HYBRIDGE

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

WP7: Error Evolution Control

Situation Awareness Error Detection

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

17 August 2004

Version: *0.3*
Task number: *7.3*
Deliverable number: *D7.3*
Contract: *IST-2001-32460 of European Commission*

[1] University of L'Aquila

DOCUMENT CONTROL SHEET

Title of document: *Situation awareness error detection*

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

Deliverable number: *D7.3*

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	31 Dec 2003	All	First draft
0.2	20 Jan 2004	All	Second draft
0.3	17 Aug 2004	All	Third 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	

Abstract

The third deliverable D7.3 of Work Package WP7 of the HYBRIDGE Project focuses on the detection of situation awareness errors. In this report, a simple ATM example, the runway crossing control problem, is considered as a case study. Six agents act in this control problem, three of which are humans that are subject to situation awareness errors. To detect these errors, we model the agents' behaviour with hybrid systems. The error detection problem can be solved by building an observer for the hybrid system obtained composing the hybrid models of the agents. This observer is designed using the theory described in Deliverable 7.2.

Contents

Chapter 1

Introduction

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 avoiding the propagation of errors and other non–nominal events in distributed systems. In an 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 that has to identify these situations and act accordingly to prevent non–nominal situations to evolve into accidents. 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. The objectives of the second task of WP7 were (i) to identify a stochastic hybrid model to describe the dynamics involved in error evolution control and to capture the essential features studied in Task 7.1 and (ii) to develop estimation methods and observer design techniques for this class of stochastic hybrid systems. The research related to the first objective of Task 7.2, pursued in collaboration with the University of Cambridge, was documented in Deliverable 1.2 [3]. As for the second objective of Task 7.2, we addressed the issue of observability and observer design for hybrid systems. In Public Deliverable 7.2 [6], we presented a unified framework to offer a perspective on the results available in the literature on observability of hybrid systems as a first step in developing a theory of observability and algorithms for observer design that can be applied to error propagation control in ATM. In particular, 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 synthesis methods for hybrid observers.

We recall the aim of Task 7.3 from the Technical Annex of the HYBRIDGE contract:

> *"Situation Awareness (SA), plays a crucial role in the identification and correction of non–nominal situations. However, one of the key problems in distributed*

safety critical systems is that humans can have errors in their situation awareness, and these errors can then evolve into the system where they may create all kind of safety critical situations. Since direct observation of human situation awareness is impossible, alternatives have to be developed. These problems are studied in Task 7.3, and a detection approach will be developed. Specific air traffic management situation awareness example(s) will be considered during this study."

In this report, our contribution is a *procedure to solve the problem of detecting situation awareness errors on a specific ATM example by means of the methods developed in Deliverable 7.2 [6].*

The example chosen here is simple enough to allow the reader to understand easily the applicability of our theoretical results. One could argue that the machinery used in this example is an overkill to obtain intuitive results. However, human 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.

The report is organized as follows. In Chapter 2, we present a hybrid model for the active runway crossing problem consisting of the compositions of hybrid models for each of the agents. In Chapter 3, we consider the application of hybrid observer design to the active runway crossing problem. We begin by reviewing some concepts related to observability of hybrid systems and to the design of observers. The definition of observer is generalized for the specific purposes of Task 7.3. Then, a particular situation of risk is analyzed, and hybrid observers are used to detect the error evolution. Finally, the proposed observer is tested and simulation results analyzed. In Chapter 4, we offer some concluding remarks.

Chapter 2

A Hybrid Model of the Active Runway Crossing System

In this section, we consider the example proposed in Stroeve *et al.* [12] of an active runway crossing. This will be a sufficiently simple case study that summarizes the main difficulties in the formulation, analysis and control of a typical accident risk situation for ATM. The active runway crossing will be decomposed into various subsystems, each with hybrid dynamics modeling its specific operations.

2.1 Description of a simple ATM framework

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

One of the key problems in distributed safety critical systems is that humans can have errors in their "Situation Awareness" (*SA*), and these errors can then evolve into the system and create safety critical situations. Situation Awareness may be defined as in [7], [12]:

> *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.*

In Deliverable 7.1 [5], a review of the work done in the literature to model and measure Situation Awareness was presented. Within the ATM system, Stroeve *et al.* [12] define an

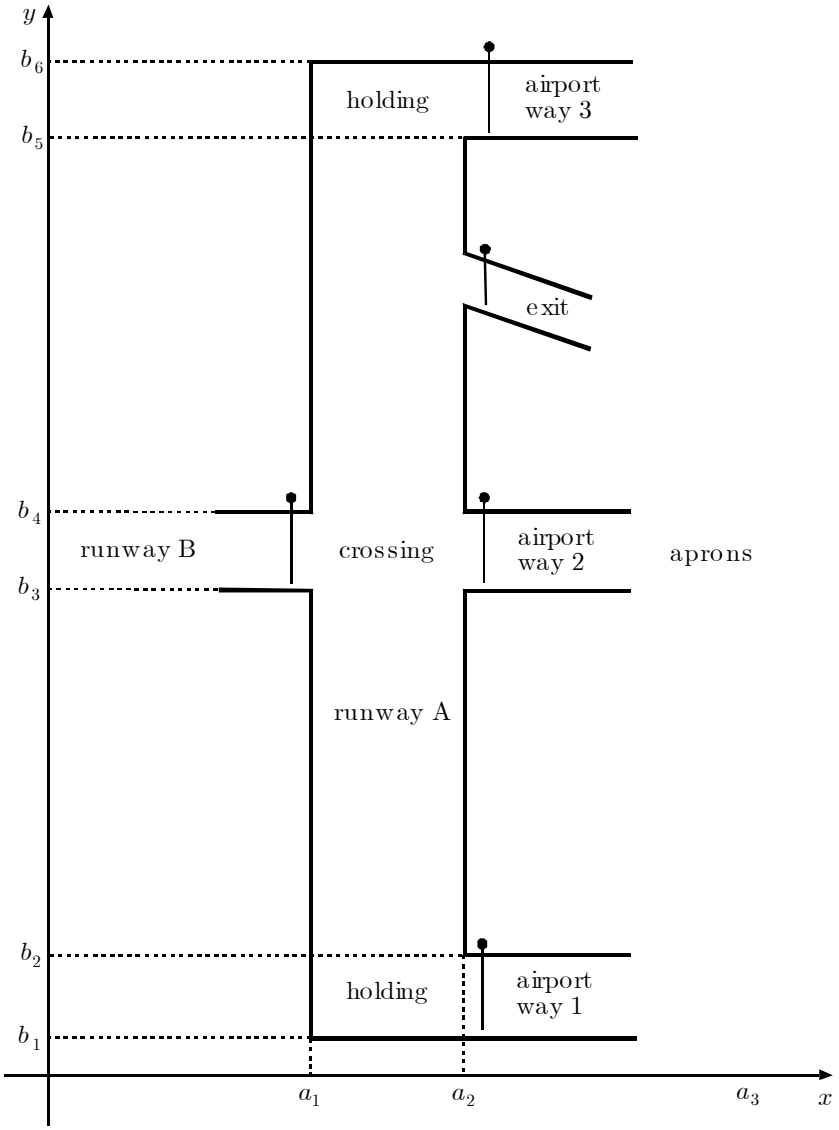

Figure 2.1: Airport configuration

Agent as an entity, such as a human operator or a technical system, which is characterized by its *SA* of the environment. Following Stroeve *et al.* [12], *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.

2.2 Agents in an active runway crossing

The runway crossing operation consists of

1. Two pilots (*PF–t*, *PF–c*) controlling respectively an aircraft taking off and one moving on the ground;
2. A runway controller (*Co*);
3. The airport technical support (*ATS*) system.

