Available online at www.sciencedirect.com
ScienceDirect
Aerospace Science and Technology 11 (2007) 183–193
Aerospace Science and Technology
www.elsevier.com/locate/aescte

Autonomous flight and navigation of VTOL UAVs: from autonomy demonstrations to out-of-sight flights

Vol et navigation autonome des hélicoptères sans pilote : des démonstrations de capacités d'autonomie aux vols hors vue

P. Fabiani *, V. Fuertes, A. Piquereau, R. Mampey, F. Teichteil-Königsbuch

Systems Control and Flight Dynamics Department, ONERA-Toulouse/DCSD 2 av. Edouard Belin, F-31055, Toulouse, France

Received 17 June 2003; received in revised form 2 May 2006; accepted 4 May 2006

Available online 13 October 2006

Abstract

Future applications of UAV systems will depend on the aircraft autonomous behavior and decision capabilities. Search and Rescue is one complex possible mission and is here taken as a case study. The ReSSAC project is a multidisciplinary project at ONERA. Its main challenges are related to the architectures and algorithms for autonomous decision and information processing onboard UAVs that perform their mission in cooperation with operators. The feasibility demonstrations and results of the project are intended to be reused and extended in further studies, projects and collaborations. A first step of the project was to develop an autonomous control architecture for our two rotorcraft. In this paper, we present the current status and preliminary achievements of the ReSSAC project, especially some records of past experimental flights with our autonomous aircraft. We further discuss ongoing studies and research perspectives.

Résumé

Les applications futures des systèmes d'UAV dèpendront des capacités de comportement et de décision autonomes de ces aéonefs. La problématique de missions de recherche et sauvetage est complexe : elle est ici prise comme cadre d'étude et de démonstration. Le projet ReSSAC est un projet multidisciplinaire à l'ONERA. Ses principaux objectifs sont liés aux architectures et aux algorithmes pour le traitement de l'information et la décision autonome à bord d'UAVs qui exécutent leur mission en coopération avec des opérateurs. Les démonstrations de faisabilité et les résultats du projet pourront être réutilisés et prolongés dans d'autres études, projets ou collaborations. Une première étape du projet était de développer une architecture autonome de commande pour nos deux hélicoptères sans pilote. Dans cet article, nous présentons l'avancement du projet ReSSAC, particulièrement certains vols expérimentaux réalisés en mode autonome. Nous discutons en outre de la poursuite des études et des perspectives de recherche.

Keywords: Flight control; Autonomous navigation; UAV; Planning; Autonomous decision making and information processing; Safe autonomous landing in an unknown area for an uninhabited rotorcraft

1. Introduction

Significant research efforts have been devoted to key technologies needed for the development of onboard mission management systems for uninhabited air vehicles (UAVs). As a matter of fact, current UAV systems are mostly remotely controlled by an operator that can control the flight plan of the aircraft by

* Corresponding author.
E-mail addresses: fabiani@cert.fr (P. Fabiani), fuertes@cert.fr (V. Fuertes), piquereau@cert.fr (A. Piquereau), mampey@cert.fr (R. Mampey), teichteil@cert.fr (F. Teichteil-Königsbuch).

1270-9638/$ – see front matter © 2006 Published by Elsevier Masson SAS.
doi:10.1016/j.ast.2006.05.005

choosing way-points or targets to be achieved by an on-board auto-pilot. Current concepts of use of UAVs are very close to the use of aircraft in the early years of aviation: missions of reconnaissance, observation, etc. These concepts of use are very likely to change with progress in the area of UAV autonomy. A number of research projects exist [1–9,12–15,20] that deal with various potential applications of UAVs such as uninhabited combat aircraft, intervention rotorcraft or general autonomous observation and reconnaissance UAVs, road traffic surveillance helicopter, pursuit helicopter, agile urban intervention rotorcraft, search and rescue helicopters, power cable inspection UAVs or forest fire surveillance aircraft. Major projects are from Georgia Tech. University, Carnegie Mellon University, Berkeley University, WITAS project of Linköping University (Sweden), US Army/NASA Rotorcraft Div. Ames, Technische Universität Berlin, Stanford University, MIT Boston ... among others [1–9,12–15,20]. Search and Rescue is a quite generic possible application of such systems. It is taken as a case study in the Autonomous Air Vehicle ReSSAC [12] project. The project is mainly devoted to study architectures and algorithms for autonomous decision and information processing onboard UAVs. Its results and feasibility demonstrations are intended to be reused in further studies, in other (not necessarily VTOL) UAV projects and are meant to be eventually transferred to the industry. This paper presents the current status and achievements of the project, the different steps toward higher autonomy levels and describes the current work both on the platforms and on the algorithms. The paper further discusses ONERA's approach to UAV autonomy for out-of-sight flights.

2. Towards UAV decisional autonomy

A scale of autonomy levels (see Fig. 1 and [6] for other scales) can be drawn according to the way the operator interacts with the system:

- *At level* 0 of interaction, the operator directly gives orders to the aircraft actuators, observes the resulting attitude and controls the aircraft stability.
- *At level* 1, the aircraft is stabilized and thus, the operator pilots the aircraft motion.
- *At level* 2, the aircraft is controlled and the operator uses the flight control laws to control the trajectory.
- *At level* 3, the guidance system allows the operator to choose the path or the sequence of way-points to be achieved. This is the first step of "operational autonomy".
- *At level* 4 ("*operational autonomy*"), the navigation system is able to define intermediate way-points by itself, which allows the operator to decide on global navigation targets to be achieved. Level 4 is achieved on ReSSAC rotorcraft since 2003.
- *At level* 5 ("*decisional autonomy*"), the operator defines the system's mission and the aircraft is provided with onboard decision and information processing capabilities. The ReSSAC project [12] further aims at achieving and demonstrating level 5 exploration in search of a landing zone and autonomous landing in an unequipped and unknown area. This goal has since been achieved in September 2006, but after the completion of this paper: it will be published in a forthcoming paper.

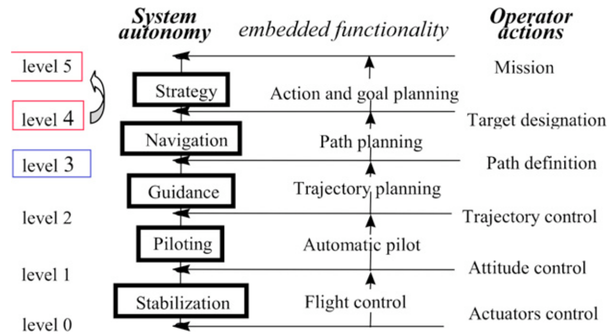

Fig. 1. Autonomy levels hierarchy.

