# A Compositional Hybrid Systems Framework for the Analysis of Air Traffic Management Systems

ALESSANDRO PETRICCONE

Doctoral Thesis
L'Aquila, Italy 2012

Dept. of Electrical Engineering and Information
Via G. Gronchi 18
67100 L'Aquila
ITALY

Academic dissertation with the permission of the University of L'Aquila to public scrutiny for the Degree of PhD Thesis in Electrical Engineering and Information on March, 27$^{th}$ 2012 at Faculty of Engineering.



Publisher: University of L'Aquila

## Abstract

The inherent complexity of Air Traffic Management (ATM) systems makes formal analysis a difficult task. Complexity of ATM systems is mainly due to their heterogeneity and to the large number of their sub–components. While heterogeneity in the diverse components of ATM systems has been effectively approached by resorting to compositional hybrid-system formalism, methods for the formal analysis of realistic large scale ATM systems are few at present. In this thesis we provide an approach to deal with the analysis of safety criticality for complex ATM systems. The approach proposed is centered on a mathematical representation of ATM systems, termed arenas of finite state machines, which are effective in modeling the behavior of ATM agents both in nominal and non–nominal modes of operation, as well as their interaction. Complexity reduction techniques are then provided that are based on a generalization of the recently introduced notion of compositional bisimulation.

*To my family and to my best friends*

# Acknowledgements

Foremost, I would like to thank my advisor Maria Domenica Di Benedetto, who accepted me as a PhD student at the Automatic Control Group at the University of L'Aquila. In a special manner, I appreciate her guidance and support throughout my research and the opportunities she offered me to extend my knowledge, by attending doctoral schools and visiting important research institutions. I owe my gratitude to my co-advisor Giordano Pola, for his restless motivation and for inspiring most of my technical work. Without their valuable comments, this thesis would probably not exist today.

I would like to thank Edoardo Filippone and Salvatore Luongo at the Centro Italiano Ricerche Aereospaziali.

I am also indebted to the coauthors of the papers who form the backbone of this work; in particular I thank Elena De Santis and Alessandro D'Innocenzo.

I wish to thank all the people of my research group in L'Aquila, including Alessandro Borri, Domenico Bianchi, Emmanule Serra, Ubaldo Tiberi and Francesco Smarra, with whom I shared a lot of nice experiences during my PhD.

I would also like to thank my friends in Avezzano that always supported me and with whom I always have good time: Stefano, Emilio, Giampiero, Davide, Vittorio, Francesco, Eugenio, Pierpaolo, Fabio, Valeria, Paolo, Giancarlo and Domenico.

The research described in this thesis is partially supported by the Center of Excellence for Research DEWS (Design methodologies for Embedded controllers, Wireless interconnect and System-on-chip), University of L'Aquila, Italy.

*Alessandro Petriccone*
L'Aquila, March 2012

# Contents

Chapter 1

# Introduction

The volume of air traffic is increasing so rapidly that a major efficiency overhaul to manage air traffic flows is necessary to maintain normal operation. This issue motivated many researchers in the area of Air Traffic Management (ATM) systems to propose new procedures that increase capacity while preserving safety. This is particularly relevant since the more capacity increases, the more complex the air traffic management system becomes. In particular the aim of the SESAR (Single European Sky Air Traffic Management Research) Programme is to improve efficiency in future European Air Traffic Management (ATM). The SESAR Programme addresses a number of research topics including long-term and innovative research which aims at supporting future European ATM and air transport industry by developing sustainable ATM research capabilities. An ATM is a joint cognitive system where a relevant number of technical systems and human agents interact with each other with the aim of coping with significant uncertainties that affect the system. Current accident statistics prove that ATM systems are very efficient from the safety perspective point of view. The increasing in the volume of air traffic that is expected in the close future, requires these joint cognitive systems to be even more efficient. Among some scientific disciplines, Resilience Engineering [55, 56, 57] deals with the design of efficient joint cognitive systems. Resilience indicates that operations and organizations are capable of resisting a wide variety of demands within their domains and capable of recovering from variations, degradations, disruptions, and any condition that may affect the stability of the operation or organization. In other words, resilience engineering addresses the design of joint cognitive systems, both in nominal and non-nominal conditions. Since ATM joint cognitive systems are complex, resilience engineering in this regard, is at an early stage of development. Psychologists mostly agree that the cognitive influence of human agents determines why external and internal disturbances are so effectively managed by the ATM. During recent years novel psychological model constructs have been proposed, which capture human cognition and its interaction with other joint cognitive system entities [55, 56, 57]. The results currently obtained demonstrated that there are non-psychological challenges when the aim is to perform a systematic analysis of

