HYBRIDGE

Distributed Control and Stochastic Analysis of Hybrid Systems
Supporting Safety Critical Real-Time Systems Design

WP7: Error Evolution Control

Error Detection within a Specific Time Horizon

Maria D. Di Benedetto, Stefano Di Gennaro,
Alessandro D'Innocenzo[1]

22 February 2005

Version: *0.4*
Task number: *7.4*
Deliverable number: *D7.4*
Contract: *IST-2001-32460 of European Commission*

[1] University of L'Aquila

DOCUMENT CONTROL SHEET

Title of document: *Error detection within a specific time horizon*

Authors of document: *Maria D. Di Benedetto, Stefano Di Gennaro, Alessandro D'Innocenzo*

Deliverable number: *D7.4*

Contract: *IST-2001-32460 of European Commission*

Project: *Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design (HYBRIDGE)*

DOCUMENT CHANGE LOG

Version #	Issue Date	Sections affected	Relevant information
0.1	30 Aug 2004	All	First draft
0.2	30 Sep 2004	All	Second draft
0.3	19 Nov 2004	All	Third draft
0.4	22 Feb 2005	All	Fourth draft

Author(s) and Reviewers		Organisation	Signature/Date
Authors	M.D. Di Benedetto	AQUI	
	S. Di Gennaro	AQUI	
	A. D'Innocenzo	AQUI	
Internal reviewers	H. Blom	NLR	
	E. De Santis	AQUI	
	G. Pola	AQUI	
	T. Lewis	BAES	
	D. Jordan	BAES	

Abstract. The fourth deliverable D7.4 of Work Package WP7 of the HYBRIDGE Project focuses on the detection of situation awareness errors within a specific time horizon. In this report, the runway crossing control problem is considered as a case study, in order to motivate the extension of the notion of observability for hybrid systems to yield the notion of *critical observability*. The hybrid model is improved with respect to the one proposed in Deliverable 7.3 and is now more realistic. Five agents are present; four are humans, modelled as hybrid systems, subject to situation awareness errors that could lead to dangerous situations. The problem is to detect the errors *immediately* to prevent them to cause catastrophic events. Hence, the classical notions of observability for hybrid systems need to be extended to consider *critical* observability, whereby hazardous states have to be detected in one step of the Finite–State Machine component of the hybrid system. Conditions for the existence of an observer for critical states are also given and a procedure for its computation presented.

1 Introduction

In an Air Traffic Management (ATM) closed–loop system with mixed computer–controlled and human–controlled subsystems, recovery from non–nominal situations implies the existence of an outer control loop which has to identify these situations and act accordingly to prevent them to evolve into accidents. The purpose of Work Package WP7, "Error Evolution Control", of the HYBRIDGE project is to develop algorithms with guaranteed performances for assisting human operators in detecting critical situation and avoiding the propagation of errors and other non–nominal events. Estimation methods and observer design techniques are essential in this regard for the design of a control strategy for error propagation avoidance and/or error recovery.

Various aspects need to be taken into account in the study of error detection for ATM:

1. Psychological models which can be used for the study of ATM;
2. Stochastic hybrid models describing the dynamics involved in error evolution control, capturing the essential features of ATM;
3. Observability and observer design for these hybrid models;
4. The applicability of theoretical results on observers to a realistic ATM situation.

In the first three tasks of WP7, the following objectives related to the first three aspects were met:

(i) Task 7.1: we dealt with a review of some of the psychological models that are in use for the study of air traffic management.

(ii) Task 7.2: we identified, in collaboration with University of Cambridge, a stochastic hybrid model to describe the dynamics involved in error evolution control and capture the essential features studied in Task 7.1 (see Deliverable 1.2 [2]). We also addressed the issue of observability and observer design for hybrid systems. In Public Deliverable 7.2 [4], we reviewed the literature on observability and observers for hybrid systems as a first step in our quest for a general hybrid system observer. We then illustrated a synthesis method for hybrid observers [1].

(iii) Task 7.3: we investigated the applicability of the theoretical results on observers obtained in Deliverable 7.2 to a realistic ATM situation, the active runway crossing control problem, for the detection of situation awareness errors (see Deliverable 7.3 [5]). The resulting observer works well for this application: an alarm is generated when a critical situation occurs, for example whenever an aircraft is about to cross the runway when another aircraft is taking off.

The work carried out in Deliverable 7.3 [5] also shows that some new theoretical investigation turns out to be necessary to represent the error detection problem better. In fact, the observer construction methods proposed in Deliverable 7.2 and applied in Deliverable 7.3 are based on the notion of K–current–state

observability [1]. A hybrid system is K–current–state observable if any discrete location of the hybrid system can be identified by the use of the discrete outputs, after a finite number $K > 0$ of discrete transitions. In this definition, the number K is generic. However, in our application, it is necessary to immediately identify those discrete locations – that we may call *critical* – that correspond to dangerous situations. If a critical state occurs before K transitions take place, then the corresponding critical situation is not identified even though the system is current–state observable. In the case of the runway crossing example, the theory applies well because, after the signatures generation, the hybrid system model is current–state observable with $K = 1$. However, this is not necessarily the case. It is therefore necessary to extend the definition of observability to a subset of critical states of the agent hybrid system, and to design an observer based on this definition to verify the observability of critical states. This will answer the objective of Task 7.4, i.e.

Fault and error detection in prescribed time horizon. Time delay in fault or error detection and identification is critical and no results are available in the literature on this particular problem for hybrid systems. Our objective is to extend previous results in order to assess fault detection within a given maximal interval of time and to design a fault/error–tolerant control strategy. Specific communication network related air traffic management problems will be considered in this study.

To solve the problem of critical observability, we build on the work presented in [1], and the one on fault and error detection in prescribed time horizon [9], [13]. To do so, we extend the definition of observability to a subset of critical states of the agent hybrid system to yield the concept of *critical observability*. We then present how to design an observer based on this definition to verify the observability of critical states.

The report is organized as follows. In Section 2, we formulate the problem and we review results on observability for hybrid systems. In Section 3, we introduce the notion of *critical observer* and we offer conditions under which critical observers may be designed. In Section 4, we apply these results to the runway crossing problem and we show experimental results based on extensive Matlab simulation. In Section 5, we offer some concluding remarks.

2 Problem Setting

We consider a hybrid system $\mathcal{H}$ with N locations $q_1, \cdots, q_N$. Each location identifies the continuous dynamics described by the equations

$$\dot{x} = A_i x + B_i u, \qquad y = C_i x, \qquad i = 1, \cdots, N \tag{1}$$

with $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times m}$, $C_i \in \mathbb{R}^{p\times n}$, $x \in X \subseteq \mathbb{R}^n$ the continuous state, $y \in Y \subseteq \mathbb{R}^p$ the continuous output, and $u \in U \subseteq \mathbb{R}^m$ the system input. As in [1], we suppose here that systems (1) are observable, although this assumption may be relaxed.

The discrete event dynamics are given by a nondeterministic generator of formal language [17]

$$\begin{aligned} q(k+1) &\in \delta(q(k), \sigma(k+1)) \\ \sigma(k+1) &\in \phi(q(k)) \\ \psi(k) &= \eta(\sigma(k)) \end{aligned} \tag{2}$$

with $q(k) \in Q$ the discrete location, $\psi(k) \in \Psi$ the output symbol, $\sigma(k) \in \Sigma$ the k^{th} input symbol, which takes place at time t_k and forces the discrete evolution. Here $Q = \{q_1, \cdots, q_N\}$, $\Sigma = \{\sigma_1, \cdots, \sigma_s\}$, $\Psi = \{\epsilon, \psi_1, \cdots, \psi_r\}$, with ϵ the null event, are the finite sets of locations, input and output symbols. Moreover,

$$\delta \colon Q \times \Sigma \to 2^Q, \qquad \phi \colon Q \to 2^\Sigma, \qquad \eta \colon \Sigma \to \Psi$$

are the transition, the input, and the output functions (in general these are partial functions). The function ϕ specifies the possible input events σ. The functions δ, η can be extended in the usual way to accept sequences $s_k = \sigma_1 \cdots \sigma_{k-1}\sigma_k \in \Sigma^*$, with Σ^* the monoid on Σ [17]:

$$\delta(q, \sigma_1 \cdots \sigma_{k-1}\sigma_k) = \bigcup_{q'} \delta(q', \sigma_k)$$

for $q' \in \delta(q, \sigma_1 \cdots \sigma_{k-1})$ and $\delta(q', \sigma_k)!$ ("!" indicates that the partial function is defined for the given arguments). If $s_m = \sigma_1\sigma_2 \cdots \sigma_m$ is an input sequence of lenght m, the measured output is $p_{\bar{m}} = \psi_1\psi_2 \cdots \psi_{\bar{m}}$, where $\bar{m} \leq m$ since some ψ_i can be the null event ϵ.

Some authors consider the output function η depending also on the state. This allows to consider different outputs for the same input σ defined for different states $q \in Q$. However, by renaming σ as different inputs, one can define a new output function depending only on the input event.

The hybrid system $\mathcal{H}$ considered here is described by systems (1), (2). The action of the discrete dynamics on the continuous ones is the change of the equations (1) when a location transition takes place. On the other hand, the action of the continuous dynamics on the discrete ones is the change of location when the continuous state x and/or the continuous control u belong to some region or when the system trajectory hits some boundary.

The event σ can be a disturbance (the corresponding transition is then called switching transition) or may be generated when some invariant condition on the continuous state x is no more satisfied (invariant transition). Switching and invariant transitions are uncontrollable. Therefore, the switching times at which the input events occur, are unknown a priori.