The first pilot proceeds towards the holding area (regular taxiway) with the intent of completing a takeoff operation, while the second pilot is approaching the crossing area. The runway controller, with the aid of visual observation of the runway and VHF communication, has the responsability of granting crossings and takeoffs, avoiding the use of the runway by two aircrafts simultaneously. Technical support systems help the pilots and the controller to communicate (VHF) and detect dangerous situations (alerts).

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

- Pilot subsystem:
    1. *Pilot flying of taking off aircraft (PF–t)*: Initially the pilot flying (*PF*) of a taking off aircraft proceeds on the airport way 1 or 3 until he reaches runway A. He begins taxiing on the taxiway and prepares to take-off. When taxiing is executed, he asks via the VHF communication system takeoff grant to the runway controller, and waits.

When he has *SA* that takeoff is allowed, he initiaties taking off, and monitors the traffic situation on the runway visually and via VHF. If a crossing aircraft is observed or in reaction to a *Co* emergency braking command the *PF–t* starts a braking action.

2. *Pilot Flying of crossing aircraft (PF–c)*: The *PF–c* proceeds on the airport way 2 until he reaches the runway. He asks to the runway controller crossing permission and crosses when granted. While proceeding towards the airport way, he may have *intent SA* that the next airport way–point is either a regular taxiway or a runway crossing. In the first case and in the second case if the *PF* has *SA* that the crossing is allowed, he enters runway A without waiting for crossing permission: in fact, if his intent *SA* is a taxiing operation, he may erroneously assume by visual monitoring process that the runway crossing is a regular taxiway. The reaction of the *PF–c* to the detection of a collision risk, due to visual observation or an active runway controller *Co* call, is an emergency braking action.

- *Active Runway Controller (Co)*: The *Co* is a human operator supported by visual observation and by the *ATS* system. If the *Co* has *SA* of a collision risk, he specifies an emergency braking action to both the crossing and taking off aircraft.

- *ATS system:* This is the technical system supporting the decisions of the Active Runway Controller. It includes three subsystems we model: a communication system, a runway incursion alert and a stopbar violation alert.

2.3 Formal Hybrid Models of the Active Runway Crossing Agents

In this section, we give a formal definition of the agents introduced in the previous section. The agent i can be a Discrete Event Dynamical System (DEDS) or an Hybrid System (HS). When a HS, Y_i is the continuous output set, coinciding with the state set X_i, and $h_i \colon X_i \times Q_i \to Y_i$ is the continuous output function. We use here the same notations as in Deliverable 7.2 [6].

Pilots Flying – Agents 1 and 2

PF–t ($i = 1$) and *PF–c* ($i = 2$), represented in Figure 2.2, can be both modeled as hybrid systems $\mathcal{H}_{PF-t}$, $\mathcal{H}_{PF-c}$ where

- $Q_i = \{q_{1,i}, q_{2,i}, q_{3,i}, q_{4,i}, q_{5,i}, q_{6,i}, q_{7,i}, q_{8,i}, q_{9,i}\}$, $i = 1, 2$, are the sets of discrete states with $q_{1,i}$ the *PF* running on an airport way, $q_{2,i}$ the *PF* crossing the runway, $q_{3,i}$ the *PF* in standby before the crossing line waiting for crossing grant, $q_{4,i}$ the *PF* taxiing on a holding,

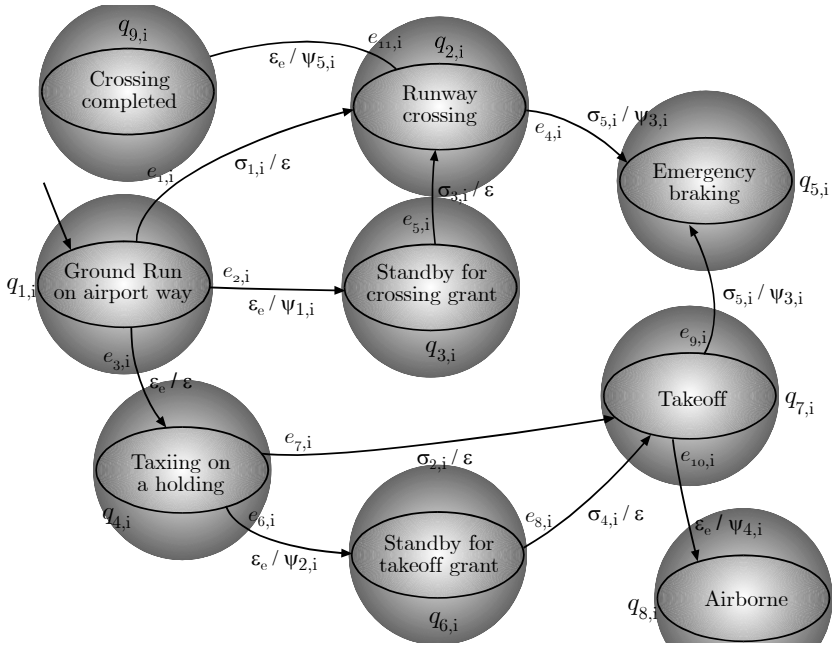

Figure 2.2: *PF* hybrid system

$q_{5,i}$ the *PF* braking for an emergency, $q_{6,i}$ the *PF* in standby before the takeoff line waiting for takeoff grant, $q_{7,i}$ the *PF* taking off, $q_{8,i}$ the *PF* airborne, $q_{9,i}$ the *PF* has completed the crossing operation;

- $P_i = \{\psi_{1,i}, \psi_{2,i}, \psi_{3,i}, \psi_{4,i}, \psi_{5,i}\} \cup \{\varepsilon\}$, $i = 1, 2$, are the sets of discrete outputs, with $\psi_{1,i}$ the crossing request, $\psi_{2,i}$ the takeoff request, $\psi_{3,i}$ the emergency braking, $\psi_{4,i}$ the takeoff completed, and $\psi_{5,i}$ the crossing completed;
- $U_{Di} = U_{Di_{EXT}} \cup U_{Di_{CONTR}} \cup \{\varepsilon_e\}, U_{Di_{EXT}} = \{\sigma_{1,i}, \sigma_{2,i}\}$, $U_{Di_{CONTR}} = \{\sigma_{3,i}, \sigma_{4,i}, \sigma_{5,i}\}$, $i = 1, 2$, are the sets of discrete inputs, where
  - $\sigma_{1,i}$, $\sigma_{2,i}$ model situation awareness errors as disturbances that cause an ungranted crossing and an ungranted takeoff operation, respectively,
  - $\sigma_{3,i}$ models the crossing grant from the *Co*, $\sigma_{4,i}$ the takeoff grant from the *Co*, $\sigma_{5,i}$ the emergency braking order by the *Co*;
- $X_i = \{(s_i, v_i) : s_i \in \mathbb{R}^2, v_i \in \mathbb{R}^2\}$, $i = 1, 2$, are the sets of the continuous state values, where s_i indicates the position and v_i the velocity of the i^{th} agent;
- $U_i = \mathbb{R}^m$, $i = 1, 2$, are the sets of the continuous input u_i values, $V_i = \mathbb{R}^p$ are those of the continuous disturbance d_i values;

- $S_{Ci} = \left\{ f_{q_{j,i}} \colon q_{j,i} \in Q_i \right\}$, $f_{q_{j,i}} \colon X_i \times U_i \times V_i \longrightarrow T_{X_i}$, $i = 1, 2$, are the sets of the continuous (simplified) dynamics

$$\begin{aligned} \dot{s}_i &= v_i \\ \dot{v}_i &= u_i(t) + d_i(t); \end{aligned}$$

and d_i represent possible disturbance forces acting on the aircraft (e.g. wind).