3. Past achievements and ReSSAC status

ONERA's know-how with respect to this autonomy hierarchy was demonstrated with the Vigilant program. Our first autonomous navigation system has been embedded successfully on the small helicopter Vigilant F2000, an aircraft weighting around 35 kg in flight and made by Techno Sud Industries (Fig. 2). The first autonomous flight occurred in year 1997 [14]. The Vigilant F2000 is presently the only UAV in France to have obtained *an* "*out-of-sight flight authorisation*" by the French Civil Aviation authorities on the civil airfield of Revel, near Toulouse, with explicit mention on general aviation maps and flight documents. All other UAVs in France are flown in military areas, under military authority. The same autonomous navigation system was successfully transferred on the FUJI-made Vigilant F5000 with an approximate weight of 300 kg in flight (Fig. 3). The autonomous flight was performed in year 2000. Neither Vigilant F2000, nor F5000 are any longer available for further developments.

The ReSSAC project [12] has generalized and extended this flight and navigation architecture, making it more generic and reusable as a "*generic autonomy kit*" currently flying on the two ReSSAC RmaXs (Fig. 4).

Two Yamaha RmaX remotely piloted rotorcraft have been acquired by ONERA in May 2002. Their first autonomous flight occurred at the end of 2003. The embedded computer architecture is based on two central units in PC 104 format linked

Fig. 2. Vigilant F2000 (35 kg).

Fig. 3. Vigilant F5000 (300 kg).

Fig. 4. ONERA's autonomous RmaX rotorcraft n^{o} 635.

by a shared memory and running under the real time operating system VXWORKS. The programming language is the object language C++. This flight and navigation architecture is the basis of the security core of our flight management systems. A security μ-controller is added for redundancy.

The current achievements of the project are:

1. The development of an autonomously flying platform and its ground station, to be reused for further experiments and studies on autonomous behavior and decision making capabilities,
2. The definition and development of a generic, safe and verifiable flight control architecture implemented and tested in an "autonomous control kit" for our RmaX helicopter.
3. The development of models and algorithms for 3D path generation on the basis of a numerical terrain model and closed-loop strategy optimisation for mission planning and on-line replanning.

We are now working on the completion of the aircraft capabilities with autonomous take-off and landing on a landing zone, first known and equipped and later unknown and unprepared. This is done especially via the integration of new sensors and through the development of robust flight and trajectory control laws for the approach and landing phases. The second point is treated in close relation with the first one.

The main challenge of the project is to integrate and demonstrate the capabilities of on-board information processing, decision making and mission management required for our autonomous exploration scenario.

We are now working on the integration of the global control and decision architecture on-board our autonomous aircraft. Two additional on-board computers, respectively for information processing and decision making, are being integrated. An "out-of-sight" exploration demonstration scenario (with "out-of-sight" flight and autonomous landing and take-off on an unprepared spot) would demonstrate decision autonomy.

4. Autonomous flights experiments

Current flights are still performed with a "line-of-sight" security pilot, and authorised on the basis of the core PC104 architecture mentioned above. The project also developed a complete mobile ground station (Fig. 5) allowing future "out-of-sight" autonomous flights with a "instrument" security pilot enabled to control the flight of the aircraft from inside the mobile ground station thanks to a number of instrument flight screens and a front camera view (Fig. 6).

Such an equipment is however already used in order to acquire geo-referenced images for vision and 3D reconstruction algorithms testing. Other power-cable inspection experiments are also on our agenda, that will require a specific authorisation and "out-of-sight" autonomous navigation capabilities.

Past flights were conducted in order to test and validate the overall autonomous flight and navigation capabilities. Examples of such flight records are given in Fig. 7, with the usual "eights" and "hippodromes" trajectories. Real flights (right side) are compared to simulated flights in the same conditions. We do not use differential GPS for the moment, and may not need it for our landing demos on an unprepared and unknown area. On the other hand, in order to obtain a true terrain reference for our vision and 3D reconstruction algorithms (cf. Robea project "Acrobate"), we will use the differential GPS. We are also testing a number of ground height sensors: laser range finder, radar telemeter, ...

Fig. 5. ReSSAC mobile ground station.

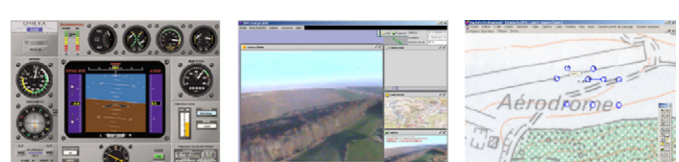

Fig. 6. ReSSAC control screens for "out-of-sight" control, navigation and capture of geo-referenced images.

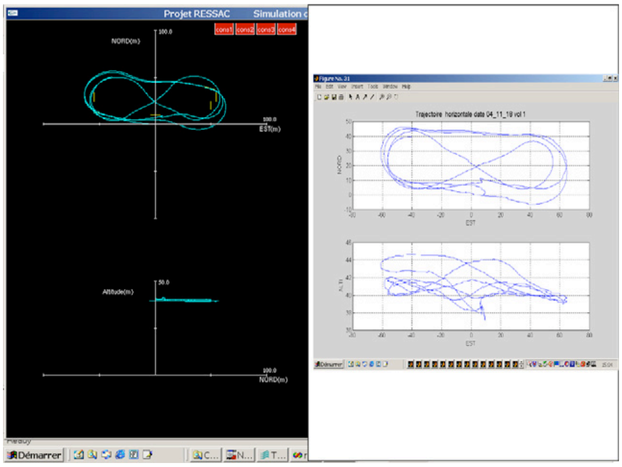

Fig. 7. Simulation (left) and autonomous flight record (right): "eight's" and "hippodromes".

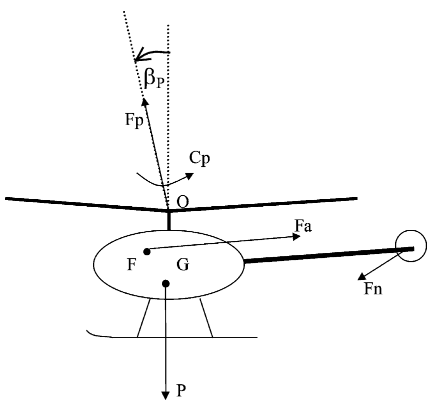

Fig. 8. Simplified model of an helicopter.