The output sequence $\psi_1\psi_2 \cdots \psi_k$ of $\mathcal{H}$ can be used to determine the current discrete state q at intermittent time instants (possibly not at each time instant). In the following definition $|\cdot|$ denotes the set cardinality. A prefix of length l of a sequence $\sigma_1\sigma_2 \cdots \sigma_l \cdots \sigma_m$ is the sequence $\sigma_1\sigma_2 \cdots \sigma_l$ of the first l events.

The definition of observability proposed in [15] states that for long enough input sequences $s_m = s_l\sigma_{l+1} \cdots \sigma_m$ generable from q, there exists a shorter input

sequence $s_l = \sigma_1 \cdots \sigma_l$ (a prefix) that takes q to a unique state $\delta(q, s_l)$ and the length of $\sigma_{l+1} \cdots \sigma_m$ is bounded. Moreover, if another sequence $\bar{s}_{\bar{l}} = \bar{\sigma}_1 \cdots \bar{\sigma}_{\bar{l}}$ from some $\bar{q}$ has the same output, then $\bar{s}_{\bar{l}}$ has to bring $\bar{q}$ to the same state to which s_l takes q. More formally,

Definition 1. [15] *The discrete state of $\mathcal{H}$ is observable if*

1. *There exists some integer k such that for all $q \in Q$ and for all the sequences $s_m = \sigma_1 \cdots \sigma_l \cdots \sigma_m$ generable from q there exists a prefix sequence $s_l = \sigma_1 \cdots \sigma_l$, with $\eta(\sigma_l) \neq \epsilon$ and $m - l \leq k$, such that $|\delta(q, s_l)| = 1$, and*
2. *For each $\bar{q} \in Q$ and for each sequence $\bar{s}_{\bar{l}} = \bar{\sigma}_1 \cdots \bar{\sigma}_{\bar{l}}$, with $\eta(\bar{\sigma}_{\bar{l}}) \neq \epsilon$, generable from $\bar{q}$ and such that $\eta(s_l) = \eta(s_{\bar{l}})$, then $\delta(q, s_l) = \delta(\bar{q}, \bar{s}_{\bar{l}})$.*

□

In [15] a procedure is proposed for the construction of a finite state machine $\mathcal{O}$ that, under appropriate conditions, allows the observation of the discrete state of $\mathcal{H}$ according to Definition 1. In ATM, an intermittent detection of the discrete state is not acceptable. Thus, we propose a construction procedure (analogous to the one used in [1], [6] and [7]) of a finite state machine $\mathcal{O}$ and give conditions under which $\mathcal{O}$ is an observer, i.e. is able to detect the discrete state $q(k)$ of $\mathcal{H}$ *for each* k greater than a positive integer K.

The procedure is based on the iterative construction of $\mathcal{O} = \{\hat{Q}, \hat{\Sigma}, \hat{\Psi}, \hat{\delta}, \hat{\phi}, \hat{\eta}\}$, where $\hat{\Sigma} = \Psi \setminus \{\epsilon\}$ is the set of inputs (namely the outputs of $\mathcal{H}$). We define the input function of $\mathcal{O}$ as

$$\hat{\phi}(\hat{q}) := \left\{ \psi \in \hat{\Sigma} : \exists\, \bar{q} \in \hat{q},\ \sigma \in \Sigma : \sigma \in \phi(\hat{q}) \text{ and } \eta(\sigma) = \psi \right\}$$

and the transition function as

$$\hat{\delta}(\hat{q}, \psi) := \left\{ q \in Q : \exists\, \bar{q} \in \hat{q},\ s \in \Sigma^* : q \in \delta(\bar{q}, s)! \text{ and } \eta(s) \in \psi\epsilon^* \right\}$$

where $\psi\epsilon^* = \{\psi, \psi\epsilon, \psi\epsilon\epsilon, \cdots\}$. We recall that [15] assumes that it is not possible for $\mathcal{H}$ to generate arbitrarily long sequences of ϵ outputs, thus $\psi\epsilon^*$ is a finite string. The output set $\hat{\Psi}$ and the output function $\hat{\eta}$ are trivially defined such that the output of $\mathcal{O}$ is the current state $\hat{q} \in \hat{Q}$. Furthermore, let

$$\hat{Q}_0 = Q_0 \bigcup \left\{ q \in Q : \exists\, \bar{q} \in Q_0,\ s \in \Sigma^* : q \in \delta(\bar{q}, s)! \text{ and } \eta(s) \in \epsilon^* \right\}$$

be the initial state $\hat{q}(0)$ of $\mathcal{O}$, where Q_0 is the set of possible initial states of $\mathcal{H}$ and $\epsilon^* = \{\epsilon, \epsilon\epsilon, \epsilon\epsilon\epsilon, \cdots\}$.

To complete the construction of $\mathcal{O}$, we determine the state set $\hat{Q}$ via the following algorithm, where $A, B \subset 2^Q$:

set $A = \hat{Q} = \not\phi$;
add $\hat{Q}_0$ *to* A *and to* $\hat{Q}$;
do until $A = \not\phi$
{
    $B = \not\phi$;
    for each $\hat{q} \in A$ *do*
    {
        for each $\psi \in \hat{\phi}(\hat{q})$ *do*
        {
            if $\hat{\delta}(\hat{q}, \psi) \notin \hat{Q}$ *then*
                add state $\{\hat{\delta}(\hat{q}, \psi)\}$ *to* B *and to* $\hat{Q}$;
        }
    }
    $A = B$;
}

Our *observation problem* is as follows:

Definition 2. *Given a hybrid system* $\mathcal{H}$*, the system* $\mathcal{O}$ *is said to be an observer for the discrete states of* $\mathcal{H}$ *if there exists an integer* K *such that*

$$\hat{q}(k) = \{q\} \qquad \text{if } q(k) = q, \ \forall k \geq K \tag{3}$$

for every initial state $(q_0, x_0) \in Q \times X$ *of the hybrid system* $\mathcal{H}$*, every continuous input function* u*, every discrete input* $s_k = \sigma_1, \cdots, \sigma_k$. □

To state the conditions under which $\mathcal{O}$ is an observer for $\mathcal{H}$, i.e. is able to detect the discrete state $q(k)$ for k greater than a positive integer, K we need the following definition.

Definition 3. [15] *A given subset* $E \subset Q$ *is invariant with respect to a function* $\delta: Q \times \Sigma \to 2^Q$ *if* $\delta(q, \sigma) \subseteq E$ *for all* $q \in E$ *and* $\sigma \in \phi(q)$*. Moreover, a state* $q \in Q$ *is* E*–prestable if every trajectory starting from* q *passes through* E *in a finite number of transitions. The state* $q \in Q$ *is* E*–stable if every state reachable from* x *is* E*–prestable. A discrete event dynamic system is* E*–stable if every* $q \in Q$ *is* E*–stable.* □

Let $\tilde{Q} := \left\{\{q\} : q \in Q\right\} \cap \hat{Q}$ be the set of singleton states of $\mathcal{O}$:

Proposition 1. *$\mathcal{O}$ is an observer for the discrete states of $\mathcal{H}$ if and only if*

1. $\exists E \subseteq \tilde{Q}$ *non–empty and invariant with respect to the dynamics of* $\hat{\delta}$*;*

2. *$\mathcal{O}$ is E–stable.* □

The conditions above are quite intuitive: the first one requires that $\mathcal{O}$ has a subset of singleton states (namely states with cardinality equal to one), and that the discrete event dynamics do not drive the state out of this set; the second one requires that any discrete evolution drives the state to the set E in finite time. These conditions are necessary and sufficient for determining, after a transient, the precise discrete state of $\mathcal{H}$.

When the conditions given in Proposition 1 are violated, it is not possible to determine the discrete state of $\mathcal{H}$ for k greater than a certain positive integer K, at least with a pure discrete event–driven observer. This is due to the fact that either an invariant set $E \subseteq \tilde{Q}$ does not exist, namely $\hat{\delta}$ drives to a state $\hat{q} = \{q_{i_1}, \cdots, q_{i_r}\}$ with cardinality greater than 1, or the evolution of $\mathcal{H}$ does not drive the observer discrete state in the invariant set of singletons E in finite time. In this case, as proposed in [1], one can exploit the knowledge coming from the continuous dynamics to create further discrete signals (called "signatures") that provide additional information to discriminate the discrete locations. Clearly, this extra information must be "rich enough" to determine an observer that satisfies the conditions of Proposition 1.

This idea is carried out in [1] as follows: appropriate Luenberger's observers are designed for each of the continuous dynamics (1). Then, the signatures $\bar{\psi}_1, \cdots, \bar{\psi}_s$ are obtained by feeding the observer outputs into a decision function block. Each label $\bar{\psi} \in \bar{\Psi} = \{\bar{\psi}_1, \cdots, \bar{\psi}_s\}$ is characteritics of a specific location q and is added as output to the arcs entering q. With this change in $\mathcal{H}$ one can obtain a finite state machine $\mathcal{O}$ that satisfies Proposition 1.

The task of the signature generator is similar to that of a fault detection algorithm and is not discussed here (see [13] for a tutorial). The key point from the observability point of view is that signatures have to be generated before the system leaves the discrete state. In [1], it is shown how the observer's gains have to be chosen so that the signatures are generated within a finite and fixed time, namely the so–called *minimum dwell–time*.