- The sets of discrete transitions are

$$E_i = \left\{ \begin{array}{lll} e_{1,i} = (q_{1,i}, \sigma_{1,i}, q_{2,i}) & e_{2,i} = (q_{1,i}, \varepsilon_e, q_{3,i}) & e_{3,i} = (q_{1,i}, \varepsilon_e, q_{4,i}) \\ e_{4,i} = (q_{2,i}, \sigma_{5,i}, q_{5,i}) & e_{5,i} = (q_{3,i}, \sigma_{3,i}, q_{2,i}) & e_{6,i} = (q_{4,i}, \varepsilon_e, q_{6,i}) \\ e_{7,i} = (q_{4,i}, \sigma_{2,i}, q_{7,i}) & e_{8,i} = (q_{6,i}, \sigma_{4,i}, q_{7,i}) & e_{9,i} = (q_{7,i}, \sigma_{5,i}, q_{5,i}) \\ e_{10,i} = (q_{7,i}, \varepsilon_e, q_{8,i}) & e_{11,i} = (q_{2,i}, \varepsilon_e, q_{9,i}) & \end{array} \right\}$$

$i = 1, 2$;

- The discrete output functions ($i = 1, 2$) are defined as follows

$$\begin{aligned} \gamma_i(e_{1,i}) &= \gamma_i(e_{3,i}) = \gamma_i(e_{5,i}) = \gamma_i(e_{7,i}) = \gamma_i(e_{8,i}) = \varepsilon \\ \gamma_i(e_{2,i}) &= \psi_{1,i} \\ \gamma_i(e_{4,i}) &= \gamma_i(e_{9,i}) = \psi_{3,i} \\ \gamma_i(e_{6,i}) &= \psi_{2,i} \\ \gamma_i(e_{10,i}) &= \psi_{4,i} \\ \gamma_i(e_{11,i}) &= \psi_{5,i} \end{aligned}$$

where the outputs corresponding to transitions due to situation awareness errors are empty and are the source of the observability problems that we need to address using techniques specifically developed for hybrid systems.

- The invariant mappings ($i = 1, 2$) are defined as follows

$$\begin{aligned}
I_{q_{1,i}} &= \{(s_i, v_i) : s_i \in [a_2, a_3] \times [b_1, b_2] \cup [b_3, b_4] \cup [b_5, b_6], \|v_i\| > 0\} \\
I_{q_{2,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_3, b_4], \|v_i\| > 0\} \\
I_{q_{3,i}} &= \{(s_i, v_i) : s_i \in [a_2, a_2 + \Delta] \times [b_3, b_4], \|v_i\| = 0\} \\
I_{q_{4,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times \{[b_1, b_2] \cup [b_5, b_6]\}, \|v_i\| > 0\} \\
I_{q_{5,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_1, b_6], \|v_i\| \geq 0\} \\
I_{q_{6,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_2 - \Delta, b_2] \cup [b_5, b_5 + \Delta], \|v_i\| = 0\} \\
I_{q_{7,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_1, b_6], \|v_i\| < v_t\} \\
I_{q_{8,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_1, b_6], \|v_i\| > v_t\} \\
I_{q_{9,i}} &= \{(s_i, v_i) : s_i \in [0, a_1] \times [b_3, b_4], \|v_i\| > 0\}
\end{aligned}$$

  where v_t is the takeoff velocity, assumed to be the same for the two agents; for simplicity we considered the same geometrical parameters for the two aircrafts; the invariant mappings establish conditions for the system to remain in the corresponding states.

- $R_i(e, x, u, v) = x$, $\forall (e, x, u, v) \in E_i \times X_i \times U_i \times V_i$ are the reset mappings, $i = 1, 2$;

- The guard mappings ($i = 1, 2$) are

$$\begin{aligned}
G_{e_{2,i}} &= \{(s_i, v_i) : s_i \in [a_2, a_2 + \Delta] \times [b_3, b_4], \|v_i\| = 0\} \\
G_{e_{3,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times \{[b_1, b_2] \cup [b_5, b_6]\}, \|v_i\| > 0\} \\
G_{e_{6,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_2 - \Delta, b_2] \cup [b_5, b_5 + \Delta], \|v_i\| = 0\} \\
G_{e_{10,i}} &= \{(s_i, v_i) : s_i \in [a_1, a_2] \times [b_1, b_6], \|v_i\| > v_t\}.
\end{aligned}$$

that establish conditions for the transitions to take place.

Active Runway Controller – Agent 3

The *Co*, represented in Figure 2.3, can be modeled as a DEDS $\mathcal{D}_{Co}$where

- $Q_3 = \{q_{1,3}, q_{2,3}, q_{3,3}, q_{4,3}\}$ is the set of discrete states, with $q_{1,3}$ the *Co* in miscellaneous operations, $q_{2,3}$ the controller granted a crossing (runway is busy), $q_{3,3}$ the controller granted a takeoff (runway is busy) and $q_{4,3}$ an emergency braking action on the runway;

- $U_{D3} = \{\sigma_{1,3}, \sigma_{2,3}, \sigma_{3,3}, \sigma_{4,3}, \sigma_{5,3}\}$ is the finite set of events, with $\sigma_{1,3}$ the takeoff request, $\sigma_{2,3}$ the crossing request, $\sigma_{3,3}$ the takeoff completed, $\sigma_{4,3}$ the crossing completed and $\sigma_{5,3}$ the stopbar violation alert or runway incursion alert;

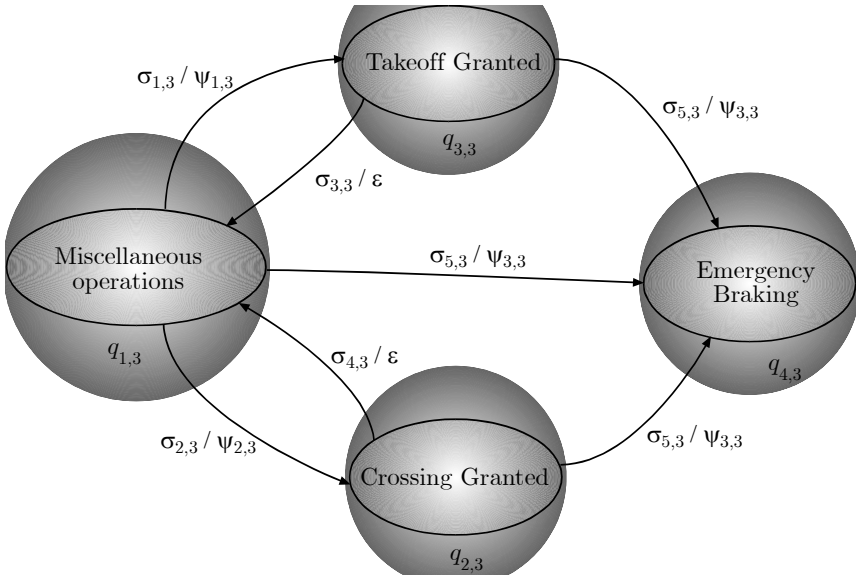

Figure 2.3: *Co* DEDS

- $P_3 = \{\psi_{1,3}, \psi_{2,3}, \psi_{3,3}\} \cup \{\varepsilon\}$ is the set of discrete outputs, with $\psi_{1,3}$ the takeoff grant, $\psi_{2,3}$ the crossing grant and $\psi_{3,3}$ the emergency braking;
- The transition function is such that

$$\begin{aligned}
&\varphi_3(q_{1,3}, \sigma_{2,3}) = \{q_{2,3}\} \qquad \varphi_3(q_{2,3}, \sigma_{5,3}) = \{q_{4,3}\} \\
&\varphi_3(q_{1,3}, \sigma_{5,3}) = \{q_{4,3}\} \qquad \varphi_3(q_{3,3}, \sigma_{3,3}) = \{q_{1,3}\} \\
&\varphi_3(q_{1,3}, \sigma_{1,3}) = \{q_{3,3}\} \qquad \varphi_3(q_{3,3}, \sigma_{5,3}) = \{q_{4,3}\} \\
&\varphi_3(q_{2,3}, \sigma_{4,3}) = \{q_{1,3}\}
\end{aligned}$$