the combinatorial many potential behaviors due to unpredictable external and internal events. Formal mathematical models and analysis methods offer a key complementary approach that is needed to render resilience engineering effectively applicable to complex joint cognitive systems, such as ATM. Formal approaches to the modeling and analysis of ATM systems are an effective auxiliary medium, towards the definition of robust novel procedures that on one hand, are efficient from the capacity point of view, and on the other hand, preserve safety of the agents operating in the scenario.

## 1.1 Background

### 1.1.1 Finite State Machines

Finite state machines (FSMs) are widely used in modeling complex systems ranging from computer and communication networks, automated manufacturing systems, air traffic management systems, distributed software systems, among many others, see e.g. [30, 5]. The increasing complexity of large scale systems demanded during the years for formal methods that can render their analysis tractable from a computational complexity point of view. Several approaches have been proposed in the literature, which include abstraction, modular verification methods, symmetry and partial order reduction, see e.g. [5]. The common goal of these approaches is to find an FSM that is equivalent to the original one, but with a set of states of smaller size. The approach followed in this context, see e.g. [31, 32], is to view a complex system as a "non–flat" system. A non–flat system is a "finite state machine" where each "state" can be either a basic state or a superstate [33] that hides inside an FSM or even a composition of FSMs. By expanding the superstates of a non–flat system to their corresponding FSMs an ordinary FSM is obtained. One of the early non–flat systems that appeared in the literature are *hierarchical state machines* (HSMs) [31]. While HSMs well capture modeling features of many design languages as for example Statecharts [33], they only consider sequential interaction among the FSMs involved. *Recursive state machines* (RSMs) [34] extend HSMs by allowing recursion in the sequential interaction of FSMs. As such, they well model sequential programming languages with recursive procedure calls. *Recursive Game Graphs*, a natural adaptation of RSMs to a game theoretic setting, have been studied in [35]; *Pushdown Graphs* have been studied in [36]. Both HSMs and RSMs do not exhibit concurrent compositional features. *Communicating hierarchical state machines* (CHSMs) [32] generalize HSMs, by allowing FSMs to interact not only sequentially but also concurrently, through the notion of parallel composition. Reachability problems and checking language and bisimulation equivalences for CHSMs are proven in [32] to fall in the class of exponential time and space complexity problems. This complexity result is in line with the ones further established in [38, 37] on complexity arising in checking a range of equivalence notions in the linear time–branching time spectrum [39] for networks of FSMs, modeled by parallel composition of FSMs. By following the conjecture of Rabinovich in [40], the work in [38, 37] strongly suggest that there

is no way to escape the so–called state explosion problem, when checking behavioral relations and in particular bisimulation equivalence, for non–flat systems exhibiting concurrent–types interaction. A novel non–flat system has been recently proposed in [48] and called Arena of Finite State Machine (AFSM), that is a collection of FSMs interacting concurrently through a communication network. For AFSMs a notion of compositional bisimulation has been proposed which allows a substantial computational complexity reduction when checking bisimulation equivalence between the flattened FSMs associated with the original AFSMs. This mathematical paradigm will be shown to be useful in the modeling and analysis of ATM systems.

### 1.1.2 Hybrid Systems

Hybrid systems are among the most natural models to deal with hierarchical and complex systems, as the ones considered in this thesis. Hybrid Systems are models that combine behaviors of purely continuous dynamics with discrete-event dynamics [28]. The state of such a system is composed of a discrete component and a continuous component. More formally, a hybrid system consists of an automaton with a finite number of discrete states (modes), and continuous dynamics associated to each mode. The model of the automaton is formalized through a Finite State Machine (FSM) (see e.g. [59], [30]) while continuous dynamics are modelled by differential equations [60]. The evolution of a hybrid system is induced by discrete events causing transitions from a mode to another. These events may be controllable or not and depend in general on the continuous state. Moreover, as a consequence of a transition from a mode to another, the continuous state may be reset instantly to a new value depending, in general, on the hybrid state before the transition. Multiple instantaneous transitions are allowed. All these features make the expressive power of hybrid systems much higher than the one of finite automata or purely continuous systems. Research efforts in these directions started, with some pioneering work, in the 1960s [61], but a general formalization of hybrid systems did not appear till the 1990s [62], [63].

