15th Triennial World Congress, Barcelona, Spain

A FORMAL APPROACH FOR MISSION PLANNING AND CONTROL OF UNMANNED AIRCRAFT

Conrado Seibel * **Jean-Marie Farines** **

* *Gyron Sistemas Autônomos Ltda.*
Florianópolis, Brazil
** *Departamento de Automação e Sistemas*
Universidade Federal de Santa Catarina
Florianópolis, Brazil

Abstract: A formal approach for the mission planning and control of unmanned aircraft is described. Linear hybrid automata are used to model the chosen aircraft, the resources needed to perform a mission and the environment in which the aircraft will operate. This *mission model* can be verified with respect to safety criteria, like not running out of fuel, or used to perform parametric analysis, allowing the instantiation of symbolic parameters in flight plans for specific missions. Later, during mission execution, an on-board flight controller is used to control the progress of the mission by reacting to relevant external and internal events.

Keywords: Autonomous vehicles, flight control, planning, automata.

1. INTRODUCTION

The use of unmanned aircraft (also known as UAVs — Unmanned Aerial Vehicles) has risen steadily over the past years, with applications ranging from law enforcement and ecological surveillance to high-altitude scientific research being reported.

Aircraft losses caused by human operator errors have traditionally been the biggest problem associated with UAVs, having even determined the cancellation of some promising UAV programs (Fulghum, 1996). Running out of fuel during a mission is the second most common cause of UAV losses, highlighting the complexity of balancing efficient use of the aircraft in pursuit of the mission objectives with conservative, safe operation.

Successfully planning and executing a mission means *achieving the mission objectives, using the resources of the given aircraft (e.g. fuel, electric energy) while complying with a set of imposed restrictions (e.g. terrain elevations, “no-fly” zones)*.

A formal approach for the mission planning and control of UAVs can greatly increase the safety of operation of this class of aircraft, while simultaneously reducing their operational cost by diminishing the workload imposed on the operator and therefore his/her training requirements.

This paper describes how hybrid automata can be used to model the mission of rotary-wing unmanned aircraft. This *mission model* can be verified with respect to safety criteria, like not running out of fuel, or used to perform parametric analysis, allowing the instantiation of symbolic parameters in flight plans for specific missions.

Later, during mission execution, a finite-state control automaton (derived from the hybrid automaton used to model the mission) is used on-board the aircraft to control the progress of the mission by reacting to relevant external and internal events.

Section 2 describes how a rotary-wing UAV and its operating environment can be modeled using hybrid automata, while section 3 presents three approaches for the construction of feasible flight plans based on the verification of the model. Section 4 shows how a single control automaton can be derived from the set of hybrid automata used to model the mission and used to control the execution of the associated flight plan. Finally, conclusions are drawn in section 5. An informal review of hybrid automata is presented in appendix A.

2. MISSION MODELING

The term *mission* is used to describe the operation of an aircraft in a given region, during a certain period of time while pursuing a specific set of mission objectives. The ordered set of movements executed by the aircraft during a mission is defined in an associated *flight plan*.

One of the ways to tackle the task of formally planning missions for unmanned aircraft is by using *hybrid automata*, a generalization of finite-state automata, equipped with a set of continuous variables (Alur *et al.*, 1993; Nicollin *et al.*, 1993). A hybrid automaton is able to model discrete events, like the the transition from one phase of the flight plan to the next one, and continuous activities governed by a set of differential equations, like the position of the aircraft, the amount of fuel remaining on board and the charge of an emergency accumulator.

Modeling a mission requires the modeling of: (1) the *aircraft dynamics* (position and velocity of the aircraft, as described in the associated flight plan); (2) the *internal conditions* of the aircraft (the resources and state of the aircraft); (3) the *external conditions* of the environment, dictated by the operational context under which the aircraft will operate and (4) a set of mandatory and optional *safety requirements*.

Items (1) and (2) will be modeled using hybrid automata, while items (3) and (4) will be modeled using the concept of *regions*, sets of states of the hybrid automata being investigated. For a more complete discussion of mission modeling for UAVs using hybrid automata, the reader is referred to (Seibel *et al.*, 1998*a*; Seibel, 2000).