5. Control of the autonomous flights

Automatic control design is made in a classical way by the establishment of a control model used for the design of the command rules and the algorithms of state estimation. The control model of the dynamic of an helicopter chosen for this study is to be rather robust and deterministic in order to be acceptable for out-of-sight flight authorisations (auto-adaptive control is not suitable for that purpose). The model of the unmanned helicopter is obtained by writing the mechanics equations of the rigid body moving in an homogeneous fluid submitted to Earth attraction, to aerodynamic forces and to control efforts of the two rotors, as illustrated in Fig. 8.

Similar models based on identification were quite recently published [7] by very mature projects such as the Autonomous Helicopter Project at Carnegie Mellon University [2]. The nonlinear model, the stabilisation and flight control design of the helicopter are described at greater length in [14] and [10], which includes a model of the Bell-Hiller bars, and takes non-stationary effects into account, contrary to usual models.

The estimation of the attitude angles is based on an inertial measurement unit including accelerometers, gyrometers and a magnetic compass. The state estimation is completed by the determination of the inertial speed and position using the data given by a GPS (global positioning system) and an altimeter. A non-stationary embedded Kalman filter, based on the cinematic relations between acceleration, speed and position is used to provide the state estimate on line and in real time.

6. SNAKE autonomous navigation

Flight plans are defined by "enriched waypoints" to be reached, each waypoint being defined by:

- its location (X, Y, Z),
- the desired velocity vector, and
- the desired heading of the aircraft at this point.

In order to determine the trajectory between each of those waypoints, we apply the Principle of the Maximum, with a minimum time criterion under the constraint of saturation $r_{\max}$ imposed on the yaw rate. In the horizontal 2D plan, this problem can be solved by geometric considerations, because of the optimality of a "bang-bang" control. This gives trajectories formed successively by a first arc of circle, a straight line and a final arc of circle, as shown in Fig. 9: the determination in real time (at each guidance step) of the parameters α_1, α_2, ε_1, ε_2 allows the real time computation of the control input in terms of yaw rate to be given to the above control loops.

The SNAKE ("Système de Navigation Autorisant une Kyrielle d'Evitements") navigation system adds to the above guidance law the capability of avoiding forbidden or dangerous (even moving) areas. This function must be compatible with the guidance algorithm and with its use in real time. For that purpose, forbidden areas or obstacles, are considered as circles, or pairs of circles, to be geometrically treated one by one by the iterative navigation algorithm. For each obstacle, two possible avoidance trajectories are considered, respectively defined by the α_d and α_g angles (Fig. 9). These two angles could soon be measured in real time by an obstacle detection sensor, but it is not the case yet. The two angles are computed from the map and compared to the current control order α_c to conclude on a possible collision and define a new route with minimal turn:

If $\alpha_d < \alpha_c < (\alpha_d + \alpha_g)/2$ then $\alpha_c = \alpha_d$

If $\alpha_g > \alpha_c > (\alpha_d + \alpha_g)/2$ then $\alpha_c = \alpha_g$

Forbidden areas and obstacles are treated by order of decreasing distance to the aircraft (the most faraway obstacle is treated first and so on). The final control order α_c is given when all the forbidden areas and obstacles between the next waypoint

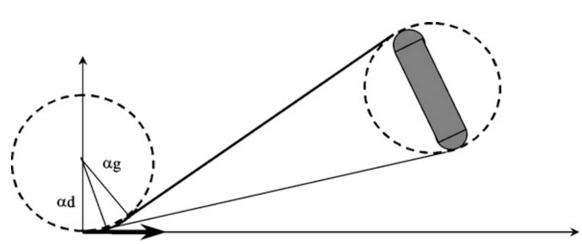

Fig. 9. Avoidance of a forbidden area.

and the current position of the aircraft have been treated. This warrants a real time behavior and a local optimality of the area avoidance trajectory, even with moving forbidden areas or obstacles: it does not always provide a globally optimal (shortest) trajectory to the next waypoint, depending on the avoidance areas configuration.

From the ground station, the operator can re-define the mission by providing the necessary data to be taken into account by the autonomous navigation system:

- Waypoints: positions (x, y, z), velocity vectors, aircraft headings.
- Forbidden areas: position of circles centres (one or two circles if they are combined), radius.
- Base: ground station for a normal landing.
- Safety base: area for a fail safe landing.
- Path: list of waypoint numbers to reach successively with possible loops on the waypoints.

7. Mission monitoring and planning

The 2D mission in Fig. 10 is designed by the operator: three waypoints, linked in two ways for a final loop realisation, and two forbidden areas.

The mission progress is monitored through the state evolution in a state graph, according to the operator requests and the safety reports. A fail-safe flight is engaged by the safety mechanisms in case of a system failure. The possible flight states corresponding to the "autonomous flight" are organized as in Fig. 11:

- three ground phases: "stop", "maintenance" and "initialisation", which correspond to the states of the helicopter when it is on the ground. The initialisation phase includes all the automatic check-list procedures before take off.
- two transition phases: "take off" and "landing".
- five flight phases: "route" and "return to base" which use the SNAKE navigation and three other phases linked with hovering flight ("position hovering flight", "velocity hovering flight", "stand by before landing").

In the flight state "fail-safe flight", the safety procedures, imposed by the use of autonomous vehicles, are described by Petri nets, such as for example in Fig. 12 for the case of a transmission failure. The autonomous flight capabilities of the helicopter allow an easy definition and execution of these procedures.

For example, it is able to return to a safety landing zone without any operator intervention nor predefined trajectory, provided that the map of known obstacles is up-to-date and sound: some verification of the map is useful. The verification of the map is made at mission planning time and should be redone as often as the mission is replanned: SNAKE is not an optimal path planner and obstacles must not overlap in the map it uses.

A path planner, as shown in Fig. 13, generates a graph of optimal itineraries by using state of the art algorithms. An itinerary planner was developed in the ReSSAC project that takes a 3D numerical terrain model and automatically searches for crests or valley waypoints on the map. It then generates a *graph* of time (or fuel) optimal itineraries between these waypoints, by taking into account the flight dynamics and the fuel consumption model of the aircraft.

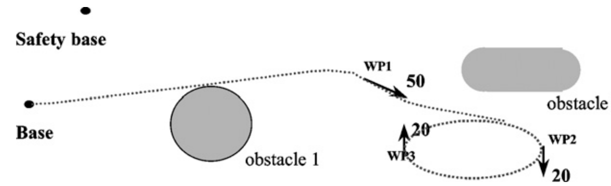

Fig. 10. Example of flight plan.