To define formally the dwell–time, we recall [12] that a hybrid time basis $\tau = \{I_j\} \in \mathcal{T}$, $j \in \mathbb{N}$, of $\mathcal{H}$ is a finite or infinite sequence of intervals $I_j = [t_j, t'_j]$ such that

1. I_j is closed if τ is infinite; I_j might be right–open if it is the last interval of a finite sequence τ;
2. $t_j \leq t'_j$ for all $j \in \mathbb{N}$ and $t'_{j-1} \leq t_j$ for $j > 0$.

The length of the hybrid time basis is $|\tau|$.

Given a hybrid system $\mathcal{H}$ and a time basis τ, we suppose that for each state $q \in Q$, there exists a minimum dwell time $\Delta_m(q)$ such that

$$0 < \Delta_m(q) \leq t'_j - t_j, \qquad \forall j \in \big[0, |\tau| - 1\big]$$

with $q(I_j) = q$, $q(I_{j+1}) \neq q$, where, with abuse of notation, $q(I_j)$ is the state for $t \in I_j$. Roughly speaking, The minimum dwell time for $\mathcal{H}$ is the minimum

time elapsed between two consecutive transitions, namely the minimum time of permanence in a given state of $\mathcal{H}$.

Our point of view on the signature generation mechanism is slightly different from [1]: instead of associating signatures to the transitions, we associate to each state $q \in Q$ an additional output value $\bar{\psi} = h(q) \in \bar{\Psi}$ depending on the state q and we suppose that $\bar{\psi}$ is generated within the minimum dwell–time $\Delta_m(q)$. In this way the generation dynamics is "hidden" inside the delay necessary to generate $\bar{\psi} = h(q)$, and we can neglect the signatures generator dynamics.

The hypothesis of generating a signature within the maximum dwell time $\Delta_m(q)$ for each state q is quite strong. We can relax it, by considering a "nonzero" signature only when it is strictly necessary for the design of the observer. This means that some signatures could possibly be the null event ϵ. For instance, if the conditions of Proposition 1 are satisfied without generating any signature, one has $h(q) = \epsilon$ for all $q \in Q$.

¿From the previous discussion, we can redefine the discrete output (2) of $\mathcal{H}$ as follows

$$\psi(k) = \begin{cases} \eta(\sigma(k)) & \text{if } h(q(k)) = \epsilon \\ h(q(k)) & \text{otherwise} \end{cases}$$

enlarging Ψ to contain $\bar{\Psi}$.

The point of view adopted here for the signatures allows us to consider into the same framework definitions such as the co–observability [16], also called eventual observability in [14].

3 Critical Observers

As already pointed out, the notion of observability introduced in the previous section does not capture the urgency of a dangerous situation that may be created by an error in an ATM system. In this case, we need to identify the states corresponding to these errors immediately, i.e., K must be 1. Consider the discrete event system $\mathcal{H}$ and the corresponding observer $\mathcal{O}$ of Figure 1. The arcs of $\mathcal{H}$ are labelled by the output symbols ("ϵ" denotes the "null" output), while the arcs of $\mathcal{O}$ are labelled by the input symbols. Suppose that $q_1$is the initial state of $\mathcal{H}$ and that q_3 is a critical state (i.e. q_3 corresponds to a dangerous situation). It is clear from the structure of $\mathcal{O}$ that it is possible to determine the current state of the system only after $K = 2$ state transitions but not for $K = 1$. In this section, we give conditions for the existence of hybrid observers that guarantee observability with $K = 1$. Similar results are presented in [14], where a definition of *immediate observability* is introduced, and necessary and sufficient conditions are given to satisfy this property. Our results differ in two aspects: (i) Immediate observability is required for all the states of the system, while here we are looking for milder conditions regarding the observability of those discrete states marked as "critical", namely connected with a possible hazardous situation for the process the system is modelling. (ii) More than on the analysis of a given system, we are interested in the extra information needed to make the property of critical observability hold.

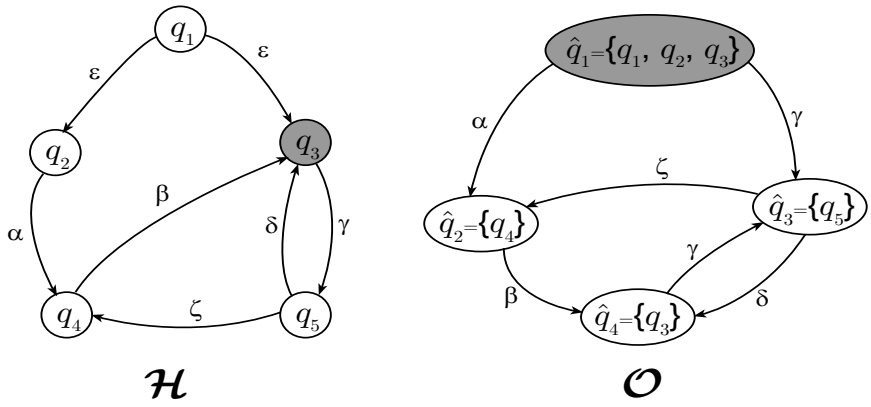

Fig. 1. $\mathcal{H}$ and the corresponding observer $\mathcal{O}$: critical states are filled in grey

A state $q_c \in Q$ is said to be *critical for a hybrid system* $\mathcal{H}$ if it corresponds to a hazardous operation. Let Q_c be the set of critical states for $\mathcal{H}$.

Definition 4. *Given a hybrid system* $\mathcal{H}$ *and a subset* $Q_c \subseteq Q$*, the system* $\mathcal{O}$ *is said to be a* critical observer *for* $\mathcal{H}$ *with respect to the set of states* Q_c *if*

$$\hat{q}(I_k) = \{q\} \qquad \forall q \in Q_c, \quad \forall I_k : q(I_k) = q, \tag{4}$$

for every initial state $(q_0, x_0) \in Q \times X$ *of the hybrid system* $\mathcal{H}$*, every continuous input function* u*, every discrete input* $s_k = \sigma_1, \cdots, \sigma_k$*.* □

A critical state for $\mathcal{H}$ induces the notion of *critical states for the observer* $\mathcal{O}$ as follows. Consider the system $\mathcal{O} = \{\hat{Q}, \hat{\Sigma}, \hat{\Psi}, \hat{\delta}, \hat{\phi}, \hat{\eta}\}$ as defined in the previous section. We recall that each discrete state $\hat{q} \in \hat{Q}$ of $\mathcal{O}$ is a non–empty set of states $q_{j_1}, \cdots, q_{j_r}$ of $\mathcal{H}$.

Definition 5. *A state* $\hat{q} \in \hat{Q}$ *is critical for* $\mathcal{O}$ *if* $\hat{q} \cap Q_c \neq \emptyset$ *and* $|\hat{q}| > 1$*.* □

where $|\hat{q}|$ denotes the cardinality of $\hat{q} \in 2^Q$ as a subset of Q.

Let $\hat{Q}_c \subseteq \hat{Q}$ be the set of the induced critical states for $\mathcal{O}$. As a consequence of the definition above, $\mathcal{O}$ is a critical observer for $\mathcal{H}$ with respect to a set $Q_c \subset Q$ if $\hat{Q}_c = \emptyset$.

The observation problem clearly arises when $\hat{Q}_c \neq \emptyset$ and $\mathcal{O}$ is in a critical state $\hat{q}_c = \{q_{j_1}, \cdots, q_{j_r}\} \in \hat{Q}_c$. In fact, in this case we need to discriminate among the states $q_{j_1}, \cdots, q_{j_r}$ to determine if the hybrid system is in a critical state of Q_c or not. The following proposition gives a condition under which the observer obtained using the constructive procedure illustrated in the previous section is critical.

Proposition 2. *$\mathcal{O}$ is a critical observer for $\mathcal{H}$ with respect to a set $Q_c \subset Q$ if for each $q_c \in Q_c$ and each induced critical state $\hat{q}_c \in \hat{Q}_c$ such that $q_c \in \hat{q}_c$*

$$h(q_c) \neq h(\bar{q}) \quad \forall \bar{q} \in \hat{q}_c, \bar{q} \neq q_c$$

□

Proof. By construction of $\mathcal{O}$.

The result above will be used in the following section for the design of a critical observer for the runway crossing system.

4 A Case Study: the Active Runway Crossing System

In this section, we review the runway crossing example analyzed in Public Deliverable D7.3 [18], that was modified following the suggestions and comments by Ted Lewis (BAE Systems) and Derek Jordan (BAE Systems). The purpose is to illustrate the results of our work on the observability theory with the help of a model of a realistic problem, without claiming that the model is close to reality.

With respect to the model proposed in D7.3, we introduced two main differences: first, the taking off aircraft can be authorized to execute power up and takeoff without stopping at holding. Thus, the observer must detect if a power up and takeoff without stopping is due to a situation awareness error or to a command of the Tower Controller. Second, the crossing grant is given by the Ground Controller, and not by the Tower Controller as in Deliverable D7.3. Thus, a situation awareness error of one of the two controllers could lead to simultaneous takeoff and crossing grants. These differences make the observability problem more complex to solve, and justify the introduction of new theoretical elements, such as the new definition of critical observability and the introduction of the shuffle product between two discrete event systems. In this section, agents will be formalized by hybrid systems or DEDS, and the observability problem will be analyzed using the techniques introduced in the previous Chapter. The active runway crossing will be decomposed into a set of subsystems, each with hybrid dynamics modeling its specific operations.

The active runway crossing environment consists of a runway A (with holdings, crossings and exits), a maintenance area and aprons. The crossings enable traffic between the aprons and the maintenance area. Crossings (on both sides) and holdings have remotely controlled stopbars to access the runway, and each exit has a fixed stopbar (see Figure 2).