## 1.2 Contributions

The contributions of this thesis are summarized in:

- Providing a sound mathematical paradigm that appropriately models agents acting in ATM procedures in both nominal and non-nominal operating modes;
- Providing a compositional framework that appropriately models the interaction among the agents involved in ATM procedures;
- Providing a formal methodology to analyse Multi–Agent Situation Awareness (MASA) inconsistencies arising in the evolution of ATM procedures, which may lead to unsafe and/or catastrophic events;
- Providing efficient algorithms for the reduction of the computational complexity arising in the analysis of MASA inconsistencies of realistic ATM scenarios, in which a large number of agents operate;

## 1.3 Thesis Outline

This thesis is organized as follows.

In Chapter 2 we recall the classes of finite state machines and hybrid systems.

In Chapter 3 we recall equivalence notions between finite state machines and compositional finite state machines and hybrid systems.

Chapter 4 presents novel results on critical observability and complexity reduction. This chapter is mainly based on the following publications:

- M.D. Di Benedetto, G. Pola, E. De Santis, A. Petriccone. *A Complexity Reduction Approach to the Detection of Safety Critical Situations in Air Traffic Management Systems*. Proc. of 49th IEEE Conference on Decision and Control, Atlanta, USA, 2010.
- E. De Santis, M.D. Di Benedetto, A. Petriccone, G. Pola. *Safety criticality analysis of complex Air Traffic Management systems via compositional bisimulation*. Submitted to 4th IFAC Conference on Analysis and Design of Hybrid Systems, Eindhoven, 6-8 June 2012.

Chapter 5 concerns the modeling and analysis of some air traffic management systems procedures. This chapter is mainly based on the work:

- E. De Santis, M.D. Di Benedetto, A. Petriccone, G. Pola. *A Compositional Hybrid System Approach to the Analysis of Air Traffic Management Systems*. EUROCONTROL Innovative Research Workshop and Exhibition, Brétigny-sur-Orge, France, 8-10 December 2009.
- M.D. Di Benedetto, A. D'Innocenzo, A. Petriccone, G. Pola. *Report on Observability Properties of Hybrid-System Composition*. Deliverable 4.2, STREP Project iFly, 31 March 2010.
- M.D. Di Benedetto, A. Petriccone, G. Pola. *Intermediate Report on Review of SESAR 2020 Conops*. Deliverable 4.1i, MAREA, 21 May 2011.
- M.D. Di Benedetto, A. Petriccone, G. Pola. *Review of SESAR 2020 Conops*. Deliverable 4.2, MAREA, 23 August 2011.

Chapter 6 offers concluding remarks.

Chapter 2

# Modeling

In this chapter we introduce the class of hybrid systems and the class of finite state machines. The chapter is organized as follows. In Section 2.1 we introduce the basic notation used in the sequel of the thesis. Section 2.2 contains the definition of finite state machines. In Section 2.3 we introduce the class of hybrid systems. Section 2.4 focuses on some subclasses of hybrid systems, as hybrid automata, rectangular automata, multi-rate automata and timed automata and we summarize main future of UPPALL for the automatic formal verification of timed automata.

## 2.1 Notation

Given a set $A$, the symbol $2^A$ denotes the set of subsets of $A$ and the symbol $|A|$ denotes the cardinality of $A$. If $|A| = 1$ then $A$ is said a singleton. Let $x = (x_1, x_2, \ldots, x_n)$ be a vector on Euclidean space $\mathbb{R}^n$, we recall the definition of Euclidean norm as follows:

$$\|x\| = \sqrt{x_1^2 + \ldots + x_n^2}.$$

A relation $R \subseteq A \times B$ is said to be total if for any $a \in A$ there exists $b \in B$ such that $(a, b) \in R$ and conversely, for any $b \in B$ there exists $a \in A$ such that $(a, b) \in R$. Given a relation $R \subseteq A \times B$, the inverse of $R$, denoted $R^{-1}$, is defined as $\{(b, a) \in B \times A : (a, b) \in R\}$. A relation $R \subseteq A \times B$ is the identity relation if $A = B$ and $a = b$ for all $(a, b) \in R$.