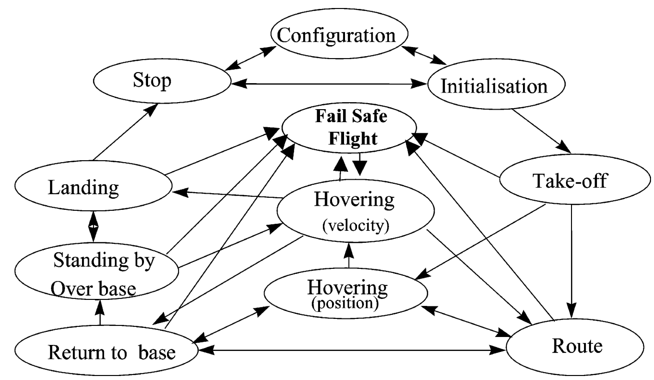

Fig. 11. Autonomous flight states.

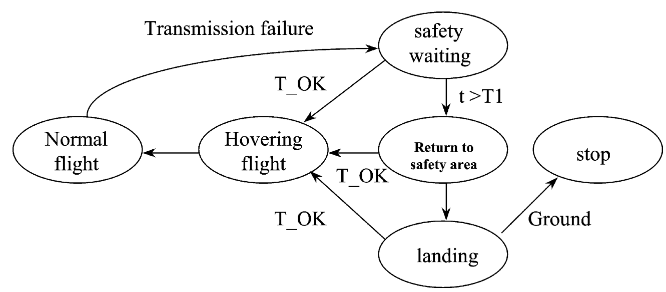

Fig. 12. Fail safe procedure.

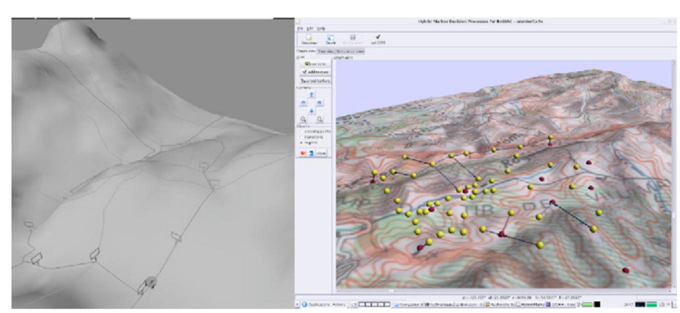

Fig. 13. Automatic waypoint and itinerary generation (left) and global mission planning interface (right).

Current work is dedicated to developing the interfaces:

- between the optimal itinerary planner and the SNAKE obstacle avoidance system on the one hand,

- between the optimal itinerary planner and a global mission (and perception) planner on the other hand.

The goal of such an integration is:

1. to enable the system to plan its mission according to the known 3D map of the environment and known obstacles on the one hand, and
2. to plan for information acquisition in order to update the current map on the other hand.

8. Towards landing in an unknown area

Last but not least, the ReSSAC project also develops tools for the perception of the environment and the updating of an unknown map of the exploration area. Vision algorithms are being developed, but in addition to that, the ReSSAC autonomous aircraft needs to be able to fly and move with heading and attitude constraints, so as to observe precisely enough the environment in order to build a coherent map of obstacles around landing areas.

A series of flights were conducted in order to improve the ReSSAC UAV flight capabilities with respect to side, up or down winds during crucial flight phases such as exploration, approach or landing. Our tests with the ground height sensors are still under progress, in order to improve the approach flight to the ground. The flight records shown on Fig. 14 show a “small square” trajectory performed with a constant heading of the rotorcraft (Yamaha training trajectories), testing sideward and backward flight capabilities. The flight record shown in Fig. 15 show a circular trajectory performed with varying heading, the nose camera of the aircraft being pointed downward to the centre of the trajectory on the ground (with no visual servoing for the moment). This kind of trajectories are useful in order to acquire geo-referenced images from different sides of an area or an object on the ground, as shown in Fig. 16, so as to perform a 3D reconstruction from monocular stereo-vision from motion.

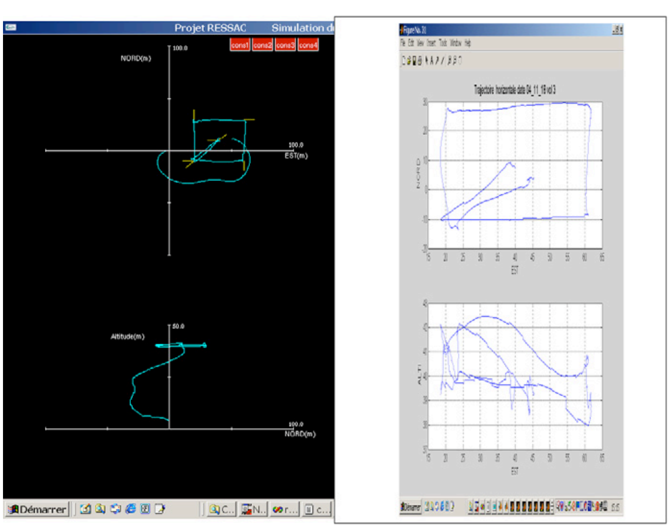

Fig. 14. Simulation (left) and autonomous flight record (right) for one “small square”.

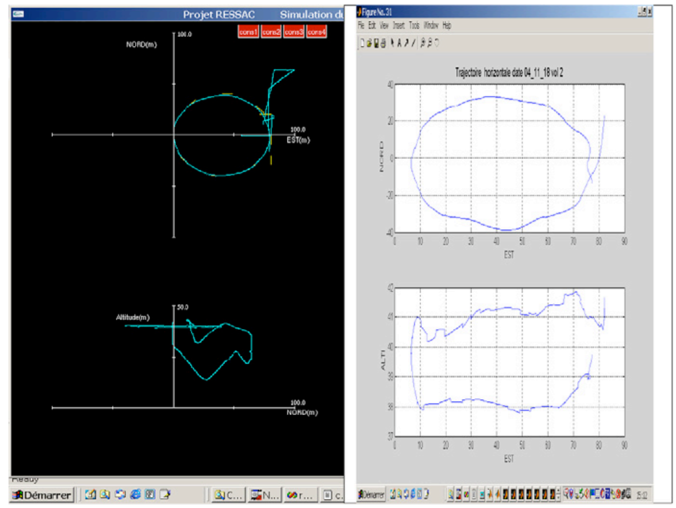

Fig. 15. Simulation (left) and autonomous flight record (right) for one “small circle”.