The following relevant areas can be defined

$$\begin{aligned}
\Omega_{Ap} &= \{(x,y) \mid x > a_4, y \in [b_1, b_6]\} \\
\Omega_{AW_1} &= \{(x,y) \mid x \in [a_3, a_4], y \in [b_1, b_2]\} \\
\Omega_{AW_2} &= \{(x,y) \mid x \in [a_3, a_4], y \in [b_3, b_4]\} \\
\Omega_{AW_3} &= \{(x,y) \mid x \in [a_3, a_4], y \in [b_5, b_6]\} \\
\Omega_{S_1} &= \{(x,y) \mid x \in [a_2, a_3], y \in [b_1, b_2]\} \\
\Omega_{S_2} &= \{(x,y) \mid x \in [a_2, a_3], y \in [b_3, b_4]\} \\
\Omega_{S_3} &= \{(x,y) \mid x \in [a_2, a_3], y \in [b_5, b_6]\} \\
\Omega_{H_1} &= \{(x,y) \mid x \in [a_1, a_2], y \in [b_1, b_2]\}
\end{aligned}$$

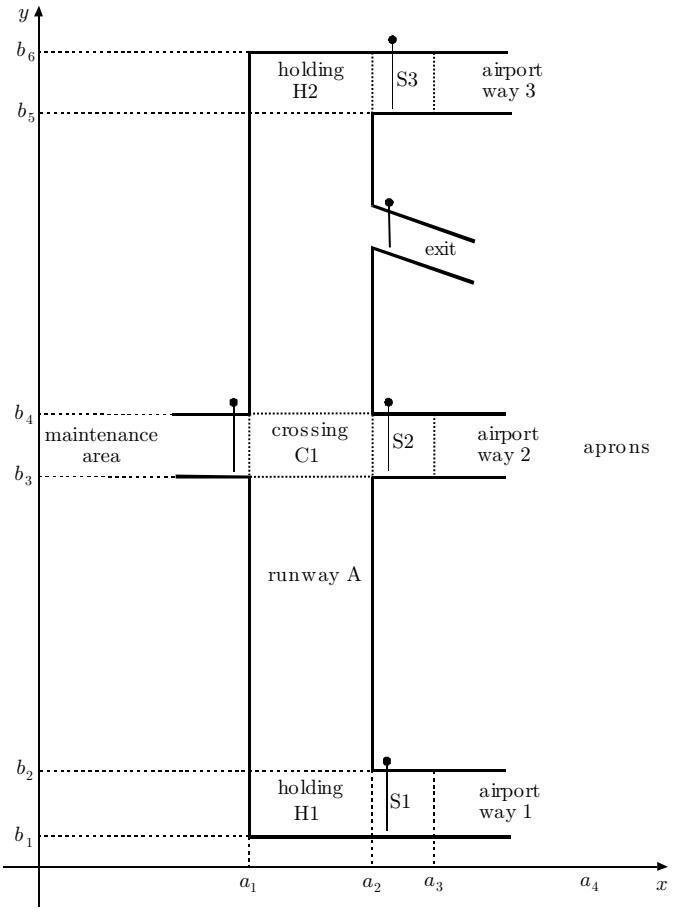

Fig. 2. Airport configuration

$$\Omega_{H_2} = \{(x, y) \mid x \in [a_1, a_2], y \in [b_5, b_6]\}$$
$$\Omega_{C_1} = \{(x, y) \mid x \in [a_1, a_2], y \in [b_3, b_4]\}$$
$$\Omega_{RW_A} = \{(x, y) \mid x \in [a_1, a_2], y \in [b_1, b_6]\}$$
$$\Omega_M = \{(x, y) \mid x < a_1, y \in [b_3, b_4]\}$$

where "Ap" stands for aprons, "AW" for airport way, "S" for stopbar, "H" for holding, "C" for crossing, "RW_A" for runway A and "M" for maintenance area. One of the key problems in distributed safety critical systems is that humans can have errors in their "Situation Awareness" (SA) [8], [18], and these errors can then evolve into the system and create safety critical situations. Situation Awareness may be defined as follows.

Definition 6. *Situation Awareness (SA) is the perception of elements in the environment within a volume of time and space, the comprehension of their*

meaning, and the projection of their status in the near future. The projection in the near future of the perception of the actual environment is referred to as intent SA. □

Within the ATM system, Stroeve *et al.* [18] define an *agent* as an entity, such as a human operator or a technical system, which is characterized by its *SA* of the environment. Similar as in [18], in [8], *SA* can be incomplete or inaccurate, due to three different situations: an entity may

1. wrongly perceive task–relevant information or miss them completely;
2. wrongly interpret the perceived information;
3. wrongly predict a future status.

An important source of error that has to be considered when analyzing multi–agent environments is the propagation of erroneous situation awareness due to agents interactions, e.g. via VHF communication.

4.1 Agents in an active runway crossing

The runway crossing operation consists of

1. a pilot flying (P_t) directed to RW_A to perform a take off operation;
2. a pilot flying (P_c) directed to the M, taxing through AW_2 and the runway crossing C_1;
3. a ground controller (C_g);
4. a tower controller (C_t);
5. the airport technical support system (ATS).

The pilot P_t proceeds towards the holding area (regular taxiway) with the intent of completing a take off operation, while the pilot P_c is approaching the crossing area. The tower controller C_t and ground controller C_g, with the aid of visual observation of the runway and VHF communication, respectively, are responsible of granting take off and crossing, avoiding the use of the runway by two aircrafts simultaneously. Technical support systems help the pilots and the controllers to communicate (VHF) and detect dangerous situations (alerts).

The specific behavior of these agents in the runway crossing operation may be described as follows

1. *Pilot flying of taking off aircraft* P_t. Initially P_t executes boarding and waits for start up grant by C_g. He begins taxiing on AW_1, stops at stopbar S_1 and communicates with the C_t at the reserved frequency to obtain take off grant. Depending on the response, P_t waits for grant or executes take off immediately. Because of a *SA* error, the take off could be initiated without grant. For simplicity, we will not consider this kind of error in this work. When the aircraft is airborne, he confirms the take off has been completed to C_t. During take off operations, P_t monitors the traffic situation on the runway visually and via VHF. If a crossing aircraft is observed or in reaction to an emergency braking command by the controller the P_t starts a braking action and so take off is rejected.

2. *Pilot Flying of crossing aircraft* P_c. When start up is granted by C_g, the P_c proceeds on the AW_2 and stops at stopbar S_2. He asks to C_g crossing permission and crosses when granted. While proceeding towards the AW, he may have the *intent SA* that the next AW point is either a regular taxiway (erroneous *intent SA*) or a runway crossing. In the first case, P_c enters RW_A without waiting for crossing permission. In the second case, P_c could have the *SA* that crossing is allowed while it is not. Then, he would enter the runway performing an unauthorized runway crossing. The reaction of P_c to the detection of a collision risk, due to visual observation or a tower controller call, is an emergency braking action.
3. *Ground Controller* C_g. C_g is a human operator supported by visual observation and by the *ATS* system. He grants start up to both to P_t and P_c, and handles crossing operations on RW_A. If C_g has *SA* of a collision risk, C_g specifies an emergency braking action to the crossing aircraft.
4. *Tower Controller* C_t. C_t is a human operator supported by visual observation and by the *ATS* system. The C_t handles take off operations on RW_A. If the C_t has *SA* of a collision risk, he specifies an emergency braking action to the taking off aircraft.
5. *ATS system*. This is the technical system supporting the decisions of the controllers, and consists of a communication system, a runway incursion alert and a stopbar violation alert.

4.2 Pilot flying observation problem

In this section, we will solve the observation problem of a non–granted runway crossing or take–off of the pilots. All the agents can be modelled either as hybrid systems or as discrete event systems [5]. In particular, P_t can be modelled as a hybrid system $\mathcal{H}_{P_t}$ with

- $Q_1 = \{q_{1,1}, q_{1,2}, q_{1,3}, q_{1,4}, q_{1,5}, q_{1,6}, q_{1,7}, q_{1,8}\}$ the set of discrete states with $q_{1,1}$ the P_t communicating with C_g and waiting for start up grant, $q_{1,2}$ the P_t taxiing on AW_1, $q_{1,3}$ the P_t aborting taxi, $q_{1,4}$ the P_t at hold H_1, $q_{1,5}$ the P_t executing an authorized take off on RW_A, $q_{1,6}$ the P_t lined up and waiting for take off grant, $q_{1,7}$ the P_t executing an unauthorized take off on RW_A, $q_{1,8}$ the P_t executing the initial climb, $q_{1,9}$ the P_t aborting take off (emergency braking);
- $\Sigma_1 = \{\sigma_{1,1}, \sigma_{1,2}, \sigma_{1,3}, \sigma_{1,4}, \sigma_{1,5}, \sigma_{1,6}, \sigma_{1,7}\}$ the set of discrete inputs, where $\sigma_{1,1}$ models the start up clearance by C_g, $\sigma_{1,2}$ the command for immediate take off by C_t, $\sigma_{1,3}$ the command to line up and wait by C_t, $\sigma_{1,4}$ the take off clearance by C_t, $\sigma_{1,5}$ an emergency braking command by C_t, $\sigma_{1,6}$ models a situation awareness error as a disturbance that causes an ungranted take off, and $\sigma_{1,7}$ is a disturbance that causes a taxi abort;
- $\Psi_1 = \{\psi_{1,1}, \psi_{1,2}, \psi_{1,3}, \psi_{1,4}, \psi_{1,5}, \psi_{1,6}, \psi_{1,7}, \psi_{1,8}\} \cup \{\epsilon\}$ the set of discrete outputs, with $\psi_{1,1}$ the start up confirmation to C_g, $\psi_{1,2}$ the take off request, $\psi_{1,3}$

