequences, defined in terms of accelerations $\left(\zeta^v = 2,\ \zeta^z = 3,\ N^\omega = 8, N^v = 3 \text{ and } N^z = 5\right)$ - Projections on the (x,y)-plane (left picture) and on the (x, z)-plane (right picture)

Quadrotor model and control strategy

Four-rotor helicopter models are derived from the description given in [26].

The rotors of each quadrotor i are located at the four corners of a square, with opposite rotors rotating in the same direction and adjacent rotors rotating in opposite directions. The true control inputs of the rotors are the signals u_i^1 to u_i^4 , classically defined as:
u_i^1 : the resulting thrust of the four rotors (which controls the motion along the z-axis of the vehicle);
u_i^2 : the difference of thrust between the left and right rotors (which controls the roll φ_i and hence contributes to the motion along the y-axis of the vehicle);
u_i^3 : the difference of thrust between the front and back motors (which controls the pitch θ_i and hence contributes to the motion along the x-axis of the vehicle);
u_i^4 : the difference of torque between the clockwise and anti-clockwise rotating rotors (which controls the yaw ψ_i of the vehicle).

It is assumed that, at each iteration k, the value of the state composed of the position, attitude angles, and the linear and angular velocities, is available for the computation of vehicle i control. The control strategy consists in applying the MPC guidance law based on a simplified 3D version of the prediction model (3), described in § "Vehicle model", to compute a desired acceleration vector $\mathbf{a}_{d,i}(k) = \left[a_{d,i}^x(k)\ a_{d,i}^y(k)\ a_{d,i}^z(k)\right]^T$.

This desired acceleration, along with a given desired value $\psi_{d,i}$ for the yaw, is then converted into vehicle control inputs (k) to $u_i^4(k)$. Note that attitude control of the quadrotor is achieved by using the approach proposed in [26].

Illustrations of the response of the controlled vehicle to an acceleration step in the x direction and to a yaw angle step are given in figure 4. The desired value is shown as a red dotted line, whereas the simulated response is shown as a plain blue line.

Mission set-up and tuning parameters

In the simulated mission, the flock must successively reach three way-points, while avoiding obstacles and collisions. The vehicles must also travel as a group at nominal velocity $v_n = 2$ m.s^{-1}. Defining the z-axis downwards, the coordinate of the ground is $z = 0$ and the altitude of a vehicle is given by -z. At all times, the vehicles must fly between the altitudes of 0m (ground) and 25 m. These constraints are materialized with two obstacles of infinite dimensions, with a vertical safety distance of 2 m.

Initial positions of the vehicles are randomly chosen in the box defined by $x \in$[-205 m; -155 m], $y \in$[-45 m; 5 m], $z \in$[-15 m; -5 m]. Their initial velocities, attitudes and attitude derivatives are set to zero and the desired yaw angle ψ_d is also set to zero for all vehicles throughout the entire mission. The guidance sampling time is $\Delta t = 0.5$ s.

Values of the constraints on velocity and acceleration are given in table I.

$v_{\max}^h$	5 m.s^{-1}
$v_{\max}^z$	1 m.s^{-1}
$a_{\max}^h$	0.5 m.s^{-2}
$a_{\max}^z$	0.25 m.s^{-2}

Table I - Velocity and acceleration constraints

Depending on the considered axis (x, y, z), different values can be assigned to distances $d_{safe}^v(ij)$, $d_{des}^v(ij)$ and $d_{loss}^v(ij)$ introduced in § "Navigation cost" and "Safety cost" and respectively defining the safety and desired distances between vehicles and the distance threshold beyond which other vehicles are no longer considered. Corresponding ellipsoids $\varepsilon_{safe}^o(i)$, $\varepsilon_{des}^o(i)$ and $\varepsilon_{loss}^v(i)$ are therefore designed, as illustrated in figure 5. Their parameterizations are given in table II. The same applies for the distances $d_{des}^o(io)$ and $d_{safe}^o(io)$ to any obstacle o, to which similar ellipsoids are associated $\left(\varepsilon_{safe}^o(o)\ \varepsilon_{des}^o(o)\right)$ and whose parameters are also given in table II. The parameters of the search procedure used for the MPC strategy (lengths of horizons, size of the sets of control sequences and shape parameters) are given in table III and, finally, the tuning parameters of the objective functions are given in table IV. Let us recall that they define the relative importance of each component of the mission. Note that the weighting parameters of the control cost are defined here in terms of the horizontal and vertical components of the desired acceleration $\mathbf{a}_{d,i}$, which is considered as the control input computed by MPC for this case study.

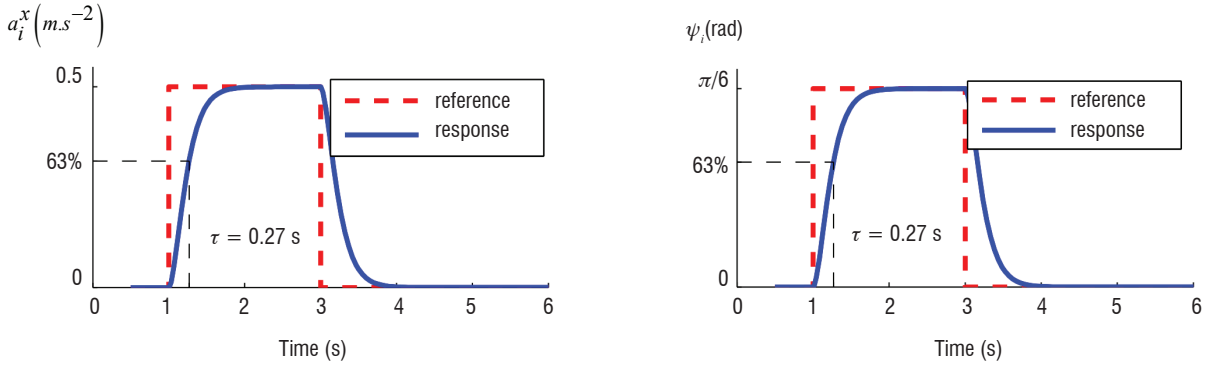


Fig. 4. Step response to a desired acceleration along x of 0.5m.s^{-2} (left picture) and to a desired yaw angle of $\pi/6$ rad (right picture)

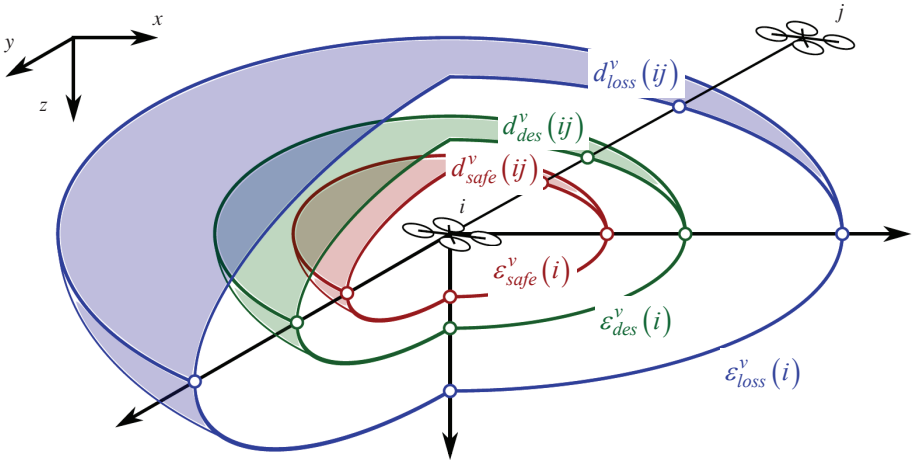


Fig. 5 - Safety $\left(\varepsilon^v_{safe}(i)\right)$, desired-locations $\left(\varepsilon^v_{des}(i)\right)$, and remoteness $\left(\varepsilon^v_{loss}(i)\right)$ ellipsoids around vehicle i

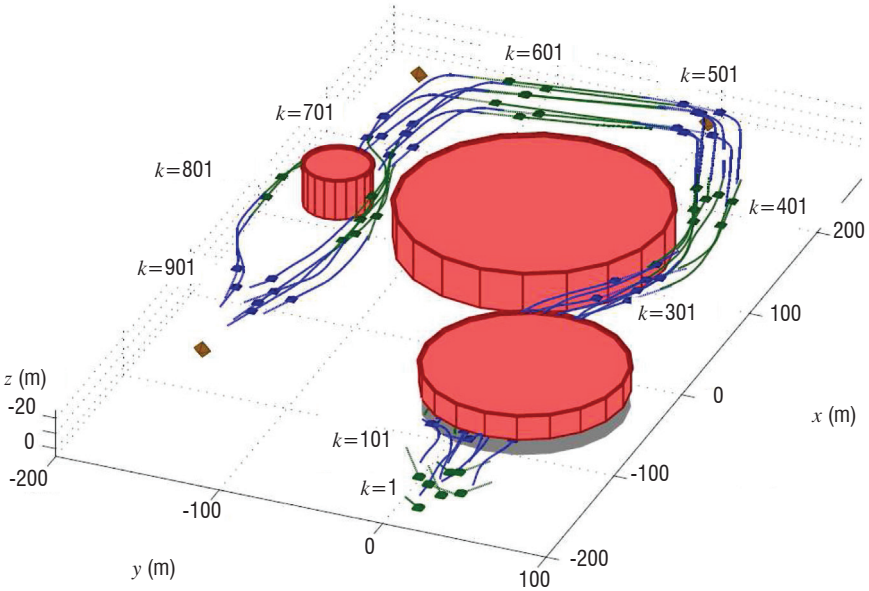


Fig. 6 - 3D view of the trajectories followed by the vehicles to complete their mission (way-points are represented by diamonds and obstacles by cylinders)

ellipsoid	semi-axis length along x	semi-axis length along y	semi-axis length along z
$safe$	10 m	10 m	5 m
ε^v_{des}	20 m	20 m	10 m
ε^v_{loss}	50 m	50 m	10 m
ε^o_{safe}	4 m	4 m	2 m
ε^o_{des}	8 m	8 m	4 m