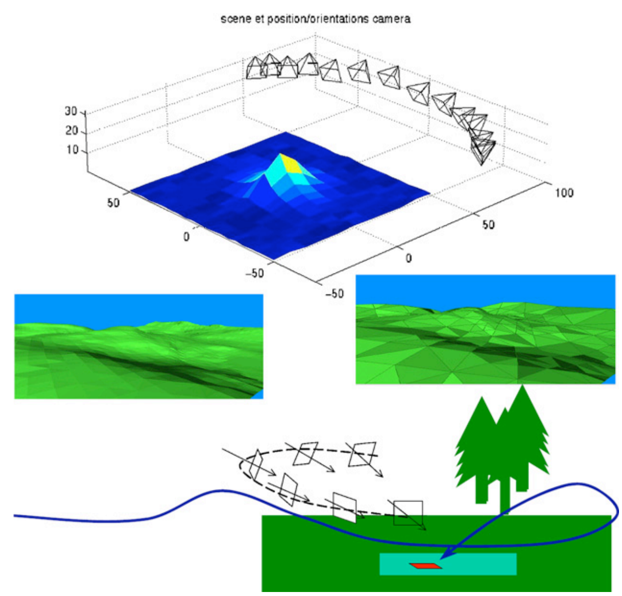

Fig. 16. Monocular stereo elevation map estimation from helicopter motion and 3D terrain models.

The result is an updating of an elevation map (a priori numerical terrain model with varying granularity) as shown in Fig. 16.

9. Towards decisional autonomy

Extending the autonomy capabilities of an unmanned aircraft is not a goal in itself. Such efforts are driven by the need for the aircraft to be able to manage by itself a situation were the control and supervision link is lost.

Preparing for this eventuality often appears as crucial when considering complex missions involving a mix of manned and unmanned assets, and especially the insertion of unmanned aircraft within an airspace populated with other aircraft. The loss of control can be due to a data-link failure with the ground control station, due to an excessive workload of the operators or due to the fact that the link with the information and decision

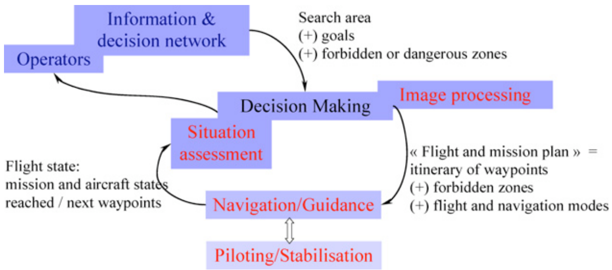

Fig. 17. Ground/On-board decision architecture.

network that normally supervises the mission, does not provide a sufficient level of situation awareness to the operators, or does not allow them to react in time.

For these reasons, we study the case were the pre-planned mission, possibly as shown in Fig. 10, should be reconfigured on-board the autonomous aircraft by its own replanning capabilities.

The basic level of such a replanning capability is well represented with the case of obstacle avoidance. Assuming that we are able to detect autonomously the obstacle, or that we are informed by other means of the presence of this obstacles, we are able to adapt to the new situation with the SNAKE navigation system, unless the obstacles come into tricky configurations so that some obstacles overlap, which is possible with moving objects. Higher level motion planning capabilities are thus required for such cases.

State of the art algorithms can provide good real time motion planning solutions to that problem in the case of one single aircraft (multiple aircraft flight planning and collision avoidance can be much more complex). Yet, most missions are not purely geometrical trajectory planning problems and combine other variables, some of them controllable and some of them random or unpredictable.

Fig. 17 shows a classical decisional architecture for autonomous agents, which is applied to the ReSSAC aircraft. It is important to notice that all the situation assessment, decision making and image processing functions need not be fully embedded on-board the aircraft, nor in the ground control station.

There is likely to be a sharing between the functions that are required to be on-board the aircraft and those that are required to be kept under control of the operator: this possibly uneasy choice is studied in the ReSSAC project.

10. Autonomous exploration problem

In order to get a first idea of the possible benefits of using Artificial Intelligence planning techniques for real-world autonomous aircraft, the case of an exploration mission is studied within the *ReSSAC* project. This example is abstracted from the initial Search & Rescue application scenario of the project.

An exploration mission is composed of a problem of navigation and a problem of information acquisition in a partially known environment. It can be addressed at different levels of modelling. Motion planning and information processing aspects are important when a sufficient flow of sensory information is available through the use of range or object detectors, 3D

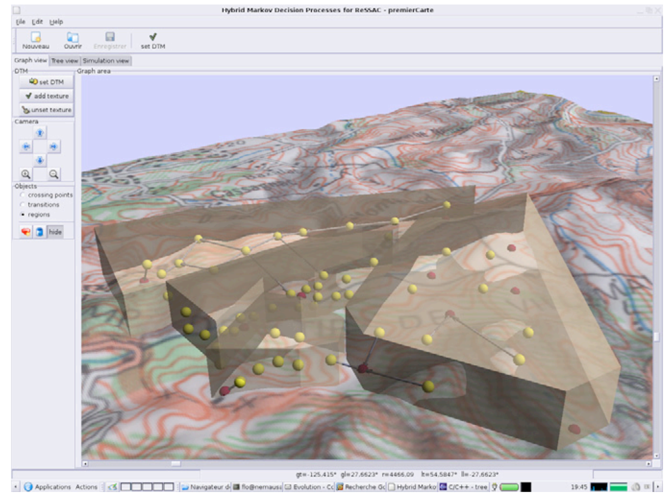

Fig. 18. Exploration mission (MDP) definition interface.

sensors, . . . etc. Our focus is on a more abstract level: on planning under uncertainty for both motion actions and information acquisition tasks, which is a crucial issue for autonomous systems.

In our exploration problem, such as in Fig. 18, different regions are identified that require to be exhaustively or partially explored, mapped or searched for the presence of persons or objects, before continuing the mission. Information acquisition may be part of intermediate goals of the mission, and may also impact on the subsequent tasks or navigation actions. Such missions can be modelled at a higher level of abstraction, as a sequence of mission phases, tasks, or macro-actions: for each mission phase, the system needs to achieve navigation and information acquisition goals, before proceeding with one of the possibly following phases.

The order in which the intermediate sub-goals or tasks must be achieved can be free or constrained. Thus, they can be seen as pre-conditions of other tasks: this implies that searching for a good mission plan (not even the optimal one) requires to explore the combinatorics of possible branches in an AND-OR graph of possible states looking like an "unfolded" version of the state graph in Fig. 10, with additional possible "information acquisition" flight phases in the middle.

Furthermore, each intermediate goal must be achieved while minimising risks and costs and thus by optimising the navigation and the action strategy while taking into account uncertainties on the values of a number of random or unpredictable variables.