- The map that specifies the possible events at each state is given by $\Phi_3 \colon Q_3 \longrightarrow 2^{U_{D3}}$, with

$$\begin{aligned}
\Phi_3(q_{1,3}) &= \{\sigma_{1,3}, \sigma_{2,3}, \sigma_{5,3}\} \\
\Phi_3(q_{2,3}) &= \{\sigma_{4,3}, \sigma_{5,3}\} \\
\Phi_3(q_{2,3}) &= \{\sigma_{3,3}, \sigma_{5,3}\} \\
\Phi_3(q_{4,3}) &= \varnothing;
\end{aligned}$$

- The output function is defined as

$$\begin{aligned}
\gamma_3(q_{1,3}, \sigma_{2,3}, q_{2,3}) &= \psi_{2,3} \\
\gamma_3(q_{1,3}, \sigma_{1,3}, q_{3,3}) &= \psi_{1,3} \\
\gamma_3(q_{1,3}, \sigma_{5,3}, q_{4,3}) &= \gamma_3(q_{2,3}, \sigma_{5,3}, q_{4,3}) = \gamma_3(q_{3,3}, \sigma_{5,3}, q_{4,3}) = \psi_{3,3} \\
\gamma_3(q_{2,3}, \sigma_{4,3}, q_{1,3}) &= \gamma_3(q_{3,3}, \sigma_{3,3}, q_{1,3}) = \varepsilon.
\end{aligned}$$

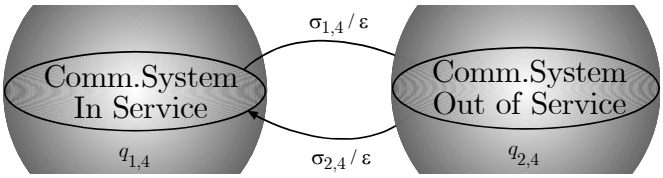

Figure 2.4: Communication System DEDS

Communication System – Agent 4

The communication system, shown in Figure 2.4, can be modeled as a DEDS $\mathcal{D}_{CS}$ where

- The state set is $Q_4 = \{q_{1,4}, q_{2,4}\}$, with $q_{1,4}$, $q_{2,4}$ the communication system in service and out of service, respectively;
- The event set is $U_{D4} = \{\sigma_{1,4}, \sigma_{2,4}\}$, where $\sigma_{1,4}$ represents a failure occurrence in the system, while $\sigma_{2,4}$ corresponds to the repair of the system;
- The discrete output set is $P_4 = \{\varepsilon\}$;
- The transition function is such that

$$\begin{aligned} \varphi_4(q_{1,4}, \sigma_{1,4}) &= \{q_{2,4}\} \\ \varphi_4(q_{2,4}, \sigma_{2,4}) &= \{q_{1,4}\}; \end{aligned}$$

- The map that specifies the possible events at each state is given by

$$\begin{aligned} \Phi_4(q_{1,4}) &= \{\sigma_{1,4}\} \\ \Phi_4(q_{2,4}) &= \{\sigma_{2,4}\} \end{aligned}$$

- The output function is defined as

$$\gamma_4(q_{1,4}, \sigma_{1,4}, q_{2,4}) = \gamma_4(q_{2,4}, \sigma_{2,4}, q_{1,4}) = \varepsilon.$$

Runway Incursion System – Agent 5

The Runway incursion system, see Figure 2.5, is modeled by a DEDS $\mathcal{D}_{RIS}$ where

- $Q_5 = \{q_{1,5}, q_{2,5}, q_{3,5}\}$ is the state set, with $q_{1,5}$, $q_{2,5}$, $q_{3,5}$ the runway incursion alert not active, active and out of service, respectively;

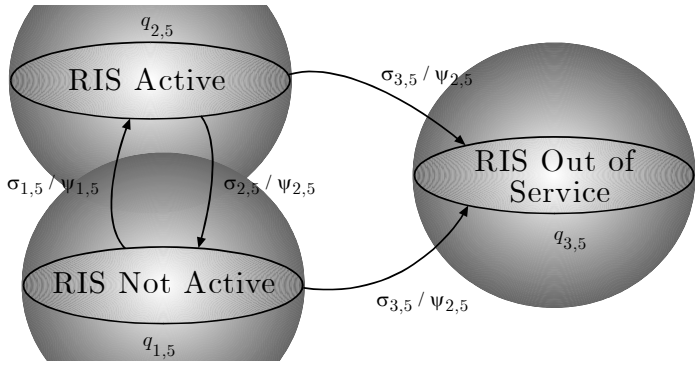

Figure 2.5: Runway Incursion Alert System DEDS

- $U_{D5} = \{\sigma_{1,5}, \sigma_{2,5}, \sigma_{3,5}\}$ is the event set, with $\sigma_{1,5}$,$\sigma_{2,5}$,$\sigma_{3,5}$ the events that determine the corresponding transition in Figure 2.5;
- The discrete output set is $P_5 = \{\psi_{1,5}, \psi_{2,5}\}$, with $\psi_{1,5}$, $\psi_{2,5}$ denoting that the alert is active and not active, respectively;
- The transition function is given by

$$\begin{aligned} \varphi_5(q_{1,5}, \sigma_{1,5}) &= \{q_{2,5}\} \\ \varphi_5(q_{2,5}, \sigma_{2,5}) &= \{q_{1,5}\} \\ \varphi_5(q_{1,5}, \sigma_{3,5}) &= \varphi_5(q_{2,5}, \sigma_{3,5}) = \{q_{3,5}\}; \end{aligned}$$

- The possible events at each state are given by

$$\begin{aligned} \Phi_5(q_{1,5}) &= \{\sigma_{1,5}, \sigma_{3,5}\} \\ \Phi_5(q_{2,5}) &= \{\sigma_{2,5}, \sigma_{3,5}\} \\ \Phi_5(q_{2,5}) &= \varnothing; \end{aligned}$$

- The output function is defined as

$$\begin{aligned} \gamma_5(q_{1,5}, \sigma_{1,5}, q_{2,5}) &= \psi_{1,5} \\ \gamma_5(q_{2,5}, \sigma_{2,5}, q_{1,5}) &= \gamma_5(q_{1,5}, \sigma_{3,5}, q_{3,5}) = \gamma_5(q_{2,5}, \sigma_{3,5}, q_{3,5}) = \psi_{2,5}. \end{aligned}$$

Stopbar Violation System – Agent 6

The Stopbar Violation System, shown in Figure 2.6, may be modeled by the DEDS $\mathcal{D}_{SVS}$ where