2.1 *Aircraft dynamics*

A flight plan can be decomposed in its component *phases*. Each phase is described either by the coordinates of a pair of *way-points* and by the speed at which the aircraft is to fly between these way-points or by its duration, an initial way-point and the speed of the aircraft. A phase is completed when the second way-point is reached by the aircraft or, in the case of a specified duration, when the associated amount of time has elapsed.[1]

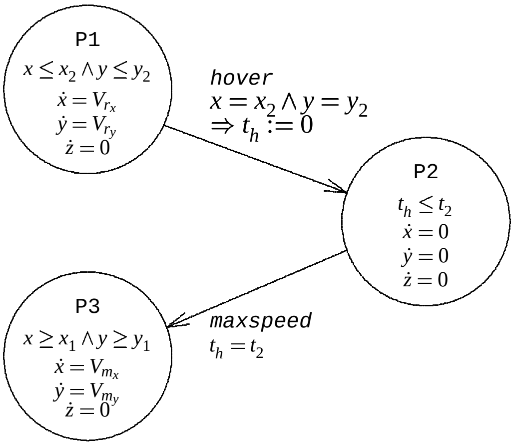

Fig. 1. A hybrid automaton used to model aircraft dynamics

A hybrid automaton, $\mathscr{D}$, is used to model the dynamics of the aircraft under the influence of the prevailing (or expected) meteorological conditions.

Consider a partial flight plan with only three phases: (1) flight at best-range speed at a constant altitude of z meters, from way-point $\mathrm{WP}_1 = (x_1, y_1, z)$ to way-point $\mathrm{WP}_2 = (x_2, y_2, z)$; (2) hover at $\mathrm{WP}_2 = (x_2, y_2, z)$ for t_2 seconds; (3) return to $\mathrm{WP}_1 = (x_1, y_1, z)$ at maximum speed.

Figure 1 shows a linear hybrid automaton which models the dynamics of the partial flight plan just described. Each control location of the automaton corresponds to one of the phases of the flight plan. The activities of each of the control locations describe the evolution of the aircraft's position on a tangent-plane coordinate system[2]. Invariants are used to force the evolution of the automaton at the end of each phase, while the transitions connecting the different control locations are guarded by tests on the aircraft's position or on a clock. Labels are used to synchronize this automaton with the rest of the model.

See appendix A for a more in-depth discussion of this example and an informal introduction to hybrid automata.

2.2 *Internal conditions*

Each aircraft has a distinctive set of resources, which are used by the operator to pursue the mission objectives. Resources have to be considered when planning a mission. They include the speeds at which the aircraft can be operated, the fuel that can be carried on board, and electric energy which can be generated and/or stored on board the aircraft. Also, an aircraft is characterized by the specific fuel consumption at each

---

[1] A *complete flight plan* contains a *primary* flight plan, comprising the phases that lead to the fulfillment of the mission objectives, and a set of *alternate* flight plans, which reflect the desired behavior of the aircraft due to different mission doctrines and/or expected failures.

[2] This is a Cartesian coordinate system whose origin is located at the take-off/landing point and which neglects the effects of the curvature of the Earth.

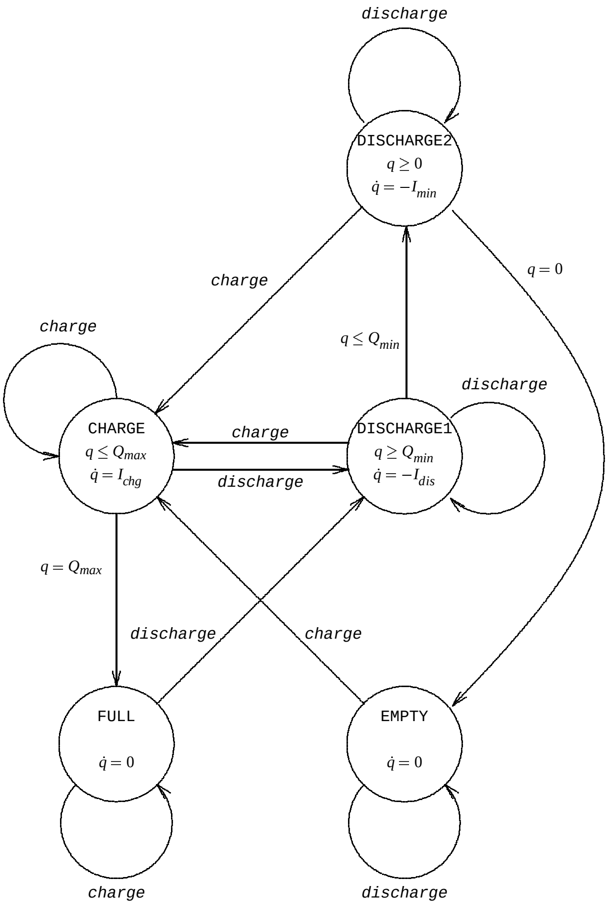

Fig. 2. A hybrid automaton used to model an emergency accumulator

speed[3], fuel tank capacity and how much electric energy is generated, stored and managed on board.