Table II - Parameters of the ellipsoids defining the characteristic distances between vehicles and to obstacles

H_c	4	N^v	3
H_p	24	N^z	5
N^ω	8	ζ^v	2
N^c	125	ζ^z	3

Table III - Parameters of the search procedure

$W^{u,h}$	2	$W^{nav,horiz}$	10
$W^{u,z}$	2	$W^{nav,vert}$	2
$W^{safe,veh}$	100	$W^{nav,direct}$	10
$W^{safe,obs}$	400	$W^{nav,final}$	20
$W^{safe,traj}$	0	$W^{nav,fleet}$	50

Table IV - Weighting coefficients of the objective functions

Mission success rate	98.5%
Collision rate	0%
% of loss of a vehicle	1.5%
Mean computation time (std)	18(1) ms

Table V - Performance results of the MPC strategy over the 200 MC simulations

Simulation results and analysis

The environment of the mission and the trajectories of the vehicles are presented in figure 6 for one realization of the mission (i.e., for given initial positions of the vehicles). This realization of the mission is considered to be a success, because all way-points have been successfully reached and vehicles have remained grouped together, while avoiding collisions. The first obstacle is avoided by flying under it, whereas the two other obstacles are avoided by turning around. Note that since $\psi_d = 0$, all vehicles are oriented along the x axis (i.e., $\psi_i(k) = 0$ for all i and k).

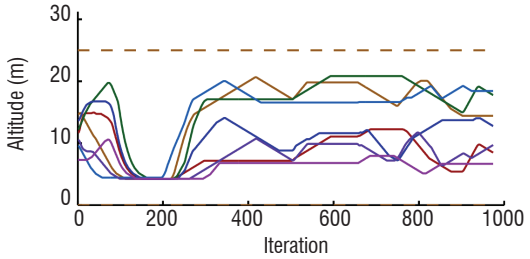


Fig. 7 - Change in the altitudes of the vehicles (the dotted lines represent the altitude constraints)

Altitude variations of the vehicles are presented in figure 7 and distances between the vehicles and to obstacles are presented in figure 8. As can be seen, the vehicles remained tightly grouped during the mission, except when they had to avoid obstacles, and spread over the vertical axis to form a tighter group while maintaining the desired distance between them. In addition, the vehicles always managed to avoid entering the safety ellipsoid of obstacles or other vehicles. Constraints on velocity and control inputs of the vehicles are satisfied throughout the mission, as can be seen in figures 9 and 10.

All of the aforementioned results concern one realization of the mission, i.e., one simulation corresponding to given initial positions of the vehicles. 200 Monte Carlo simulations have been run, randomly choosing these initial positions in the box defined by $x \in$ [-205 m; -155 m], $y \in$ [-45 m; 5 m], $z \in$ [-15 m; -5 m]. Table V provides the rate of success, collision and loss-of-a-vehicle (i.e., when the distance between two vehicles becomes greater than d^v_{loss}) over these 200 simulations. The mean value and standard deviation of the computation time (using Matlab on a standard PC) are also presented, over the 200 simulations. Note

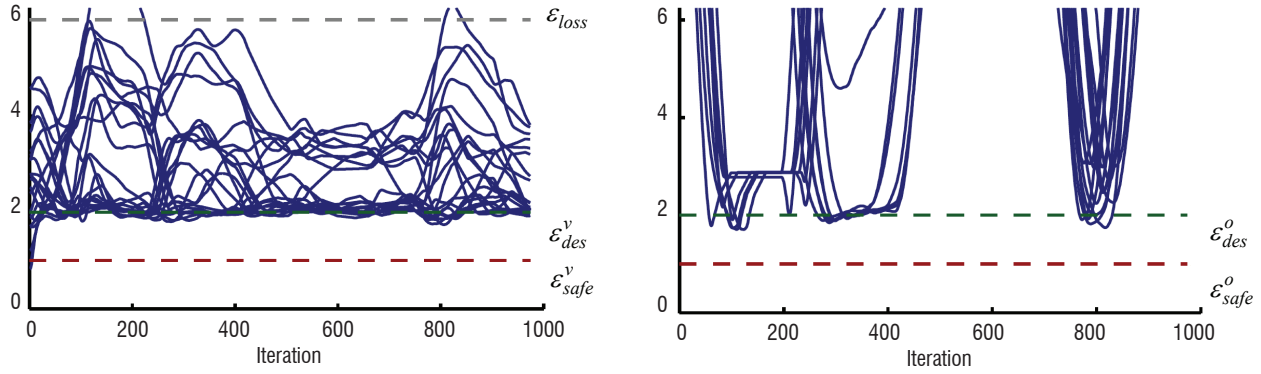


Fig. 8 - Distances between vehicles (left picture) and to each obstacle (right picture)

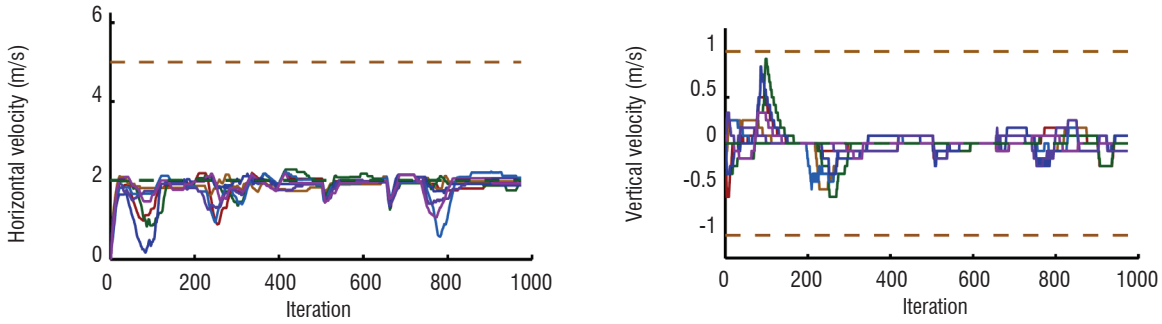


Fig. 9 - Constraints on the horizontal and vertical components of the velocity

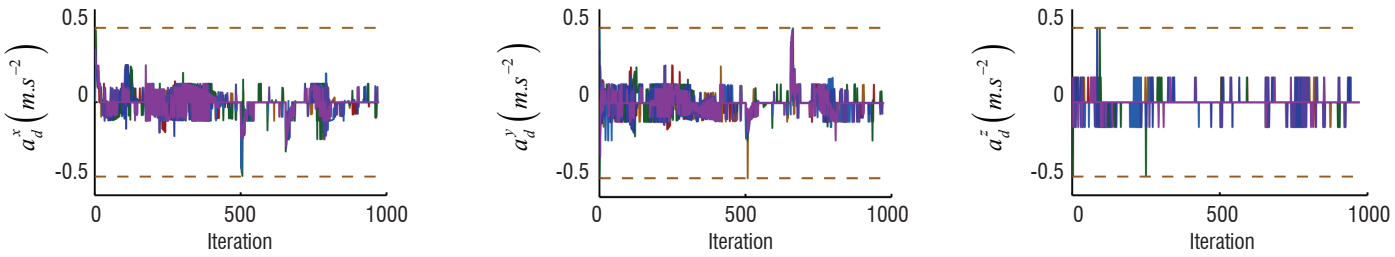


Fig. 10 - Components of the desired acceleration (control input)

that, for a given mission, the computation time remains constant throughout the whole mission.

Other studies on the influence of search parameters (number of control sequence candidates) and comparison to a traditional optimization procedure (SQP) have also been conducted [14], confirming the good performance and robustness of the proposed MPC strategy.

Area exploration via cooperative grid allocation

The cooperative exploration problem addressed in this section consists in zone surveillance. The zone is described by a value grid, as in [27] or [28]. Each point of the grid must be visited at least once by one of the vehicles belonging to the fleet.