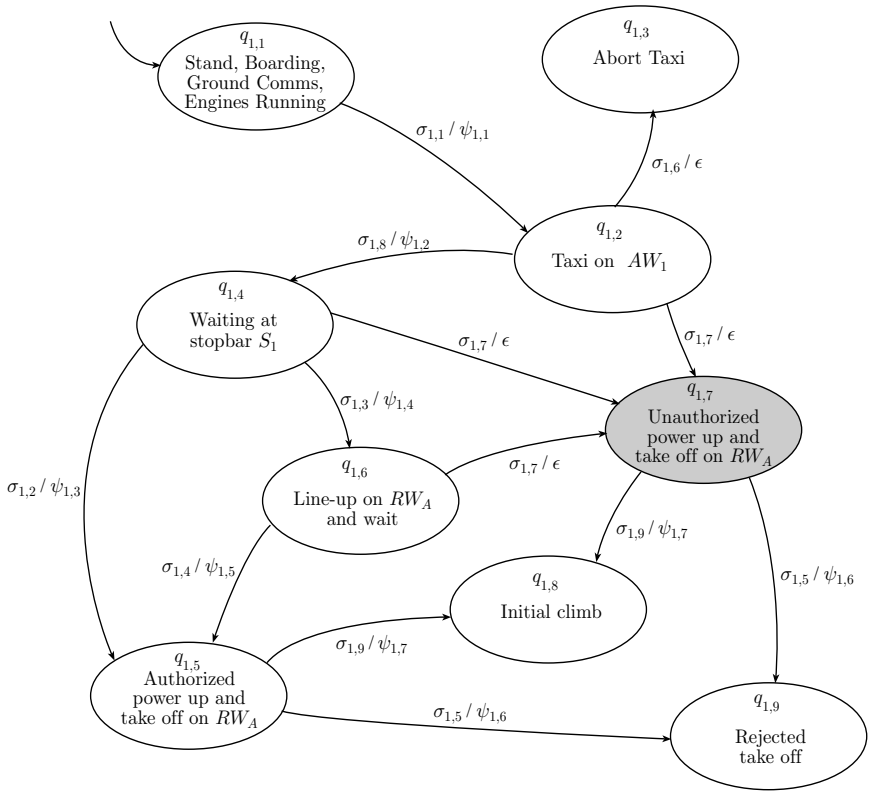

Fig. 3. Hybrid system $\mathcal{H}_{P_t}$ modelling P_t

the immediate take off confirmation, $\psi_{1,4}$ the line–up and wait confirmation, $\psi_{1,5}$ the take off confirmation, $\psi_{1,6}$ the emergency braking confirmation, $\psi_{1,7}$ the airborne confirmation;

- $X_1 = \{(s_1, v_1) : s_1 \in \mathbb{R}^2, v_1 \in \mathbb{R}^2\}$, is the set of the continuous state values, where s_1 indicates the position and v_1 the velocity of the agent;
- $U_1 = \mathbb{R}^m$, is the set of the continuous input u_1 values, $D_1 = \mathbb{R}^p$ is that of the continuous disturbance d_1 values;
- $S_{C_1} = \{f_{q_{j,1}} : q_{j,1} \in Q_1\}$, $f_{q_{j,1}} : X_1 \times U_1 \times D_1 \to T_{X_1}$, the sets of the continuous (simplified) dynamics $\dot{s}_1 = v_1$, $\dot{v}_1 = u_1(t) + d_1(t)$, where d_1 represents possible disturbance forces acting on the aircraft (e.g. wind);
- E_1 the sets of discrete transitions, given by the graph in Figure 3;
- $\eta_1(\cdot)$ the discrete output function, defined by the graph in Figure 3, where the outputs corresponding to transitions due to situation awareness errors ($e_{1,7}$, $e_{1,8}$ and $e_{1,9}$) are null;

– The invariant conditions are defined as

$$
\begin{aligned}
I_{q_{1,1}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{Ap}, \|v_1\| = 0\} \\
I_{q_{1,2}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{AW_1} \cup \Omega_{S_1}, \|v_1\| > 0\} \\
I_{q_{1,3}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{AW_1} \cup \Omega_{S_1}, \|v_1\| = 0\} \\
I_{q_{1,4}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{S_1}, \|v_1\| = 0\} \\
I_{q_{1,5}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{RW_A}, \|v_1\| > 0\} \\
I_{q_{1,6}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{H_1}, \|v_1\| \geq 0\} \\
I_{q_{1,7}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{RW_A} \cup \Omega_{S_1}, \|v_1\| > 0\} \\
I_{q_{1,8}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{RW_A}, \|v_1\| > v_t\} \\
I_{q_{1,9}} &= \{(s_1, v_1)\colon s_1 \in \Omega_{RW_A}, \|v_1\| \geq 0\}
\end{aligned}
$$

  where v_t is the takeoff velocity and Ω's are defined by the airport configuration geometry;

– $R_1(e, x, u, v) = x$, $\forall (e, x, u, v) \in E_1 \times X_1 \times U_1 \times D_1$ are the reset mappings;

– The guard conditions are

$$
\begin{aligned}
G_{e_{1,3}} &= \{(s_1, v_1)\colon s_1 \in S_1, \|v_1\| = 0\} \\
G_{e_{1,10}} &= G_{e_{1,11}} = \{(s_1, v_1)\colon s_1 \in RW_A, \|v_1\| > v_t\}.
\end{aligned}
$$

– The initial discrete state is $q_{1,1}$

Analogously, P_c can be modelled by a hybrid system with

– $Q_2 = \{q_{2,1}, q_{2,2}, q_{2,3}, q_{2,4}, q_{2,5}, q_{2,6}, q_{2,7}\}$, are the sets of discrete states where $q_{2,1}$ corresponds to P_c communicating with C_g and waiting for start up grant, $q_{2,2}$ to P_c taxiing on AW_2, $q_{2,3}$to P_c waiting at stopbar S_2, $q_{2,4}$ to P_c executing an authorized crossing of RW_A, $q_{2,5}$ to P_c executing an unauthorized crossing of RW_A, $q_{2,6}$ to P_c crossing towards M, $q_{2,7}$ to P_c performing an emergency braking operation;

– $\Sigma_2 = \{\sigma_{2,1}, \sigma_{2,2}, \sigma_{2,3}, \sigma_{2,4}, \sigma_{2,5}\}$, is the set of discrete inputs, where $\sigma_{2,1}$ models the start up clearance by the C_g, $\sigma_{2,2}$ the command by C_g to wait at stopbar S_2, $\sigma_{2,3}$ the crossing grant by C_g, $\sigma_{2,4}$ the emergency braking command by C_g, $\sigma_{2,5}$ models situation awareness error as a disturbance that causes an ungranted crossing;

– $\Psi_2 = \{\psi_{2,1}, \psi_{2,2}, \psi_{2,3}, \psi_{2,4}, \psi_{2,5}\} \cup \{\epsilon\}$, is the set of discrete outputs, with $\psi_{2,1}$ the start up confirmation, $\psi_{2,2}$ the crossing request, $\psi_{2,3}$ the RW_A crossing grant confirmation, $\psi_{2,4}$ the crossing complete confirmation, $\psi_{2,5}$ the emergency braking confirmation;

– $X_2 = \{(s_2, v_2)\colon s_2 \in \mathbb{R}^2, v_2 \in \mathbb{R}^2\}$, is the set of the continuous state values, where s_2 indicates the position and v_2 the velocity of the agent;

– $U_2 = \mathbb{R}^m$, is the set of the continuous input u_2 values, $V_2 = \mathbb{R}^p$ is that of the continuous disturbance d_2 values;

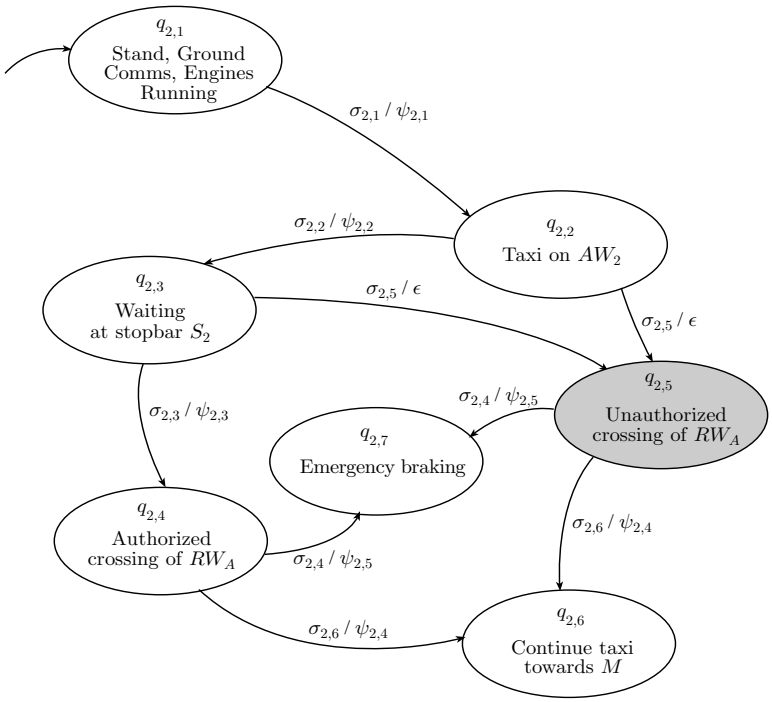

Fig. 4. Hybrid system $\mathcal{H}_{P_c}$ modelling P_c

- $S_{C_2} = \{f_{q_{j,2}} : q_{j,2} \in Q_2\}$, $f_{q_{j,2}} : X_2 \times U_2 \times V_2 \to T_{X_2}$, $j = 1, 2$, are the sets of the continuous (simplified) dynamics $\dot{s}_2 = v_2$, $\dot{v}_2 = u_2(t) + d_2(t)$, and d_2 represents possible disturbance forces acting on the aircraft (e.g. wind);
- E_2 the sets of discrete transitions, given by the graph in Figure 4;
- $\eta_2(\cdot)$ the discrete output function, defined by the graph in Figure 4, where the outputs corresponding to transitions due to situation awareness errors ($e_{2,4}$ and $e_{2,5}$) are empty and are the source of the observability problems that we need to address.
- The invariant conditions are defined as follows