The term *internal conditions* is used in this article to collectively describe the resources and characteristics of the aircraft used to perform the mission. Internal conditions can be directly modeled by hybrid automata. In our models, we use at least the $\mathscr{F}$ and $\mathscr{Q}$-automata to model fuel consumption and electric energy stored in an emergency accumulator, respectively.

Figure 2 shows a typical $\mathscr{Q}$-automaton. The location `CHARGE` is used to model the charging of the accumulator with a constant current I_{chg}. Discharging the accumulator is modeled by the locations `DISCHARGE1` (where the current drained from the accumulator is I_{dis}) and `DISCHARGE2` (where the current drained is $I_{min} < I_{dis}$). Two additional locations, `FULL` and `EMPTY`, are used to model the fact that the charge of an accumulator cannot be greater than its nominal capacity or lesser than zero.

Again, labels are used to synchronize this automaton with the rest of the model, so that the energy used during the initialization of the aircraft (when the engine is not running and therefore the generator is not producing any energy) can be considered and failures of the generator/power conditioner can be modeled.

2.3 *External conditions*

The external conditions describe the context under which the aircraft will operate.

The prevailing (or expected) *meteorological conditions* at the time of the flight have to be considered since the aircraft's dynamics are affected by the speed and direction of the wind. This influence is most evident in small UAVs, which are often under-powered. Meteorological conditions are modeled by the $\mathscr{D}$-automaton.

Mission modeling also has to consider the geographical region were the flight will be performed, i.e., the elevations of the terrain and the existence of "no-fly" zones, zones were the operation of the aircraft is not allowed at all or only during certain periods of time.

The geographical region were the flight is performed also influences the propagation of the radio waves used for communication between the aircraft and its ground control station. Particularly, the terrain profile can generate additional conditions when using line-of-sight communication links.

External conditions can be described by using inequalities over a tangent-plane coordinate system to define a set of "good" and "bad" 3-dimensional regions. Communications coverage zones are considered "good" regions, while terrain elevation and "no-fly" zones are considered "bad" regions, since they should be avoided by the aircraft at all times.

2.4 *Safety requirements*

Mission planning has to take a set of *safety requirements* into account. Mandatory safety requirements, like not running out of fuel, have to be satisfied to avoid the loss of the aircraft. Optional safety requirements, like keeping minimum distances from the ground during the flight, increase the safety of the operation.

As with external conditions, safety criteria can be modeled using the concept of regions[4]. All the safety criteria that flight plans for UAVs have to satisfy can be expressed in terms of inclusion in a "good" region and intersection with a "bad" region.

---

[3] Rotary-wing aircraft normally operate at one of the following speeds: (1) best-range speed, the speed at which the *distance* traveled or *area* covered is maximized; (2) endurance speed, the speed at which *airborne time* is maximized, (3) maximum speed and (4) low speeds used for hover.

[4] Regions used to model safety criteria are not necessarily restricted to the three dimensions of the tangent-plane coordinate system. They include, for example, inequalities about the amount of fuel remaining on board the aircraft.

3. MISSION PLANNING

In the following we present three approaches to mission planning for unmanned aerial vehicles: (1) *verification* of a given flight plan, (2) *instantiation* of parameters in a parameterized flight plan and *incremental construction* of flight plans.

3.1 *Analysis of linear hybrid automata*

All three approaches are based on a methodology for analysis of linear hybrid automata developed in (Alur *et al.*, 1993; Nicollin *et al.*, 1993). The methodology is based on predicate transformers for computing the step predecessors and step successors of a given set of states.

Using successive approximation, the methodology allows the computation of $post^*(W)$, the *reachable region* of a region W, i.e., the set of all states that are reachable from states in W. Conversely, it is possible to compute $pre^*(W)$, the *initial region* of W, the set of all states from which a state in W is reachable.