Adaptation of navigation and objective criteria

The first step is to allocate the grid points among the vehicles. In order to maintain a distributed control design approach, the procedure suggested here consists in allocating to each vehicle two grid points selected at each iteration step among the remaining points, using the following selection criterion. This criterion, denoted $J^{attr}_{i,\{p_1,p_2\}}$, is built as the sum of the three following terms:

- J^{pec}_{i,p_1} estimates the maneuvering cost of the vehicle to reach the first grid point of the allocated pair,
- $J^{pec}_{i,p_2|p_1}$ estimates the additional maneuvering cost for the vehicle to reach the second grid point, while progressing to the first allocated point,
- $J^{dev}_{i,\{p_1,p_2\}}$ estimates the additional cost for the vehicles that were initially directed to p_1 or p_2 and have to be redirected as they are allocated to vehicle i.

The first component of the criterion J^{pec}_{i,p_1} is determined as a function of the distance between the first grid point and the predicted trajectory defined over the horizon of prediction. It is expressed as

$$J^{pec}_{i,p_1}(k) = \widehat{d^{\min}_{i,p_1}}(k) \tag{33}$$

$$\widehat{d^{\min}_{i,p_1}}(k) = \min_{n\in[k+1,k+H_p]} \left\| \widehat{\mathbf{p}}_i(n) - \mathbf{p}_{p_1} \right\| \tag{34}$$

The criterion $J^{pec}_{i,p_2|p_1}$ expression varies, whether the previous grid point p_1 is reached before, after half of the duration of the prediction horizon, or not reached at all. In the first case, $J^{pec}_{i,p_2|p_1}$ is defined as

$$J^{pec}_{i,p_2|p_1}(k) = \widehat{d^{\min}_{i,p_2}}(k) \tag{35}$$

In the second case, the former expression is transformed into

$$J^{pec}_{i,p_2|p_1}(k) = \left(1+\lambda_\theta.\widehat{\theta^{rel}_{i,p_2}}\left(k+H_p\right)\right).\widehat{d_{i,p_2}}\left(k+H_p\right) \tag{36}$$

where $\widehat{d_{i,p_2}}\left(k+H_p\right)$ is the distance between the predicted position of the vehicle at H_p and the second grid point, and $\widehat{\theta^{rel}_{i,p_2}}\left(k+H_p\right)$ is the variation of the predicted attitude of the vehicle $\widehat{\chi_i}\left(k+H_p\right)$ and the direction of the line of sight $\arg\left(\mathbf{p}_{p_2} - \widehat{\mathbf{p}}_i\left(k+H_p\right)\right)$.

In the latter case, the criterion is of similar form:

$$J^{pec}_{i,p_2|p_1}(k) = \left(1+\lambda_\theta.\widehat{\theta^{rel}_{i,p_2|p_1}}\left(k+\frac{H_p}{2}\right)\right).d_{p_1,p_2} \tag{37}$$

where d_{p_1,p_2} is the distance between p_1 and p_2.

The last component $J^{dev}_{i,\{p_1,p_2\}}$ of the cost $J^{attr}_{i,\{p_1,p_2\}}$ requires each vehicle to establish a list of the potential pair of allocated points, to cross-check with the other vehicles whether they are at risk of being deviated from their initial choice. The list contains the available pairs of grid points classified by decreasing order of sum of J^{pec}_{i,p_1} and $J^{pec}_{i,p_2|p_1}$.

Simultaneously to the selection of objectives, a navigation criterion must be computed. The functional developed for this purpose is derived from the cooperative MPC strategy presented in § " Guidance of a fleet toward predefined objectives". The differences between way-point guidance and way-point allocation are, first, that the way-points describing a grid zone are closer to each other than the way-points used to indicate a global path to the fleet and, second, that the trajectories are defined to be close to the grid points instead of being directed towards them. Hence, the navigation criterion $J^{nav,final}$ must be adapted accordingly. In this context, it depends on the minimal distance $\widehat{d^{\min}_{i,p_1}}(k)$ defined by (34) between predicted positions of the vehicle and the way-point that it must explore during the prediction horizon. The resulting expression of the criterion is

$$J^{nav,final}_i(k) = \begin{cases} W^{nav,final} \times \left(\widehat{d^{\min}_{i,p_1}}(k)\right)^2 \\ if\ d_{i,p_1}(k) \le H_p.\Delta t.v_n \\ W^{nav,final} \times \left(d\left(\widehat{\mathbf{p}}_i\left(k+H_p\right), B^{ref}_{i,p_1}(k)\right)\right)^2 \\ otherwise \end{cases} \tag{38}$$

Application example

Figure 11 presents the trajectories obtained for an exploration mission on a grid zone, realized by four vehicles. The duration of the mission is 400 s. At the end of the mission, all vehicles must reach an exit point of coordinates x =-200, y =250.

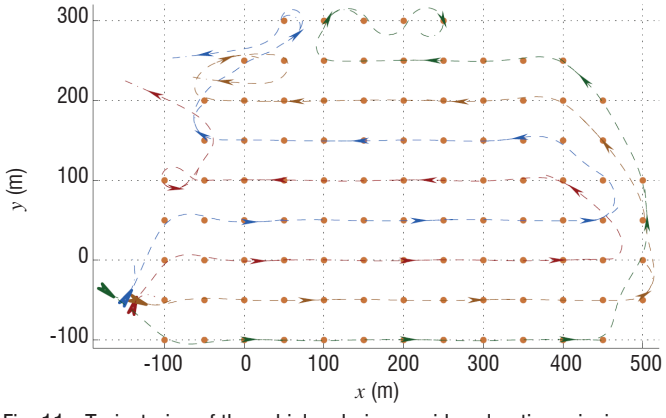


Fig. 11 - Trajectories of the vehicles during a grid exploration mission

It illustrates some of the effects of the approach described. At the beginning, the vehicles separate to reach a different row of grid points. While the number of remaining grid points remains high enough to limit multiple allocations, the vehicles follow straight lines. When allocation is sparse, the trajectories may present oscillations, since the evaluation and comparison of costs require increased communications and the updating of allocated grid points.

Area exploration via a cost-oriented approach

Zone watching is defined in this section as a cooperative problem, where a number of autonomous vehicles must explore a wide area in a limited amount of time without any way-point defined in advance. In addition to zone coverage, the dynamical allocation of exit locations is considered.

Each vehicle defines its own trajectory online to achieve the cooperative mission objectives:

- maximize the cumulated area covered,
- allocate and reach exit points at the end of the mission,

while respecting the constraints:

- collision avoidance,
- limited mission time,
- limited number of vehicles at a given exit.

The choice of optimal control entries should thus take into account four main aspects: collision avoidance, minimum control energy, map exploration and exit point assignment. The associated global cost function for this application is

$$J_i = J_i^{safety} + J_i^{u} + J_i^{expl} + J_i^{nav,direct} \tag{39}$$

The costs J_i^{safety} to avoid collisions and J_i^{u} to limit the energy spent by the vehicles are those defined respectively in § "Safety cost" and "Control cost". The cost J_i^{expl} is specific to the exploration problem considered (see next §). The cost $J_i^{nav,direct}$ (defined in § "Navigation cost") is used to guide the vehicle to its allocated exit point, which is computed online, as indicated in § "Exit point (re)allocation". A dynamic weight gives more importance to this last cost when the mission time approaches its limit (§ "Weighting of the functions").

Zone coverage

The cost function J_i^{expl} should reflect the gain in terms of map exploration for a potential trajectory. Each vehicle is assumed to have a seeker capability, described by a function f_{expl} of the relative position between the observed point and the vehicle.

The cooperatively explored area at time k is:

$$\Omega = \bigcup_{\substack{t=1,\dots,k \\ i=1,\dots,N}} \mathcal{D}_i(t) \tag{40}$$

where $\mathcal{D}_i(t)$ is the sensing footprint of vehicle i at timestep t. Since this representation is impractical, the mission field is approximated as a grid with spacing d_{grid}. A matrix G stores the level of exploration of each square of the grid. Each element $G_{l,m}$ (where l, m are the integer coordinates of the square in the grid) ranges between 0 when no vehicle has explored this location and 1 when it has been entirely observed. Each vehicle stores a copy of this exploration map and updates it with the information from the rest of the fleet. The precision of the representation only depends on the parameter d_{grid}. When a vehicle comes at a distance d from the center of square (l, m), the exploration level is updated as

$$G_{l,m}^{+} = \max\left(G_{l,m}, f_{expl}(d)\right)$$

The exploration index is increased only if the vehicle is close enough. The function f_{expl} is chosen to be continuous and identically 0 for $d > r_{sensor}$. Here,

$$f_{expl}(d) = \begin{cases} 0 & if\ d \geq r_{sensor} \\ \frac{1}{2}\left(1+\cos\frac{\pi d}{r_{sensor}}\right) & if\ d < r_{sensor} \end{cases}$$

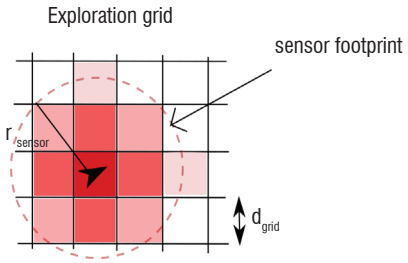


Fig. 12 - Illustration of exploration cost: colors reflect the exploration level

The cost function rewards trajectories that cooperatively increase the global level of exploration of the map. It is defined as

$$\begin{aligned} J_i^{expl} &= -W^{expl} \sum_{l,m} \left(G_{l,m}(k+H_p) - G_{l,m}(k)\right) \\ &= -W^{expl}\,\mathbf{1}^T.\left(G(k+H_p) - G(k)\right).\mathbf{1} \end{aligned} \tag{41}$$

where $G(k + H_p)$ is the predicted exploration map associated to the vehicle trajectory and $\mathbf{1}$ is the vector whose components are all 1. This cost function represents the total increase of the global exploration level resulting from a predicted trajectory. Since the vehicles share information, flying in already explored zones is therefore penalized.