Our exploration planning problem contains a number of possible tasks, associated with rewards, each of which can be obtained once in turn, no matter the order, before reaching the goals, and final rewards. To model this, we need to introduce binary state variables depending on whether the goals have already been achieved or not. Problems that are modelled with such an hybrid structure combining a navigation state space and orthogonal mission of internal state variables have been studied in [19]. Such problems of sequential decision under uncertainty can be modelled using Dynamics Bayes Nets (DBNs) such as in Fig. 19, or in many problems of the ICAPS'2004 probabilistic

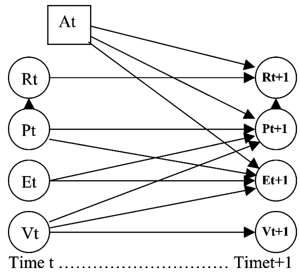

Fig. 19. Dynamic Bayes Net for variables Rt (region), Pt (waypoint), Et (energy level), Vt (wind), At (action).

track planning competition in which we participated with preliminary versions of our algorithms [17]. In Fig. 19, the arrows arriving to $Et + 1$ mean that the probability distribution on the possible values of E (energy level of the aircraft) at time $t + 1$ is given as a conditional probability depending on Vt, Et, Pt and the choice of the action At chosen at time t.

Our exploration planning problem is thus modelled in a factored form [16,17,19], and difficult to represent graphically, except for the interface defined in Fig. 18. The underlying navigation problem is still present, but the dimension of the state space is increased. The navigation goals to be achieved depend on the values of the other variables. The level of energy autonomy is a good example of an additional state variable with a high impact on the navigation. Flight is thus limited by energy autonomy, and the aircraft can decide to return to its base, either if all its goals have been achieved, or if it is running short of fuel, in which case it may as well go to its failsafe or emergency crash base. Other random variables may also impact the navigation decisions, such as the ground height, the presence of objects, the distance to obstacles (especially in unknown environments), the local winds and turbulence conditions which may allow an approach to landing or not, etc. Yet, such factored MDP formulations of the problem may lead to optimisation problems of untractable complexity. We tested many algorithms [16,17] and developed a symbolic focused dynamic programming (SFDP) algorithmic scheme, which enables us to efficiently find a feasible solution for very large factored MDP [18], that optimal algorithms cannot tackle (Fig. 20).

We developed a incrementally defocused approach that first finds a feasible solution to the problem and then incrementally imposes increasing constraints on the optimality of the solution, until optimality [11].

Current work on decisional autonomy is focused on:

- the integration of an overall mission management architecture allowing to extend the decisional autonomy of the aircraft (the decision and information processing capabilities will first be implemented and tested on the ground before any attempt of on-board integration),
- the integration of motion planning algorithms with the navigation system,
- the development of optimisation algorithms to tackle factored probabilistic planning problems,

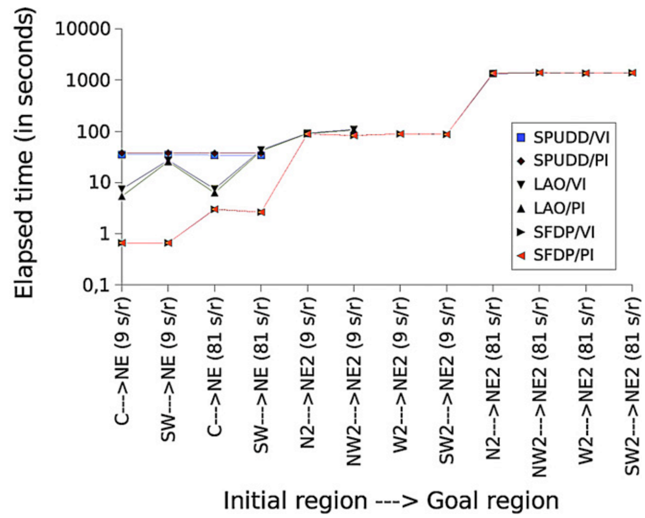

Fig. 20. SFDP can tackle the problems beyond N° 7 (from left to right).

Fig. 21. SFDP incrementally improves the solution.

- general work toward a final demo including exploration, landing zone selection and mapping, autonomous landing in an unprepared area and flight back to the base.

11. Conclusion

We have presented the current status and achievements of the Autonomous Air Vehicle ReSSAC project at ONERA. The project has now reached a maturity stage, with two aircraft capable of autonomous navigation under the supervision of an operator. This article gives some records of these achievements. Yet, the presented results must still be considered as preliminary since the real steps forward are currently being worked on, compared to past achievements at ONERA and in other projects.

The ReSSAC project, quite ambitious, shares a number of challenges, and is in competition, with other ongoing projects that involve teams of talented researchers: exploration or Search & Rescue missions on the one hand, autonomous landing in a non-cooperative unprepared unknown area on the other hand. We therefore believe that we are not completely wrong in that research directions. Co-operations with the LAAS-CNRS Robotics group in Toulouse are alive. We are open to possible collaborations with people that may be facing similar problems as we do in our research, or with researchers that would find an interesting field of application of their own work in our problematic or could exchange with us by contributing to our experimental platforms.

Appendix A. Automatic flight control

Automatic control design is made in a classical way by the establishment of a control model used for the design of the command rules and the algorithms of state estimation. According to the specification which are more important on the stability than on the performances, this design is made axis by axis without any coupling between them, thus giving the basic safe and verified layers of our control architecture.

This point is crucial to obtain again the *out-of-sight flight authorisations* from the French civil Aviation authorities, similarly to the Vigilant F2000 in 1998.

A.1. Modelling

The flight dynamics model of our unmanned helicopters is chosen to be non-linear in order to provide efficient control laws. Such a non-linear model is obtained by writing the mechanics equations of the rigid body moving in an homogeneous fluid submitted to the attraction of Earth, to the aerodynamic forces and to the control efforts of the two rotors, as illustrated in Fig. 8.

Aerodynamic parameters are obtained via the identification of linear models. Similar linear models are used and were recently published [7] in other unmanned helicopter projects such as the Autonomous Helicopter Project at Carnegie Mellon University [2].

For the longitudinal and lateral axis, the simplified model is:

$$I\ddot{\theta} = \vec{F}_p \overrightarrow{OG} \sin\beta_P \approx \vec{F}_p \overrightarrow{OG}\beta_P$$

$$M\ddot{x} = \vec{F}_p \sin(\theta + \beta_P) \approx \vec{F}_p(\theta + \beta_P) \approx -Mg(\theta + \beta_P)$$

For the motion along the vertical axis, the control model is described by a simplified representation of the main rotor, with a first order approximation giving its lift force F_p as a function of the variation of the induced velocity V_u near the equilibrium in hover:

$$F_p \approx -2\rho S V_f (V_f + 2V_u + V_z)$$

with:

$$V_f = \sqrt{\frac{Mg}{2\rho S}}$$

$V_z =$ vertical speed

$V_u =$ variation of induced velocity near hover equilibrium

The expression for V_f is according to the momentum theory in hover. V_u is assumed to be linearly controlled by a commanded variation u of DT0 (see in Fig. 26), the collective pitch angle, near the equilibrium in hover.