Tools for the automatic analysis of hybrid systems, like HYTECH (Henzinger and Ho, 1995), can be used to perform the parallel composition of hybrid automata and to compute the reachable region of the resulting hybrid system. HYTECH is also able to manipulate regions, computing their intersection and verifying if a region is included within another one.

Existing algorithms only allow the verification of linear hybrid automata. We therefore establish the following five hypothesis, which allow us to describe flight plans for unmanned aircraft by a set of linear hybrid automata (Seibel and Farines, 1997; Seibel, 2000):

- Position and velocity of the aircraft are described in a *tangent plane coordinate system*.
- The air-speed of the aircraft is constant during each phase of the flight.
- The transitions from one phase of the flight plan to the next one are much shorter than the duration of the phases themselves.
- Non-linearities of the specific fuel consumption can be removed by using the *rate conversion* technique described in (Ho, 1995).
- Terrain elevations, communications-covered regions and exclusion zones can be approximated with satisfactory resolution by a set of first-order inequalities.

3.2 *Verification of flight plans*

The feasibility of a proposed flight plan can be *verified*. The result of the verification process is either “flight plan is feasible” or “flight plan is not feasible”. Verification can be accomplished by the following algorithm:

(1) Construct an automaton that describes the mission (the $\mathscr{M}$-automaton). This is the parallel composition of the $\mathscr{D}$, $\mathscr{F}$ and $\mathscr{Q}$-automata.
(2) Starting with initial conditions at the take-off point (region I), compute the forward reachable region of the $\mathscr{M}$-automaton.
(3) Compute the intersection of the reachable region with the “bad” region (defined by the external conditions in the form of inequalities on the tangent plane coordinate system). If the intersection is not empty, the feasibility of the flight plan cannot be guaranteed.
(4) Verify if the reachable region is contained within the “good” region. If not, the feasibility of the flight plan cannot be guaranteed.

Let us consider a sample mission: over-flying (at constant altitude) a couple of islands off the coast of Santa Catarina in the southern part of Brazil, while investigating outlawed fishing activities on the islands. The aircraft will take-off from 27°40.9′ S, 48°33.8′ W and fly directly to Island Francisca, located at 27°42.2′ S, 48°33.9′ W. From there it will fly to Island do Largo, at 27°42.4′ S, 48°35.6′ W and return to the take-off point for landing.

Tangent plane coordinates for take-off/landing point, Island Francisca and Island do Largo are, respectively, $(0,0)$, $(-2247,-118)$ and $(-3051,-3080)$[5]. The aircraft will be flown at its bestrange speed, $V_r = 15$ m/s. Specific fuel consumption, f, is less than 0.56 g/s under such conditions.

We want to verify if an initial amount of fuel $F = 1000$ g is sufficient to accomplish the mission while avoiding the approach corridor of the nearby airfield, described (in tangent plane coordinates) by

$$-2/5y + 700 < x < -2/5y + 3700$$

Using HYTECH we find that the reachable region of the $\mathscr{M}$-automaton is

$$\begin{aligned} post^*(I) = {} & 0 = x + 15t \wedge 4x = 75y \wedge \\ & 27f = x + 27000 \wedge \\ & 0 >= x \wedge x + 2250 >= 0 \\ \vee\ & 0 = x + 4t + 1650 \wedge \\ & 8y = 29x + 64290 \wedge \\ & 36f = 5x + 44250 \wedge \\ & 0 >= x + 2250 \wedge x + 3045 >= 0 \\ \vee\ & 32t = 3x + 20295 \wedge \\ & 63x = 64y + 285 \wedge \\ & 5x + 96f = 62175 \wedge \\ & x + 3045 >= 0 \wedge 0 >= x \end{aligned}$$

---

[5] tangent plane coordinates are expressed in meters

$$\begin{aligned} &\vee\ x = 0 \wedge 64y + 285 = 0 \wedge \\ &\quad 32f = 20725 \wedge 32t >= 20295 \end{aligned}$$

The above result shows that the aircraft uses $\frac{20725}{32}$ g of fuel (which is less than $F = 1000$ g) and that the intersection of the reachable region of the $\mathscr{M}$-automaton and the “no-fly” zone defined by the approach corridor to the airfield is empty. We therefore conclude that the proposed flight plan is feasible.

3.3 *Instantiation of parameterized flight plans*

The discussed methodology for the analysis of hybrid systems is able to manipulate not only numerical quantities but also *symbolic parameters*.