Exit point (re)allocation

Two cases are studied. In the first one, the number of vehicles N is identical to the number of exits n_t and a given exit point c_j can shelter at the most one vehicle. In the second, the number of vehicles exceeds the number of exits and at most n_{max} vehicles can reach a given exit point.

Case $n_t = N$

The aim is to define a cost that balances the distance to the exit and the cost in the control inputs (in other words, penalizes deviations from the trajectory to an exit point c_j of position $\mathbf{p}_{cj}$). This cost will serve as a measure of the interest for a given vehicle to go to an exit and will support the decision of dispatching the vehicles to the exits. Therefore, it must discriminate efficiently between different vehicles aiming for the same exit.

For each pair {vehicle, exit point}, the cost function

$$J_{i,j}^{af}\left(\mathbf{u}_i(k), \mathbf{x}_i(k)\right) = \sum_{t=k}^{k+H_p-1} W^d \left\|\mathbf{p}_i(t) - \mathbf{p}_{cj}\right\|^2 \tag{42}$$

is minimized and an assignment cost matrix $R = \left(r_{ij}\right)_{\substack{i=1,\dots,N \\ j=1,\dots,n_t}}$ is obtained.

$$r_{ij} = \min_{\mathbf{u}_i} J_{ij}^{af}\left(\mathbf{u}_i, \mathbf{x}_i\right)$$

The optimal assignment is obtained by the Hungarian algorithm [29].

Case $n_t < N$

In this case, at the most n_{max} vehicles can go to the same exit point at the end of the mission. The $(N \times n_t)$ matrix R is built. A basic consensus mechanism is used to find a good admissible assignment: each vehicle forms a list of wishes based on its cost evaluations. These costs are centralized and whether no more than n_{max} vehicles aim for each exit is checked. In case of conflict, the admissible alternative exits are considered. The minimizing costs among these are chosen for each conflicting vehicle, consecutively.

In these two cases, the construction of the cost matrix is decentralized but information must be centralized to perform the actual assignment.

The vehicles are now able to explore a zone and reach an exit at the end of the allocated time. The final constraints on the positions require a terminal allocation at all times, merely to ensure satisfaction of the constraints on the maximal number of vehicles for each exit. Nevertheless, if enough time remains, the vehicles should focus on exploration. Therefore, the initial assignment could be reconsidered after some time: a reallocation of the vehicle may prove beneficial. One option is to repeat the assignment procedure presented in the previous subsection during the mission. However, it could lead to an undesirable situation where, in order to decrease the total cost, the global optimization assigns to a vehicle an exit that cannot be reached before the end of the mission. Consequently, a penalty linking the time needed to reach the exit and the remaining mission time is added. It is expressed as follows:

$$s_{ij} = \begin{cases} 0 & \text{if } T \geq T_{safe} \\ \left(\dfrac{T_{safe} - T}{T_{safe} - T_{danger}}\right)^2 & \text{if } T < T_{safe} \end{cases} \quad (43)$$

where T is the remaining time, $T_{danger} = \dfrac{\left\|\mathbf{p}_i - \mathbf{p}_{cj}\right\|}{v_n}$ and $T_{safe} = f_{safe} T_{danger} + T_{margin}$, $\mathbf{p}_{cj}$ is the position of the exit, and T_{margin} and f_{safe} are predefined parameters. The matrix used for the global assignment is $R' = \left(r'_{ij}\right)_{\substack{i=1,\dots N \\ j=1,\dots,n_t}}$ with $r'_{ij} = r_{ij} + s_{ij}$. The continuous variation of the penalty prevents vehicles from choosing unreachable exit points, provided that the reallocation is performed often enough. Repeating the allocation procedure frequently represents a large computational load, therefore instead of using the nonlinear dynamical model a simple linear model (double integrator) is used to approximate the dynamics. This does not significantly deteriorate the performance, because only estimates of the costs to go are required in order to choose a reasonable assignment. The linear approximation and the constraint translation is based on [13] and the reallocation can only be repeated at large time intervals.

Weighting of the functions

Each penalty function and its subcomponents are weighted with a coefficient $W\bullet = k\bullet . w\bullet$, with $k\bullet$ a normalization coefficient and $w\bullet$ a weighting coefficient. The $k\bullet$ (Table VI) coefficients are chosen so that without weighting, the worst case cost would be around 1.

k^{safety}	$\dfrac{2}{H_p}$	k^{expl}	$\left(\dfrac{2 r_{sensor} . v_n . H_p}{d_{grid}}\right)^{-1}$
k^{u}	$\dfrac{1}{\left(H_c . v_{\max} . \omega_{\max}\right)^2}$	k^{nav}	$\dfrac{1}{\overline{dist}^2}$

Table VI - Renormalization coefficients

Note that k^{nav} is chosen so that exit allocation can be fairly performed between the vehicles: $\overline{dist} = \dfrac{1}{n_t . N} \sum_{i=1}^{N} \sum_{j=1}^{n_t} \left\|\mathbf{p}_i - \mathbf{p}_{c_j}\right\|$ is the average distance between vehicles and exits.

Since the total time allocated for the mission is known, it is preferable to rejoin the exit point only when the vehicles run out of time. A dynamic weighting procedure is proposed: the exploration and exit rejoining costs are weighted with respect to the difference between the estimated time to reach the exit and the actual remaining time. A scheme based on [30] is adopted: the exploration of the map is initially favored in the cost function, whereas exit points progressively take more importance in the cost function as the remaining time decreases. This translates into the algorithms by means of balancing coefficients C^{expl}, C^{u} which multiply W^{expl}, W^{u} as reported in Algorithm 1.

Algorithm 1: Calculation of the weighting coefficients

1) Compute $d = dist(\mathbf{p}_i, \mathbf{p}_{ci})$ distance between vehicle i and its exit point.

2) Compute $T_{danger} = \dfrac{d}{v_n}$, the estimated minimal time to reach the exit assuming a straight path and nominal speed. Compute $T_{safe} = f_{safe} . T_{danger} + T_{margin}$, an overestimate of T_{safe} considered as comfortable to reach the exit.

3) Compute:

$$C^{expl} = \begin{cases} 0 & \text{if } T \leq T_{danger} \\ \dfrac{T - T_{danger}}{T_{safe} - T_{danger}} & \text{if } T_{danger} < T \leq T_{safe} \\ 1 & \text{if } T > T_{safe} \end{cases}$$

$$C^{u} = \begin{cases} 1 & \text{if } T \leq T_{danger} \\ \dfrac{C_{\min}\left(T - T_{danger}\right) + T_{safe} - T}{T_{safe} - T_{danger}} & \text{if } T_{danger} < T \leq T_{safe} \\ C_{\min} & \text{if } T > T_{safe} \end{cases}$$

Simulation results

Simulation parameters are grouped in table VII. The various requirements of the mission are first illustrated individually and quantitative simulation results are then given.

Collision avoidance and exit rejoining

The vehicles are positioned so that they have to cross paths to reach their exit points. No reallocation is allowed and map exploration is not taken into account. The dashed lines denote the past trajectories of the vehicles, whereas the dotted short lines depict the predicted trajectories at the current time. The circles denote the danger zones around the vehicles. Figure 13 shows that vehicles can reach agreements, even in complex situations, to cross ways without endangering themselves or the other vehicles.

v_{min}	0.3	v_n	0.7	v_{max}	1.0
$\Delta\omega_{max}$	0.1	Δv_{max}	0.1	ω_{max}	0.5
d^v_{safe}	4	d^v_{des}	8	w^{safety}	5
w^{tt}	0.5	$w^{nav,horiz}$	1	w^{expl}	2
w^{el}	2	f_{safe}	1.1	C_{min}	0.2
H_c	3	H_p	21	r_{max}	5
T_{margin}	15	f_{safe}	1.1	d_{grid}	2.5

Table VII - Simulation parameters for area exploration by cost-oriented approach

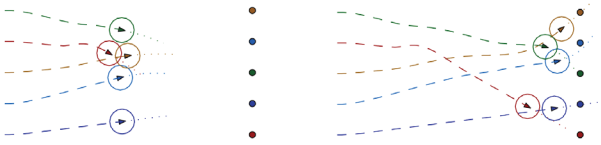
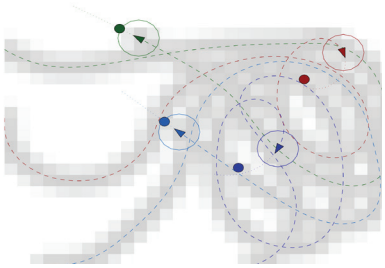
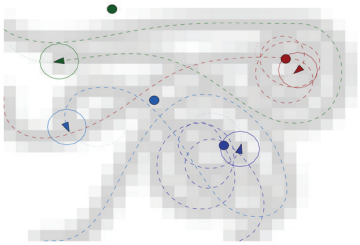


Fig. 13 - Illustration of the collision avoidance

Map exploration and exit assignment



(a) Weighting factors are dynamic: exploration is favored first and the exiting cost progressively prevails



(b) Weighting factors are constant

Fig. 14 - Comparison of different exploration strategies: the colors of the vehicles correspond to the assigned target

Map exploration and exit assignment are illustrated with a 4-vehicle scenario presented in figure 14. Exits are chosen randomly for each vehicle and no reallocation is allowed. It compares the behavior of the vehicles in two different settings: (a) dynamic weighting of exploration and exit assignment with respect to remaining time versus (b) constant weights. The main difference is that, in case (a), the vehicles can go far away from their exit point as long as time remains and, consequently, it is easier for them to find new zones to explore, while in case (b) vehicles tend to stay close to their exit point.