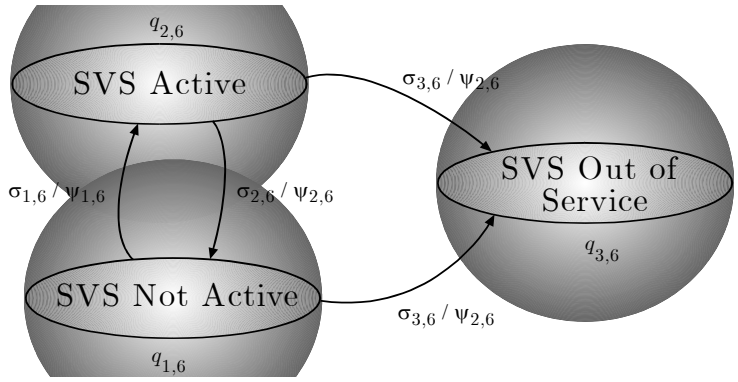

Figure 2.6: Stopbar Violation System DEDS

- The state set is $Q_6 = \{q_{1,6}, q_{2,6}, q_{3,6}\}$, where $q_{1,6}$, $q_{2,6}$, $q_{3,6}$ denote that the stopbar violation alert is not active, active and out of service, respectively;
- The event set is $U_{D6} = \{\sigma_{1,6}, \sigma_{2,6}, \sigma_{3,6}\}$, with $\sigma_{1,6}, \sigma_{2,6}, \sigma_{3,6}$ meaning the events that determine the corresponding transition in Figure 2.6;
- The discrete output set is $P_6 = \{\psi_{1,6}, \psi_{2,6}\}$, where $\psi_{1,6}$, $\psi_{2,6}$ indicate that the alert active and not active, respectively;
- The transition function is

$$\begin{aligned} \varphi_6(q_{1,6}, \sigma_{1,6}) &= \{q_{2,6}\} \\ \varphi_6(q_{2,6}, \sigma_{2,6}) &= \{q_{1,6}\} \\ \varphi_6(q_{1,6}, \sigma_{3,6}) &= \varphi_6(q_{2,6}, \sigma_{3,6}) = \{q_{3,6}\}; \end{aligned}$$

- The events at each state are

$$\begin{aligned} \Phi_6(q_{1,6}) &= \{\sigma_{1,6}, \sigma_{3,6}\} \\ \Phi_6(q_{2,6}) &= \{\sigma_{2,6}, \sigma_{3,6}\} \\ \Phi_6(q_{2,6}) &= \varnothing; \end{aligned}$$

- The output function is

$$\begin{aligned} \gamma_6(q_{1,6}, \sigma_{1,6}, q_{2,6}) &= \psi_{1,6} \\ \gamma_6(q_{2,6}, \sigma_{2,6}, q_{1,6}) &= \gamma_6(q_{1,6}, \sigma_{3,6}, q_{3,6}) = \gamma_6(q_{2,6}, \sigma_{3,6}, q_{3,6}) = \psi_{2,6}. \end{aligned}$$

Chapter 3

Hybrid Observers for the Active Runway Crossing System

Results on observability of hybrid systems are scant. This situation makes the design of a controller for hybrid systems challenging. In this section, we first review the results presented in [6] on the design of hybrid observers. Then, we show how these results can be applied to solve the problem at hand.

3.1 Observer Design for General Hybrid Systems

In this section, we describe a methodology for designing a hybrid observer that recovers the hybrid state evolution of a hybrid system on the basis of its observed output. These results are due to Balluchi et al. [2], [1] and are reviewed in Deliverable 7.2 [6]. We use here the same notations as in Deliverable 7.2.

We consider hybrid systems $\mathcal{H}$ that are linear in the continuous dynamics and where the reset map is an affine function of the continuous state before the transition and of the discrete states involved in the transition. Discrete transitions may be either switching, controllable or invariance transitions (see [6] for a formal definition of the hybrid systems under consideration).

The hybrid observer is a hybrid system itself, denoted $\mathcal{H}_O$, whose task is to provide an estimate $\hat{q}(k)$ and an estimate $\hat{x}(t)$ for the current location $q(k)$ and continuous state $x(t)$ of the hybrid plant. Its inputs are the continuous input and output, $u_c(t)$ and $y_c(t)$, and the discrete output $y_d(k)$.

The observer has to satisfy the following property:

Definition 1 *Given a hybrid system $\mathcal{H}$, a hybrid system $\mathcal{H}_O$ is said to be an observer for $\mathcal{H}$ with respect to the set of states $\bar{Q} \subseteq Q$ if there exist some constants $c \geq 1$, $\mu > 0$ and $b \geq 0$ such*

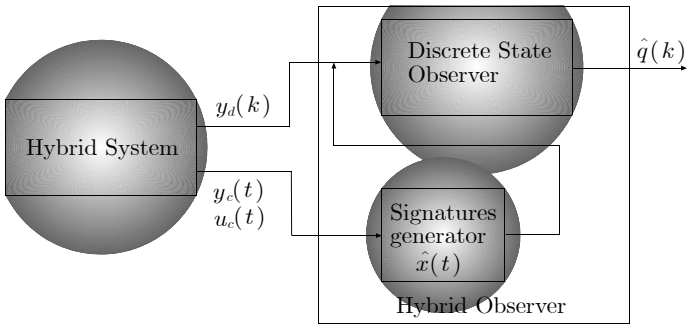

Figure 3.1: Observer structure: location observer and continuous observer

that

$$\begin{aligned} \hat{q}(k) &= q(k) && \forall q \in \bar{Q} \\ \|\hat{x}(t) - x(t)\| &\leq c\,\|\hat{x}(t_k) - x(t_k)\|\, e^{-\mu(t-t_k)} + b && \forall t > t_k \end{aligned}$$

for every initial hybrid state $(q(0), x(0)) \in Q \times X$*, every continuous input* $u(\tau)$ *with* $\tau \in [0,t]$*, every possible input sequence* $\sigma(1), \cdots, \sigma(k)$ *and output sequence* $\psi(1), \cdots, \psi(k)$*. Here* t_k *is the time instant corresponding to the input* $\sigma(k)$*,* μ *is the rate of convergence and* b *is the ultimate bound. If* $b = 0$*, the observer is said to be exponentially convergent.*

Given a hybrid plant with state $\begin{pmatrix} x \\ q \end{pmatrix}$, the structure of the proposed hybrid observer $\mathcal{H}_O$ is illustrated in Figure 3.1. The *location observer* describes the evolution of the discrete location of $\mathcal{H}_O$while the *continuous observer* governs the evolution of the continuous state of $\mathcal{H}_O$.

The *location observer* receives as input the continuous input $u_c(t)$ and the continuous and discrete outputs $y_c(t)$, $y_d(k)$. Its task is to provide the estimate $\hat{q}(k)$ of the discrete location $q(k)$ of the hybrid plant at the current time. Based on the discrete evolution of the location observer, the *continuous observer* constructs an estimate $\hat{x}(t)$ of the plant continuous state that converges exponentially to $x(t)$. The continuous plant input $u(t)$ and output $y_c(t)$ are used by the continuous observer for this purpose. The continuous disturbance $\delta(t)$ is assumed to be measurable.

When the evolutions of the discrete inputs and outputs of the hybrid plant $\mathcal{H}$ are sufficient to estimate the current discrete location, $\mathcal{H}$ is said to be current–state observable and the location observer is a Discrete Event Dynamical System (DEDS) that can be constructed as shown in Chapter 3 of Deliverable 7.2 [6]. If this is not the case, the continuous plant inputs and outputs can be used to obtain some additional information that may be useful for the identification of the plant current location. A transition of the hybrid plant can be detected by observing

the corresponding change of the continuous dynamics. Continuous dynamics changes can be identified by comparing the evolution of the continuous inputs and outputs of the hybrid plant with the evolutions that correspond to the dynamics associated to the locations to be identified. In this way additional discrete signals, to be used as extra inputs to the DEDS observer, are produced. These signals are referred to as *signatures*. A methodology for selecting where the continuous information should be supplied and how to process it, was described in Chapter 6 of Deliverable 7.2 [6]. The processing of the continuous signals of the plant gives reliable discrete information only after some delay with respect to plant location switchings.