For the purpose of this paper, we therefore introduce the concept of a *parameterized flight plan*, a plan with at least one symbolic parameter. Symbolic parameters in a parameterized flight plan can be *instantiated* so that the resulting flight plan is feasible. The result of the instantiation process is “flight plan will be feasible if ...”.

Instantiation can be accomplished by the following algorithm:

(1) Construct an automaton that describes the mission (the $\mathscr{M}$-automaton). This is the parallel composition of the $\mathscr{D}$, $\mathscr{F}$ and $\mathscr{Q}$-automata.
(2) Compute the backward reachable region of the $\mathscr{M}$-automaton, starting at the overall “bad” region.
(3) Intersect the result with the initial region I and instantiate all symbolic parameters. The values obtained at this step are those that will lead from the initial region to the “bad” region.
(4) Complement the parameters obtained in the previous step. These are the values for which the mission is feasible.

3.4 *Incremental construction of flight plans*

Flight plans can be incrementally constructed by computing the safely reachable region as each phase is added to the partially constructed flight plan. The overall feasibility of a flight plan constructed in this manner can therefore be guaranteed in advance.

Incrementally constructing a flight plan can be done by continuously computing and displaying the reachable region of the $\mathscr{M}$-automaton over a map of the terrain where the flight will be performed.

The consequences of adding or removing way-points to the flight plan are immediately reflected in the shape of the reachable region, assisting the operator in the determination of the next phase of the flight plan being constructed.

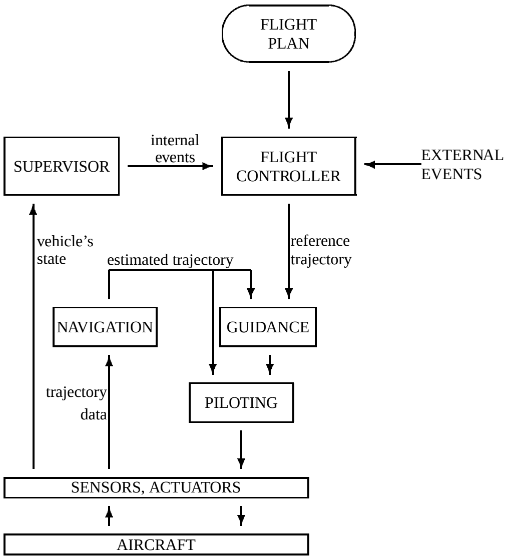

Fig. 3. The on-board flight controller

See (Seibel *et al.*, 1998*b*) for further details of a *graphical flight plan editor* with an underlying verification engine.

4. MISSION CONTROL

Mission control comprises the step-by-step execution of the resulting flight plan, while considering the aircraft’s internal state, relevant external events and the possible early success of the mission.

A reactive flight controller is used to control the evolution of a finite-state automaton on-board the aircraft during the execution of the mission. The finite-state automaton $\mathscr{C}$ used by the flight controller can be derived from the mission automaton $\mathscr{M}$ by stripping it of its continuous parts.

A Petri net is used to represent the $\mathscr{C}$-automaton on-board the aircraft. The flight controller is implemented around a general-purpose Petri net token-player and passes reference values to the lower-level piloting and guidance controllers. A supervisor is responsible for generating internal events based on the state and position of the vehicle, see figure 3.

5. CONCLUSIONS

Hybrid automata are well suited to model unmanned rotary-wing aircraft and their operating environment. At the same time, the usage of regions is very natural for the verification of safety requirements.

The approach could be used for the mission planning of fixed-wing aircraft, underwater autonomous vehicles or any other class of vehicles that operate at constant speed between way-points, follow a straight line from one way-point to the next one and where the error introduced by neglecting the curvature of the earth can be ignored.

Appendix A. MODELING AND VERIFICATION USING HYBRID AUTOMATA

A.1 *Hybrid Automata*

A hybrid automaton, as defined in (Alur *et al.*, 1993; Nicollin *et al.*, 1993), is constructed by the generalization of a finite-state automaton, equipped with a set of continuous variables. A hybrid automaton is able to model discrete events and continuous activities, governed by a set of differential equations.

Hybrid automata are described by a finite set of real-valued variables X and by a labeled multi-graph (V,E). The standard notation $\dot{X}$ is used to denote the first-order derivatives of X. The edges E represent the discrete events and are labeled with restrictions on the values of X *before* and *after* the execution of the corresponding actions. The vertices V represent the continuous activities and are labeled with restrictions on X and $\dot{X}$ *during* the corresponding activity. Therefore, the state of a hybrid automaton is modified by discrete events and by the passing of time.