Dynamic reallocation

Dynamic reassignment is illustrated in figure 15. The current assignment in the figures is depicted by matching colors. In this particular instance, the vehicle beginning in the top left corner keeps his initial assignment during the mission, whereas the two others do not. One of them first changes its exit, whereas the last one also changes its plan later on.

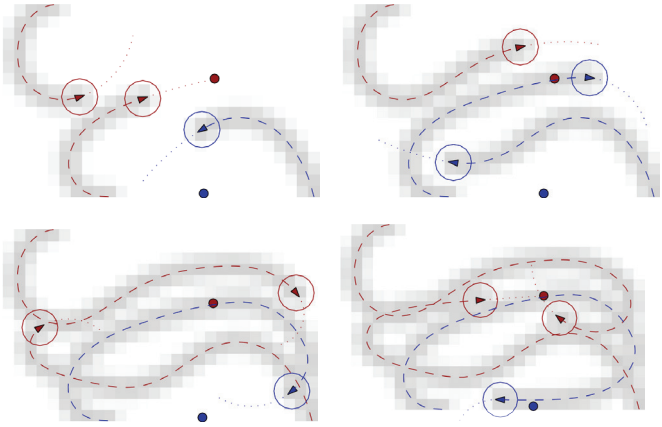


Fig. 15 - Online reassignment of the exit points: the colors of the vehicles correspond to the assigned target

Performances

To evaluate the performance of the strategies, a set of 70 missions was simulated with different configurations. The settings were:

- No exploration is considered: each vehicle chooses an exit and rejoins it as soon as possible (A)
- Exploration is considered, but weightings of exploration and exit rejoining in the cost function are fixed throughout the mission (B)
- Exploration is valued at the beginning of the mission and exit reaching progressively becomes the dominant cost (C)
- Configuration is identical to (C) but reassignment is granted (D)

In each mission, a 78 m x 78 m field is explored by 4 vehicles, with a mission time of 300s. The position of the vehicles and the 4 exit locations are chosen randomly for each run. The mission is performed for the 4 configurations and the results are given in table VIII. Expl(%) is the portion of the map that has been explored during the mission. It takes into account both the number of squares explored and their respective level of exploration. Danger(%) gives the ratio between the time during which a dangerous situation has occurred and the mission duration, that is, when 2 vehicles come closer than a distance of d^v_{safe} at some point during the mission. d_{exit} gives the average distance of the vehicles to their exit targets at the end of the mission. We can observe that map exploration costs allow a better exploration and dynamic weighting increases the efficiency of the exploration even more, as expected. However, it also

increases the number of dangerous situations: adding the exploration cost increases the chances of several vehicles coming into the same zone and therefore increases the collision risks. Furthermore, dynamical reallocation does reduce these risks significantly, while preserving the exploration efficiency. The results presented show that a global complex mission can be satisfactorily fulfilled by using a short-sighted and distributed control architecture. The proposed method renders the problem tractable and allows the actions to be taken online to be chosen.

	Expl (%)	danger (%)	$d_{exit}\,(m)$
A	20.9 (3.0)	1.4	4.2 (0.2)
B	44.4 (5.0)	5.7	8.4 (2.7)
C	58.3 (3.2)	11.4	6.2 (1.7)
D	59.1 (3.3)	8.5	6.5 (2.5)

Table VIII - Simulation results (standard deviations are given in brackets)

Formation flying using an adaptable virtual structure

Another guidance law is derived in this section to achieve formation flight toward a way-point for a fleet of autonomous vehicles. The formation is now defined by a virtual geometrical structure - here, an ellipse - that can modify its shape and orientation to avoid collision with obstacles in the environment. The proposed guidance law is divided into two layers, with a MPC scheme at each level. The higher layer controls the structure itself, to fulfill the goals and constraints of the required mission. The trajectory of the fleet is built on-line using this layer, as well as the adaptation of the structure to the environment. The lower layer controls the vehicles, so as to attract and keep them within the structure without side collision.

Virtual structure control

The first layer of the guidance law generates the change in the virtual structure and adapts its shape so that it does not collide with the obstacles on its way to a way-point. It has been chosen to describe the formation shape as an ellipse, represented only by its center and characteristic matrix (this description can be steadily extended to that of an ellipsoid in 3D).

Model of the virtual structure

An ellipse with center $\mathbf{p}_c = [x_c\ y_c]^T$ and characteristic matrix $\mathbf{M}$ is defined by all points $\mathbf{p} = [x\ y]^T$ such that

$$(\mathbf{p}-\mathbf{p}_c)^T\,\mathbf{M}^{-1}(\mathbf{p}-\mathbf{p}_c) \le 1 \tag{44}$$

The characteristic matrix M of the ellipse can be written as

$$\mathbf{M} = \begin{bmatrix} \cos\chi_e & -\sin\chi_e \\ \sin\chi_e & \cos\chi_e \end{bmatrix} \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \begin{bmatrix} \cos\chi_e & -\sin\chi_e \\ \sin\chi_e & \cos\chi_e \end{bmatrix}^T \tag{45}$$

where the parameters of the ellipse are:

- χ_e, the angle between the first principal axis and the horizontal;
- a, the length of the first principal axis of the ellipse;
- b, the length of the second principal axis of the ellipse;
- $\mathcal{A}$, the area of the ellipse, equal to πab.

The dynamical evolution of the ellipse is modeled by

$$\begin{cases} x_c(k+1) = x_c(k) + \Delta t.v_c(k).\cos(\alpha_c(k)) \\ y_c(k+1) = y_c(k) + \Delta t.v_c(k).\sin(\alpha_c(k)) \\ v_c(k+1) = v_c(k) + \Delta t.u_v(k) \\ \alpha_c(k+1) = \alpha_c(k) + \Delta t.u_\alpha(k) \\ \chi_e(k+1) = \chi_e(k) + \Delta t.u_X(k) \\ a(k+1) = a(k) + \Delta t.u_a(k) \\ b(k+1) = b(k) + \Delta t.u_b(k) \end{cases} \tag{46}$$

with

$$\begin{aligned} &u_{v-} \le u_v \le u_{v+},\ u_{\alpha-} \le u_\alpha \le u_{\alpha+} \\ &u_{\chi-} \le u_\chi \le u_{\chi+},\ u_{a-} \le u_a \le u_{a+}, \\ &u_{b-} \le u_b \le u_{b+} \end{aligned} \tag{47}$$

where x_c, y_c, χ_e, a, b are the aforementioned parameters of the ellipse, v_c is the speed of the center and α_c is its orientation (see figure 16). The control inputs u_v, u_α govern the movement of the center of the ellipse $\mathbf{p}_c$ by acting on its speed and angular velocity, while the control inputs u_χ, u_α; u_b modify the characteristic matrix $\mathbf{M}$ (shape and orientation).

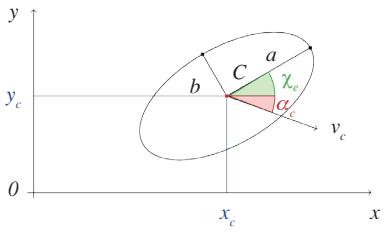


Fig. 16 - Ellipse parameterization

This dynamical model should be related to the dynamics of the UAVs so that it does not scatter the formation. The control inputs must thus be selected within a suitable range and suitable dynamics. This is yet a clear advantage over methods that abruptly modify the virtual structure and, as a result, do not take into account UAV constraints.