$$
\begin{aligned}
I_{q_{2,1}} &= \{(s_2, v_2) : s_2 \in \Omega_{Ap}, \|v_2\| = 0\} \\
I_{q_{2,2}} &= \{(s_2, v_2) : s_2 \in \Omega_{AW} \cup \Omega_{S_2}, \|v_2\| > 0\} \\
I_{q_{2,3}} &= \{(s_2, v_2) : s_2 \in \Omega_{S_2}, \|v_2\| = 0\} \\
I_{q_{2,4}} &= \{(s_2, v_2) : s_2 \in \Omega_{C_1}, \|v_2\| > 0\} \\
I_{q_{2,5}} &= \{(s_2, v_2) : s_2 \in \Omega_{S_2} \cup \Omega_{C_1}, \|v_2\| > 0\} \\
I_{q_{2,6}} &= \{(s_2, v_2) : s_2 \in \Omega_M, \|v_2\| > 0\} \\
I_{q_{2,7}} &= \{(s_2, v_2) : s_2 \in \Omega_{C_1}, \|v_2\| \geq 0\}
\end{aligned}
$$

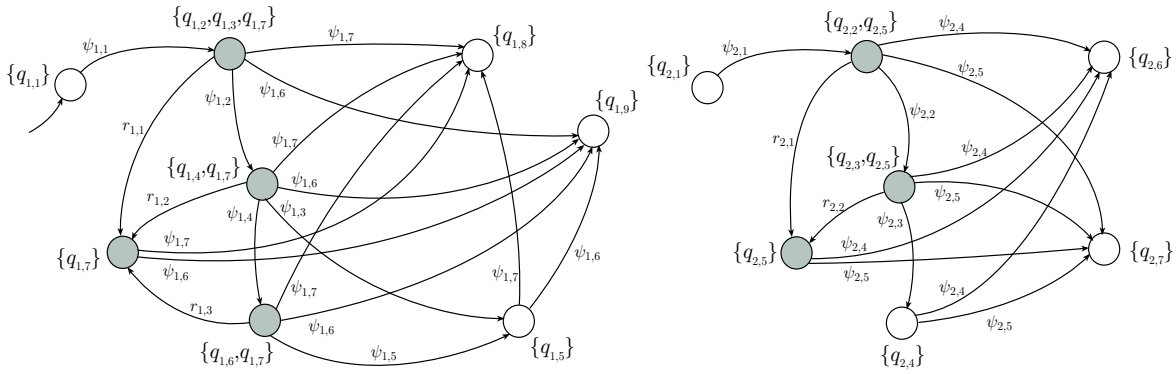

Fig. 5. Observers $\mathcal{O}_1$ and $\mathcal{O}_2$

- $R_2(e, x, u, v) = x$, $\forall (e, x, u, v) \in E_2 \times X_2 \times U_2 \times D_2$ are the reset mappings;
- The guard conditions are

$$G_{e_{2,6}} = G_{e_{2,7}} = \{(s_2, v_2) \colon s_2 \in M, \|v_2\| > 0\}.$$

- The initial discrete state is $q_{2,1}$

The simple model presented above does not consider an unsafe situation involving an emergency braking action that could result into a halt of the aircraft on the runway. However, the methods developed to solve the observability problem may be applied to complex discrete event systems that model all unsafe situations.

The critical situations for P_t, P_c, which we wish to detect, are related to the states $q_{1,7}$ and $q_{2,5}$ that represent respectively unauthorized take off and unauthorized crossing operations.

Consider now the observers $\mathcal{O}_1$ and $\mathcal{O}_2$ for $\mathcal{H}_{P_t}$ and $\mathcal{H}_{P_c}$, shown in Figure 5, and constructed using the discrete output information. Consider the critical observer state $\{q_{1,2}, q_{1,3}, q_{1,7}\}$ of $\mathcal{O}_1$. If $s_1 \in \Omega_{RW_A}$, a signature $r_{1,1}$ may be generated to distinguish $q_{1,7}$ from $q_{1,2}$ and $q_{1,3}$. In a similar way, for the other critcal states, we generate the signatures $r_{1,2}$ if $s_1 \in \Omega_{RW_A}$, $r_{1,3}$ if $s_1 \in \Omega_{RW_A} \setminus \Omega_{H_1}$, $r_{2,1}$ and $r_{2,2}$ if $s_1 \in \Omega_{C_1}$. Then, the condition of Proposition 3 is satisfied for all critical states, and $\mathcal{O}_1$ and $\mathcal{O}_2$ are critical observers for $\mathcal{H}_{P_t}$ and $\mathcal{H}_{P_c}$ w.r.t. the critical states $\{q_{1,7}\}$ and $\{q_{2,5}\}$. The discrete outputs of the observers α_1 and α_2 are alarm signals. This shows how we can solve the problem of the detection of the current location for the two pilots. In a similar way, one may solve the critical observability problem for the two pilots acting together, by considering the shuffle product of the single models $\mathcal{H}_{P_t}$ and $\mathcal{H}_{P_c}$ [10], and determine the induced critical states on this new system $\mathcal{H}$.

4.3 Controller observation problem

Consider now the observation problem of the controllers.

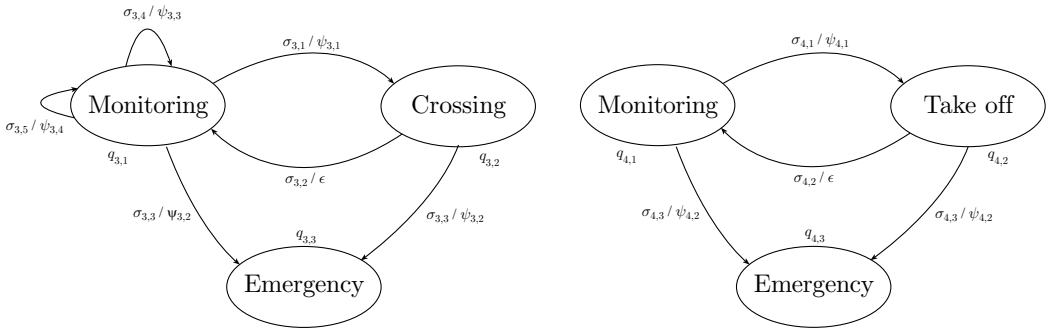

Fig. 6. DEDS modelling $\mathcal{D}_{C_g}$ and $\mathcal{D}_{C_t}$

The ground controller C_g can be modelled by a discrete event dynamic systems (DEDS) $\mathcal{D}_{C_g}$ where:

- $Q_3 = \{q_{3,1}, q_{3,2}, q_{3,3}\}$ is the set of discrete states, with $q_{3,1}$ corresponding to C_g in miscellaneous monitoring operations, $q_{3,2}$ to C_g having granted crossing, $q_{3,3}$ to an emergency braking action on the runway;
- $\Sigma_3 = \{\sigma_{3,1}, \sigma_{3,2}, \sigma_{3,3}, \sigma_{3,4}, \sigma_{3,5}\}$ is the finite set of input events, with $\sigma_{3,1}$ the decision to give a crossing grant, $\sigma_{3,2} = \psi_{2,4}$ the crossing completed confirmation, $\sigma_{3,3}$ the stopbar violation alarm on, $\sigma_{3,4}$ the decision to give a start up, $\sigma_{3,5} = \psi_{2,2}$ the crossing request;
- $\Psi_3 = \{\psi_{3,1}, \psi_{3,2}, \psi_{3,3}, \psi_{3,4}\} \cup \{\varepsilon\}$ is the set of discrete outputs, with $\psi_{3,1} = \sigma_{2,3}$ the crossing grant, $\psi_{3,2} = \sigma_{2,4}$ the emergency braking command, $\psi_{3,3} = \sigma_{1,1} = \sigma_{2,1}$ the start up grant, $\psi_{3,4} = \sigma_{2,2}$ the command to wait for crossing grant at stopbar S2;
- The input, transition and output functions ϕ_3, δ_3 and η_3 are defined by the graph in Figure 6;

The tower controller C_t can also be modelled by a DEDS $\mathcal{D}_{C_t}$ where:

- $Q_4 = \{q_{4,1}, q_{4,2}, q_{4,3}\}$ is the set of discrete states, with $q_{4,1}$ corresponding to C_t in miscellaneous operations, $q_{4,2}$ to C_t having granted take off, $q_{4,3}$ an emergency braking action on the runway;
- $\Sigma_4 = \{\sigma_{4,1}, \sigma_{4,2}, \sigma_{4,3}\}$ is the finite set of input events, with $\sigma_{4,1} = \psi_{1,2}$ the take off request, $\sigma_{4,2} = \psi_{1,7}$ the take off completed confirmation, $\sigma_{4,3}$ the runway incursion alert on;
- $\Psi_4 = \{\psi_{4,1}, \psi_{4,2}\} \cup \{\varepsilon\}$ is the set of discrete outputs, with $\psi_{4,1} = \sigma_{1,2}$ the take off grant, $\psi_{4,2} = \sigma_{1,5}$ emergency braking command;
- The input, transition and output functions ϕ_4, δ_4 and η_4 are defined by the graph in Figure 6;