3.2 Application to the Runway Crossing Problem

Consider first the *PF* Hybrid system $\mathcal{H}_{PF}$. The hazard situations we wish to detect are related to the transitions $\{q_{2,i}, q_{3,i}\}$ and $\{q_{6,i}, q_{7,i}\}$ that represent respectively a crossing and a take–off when the respective pilots have not been granted permission to take this action.

Consider now the DEDS observer shown in Figure 3.2 constructed using only the discrete output information.

We see that this system is not an observer for $\mathcal{H}_{PF}$ with respect to the set of states $\{q_{2,i}, q_{3,i}\}$and $\{q_{6,i}, q_{7,i}\}$ because of the "zero output" ε associated with some of the transitions of the PF system shown in Figure 2.2. Hence, $\mathcal{H}_{PF}$ is not current–state observable *in the transient (there exists K such that we can observe the discrete state, but it is not true for $K = 1$).*

As a consequence, it is impossible to construct an observer for the entire active runway crossing system using the discrete information only. We need to extract additional information from the continuous output. To make the hazard situation observable using the techniques presented in the previous section, we need to generate signatures that would make the system current–state observable with respect to the set of states $\{q_{2,i}, q_{3,i}\}$ and $\{q_{6,i}, q_{7,i}\}$.

Referring to one of the agents $i = 1, 2$, signature $r_{1,i}$ indicates a transition of the *PF* from the "Ground run on airport way" state to the "Runway crossing" state, or from the "Standby for crossing grant" state to the "Runway crossing" state, and is generated when the position of the aircraft is inside a crossing area.

Signature $r_{2,i}$ indicates a transition of the *PF* from the "Ground run on airport way" state to the "Taxiing on a holding" state, and is generated when the position of the aircraft is inside the holding.

Signature $r_{3,i}$ indicates a transition of the *PF* from the "Taxiing on a holding" state to the

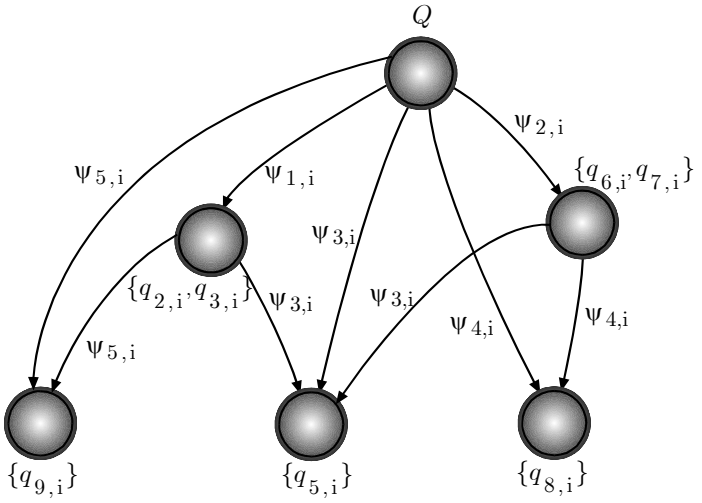

Figure 3.2: *PF* observer without signatures

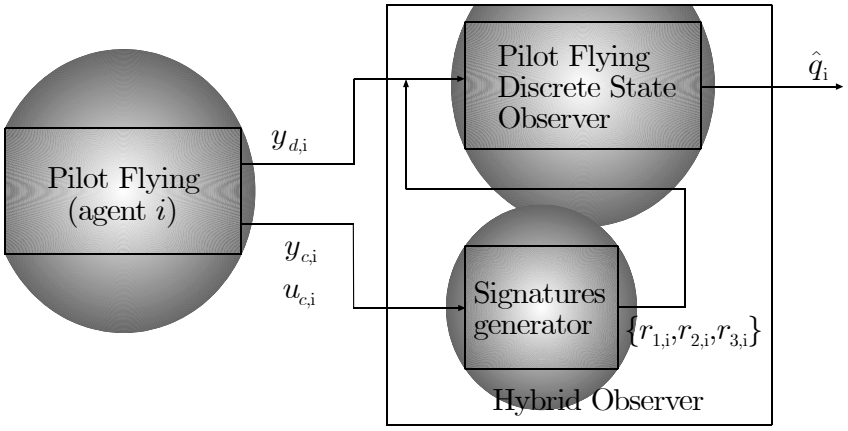

Figure 3.3: Hybrid observer block diagram

"Takeoff" state, or from the "Standby for takeoff grant" state to the "Takeoff" state, and is generated when the aircraft exits the taxiway and enters in the takeoff area of the runway.

The structure of the resulting PF observer with signatures is schematically represented in Figure 3.3. The PF observer with signatures is described by the DEDS $\mathcal{D}_{PFO}$ shown in 3.4, where

- The state set is $Q = Q_i = \{q_{1,i}, q_{2,i}, q_{3,i}, q_{4,i}, q_{5,i}, q_{6,i}, q_{7,i}, q_{8,i}, q_{9,i}\}$, $i = 1, 2$, where each state represents the actual discrete state of the PF Hybrid System;
- The event set is $U_D = U_{Di} = \{\psi_{1,i}, \psi_{2,i}, \psi_{3,i}, \psi_{4,i}, \psi_{5,i}\} \cup \{\varepsilon_e\}$, $i = 1, 2$;
- The discrete output set is $P = \{\alpha_{1,i}, \alpha_{2,i}\}$, $i = 1, 2$, where $\alpha_{1,i}, \alpha_{2,i}$ represent alarm signals, described hereinafter;
- The transition function is defined by means of

$$\begin{array}{lll}
\varphi(q_{1,i}, r_{1,i}) = \{q_{2,i}\} & \varphi(q_{3,i}, r_{1,i}) = \{q_{2,i}\} & \varphi(q_{7,i}, \psi_{3,i}) = \{q_{5,i}\} \\
\varphi(q_{1,i}, \psi_{1,i}) = \{q_{3,i}\} & \varphi(q_{4,i}, \psi_{2,i}) = \{q_{6,i}\} & \varphi(q_{7,i}, \psi_{4,i}) = \{q_{8,i}\} \\
\varphi(q_{1,i}, r_{2,i}) = \{q_{4,i}\} & \varphi(q_{4,i}, r_{3,i}) = \{q_{7,i}\} & \varphi(q_{2,i}, \psi_{5,i}) = \{q_{9,i}\} \\
\varphi(q_{2,i}, \psi_{3,i}) = \{q_{5,i}\} & \varphi(q_{6,i}, r_{3,i}) = \{q_{7,i}\} &
\end{array}$$

- The event sets are

$$\begin{array}{lll}
\Phi(q_{1,i}) = \{\psi_{1,i}, r_{1,i}, r_{2,i}\} & \Phi(q_{4,i}) = \{\psi_{2,i}, r_{3,i}\} & \Phi(q_{7,i}) = \{\psi_{3,i}, \psi_{4,i}\} \\
\Phi(q_{2,i}) = \{\psi_{3,i}, \psi_{5,i}\} & \Phi(q_{5,i}) = \varnothing & \Phi(q_{8,i}) = \varnothing \\
\Phi(q_{3,i}) = \{r_{1,i}\} & \Phi(q_{6,i}) = \{r_{3,i}\} & \Phi(q_{9,i}) = \varnothing
\end{array}$$