A *hybrid system* is described by a collection of hybrid automata, one for each component of the system. The constituent automata operate in a concurrent and coordinated way, sharing a set of common variables X, and synchronizing on the common set $syn_1 \cap syn_2$ of synchronization labels.

A hybrid automaton is said to be linear if all its activities, invariants and transition relations can be described by linear expressions over the set of the automaton variables, X. This implies that for all locations $v \in V$, the activities $act(v)$ are defined by a set of differential equations of the form $\dot{x} = k_x$, one for each variable $x \in X$, where $k_x \in Z$ is an integer constant.

A.2 *Modeling aircraft dynamics with hybrid automata*

The following is a more detailed discussion of the example presented in section 3.2.

The automaton starts at location `P1`. At this location, the aircraft's position in the tangent plane coordinate system is described by the differential equations $\dot{x} = V_{r_x}$ and $\dot{y} = V_{r_y}$. V_{r_x} and V_{r_y} are the north and east components of the aircraft's best-range speed V_r after considering the influence of wind:

$$V_{r_x} = V_r \cos\psi + V_w \cos\psi_w$$
$$V_{r_y} = V_r \sin\psi + V_w \sin\psi_w$$

where $\psi = \arctan((y_2 - y_1)/(x_2 - x_1))$ is the course that leads from WP_1 to WP_2, ψ_w is the direction to where the wind blows and V_w is the wind's speed. It is important to note that the model accepts constant values for the wind's speed and direction as well as ranges of minimum/maximum values.

At $\mathrm{WP}_2 = (x_2, y_2, z)$, the transition that leads to `P2` is enabled and the invariant of `P1` turns false. Firing the transition sets clock $t_h := 0$, which will be used to time the duration of the hover. The synchronization label *hover* is used to synchronize other automata used to model the aircraft.

The automaton remains at location `P2` for t_2 seconds. Note that, during the hover, $\dot{x} = \dot{y} = 0$. After t_2 seconds have elapsed, the automaton progresses to location `P3`, which is used to model the return to WP_1 at maximum speed (*maxspeed*).

Appendix B. REFERENCES

Alur, R., C. Courcoubetis, T. Henzinger and P. Ho (1993). Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: *Hybrid Systems* (R. L. Grossman, A. Nerode, A. P. Ravn and H. Rischel, Eds.). Vol. 736 of *Lecture Notes in Computer Science*. pp. 209–229. Springer Verlag.

Fulghum, David A. (1996). Outrider UAV tackles army, navy requirements. *Aviation Week & Space Techology*.

Henzinger, T. and P. Ho (1995). HYTECH: The Cornell HYbrid TECHnology tool. In: *Proceedings of 1994 Workshop on Hybrid Systems and Autonomous Control*. Springer Verlag.

Ho, Pei-Hsin (1995). Automatic Analysis of Hybrid Systems. PhD thesis. Cornell University. Department of Computer Science.

Nicollin, X., A. Olivero, J. Sifakis and S. Yovine (1993). An approach to the description and analysis of hybrid systems. In: *Hybrid Systems* (R. L. Grossman, A. Nerode, A. P. Ravn and H. Rischel, Eds.). Vol. 736 of *Lecture Notes in Computer Science*. pp. 149–178. Springer Verlag.

Seibel, C. (2000). Uma Metodologia Formal para o Planejamento e Controle de Missões de Aeronaves Não-Tripuladas. PhD thesis. Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina. In Portuguese.

Seibel, C. and J.M. Farines (1997). Formal verification and dimensioning of flight plans for rotary-wing unmanned aerial vehicles. In: *Proceedings of the 24th Annual Technical Symposium of the Association for Unmanned Vehicle Systems International*. Baltimore, MD.

Seibel, C., J.M. Farines and José E. R. Cury (1998*a*). Towards using hybrid automata for the mission planning of unmanned aerial vehicles. In: *Hybrid Systems V*. Lecture Notes in Computer Science. Springer Verlag.

Seibel, C., J.M. Farines, L. C. Corrêa and G. C. Dallagnolo (1998*b*). Generating executable flight plans for unmanned aircraft: A formal mission planning tool. In: *Proceedings of the 13th Bristol International Conference on RPVs/UAVs*. Bristol, UK.