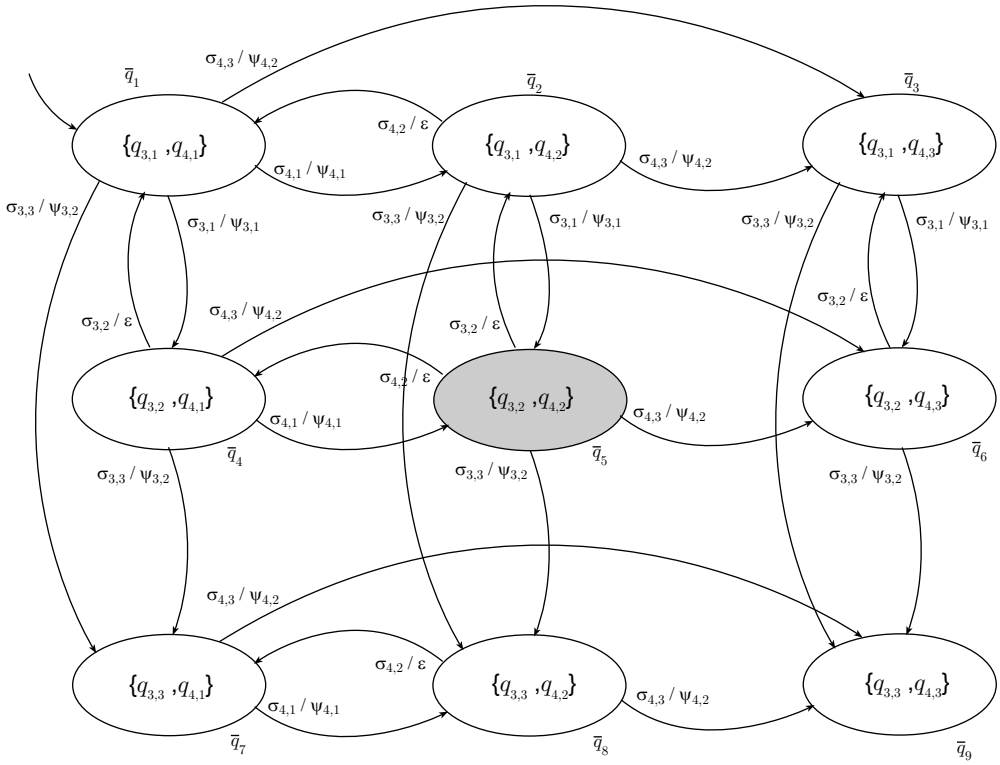

Fig. 7. Shuffle product of $\mathcal{D}_{C_g}$ and $\mathcal{D}_{C_t}$

The hazardous situation of a crossing grant given by C_g and a take off grant simultaneously given by C_t should be detected. However, the discrete event dynamic systems $\mathcal{D}_{C_g}$ and $\mathcal{H}_{C_t}$ have no critical states, because the hazardous situation arise when a crossing grant is given by C_g simultaneously with a take off grant given by C_t. Hence, it is necessary to apply the observation problem has to be applied to the shuffle product of $\mathcal{D}_{C_g}$ and $\mathcal{D}_{C_t}$ and represented in Figure 7.

The state $\bar{q}_5 = \{q_{3,2}, q_{4,2}\}$ that corresponds to simultaneous crossing grant and take off grant, is critical. The observer for this system is illustrated in Figure 8. One can see that additional information is needed in order to detect the critical state $\bar{q}_5$.

However, in a discrete event system, no continuous information is available for the generation of signatures. Hence, the only way for solving the critical state observability problem is to introduce new discrete outputs, for example, in this case, the confirmation that crossing ($\bar{\psi}_3$) or take off ($\bar{\psi}_4$) are completed. The system modified with the addition of the new discrete outputs is represented in Figure 9. This action corresponds to a change in the procedure the controllers have to follow.

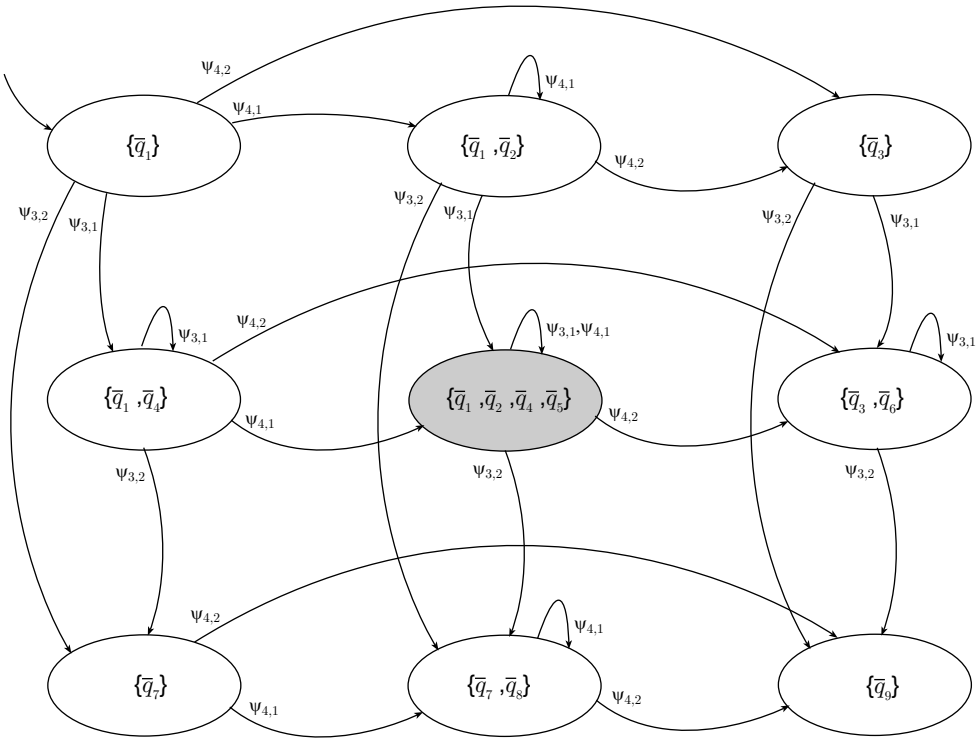

Fig. 8. Controller observer

After the addition of the new outputs, the observer of the shuffle product satisfies the critical observability condition with respect to the critical state $\bar{q}_5$, since it has no critical states. This observer is represented in Figure 10. In this case, the observer coincides with the discrete event system to be observed, because every transition has a non–null discrete output.

4.4 Simulation

Starting from Matlab executions of the mathematical models previously described, we developed a framework for generating an animated simulation of the runway crossing. The tools used to realize a graphical environment are Matlab and Visual Nastran. This choice is justified by the idea to develop separately the generation of data relative to aircraft position and dynamics (Matlab) and the visualization of the graphical output (Visual Nastran). More precisely, the choice of Matlab is motivated by the fact that all previous simulations of the Active Runway Crossing procedure were developed using "State Flow", a Matlab tool useful to model automata. Visual Nastran was chosen for two reasons: first, its graphical capabilities are superior with respect to the graphical environment

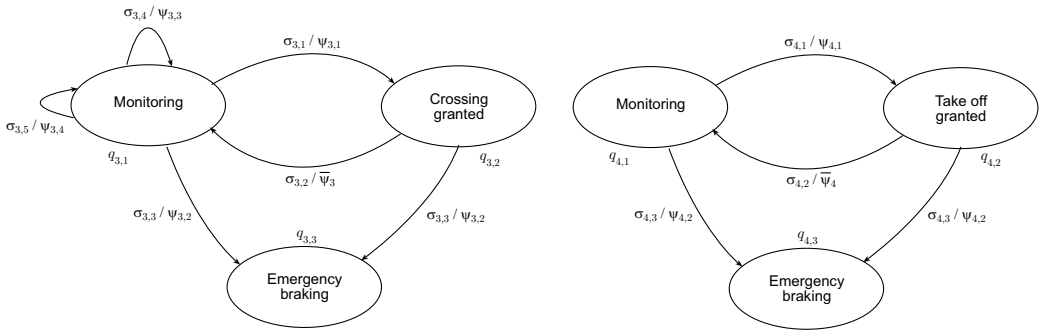

Fig. 9. Modified controllers modal states

included in Matlab; second, it may be easily interfaced with Matlab, using a proprietary interface data file.

Once created a Matlab simulation of the Runway Crossing agents, an interface between Matlab simulation data and Visual Nastran graphical generation was defined. Then, the following steps were necessary to produce an AVI graphical simulation of a Matlab system execution: first, the construction of the static model to animate, where a solid model of an aircraft (Figure 11) and of the runway configuration (see Figure 12) were generated, and each solid model of the aircraft was associated to a dynamic; thus, the animation of the static model according to the Matlab simulation. Once all these steps are executed, it is possible to analyze the simulation step by step, and to generate the animation of the Runway Crossing Procedure as an AVI file.

5 Conclusions

In this Deliverable D7.4, we introduced the notion of critical observability for hybrid systems to solve the problem of error detection in prescribed time-horizon. In particular, we gave conditions for the existence of a hybrid observer for critical states corresponding to hazardous situations. We showed how critical observability could be used in the runway crossing problem where four human agents interact in a system consisting of five subsystems. The human agents are subject to errors that may lead to catastrophic situations and are modeled as hybrid systems. We developed a hybrid observer to detect the hazardous situations corresponding to critical states and demonstrated its use with extensive Matlab simulations.