- The output function is given by

$$\begin{aligned}
\gamma(q_{1,i}, r_{1,i}, q_{2,i}) &= \alpha_{1,i} \\
\gamma(q_{4,i}, r_{3,i}, q_{7,i}) &= \alpha_{2,i} \\
\gamma(q_{1,i}, \psi_{1,i}, q_{3,i}) &= \gamma(q_{1,i}, r_{2,i}, q_{4,i}) = \gamma(q_{2,i}, \psi_{3,i}, q_{5,i}) = \gamma(q_{3,i}, r_{1,i}, q_{2,i}) = \\
&= \gamma(q_{4,i}, \psi_{2,i}, q_{6,i}) = \gamma(q_{6,i}, r_{3,i}, q_{7,i}) = \gamma(q_{7,i}, \psi_{3,i}, q_{5,i}) = \\
&= \gamma(q_{7,i}, \psi_{4,i}, q_{8,i}) = \gamma(q_{2,i}, \psi_{5,i}, q_{9,i}) = \varepsilon.
\end{aligned}$$

This shows how the problem of current location determination for the PF can be solved. Since for all of the other active runway crossing system agents the states can be easily determined, this result makes the problem of observing the states of the overall system solved.

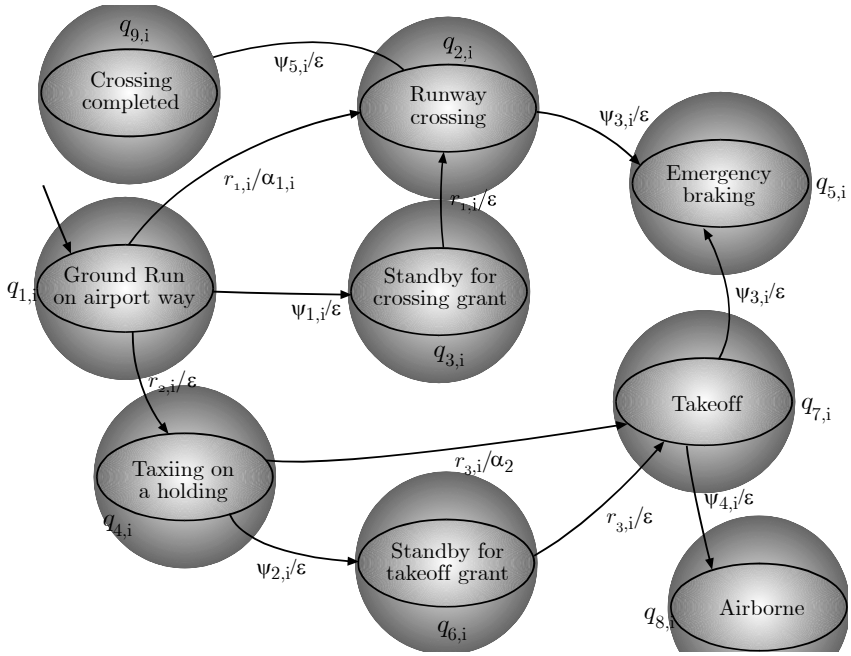

Figure 3.4: Pilot flying observer with signatures

3.3 Simulation results

We will now analyze the specific situation of a *PF–t* ready to take off, while a *PF–c* proceeds on the airport way 2 with intent *SA* that the next way-point is a taxiway, and thus crosses the runway without communicating with the *Co*. This hazardous operation causes the activation of the stopbar violation alert, and the *Co,* if aware that the alarm is on, orders a braking action to the pilots. Instead, if the stopbar violation alert is out of service or the *Co* has no *SA* of the stopbar violation alert, a takeoff could be granted while a *PF–c* is crossing the runway, potentially leading to a collision.

A simulator based on Matlab 6.1 has been realized: among the simulations executed to test the functionality of the observer, two of them are interesting to demonstrate how the observer may help the *Co* to detect risk situations.

In the first simulation (Figure 3.5), the stopbar violation system is "in service". The hazardous situation starts at time t_0 and the information regarding its occurrence is available to the *Co* at time t_1. Hence, after a reaction time, the *Co* has *SA* that a stopbar violation has occurred and, after a time $t_2 - t_1$ due to its reaction time and elaboration of the information received, specifies at time t_2 a hold clearance to both crossing and taking off aircraft.

The observer, with the alarm α_1, informs the *Co* not only that an aircraft has passed the

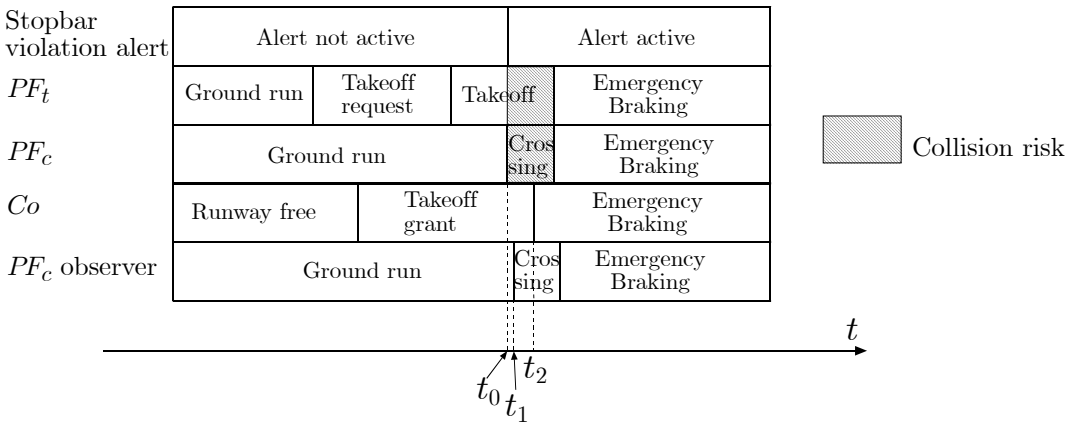

Figure 3.5: Simulation 1 – Stopbar violation system in service

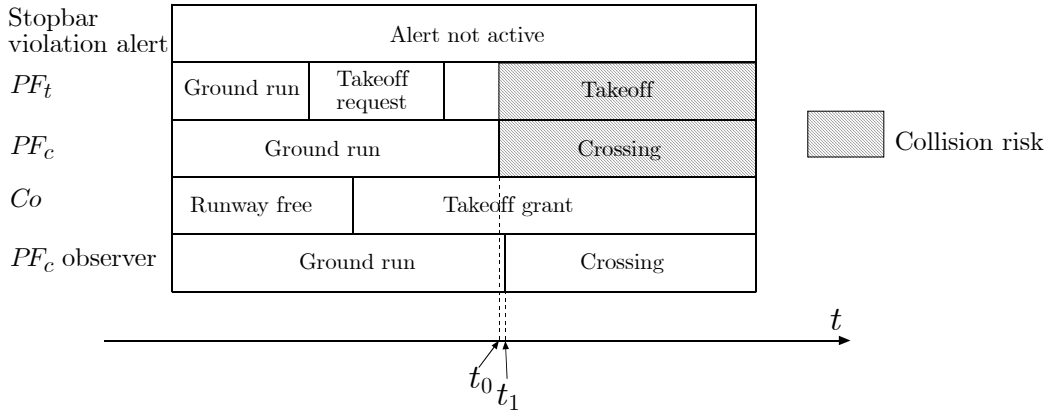

Figure 3.6: Simulation 2 – Stopbar violation system out of service

stopbar (information given by the stopbar violation alert): the alarm is more specific, and consists of a warning that a pilot is executing a crossing without passing through the "Standby for crossing request" state.