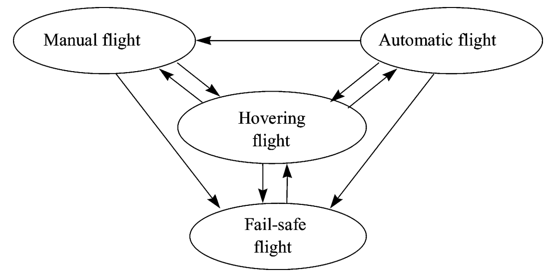

Fig. 22. States of flight and flight modes.

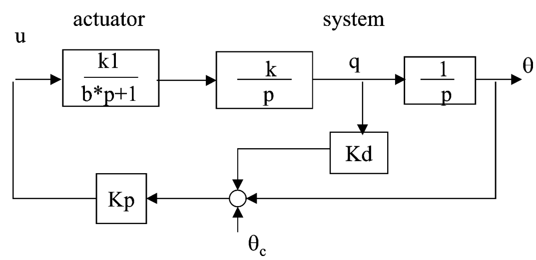

Fig. 23. Pitch (or roll) stabilisation.

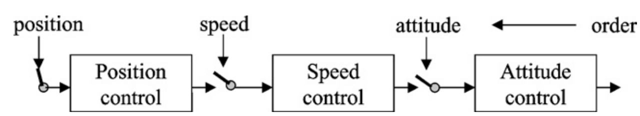

Fig. 24. Horizontal control.

A.2. Control design

A.2.1. Stabilisation

The stabilisation of the helicopter consists in the control of the pitch and roll angles and of the yaw speed with the bloc diagram presented on Fig. 23. On the pitch and roll axis, the stabilisation is performed with a proportional derivative controller, and for the yaw control with a proportional controller on the yaw rate.

A.2.2. Control of the roll, pitch and yaw angles

The following equations apply for the control laws in both roll and pitch axis, with the generic state variable α: $\ddot{\alpha} = -g\theta_c$ is used to obtain the desired first derivative $\dot{\alpha}_c$ where the control input θ_c is then as follows:

$$\theta_c = k_v(\dot{\alpha}_c - \dot{\alpha}) + k_{vi}\int (\dot{\alpha}_c - \dot{\alpha})\,\mathrm{d}t$$

The desired first derivative of α can either be a direct pilot input or a function of the error of α compared to the desired value α_c:

$$\dot{\alpha}_c = k_x(\alpha_c - \alpha)$$

There are then three possibilities for the operator to control the roll and pitch angles of the helicopter (as shown in Fig. 24):

- attitude control (second derivative of α),
- speed control (derivative of α),
- or position control (value of α).

The yaw angle control scheme is simpler (Fig. 25) and can be either:

- a yaw speed control, based on the gyrometer measurements,
- or a heading control, using the filtered and calibrated magnetometer measurements.

A.2.3. Vertical speed and altitude control

The linearized low speed model is used to design the vertical speed and altitude control law around the hover equilibrium. The variable u is the variation of collective control input which is supposed linked with the induced velocity by a first order transfer.

The operator can choose to control either the vertical speed, or the altitude (Fig. 26).

A.3. State estimation

The estimation of the attitude angles is based on an inertial measurement unit including accelerometers, gyrometers and a magnetic compass for the heading. The state estimation is completed by the determination of the inertial speed and position using the data given by a GPS (global positioning system) and an altimeter.

A Kalman filter, based on the cinematic relations (Fig. 27) between acceleration, speed and position is used to provide the state estimate on line and in real time.

For a greater capacity of adaptation to variations in the quality of the measurement and in the data rate, a non-stationary form of the filter is encoded and runs on the embedded computer.

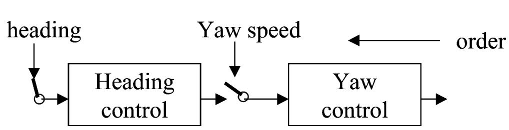

Fig. 25. Yaw axis control.

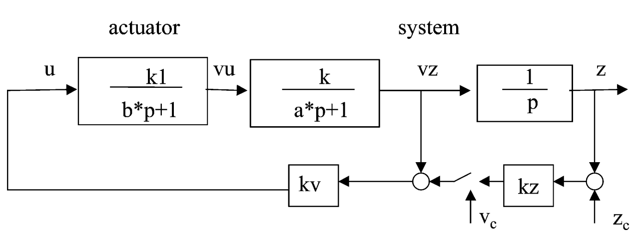

Fig. 26. Vertical controller.

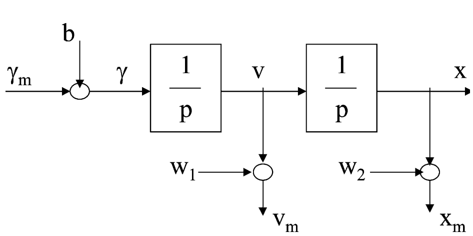

Fig. 27. Cinematic model.

The equations then follow from the cinematic model shown in Fig. 27 and give the following expression:

- $$\begin{pmatrix} V \\ X \end{pmatrix}_{n+1} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V \\ X \end{pmatrix}_n + \begin{pmatrix} 0 \\ T \end{pmatrix} \gamma_n + \begin{pmatrix} 0 \\ b \end{pmatrix}$$
- with the measurement equation:
$$\begin{pmatrix} X_m \\ V_m \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ V \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$
- with the transformation of the GPS information (*lat*, *long*, *alt*) in the Cartesian vector X (North, East, Down) with a origin point (lat_0, $long_0$, alt_0), as follows:
$$x_m = (lat - lat_0) R_0$$
$$y_m = (long - long_0) R_0 \cos(lat_0)$$
$$z_m = alt_0 - alt$$
- with the specific altimeter equation:
$$z_{alt} = (1 \quad 0) \begin{pmatrix} z \\ v \end{pmatrix} + w_3$$
This gives the following predictive equations:
- $$\begin{pmatrix} \hat{X} \\ \hat{V} \end{pmatrix}_{n+1/k} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{X} \\ \hat{V} \end{pmatrix}_{n/k} + \begin{pmatrix} 0 \\ T \end{pmatrix} \gamma_n$$
- $$P_{n+1/k} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} P_{n/k} \begin{pmatrix} 1 & 0 \\ T & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & q \end{pmatrix}$$
- with, for the GPS measurement:
  - Kalman coefficient with r_1 and r_2 respective quality of position and speed measure:
$$\begin{pmatrix} k_{pp} & k_{pv} \\ k_{vp} & k_{vv} \end{pmatrix} = P_{l/k} \left(\begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} + P_{l/k} \right)^{-1}$$
  - Kalman correction:
$$\begin{pmatrix} \hat{X} \\ \hat{V} \end{pmatrix}_{1/k+1} = \begin{pmatrix} \hat{X} \\ \hat{V} \end{pmatrix}_{l/k} + \begin{pmatrix} k_{pp} & k_{pv} \\ k_{vp} & k_{vv} \end{pmatrix} \left(\begin{pmatrix} X_m \\ V_m \end{pmatrix} - \begin{pmatrix} \hat{X} \\ \hat{V} \end{pmatrix}_{l/k} \right)$$
  - Covariance matrix of the estimate error:
$$P_{1/k+1} = \begin{pmatrix} 1 - k_{pp} & -k_{pv} \\ -k_{vp} & 1 - k_{vv} \end{pmatrix} P_{k+l/k} \times \begin{pmatrix} 1 - k_{pp} & -k_{vp} \\ -k_{pv} & 1 - k_{vv} \end{pmatrix} + \begin{pmatrix} k_{pp} & k_{pv} \\ k_{vp} & k_{vv} \end{pmatrix} \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \begin{pmatrix} k_{pp} & k_{vp} \\ k_{pv} & k_{vv} \end{pmatrix}.$$
- with for the altimeter measurement:
  - Kalman coefficient:
$$\begin{pmatrix} k_{zalt} \\ k_{valt} \end{pmatrix} = \frac{P_{l/k} \binom{1}{0}}{\left(r_{alti} + (1 \quad 0) P_{l/k} \binom{1}{0}\right)}$$
  - Kalman correction:
$$\begin{pmatrix} \hat{x} \\ \hat{v} \end{pmatrix}_{1/k+1} = \begin{pmatrix} \hat{x} \\ \hat{v} \end{pmatrix}_{l/k} + \begin{pmatrix} k_{zalt} \\ k_{valt} \end{pmatrix} \left(Z_{alt} - (1 \quad 0) \begin{pmatrix} \hat{x} \\ \hat{v} \end{pmatrix}_{l/k} \right)$$

– Covariance matrix of the estimate error:

$$P_{1/k+1} = \begin{pmatrix} 1 - k_{zalt} & 0 \\ k_{valt} & 1 \end{pmatrix} P_{k+l/k} \begin{pmatrix} 1 - k_{zalt} & k_{valt} \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} k_{zalt} \\ k_{valt} \end{pmatrix} r_{alti} \begin{pmatrix} k_{zalt} & k_{valt} \end{pmatrix}$$

References

[1] The Hummingbird project, Stanford university autonomous helicopter project, http://sun-valley.stanford.edu/heli/.
[2] Helicopter projects, Massachusetts Institute of Technology, http://www.gewurtz.mit.edu/research/heli.htm.
[3] The Autonomous helicopter project, Carnegie Mellon University, http://www.ri.cmu.edu/projects/project_93.html.
[4] BEAR, Berkeley university aerobot project, http://robotic.eecs.berkeley.edu/bear/overview.html, 2001.
[5] The software enabled control project, Georgia Technology Inst., http://controls.ae.gatech.edu/sec/, 2002.
[6] M. Idan, M.D. Johnson, M.J. Calise, A hierarchical approach to adaptive control for improved flight safety, AIAA Journal of Guidance, Control, and Dynamics 25 (6) (2002) 1012.
[7] B. Mettler, M.B. Tischler, T. Kanade, System identification modeling of a small scale unmanned rotorcraft for flight control design, Journal of the American Helicopter Society (January 2002).
[8] J. Del Cerro, A. Barrientos, P. Campoy, P.J. Garcia, An autonomous helicopter guided by computer vision for inspection of overhead power cable, in: IROS Workshop on Aerial Robotics, Lausanne, 2002.
[9] P. Doherty, G. Granlund, Kr. Kuchcinski, E. Sandewall, K.L. Nordberg, E. Skarman, J. Wicklund, The WITAS unmanned aerial project, in: ECAI2000, 2000.
[10] P. Fabiani, A. Piquereau, V. Fuertes, P.-M. Basset, American Helicopter Society Specialists' Meeting on Unmanned Rotorcraft, Chandler, Arizona, USA, January 18–20, 2005.
[11] P. Fabiani, F. Teichteil-Königsbuch, Symbolic focused dynamic programming for decision-theoretic planning, in: 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 30 July–5 August 2005.
[12] P. Fabiani, The ReSSAC autonomous air vehicle: challenges and demonstration platforms, in: IROS Workshop on Aerial Robotics, Lausanne, 2002.
[13] K.L. Nordberg, P. Doherty, G. Farnebäck, P.-E. Forssén, G. Granlund, A. Moe, J. Wicklund, Vision for a UAV helicopter, in: IROS2002 Workshop on Aerial Robotics, Lausanne, 2002.
[14] A. Piquereau, V. Fuertes, P. Fabiani, Autonomous flight and navigation of VTOL UAVs, in: ODAS'2003 Toulouse, France, Navigation autonome du Vigilant F2000, ONERA/DCSD, Final Report, 1998.
[15] V. Remuß, M. Musial, G. Hommel, MARVIN an autonomous flying robot based on mass market components, in: IROS2002 Workshop on Aerial Robotics, Lausanne, 2002, Techn. Universität Berlin.
[16] F. Teichteil-Königsbuch, P. Fabiani, Influence of modeling structure in probabilistic sequential decision problems in: RAIRO Operations Research, 2005, in press.
[17] F. Teichteil-Königsbuch, P. Fabiani, Probabilistic reachability analysis for structured Markov decision processes, in: The Probabilistic Planning Track of the ICAPS 2004 International Planning Competition, Whistler, British Columbia, Canada, 3–7 June 2004.
[18] F. Teichteil-Königsbuch, P. Fabiani, Symbolic heuristic policy iteration algorithms for structured decision-theoretic exploration problems, in: Workshop "Planning under Uncertainty for Autonomous Systems", Monterey, California, USA, 6–7 June 2005.
[19] F. Teichteil-Königsbuch, P. Fabiani, Un modèle hybride en planification probabiliste d'exploration autonome, in: Quatorzième Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, Toulouse, France, 28–30 janvier 2004. LAAS-CNRS, IRIT.
[20] R. Vidal, Sh. Sastry, H.J. Kim, O. Shakernia, D.H. Shim, The Berkeley aerial robot project (BEAR), in: IROS2002 Workshop on Aerial Robotics, Lausanne, 2002.