Guidance law design

The cost function J^z associated with the motion and shape of the virtual structure is composed of terms dealing with the mission objectives and the constraints on the structure itself. The optimal control inputs at time k should minimize the cost function J^z, such that

$$u_\chi^*, u_a^*, u_b^*, u_v^*, u_\alpha^* = \arg \min_{\substack{u_v, u_\alpha \\ u_\chi, u_a, u_b}} J^z \tag{48}$$

Where

$$J^z = J^{nav,direct} + J^{nav,horiz} + J^v + J^{abmin} + J^c \tag{49}$$

The components of J^z are designed in such a way that

- $J^{nav,direct}$ drives the ellipse to a way-point and $J^{nav,horiz}$ constrains its speed to a desired value (costs defined in § "Costs used for all types of missions" and applied here to the ellipse center and velocity);

• J^v keeps the ellipse area close to the initial one, $\mathcal{A}$;
• J^{abmin} maintains a and b greater than a boundary value, so as to avoid the flattening of the ellipse along one of its axes;
• J^c modifies the matrix $\mathbf{M}$ to avoid obstacles.

Costs related to ellipse constraints

$$J^v = w_v \sum_{t=1}^{H_p} \left\| \hat{a}(k+t)\hat{b}(k+1)\pi - \mathcal{A} \right\| \quad (50)$$

$$J^{abmin} = w_{abmin} \sum_{t=1}^{H_p} \left\{ f\left(\hat{a}(k+t), l_1, l_2\right) + f\left(\hat{b}(k+t), l_1, l_2\right) \right\} \quad (51)$$

where

$f(\hat{a}(k+t), l_1, l_2)$ is a continuous function, and l_1 and l_2 are user-defined parameters, such that
• f takes the value 1 when $\hat{a}(k+t) \leq l_1$,
• f takes the value 0 when $\hat{a}(k+t) \geq l_1 + l_2$,
• f undergoes a continuous change between these two extreme values.

For example, an appropriate choice for f is the function presented in figure 17, which is related to the one used in § "Applications".

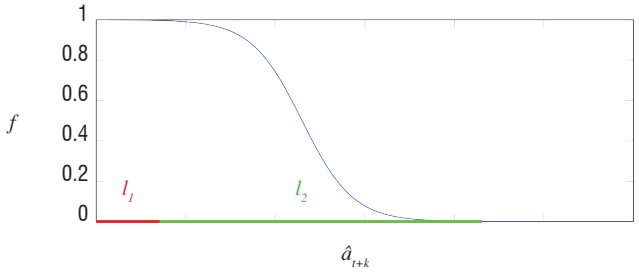


Fig. 17 - Function f

Costs related to obstacle avoidance

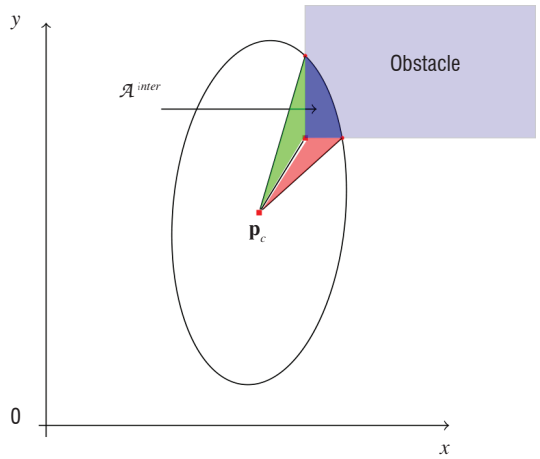


Fig. 18 - Ellipse-obstacle intersection

The structure should maneuver to avoid collision with obstacles in the environment. Assuming that the obstacles are described as convex surfaces (volumes in a 3D case), the intersection area between the virtual structure and the obstacles is computed to detect and quantify potential collisions (figure 18). Using this value as a penalization in the criterion makes it possible to find a path that minimizes this intersection and thus the risk of possible collision.

The collision avoidance term J^c uses the intersection area $\mathcal{A}_l^{inter}(k+t)$ at time $k+t$ for each obstacle l (N^o being the number of obstacles in the neighborhood). The weight is chosen to give greater importance to the first prediction steps rather than the future ones.

$$J^c = w_c \sum_{t=1}^{H_p} \sum_{l=1}^{N^\circ} \frac{H_p - 1}{H_p} . \mathcal{A}_l^{inter}\left(k+t\right) \quad (52)$$

UAV control

The UAV control layer computes the motion of each vehicle so that it remains within the ellipse and avoids collision with the other vehicles. It thus has three goals:
• to attract the UAV inside the area;
• to allocate each UAV inside the area;
• to avoid collision between UAVs (using the cost defined in § "Safety cost").

This control is decentralized (each UAV determines its own control inputs), yet it uses the prediction of the future state of the virtual structure, which is available using the developments from § "Virtual structure control". MPC is used again, since allocation and collision avoidance may benefit from a prediction of the impact of control inputs on the future states of the vehicles.

The fleet is still composed of N identical UAVs that are assumed to have instantaneous access to the state of all the other vehicles, which is $\mathbf{x}_i = [x_i, y_i, v_i, \chi_i]^T$ for the i-th vehicle.

For each UAV, the control inputs u_i^v and u_i^ω are determined at each time k in such a way that

$$u_i^{v*}, u_i^{\omega*} = \arg\min_{u_i^v, u_i^\omega} J_i \quad (53)$$

where for this application

$$J_i = J_i^{safe,veh} + J_i^u + J_i^t + J_i^e \quad (54)$$

The components of J_i are designed in such a way that
• $J_i^{safe,veh}$ modifies the direction and the speed to avoid collision with other UAVs (§ "Safety cost"),
• J_i^u minimizes the energy consumption in terms of control inputs (defined in § "Control cost"),
• J_i^t drives the UAV within the area,
• J_i^e keeps the speed and orientation of the UAV close to those of the center of the structure.

Attraction and allocation of the UAVs within the structure

The Mahalanobis distance [31] evaluates the norm between a point $\mathbf{p} = [x\ y]^T$ and the center $\mathbf{p}_c = [x_c\ y_c]^T$ of an ellipse, weighted by a function of the length of its main axis (see figure 19):

$$d_M(\mathbf{p}) = \sqrt{(\mathbf{p} - \mathbf{p}_c)^T \mathbf{M} (\mathbf{p} - \mathbf{p}_c)} \quad (55)$$

where $\mathbf{M}$ is the characteristic matrix of the ellipse.

The term J_i^t is used to lead the UAVs within the virtual structure. The Mahalanobis distance is used to reflect the shape of the ellipse.

$$J_i^t = w_t \sum_{h=1}^{H_p} \frac{(H_p - h)}{H_p} g(i,h) \tag{56}$$

where the function g is defined as

$$g(i,h) = \begin{cases} d_M\left(\hat{\mathbf{p}}_i(k+h)\right) & \text{if } d_M\left(\hat{\mathbf{p}}_i(k+h)\right) > 1 \\ d_M\left(\hat{\mathbf{p}}_i(k+h)\right) - 1 & \text{if } d_M\left(\hat{\mathbf{p}}_i(k+h)\right) < 1 \end{cases} \tag{57}$$

The weight $(H_p - h)/H_p$ is meant to give more importance to the first predictions than the later ones. The function $g(i, h)$ is built on the basis of the Mahalanobis distance of the UAV position to the ellipse center. This function introduces a potential field that guides the UAV within the area. A discontinuity has been added, to make a stronger difference at the boundary of the virtual structure. A projection of function g is provided in figure 20.

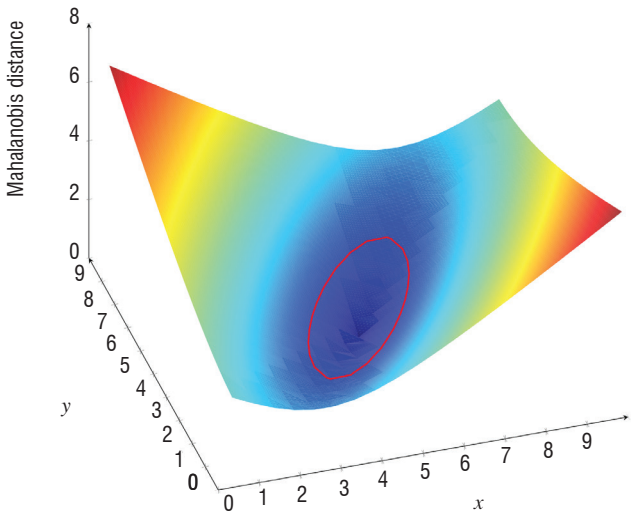


Fig. 19 - Mahalanobis distance to an ellipse (in red) over the position space

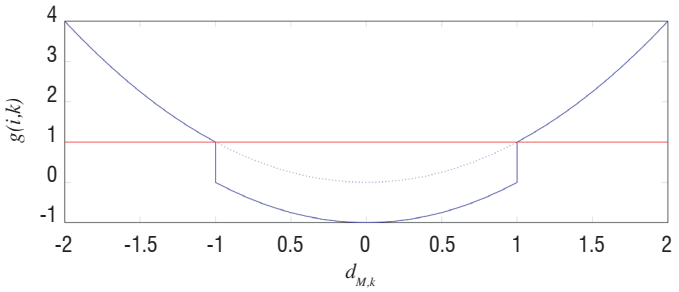


Fig. 20 - Shape of the function g (2D projection)

Formation consistency

The cost J_i^e is used to keep the speed and orientation of the UAV close to those of the ellipse v_c and α_c.

$$J_i^e = \sum_{t=1}^{H_p} w_n^t \left[w_{ni}\left(v_i(k+t)\right) - v_c(k+t)\right)^2 + w_{n2}\left(\chi_i(k+t) - \alpha_c(k+t)\right)^2 \right] \tag{58}$$

The weights w_n^t depend on the Mahalanobis distance between the UAV and the ellipse. When the distance is less than a value l_3, w_n^t is equal to one and when it is greater than l_4, w_n^t is equal to 0. The function f (figure 17) is used again.

$$w_n^t = f\left(d_M\left(\mathbf{p}_i(t)\right), l_3, l_4\right) \tag{59}$$

Simulation results

A simple scenario has been defined to test the guidance law. The formation should reach a predefined way-point $\mathbf{p}_{target}$ starting from $\mathbf{p}_0$, with an initial orientation of the ellipse perpendicular to the motion direction. Two rectangular obstacles cross the trajectory of the formation. In order to avoid collision with these obstacles, the virtual structure must modify its shape so as to pass the obstacles safely. Only the deformation is considered here ($u_\chi = 0$), but an additional rotation of the structure could be handled similarly.