In the second case (Figure 3.6) the stopbar violation alert is considered out of service, so the *Co* has no *SA* of the crossing and grants a takeoff operation. Therefore, the *Co* can be aware of the crossing only using the aid of the observer alerts.

In both simulations the observer of the *PF-c* tracks the transition from "Ground run on airport way" to "Runway crossing" after a delay, necessary to compute the position of the aircraft and to generate the signature. Thus, the observer is able to advise the *Co* that a crossing operation without grant is in progress switching on the alarm α_1 simultaneously to the crossing operation.

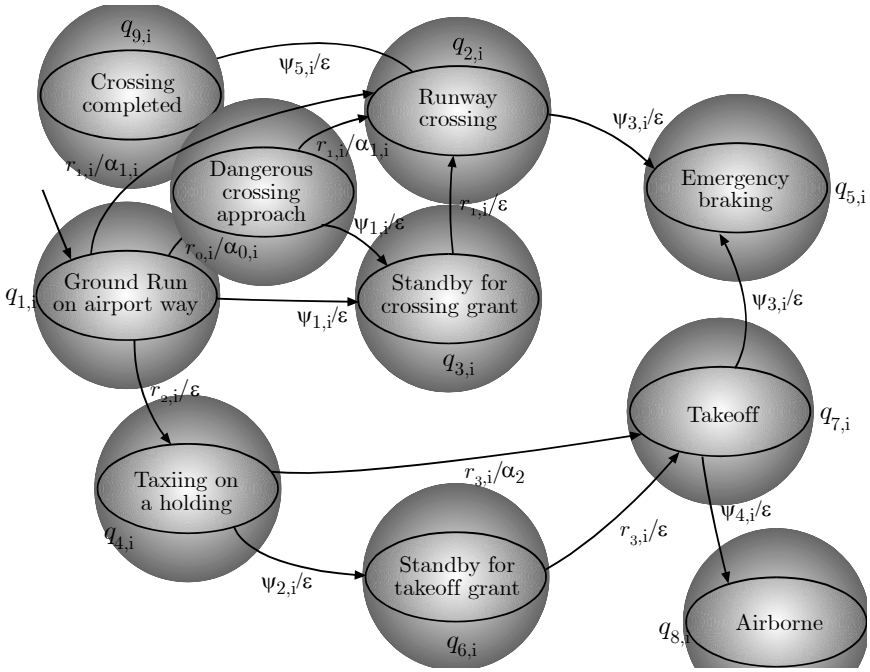

Figure 3.7: Extended pilot flying observer

Furthermore, we could conceive to initiate a pre–alarm condition trying to predict the intent *SA* of the *PF–c*. It is reasonable to assume that, if the intent *SA* of the *PF-c* is a crossing operation, he will decelerate before the stopbar: we can add a new state in the *PF–c* observer which represents a "dangerous approach to crossing" state, which is visited when position and speed dynamics of the aircraft do not match a standard behaviour (e.g. the deceleration curve is not steep enough). From this "pre–alarm" state (see Figure 3.7) the *PF–c* can stop and ask for crossing grant or proceed to the crossing area and generate an alarm α_1 for "unauthorized crossing". This clearly would increment the performance of the *Co* with a pre–alarm condition which warns of a potentially erroneous intent *SA*.

Chapter 4

Conclusions

In this report, the example of a simple but significant ATM case, the active runway crossing control problem, was examined with the intent of testing the applicability of the theoretical results on observers obtained in Deliverable 7.2 to a realistic ATM situation for the detection of situation awareness errors. In this example, we considered six agents, three of which are humans subject to situation awareness errors. We defined a suitable hybrid system framework capable of capturing the essential observability problems of this example, including erroneous maneuvers of the aircrafts involved, and faults of control apparatuses.

The construction of an observer for detecting the hazardous transitions that may lead to an accident is a non-trivial problem, since the discrete outputs cannot resolve alone ambiguities that make the observability problem unsolvable. We therefore turned to the techniques developed in [2] and [6]: by generating the "signatures" of the continuous dynamics, we made the observability problem solvable. 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.

One could argue that the machinery used in this example is an overkill to obtain results that seem to be easily obtainable by intuition. Indeed, in this particular example, an intuitive design would have solved the problem. However, 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 failures of complex systems can be traced back to unforeseen circumstances that are trivial to analyze *after* they become visible. The idea of using formal techniques in design is mostly successful when corner cases are numerous and difficult to enumerate. We are investigating some ATM cases to demonstrate how difficult it is to enumerate the corner cases of real applications.

An important outcome of our work is the realization that common observability notions such

as $K-$current-state observability that we used as the basis for Deliverable 7.2 may not represent accurately the problem we are trying to solve. $K-$current-state observability means that any discrete location of the hybrid system can be identified by the use of the discrete outputs, after a finite number K (> 0 and generic) of discrete transitions. In the error detection problem, it is necessary to identify those discrete locations - we may call them "*critical*" - that correspond to dangerous situations. If a critical state occurs before K transitions take place, then, even though the system is current-state observable, the critical situation is not identified. In the case of the runway crossing example, the theory applies well because, after signature generation, the hybrid system model is current-state observable with $K = 1$. However, this is not always 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 work will be done in Deliverable 7.4.

Bibliography

[1] A. Balluchi, L. Benvenuti, M.D. Di Benedetto, A.L. Sangiovanni- Vincentelli: A Hybrid Observer for the Driveline Dynamics. *European Control Conference ECC'01*, pp.618–623, Porto (Portugal), Sept. 4–7, 2001.

[2] A. Balluchi, L. Benvenuti, M.D. Di Benedetto, A.L., Sangiovanni-Vincentelli, Design of Observers for Hybrid Systems. In Lecture Notes in Computer Science 2289, C.J. Tomlin and M.R. Greensreet Eds., pp. 76-89, Springer-Verlag, 2002.

[3] M.L. Bujorianu, J. Lygeros, W. Glover and 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.

[4] E. De Santis, M.D. Di Benedetto, M.D., G. Pola, On observability and detectability of continuous–time linear switching systems. *42nd IEEE Conference on Decision and Control CDC 2003*, Maui, Hawaii, Dec. 2003, pp.

[5] M.D. Di Benedetto, G. Pola, Inventory of Error Evolution Control Problems in Air Traffic Management, Deliverable D7.1, Project IST-2001-32460 HYBRIDGE, November 4, 2002.

[6] E. De Santis, S. Di Gennaro, M.D. Di Benedetto, 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

[7] Endsley, M.R., Towards a theory of situation awareness in dynamic system, *Human Factors*, vol. 37, n°1, pp.32-64, 1995.

[8] D.G. Luenberger, An introduction to observers, *IEEE Transactions on Automatic Control*, vol.16, 6, pp. 596-602, Dec, 1971.

[9] C. M.Ozveren and A. S. Willsky, Observability of discrete event dynamic systems, *IEEE Trans. on Automatic Control*, 35, 7, 797-806, July, 1990.

[10] C. M. Ozveren and A. S. Willsky and P. J. Antsaklis, Stability and stabilizability of discrete event dynamic systems, *Journal of the Association for Computing Machinery*, 38, 3, pp. 730-752, July, 1991.

[11] P. J. Ramadge, Observability of discrete event–systems, *25th IEEE Conference on Decision and Control*, pp. 1108–1112, Athens, Greece, 1986.

[12] Stroeve, S., Blom, H.A.P., van der Park, M., Multi-Agent Situation Awareness Error Evolution in Accident Risk Modelling, draft in FAA-Eurocontrol, ATM2003, June 2003.