Given the relative simplicity of the case study involving the runway crossing example, the results seem rather obvious. We used this example to illustrate and verify our methodology (indeed it worked as expected). In a more complex case, intuition would not have helped: errors that we try to prevent often originate from interactions among distributed systems that, albeit simple, can create risky situations that are difficult to discern without the help of automation. Several

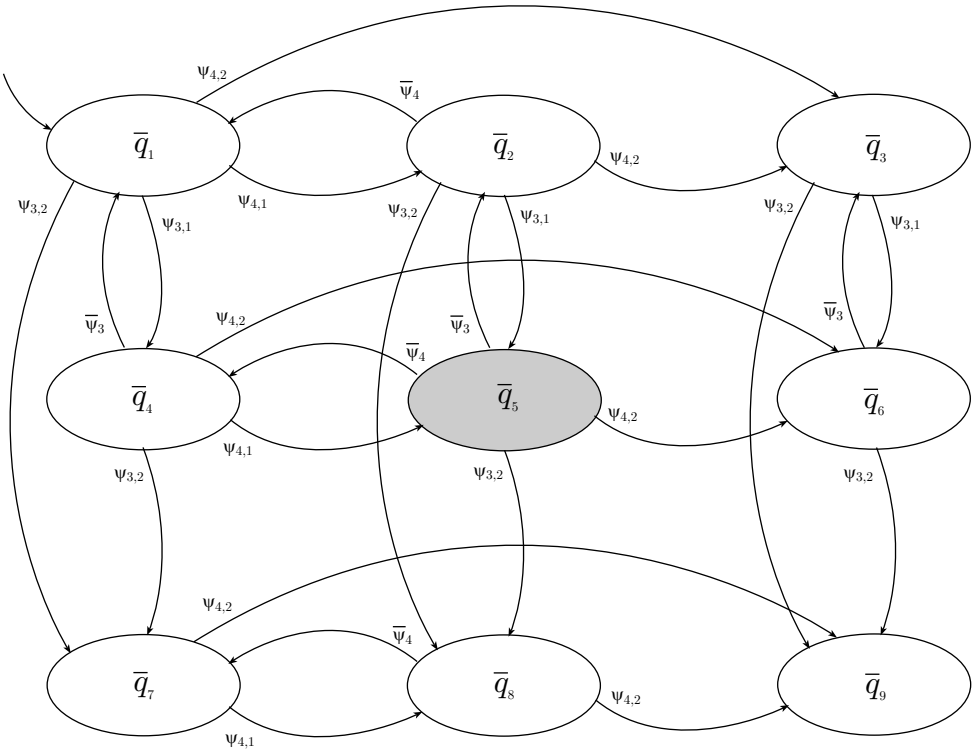

Fig. 10. Modified controllers shuffle observer

failures of complex systems can be traced back to unforeseen circumstances that are trivial to analyze after they become visible. We are now investigating some more complex ATM cases to demonstrate how difficult it is to enumerate the corner cases of real applications.

Acknowledgments

The authors wish to thank Ted Lewis and Derek Jordan (BAES) for providing the scenario illustrated in Section 4, based on present procedures relying on the UK Radio Telephony (RT) procedures CAP 413(2002) that are largely similar for light and commercial aircraft.

References

1. A. Balluchi, L. Benvenuti, M. D. Di Benedetto, A. L. Sangiovanni–Vincentelli, Design of Observers for Hybrid Systems, In Claire J. Tomlin and Mark R. Greenstreet, Editors, *Hybrid Systems: Computation and Control*, Vol. 2289 of Lecture Notes in Computer Science, pp. 76–89, Springer-Verlag, Berlin Heidelberg New York, 2002.

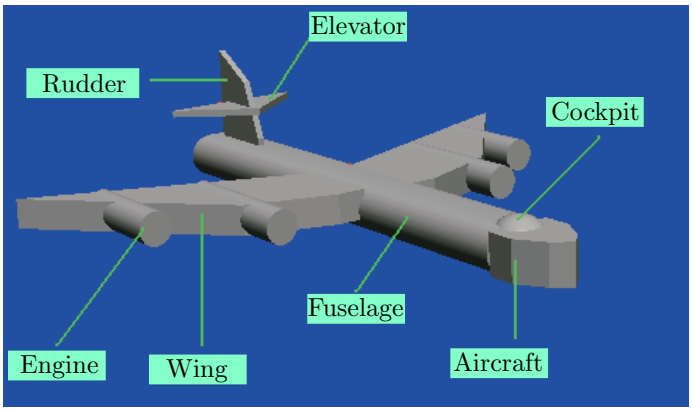

Fig. 11. Solid model of the aircraft

2. M.L. Bujorianu, J. Lygeros, W. Glover, G. Pola, A Stochastic Hybrid System Modeling Framework, Deliverable 1.2, Project IST–2001–32460 HYBRIDGE, February 1, 2003, http://www.nlr.nl/public/hosted–sites/hybridge.
3. M. Cüneyt, C.M. Özveren, A.S. Willsky, Stability and Stabilizability of Discrete Event Dynamic Systems, *Journal of the Association for Computing Machinery*, Vol. 38, No. 3, pp. 730–752, July 1991.
4. E. De Santis, M. D. Di Benedetto, S. Di Gennaro, G. Pola, Hybrid Observer Design Methodology, Public Deliverable D7.2, Project IST–2001–32460 HYBRIDGE, August 19, 2003, http://www.nlr.nl/public/hosted–sites/hybridge.
5. M. D. Di Benedetto, S. Di Gennaro, A. D'Innocenzo, Situation Awareness Error Detection, Public Deliverable D7.3, Project IST–2001–32460 HYBRIDGE, August 18, 2004, http://www.nlr.nl/public/hosted–sites/hybridge.
6. S. Di Gennaro, Nested Observers for Hybrid Systems, *Proceedings of the Latin–American Conference on Automatic Control CLCA 2002*, Guadalajara, México, December 3–6, 2002.
7. S. Di Gennaro, Notes on the Nested Observers for Hybrid Systems, *Proceedings of the European Control Conference 2003 – ECC 03*, Cambridge, UK, 2003.
8. M. R. Endsley, Towards a Theory of Situation Awareness in Dynamic Systems, *Human Factors*, Vol. 37, No. 1, pp. 32–64, 1995.
9. P. M. Frank, Fault Diagnosis in Dynamic Systems using Analytical and Knowledge–Based Redundancy – A Survey and Some New Results, *Automatica*, Vol. 26, No. 3, pp. 459–474, 1990.
10. J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison–Wesley, Reading, MA, 1979.
11. T. Lewis, D. Jordan, Personal communication, BAE Systems, 2004.
12. J. Lygeros, C. Tomlin, S. Sastry, Controllers for reachability specifications for hybrid systems, *Automatica*, Special Issue on Hybrid Systems, vol. 35, 1999.
13. M. A. Massoumnia, G. C. Verghese, A. S. Willsky, Failure Detection and Identification, *IEEE Transactions on Automatic Control*, Vol. 34, No.3, pp. 316–321, 1989.
14. M. Oishi, I. Hwang, C. Tomlin, Immediate Observability of Discrete Event Systems with Application to User–Interface Design, *Proceedings of the* 42^{nd} *IEEE Conference on Decision and Control*, Maui, Hawaii USA, pp. 2665–2672, 2003

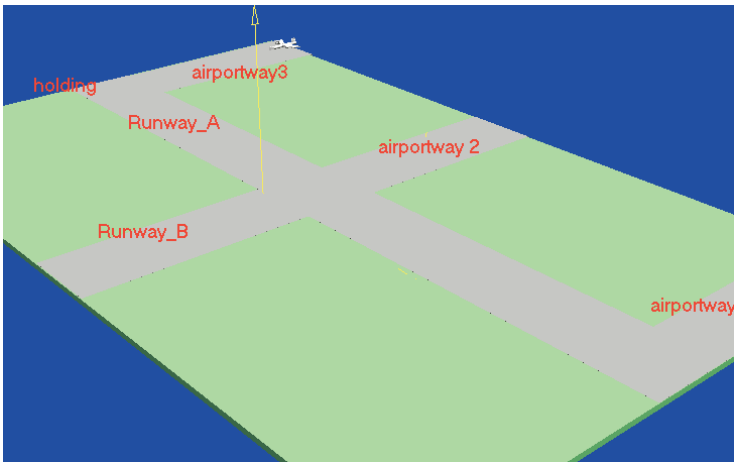

Fig. 12. Model of the runway configuration

15. C.M. Özveren, A.S. Willsky, Observability of Discrete Event Dynamic Systems, *IEEE Transactions on Automatic Control*, Vol. 35, pp. 797–806, 1990.
16. P. Ramadge, Observability of Discrete Event Systems, *Proceedings of the* 25^{th} *IEEE Conference on Decision and Control*, Athens, Greece, pp. 1108-1112, 1986.
17. P. J. Ramadge, W. M. Wonham, Supervisory Control of a Class of Discrete–Event Processes *SIAM Journal of Control and Optimization*, Vol. 25, No. 1, pp. 206–230, Jan. 1987.
18. S. Stroeve, H.A.P. Blom, M. van der Park, Multi–Agent Situation Awareness Error Evolution in Accident Risk Modelling, FAA–Eurocontrol, ATM2003, June 2003, http://atm2003.eurocontrol.fr/