The initialization parameters of the simulation are given in table IX. Note that the virtual structure has a longer prediction horizon than the UAVs, since it holds more information on the final destination and the target.

$w_{nav} = 10^{-2}$	$w_v = 10^{-4}$	$w_{ab} = 10$	$w_{abmin} = 10^{-3}$
$N = 8$	$v_n = 4$	$v_{min} = 2$	$v_{max} = 6$
$d\alpha_{max} = 0.3$	$dt = 1$	$d^v_{safe} = 6$	$d^v_{des} = 6$
$H_{p,uav} = 10$	$H_{c,uav} = 5$	$H_{p,ell} = 30$	$H_{c,ell} = 5$
$l_1 = 70$	$l_2 = 90$	$l_3 = 5$	$l_4 = 10$
$a_0 = 200$	$b_0 = 100$	$v_c = 4$	$N° = 2$
$\alpha_0 = \pi/2$	$\mathbf{p}_0 = [100\ 0]$	$\mathbf{p}_{target} = [2000\ 0]$	

Table IX - Simulation parameters

An example of gathering of the UAVs within the virtual structure is shown in figure 21.

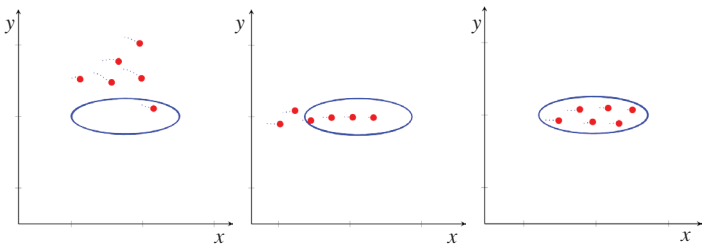


Fig. 21 - Gathering of UAVs within the ellipse structure

The complete scenario is illustrated by the sequence in figure 22. The ellipse modifies its shape accordingly when approaching the obstacles and no collision has been reported. The UAVs were initially in a vertical formation inside the ellipse. When the ellipse changes, the formation is modified to keep all of the UAVs within the structure. It can be seen that, since the range of the ellipse control inputs has been chosen to cope with the UAV dynamics, the vehicles have sufficient time to remain within the virtual structure when it is modified. The area of the ellipse is also kept close to its initial value. Figure 23 shows the values of the control inputs u_a and u_b that govern the deformation of the structure over time. These input values modify the length of the two principal axes simultaneously and almost symmetrically, to cope with the area constraint.

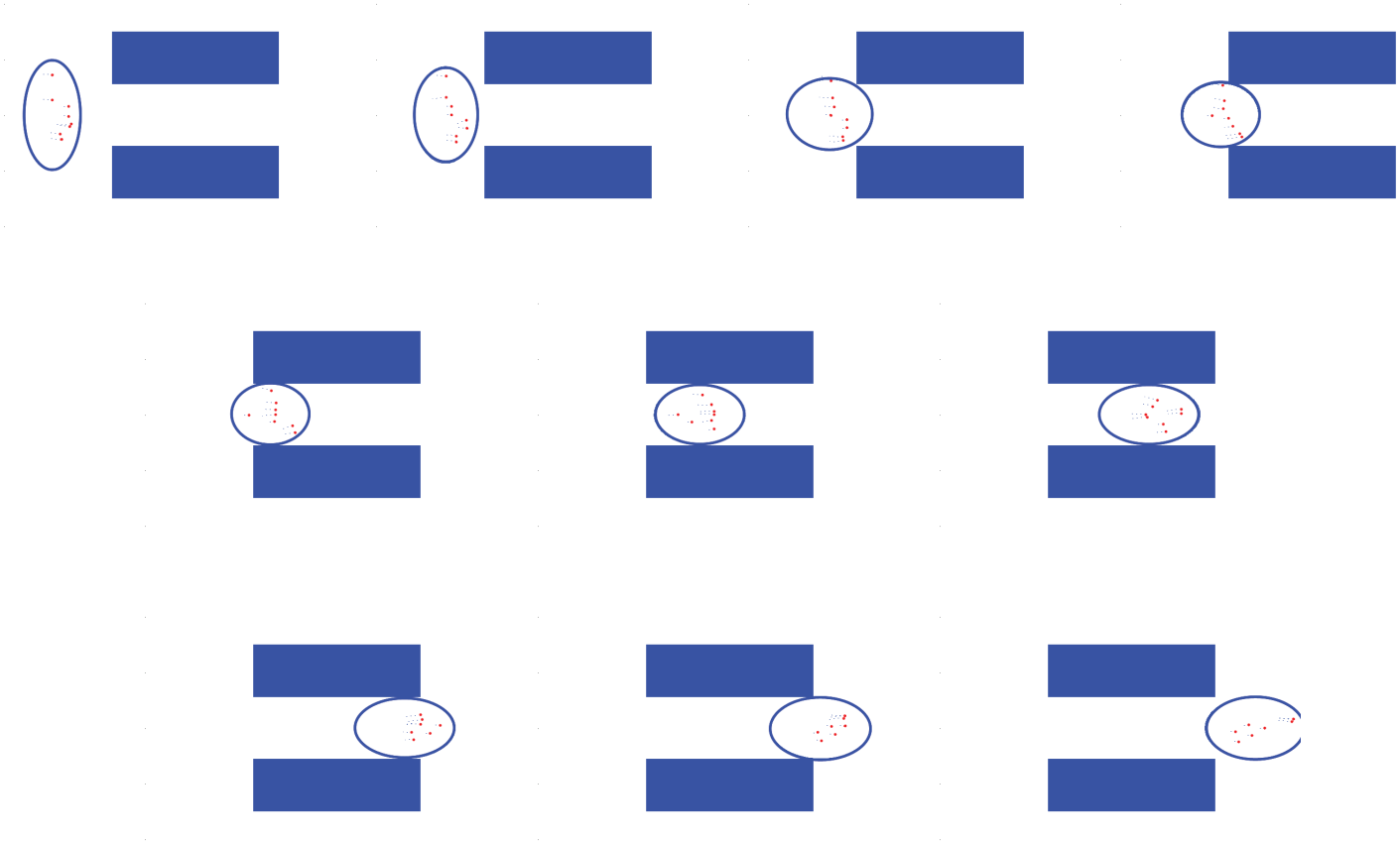


Fig. 22 - Obstacle avoidance by deformation

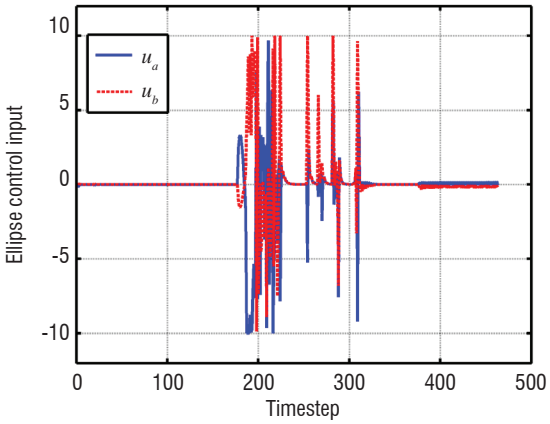


Fig. 23 - Control inputs for ellipse deformation

The proposed method thus makes it possible to maintain the UAVs within an elliptical virtual structure with collision avoidance, using the two-layer guidance law. The higher layer modifies the characteristics of the virtual structure with only knowledge of the obstacles and target, while the lower layer modifies the formation and distribution of the UAVs in a decentralized way, based only on the knowledge of the actions from the upper layer. Other shapes for the virtual structure could be taken into account within this guidance scheme by modifying the dynamical model of the structure and the criteria that govern the shape modification.

Conclusions and perspectives

In this paper, the design of distributed cooperative control laws for a fleet of autonomous vehicles has been presented using Model Predictive Control. This approach proves very flexible for taking into account mission objectives and safety and reliability constraints. The use of an 'any-time' optimization procedure guarantees that a control value will be obtained in a given amount of time, depending on the computational ability of the vehicles. Future developments include detection and rejection of outlying data, definition of suitable observers taking the cooperativeness of the vehicles into account and demonstration of robustness properties of the resulting control laws ■

Acknowledgements

The authors would like to thank Dominique Beauvois, Didier Dumur, Tomasz Gorecki, Guillaume Broussin, Mathieu Touchard and Mathieu Balesdent who contributed to this research program.

References

[1] R. OLFATI-SABER - *Flocking for Multi-agent Dynamic Systems: Algorithms and Theory.* IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 401–420, 2006.

[2] A. JADBABAIE, J. LIN, A. S. MORSE - *Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules.* IEEE Transactions on Automatic Control, vol. 48, no. 6, pp. 988–1001, 2003.

[3] J. A. FAX, R. M. MURRAY - *Information Flow and Cooperative Control of Vehicle Formations.* IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] D. GUNDUZ AND E. ERKIP - *Opportunistic Cooperation by Dynamic Resource Allocation.* IEEE Transactions on Wireless Communications, vol. 6, no. 4, pp. 1446–1454, 2007.

[5] T. VICSEK, A. CZIRÓK, E. BEN-JACOB, I. COHEN, O. SHOCHET - *Novel Type of Phase Transition in a System of Self-driven Particles.* Physical Review Letters, vol. 75, no. 6, p. 1226, 1995.

[6] B. GERKEY, M. MATARIC - *Sold!: Auction Methods for Multirobot Coordination.* IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 758–768, 2002.

[7] Y. GUO, L. E. PARKER - *A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots.* IEEE International Conference on Robotics and Automation, vol. 3, 2002, pp. 2612–2619.

[8] M. JUN, R. D'ANDREA - *Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments.* Cooperative Control: Models, Applications and Algorithms, Springer, 2003.

[9] W. DUNBAR, R. MURRAY - *Distributed Receding Horizon Control for Multi-vehicle Formation Stabilization.* Automatica, vol. 42, pp. 549–558, 2006.

[10] R. OLFATI-SABER, W. DUNBAR, R. MURRAY - *Cooperative Control of Multi-vehicle Systems Using Cost Graphs and Optimization.* Proceedings of the American Control Conference, Denver, Colorado, USA, vol. 3, 2003, pp. 2217–2222.

[11] W. DUNBAR, R. MURRAY - *Receding Horizon Control of Multi-vehicle Formations: A Distributed Implementation.* Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Bahamas, vol. 2, 2004, pp. 1995–2002.

[12] R. SCATTOLINi - *Architectures for Distributed and Hierarchical Model Predictive Control–a Review.* Journal of Process Control, vol. 19, no. 5, pp. 723–731, 2009.

[13] A. RICHARDS, J. P. HOW - *Robust Distributed Model Predictive Control.* International Journal of Control, vol. 80, no. 9, pp. 1517–1531, 2007.

[14] Y. ROCHEFORT, S. BERTRAND, H. PIET-LAHANIER, D. BEAUVOIS, D. DUMUR - *Cooperative Nonlinear Model Predictive Control for Flocks of Vehicules.* Proceedings of the IFAC Workshop on Embedded Guidance, Navigation and Control in Aerospace, Bangalore, India, 2012.

[15] E. HANSEN - *Global Optimization Using Interval Analysis*. CRC Press, 2004.

[16] P. POTOCNIK, I. GRABEC - *Nonlinear Model Predictive Control of a Cutting process.* Neurocomputing, vol. 43, no. 1-4, pp. 107–126, 2002.

[17] R. A. BARTLETT, A. WACHTER, L. BIEGLER - *Active Set vs. Interior Point Strategies for Model Predictive Control.* Proceedings of the 2000 American Control Conference, vol. 6, 2000, pp. 4229–4233.

[18] F. MARTINSEN, L. T. BIEGLER, B. A. FOSS - *A New Optimization Algorithm with Application to Nonlinear MPC*. Journal of Process Control, vol. 14, pp. 853–865, 2004.

[19] M. DIEHL, H. G. BOCK, J. P. SCHLODER, R. FINDEISEN, Z. NAGY, F. ALLGOWER - *Real-time Optimization and Nonlinear Model Predictive Control of Processes Governed by Differential-algebraic Equations.* Journal of Process Control, vol. 12, pp. 577–585, 2002.

[20] A. CHEMORI, M. ALAMIR - *Multi-step Limit Cycle Generation for Rabbit's Walking Based on a Nonlinear low Dimensional Predictive Control Scheme.* International Journal of Mechatronics, vol. 16, no. 5, pp. 259–277, 2006.

[21] M. ALAMIR, A. MURILO - *Swing-up and Stabilization of a Twin-pendulum under State and Control Constraints by Fast NMPC Scheme*. Automatica, vol. 44, pp. 1319–1324, 2008.

[22] E. W. FREW - *Receding Horizon Control Using Random Search for uav Navigation with Passive, Non-cooperative Sensing*. University of Colorado at Boulder, Tech. Rep., 2006.

[23] Y. ROCHEFORT - *Méthodes pour le guidage coopératif.* Ph.D. dissertation, Université Paris Sud - Supélec - Onera, 2013.

[24] T. GORECKI, H. PIET-LAHANIER, J. MARZAT, M. BALESDENT - *Cooperative Guidance of UAVs for Area Exploration with Final Target Allocation.* Proceedings of the 19th IFAC Symposium on Automatic Control in Aerospace, Würzburg, Germany, 2013.

[25] A. KAHN, J. MARZAT, H. PIET-LAHANIER - *Formation Flying Control Via Elliptical Virtual Structure*. Proceedings of the 10th IEEE International Conference on Networking, Sensing and Control, Evry, France, 2013, pp. 158–163.

[26] A. MOKHTARI, A. BENALLEGUE - *Dynamic Feedback Controller of Euler Angles and Wind Parameters Estimation for a Quadrotor Unmanned Aerial Vehicle.* IEEE International conference on robotics and automation, vol. 3, 2004, pp. 2359–2366.

[27] K. SINGH, K. FUJIMURA - *Map Making by Cooperating Mobile Robots.* IEEE International conference on robotics and automation, vol. 2, 1993, pp. 254–259.

[28] J. YUAN, Y. HUANG, T. TAO, F. SUN - *A Cooperative Approach for Multi-Robot Area Exploration*. IEEE International Conference on Intelligent Robots and Systems, 2010, pp. 1390–1395.

[29] J. MUNKRES - *Algorithms for the Assignment and Transportation Problems*. Journal of the Society of Industrial and Applied Mathematics, vol. 5, no. 1, pp. 32–38, March 1957.

[30] A. AHMADZADEH, G. BUCHMAN, P. CHENG, A. JADBABAIE, J. KELLER, V. KUMAR, G. PAPPAS - *Cooperative Control of UAVs for Search and Coverage*. Proceedings of the AUVSI conference on Unmanned Systems, Orlando, FL, USA, 2006.

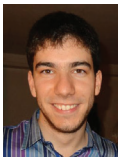
[31] R. D. MAESSCHALCK, D. JOUAN-RIMBAUD, D. MASSART - *The Mahalanobis Distance.* Chemometrics and Intelligent Laboratory Systems, vol. 50, no. 1, pp. 1–18, 2000.

AUTHORS

Sylvain Bertrand received his Engineering degree in 2004 from the Ecole Centrale de Lille, and his PhD in 2007 from the University of Nice Sophia Antipolis. Since 2007 he has been working as a research engineer at Onera on optimization based control and estimation for dynamic systems, guidance of Unmanned Aerial Vehicles, orbit determination and performance evaluation of space systems.

Julien Marzat graduated as an engineer from ENSEM (INPL Nancy) in 2008. He completed his PhD Thesis from the University Paris Sud XI at the end of 2011. Since 2011, he has been working as a research engineer at Onera. His research interests include fault diagnosis and control of autonomous vehicles, as well as surrogate-based optimization for engineering design.

Hélène Piet-Lahanier is a Senior Scientist (MR2) at Onera. She graduated from SupAero (Toulouse) and obtained her PhD in Physics and her Habilitation à Diriger les Recherches (HDR), both from the University ParisXI, Orsay. Her research interests include modelling under uncertainty, bounded error estimation, cooperative guidance and autopilot design. She is a member of the IFAC technical committee on Aerospace.



Arthur Kahn received his Engineer degree from Polytech Paris UPMC and a master degree from University Paris VI both in 2012. He is currently preparing a Ph.D. degree in automation at ONERA, Palaiseau, France, in the System Design and Performance evaluation Department (DCPS). His fields of research are: control laws for autonomous vehicles, multi agent systems and robust estimation using sensor networks.

Yohan Rochefort graduated as an automation engineer from the Ecole National Supérieure de Physique de Strasbourg (ENSPS) in 2009. He obtained a PhD in automation from the Ecole Supérieure d'Electricité (SUPELEC) in 2013 on design of control laws for cooperative guidance of autonomous vehicles. He is currently working as a system engineer in the field of navigation.