A directed graph is a tuple $G = (V, E)$ where $V$ is the set of vertices and $E$ is the set of edges. We denote by $\mathbb{N}$ the set of positive integers and by $\mathbb{R}$ the set of real numbers.

A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is said to be Lipschitz continuous if there exists a constant $L > 0$ so that $\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$ for any $x_1, x_2 \in \mathbb{R}^n$.

We recall that a subset $B$ of the Euclidean space $\mathbb{R}^n$ is said to be rectangular if it can be represented as:

$$B = B_1 \times B_2 \times \cdots \times B_n,$$

where $B_i$ are bounded or unbounded intervals of the real line $\mathbb{R}$. In particular, given $a_i, b_i \in \mathbb{R} \cup \{+\infty\} \cup \{-\infty\}$, the set $B_i$ can be either of the form $[a_i, b_i]$, or $[a_i, b_i[$, or $]a_i, b_i]$, or $]a_i, b_i[$ where the real numbers $a_i \dots b_i$ can also coincide. Given two sets X and Y, we define the shuffle product as follows:

$$X \times Y = \{(x, y) | x \in X \wedge y \in Y\}$$

## 2.2 Finite State Machines

In this section we review the notion of finite state machines. A Mealy machine is a finite state machine that generates an output based on its current state and input. This means that the state diagram will include both an input and output signal for each transition edge. A Moore machine is a finite state machine where the outputs are determined by the current state alone and do not depend directly on the input. The state diagram for a Moore machine will include an output signal for each state.

**Definition 2.2.1.** *A Mealy Finite State Machine is a tuple*

$$M = (Q, q_0, \Sigma, \Psi, \eta, E), \tag{2.1}$$

*where:*

- $Q$ *is a finite set of states;*
- $q_0 \in Q$ *is an initial state;*
- $\Sigma$ *is a finite set of input symbols;*
- $\Psi$ *is a finite set of output symbols;*
- $\eta : E \to \Psi$ *is an output map;*
- $E \subseteq Q \times \Sigma \times Q$ *is a transition relation.*

**Definition 2.2.2.** *A Moore Finite State Machine is a tuple*

$$M = (Q, q_0, \Sigma, \Psi, \eta, E), \tag{2.2}$$

*where:*

- $Q$ *is a finite set of states;*
- $q_0 \in Q$ *is an initial state;*
- $\Sigma$ *is a finite set of input symbols;*
- $\Psi$ *is a finite set of output symbols;*
- $\eta : Q \to \Psi$ *is an output map;*

- $E \subseteq Q \times \Sigma \times Q$ *is a transition relation.*

The use of a Mealy FSM leads often to a reduction of the number of states. A Moore finite state machine output is shown inside the state bubble, because the output remains the same as long as the state machine remains in that state. The output can be arbitrarily complex but must be the same every time the machine enters that state. The advantage of the Moore model is a simplification of the behavior. Despite classical formulations of Moore finite state machines, in this thesis we model the transition relation $E$ as a subset of $Q \times 2^{\Sigma} \times Q$ and the output function $\eta$ as a function from $Q$ to $2^{\Psi}$. By this choice, multiple interactions of FSMs can be considered; this will useful in the sequel to model agents acting in air traffic management system's scenarios.

**Definition 2.2.3.** *[42] A Finite State Machine (FSM) is a tuple*

$$M = (Q, q_0, \Sigma, \Psi, \eta, E), \tag{2.3}$$

*where:*

- $Q$ *is a finite set of states;*
- $q_0 \in Q$ *is an initial state;*
- $\Sigma$ *is a finite set of input symbols;*
- $\Psi$ *is a finite set of output symbols;*
- $\eta : Q \to 2^{\Psi}$ *is an output map;*
- $E \subseteq Q \times 2^{\Sigma} \times Q$ *is a transition relation.*

In the sequel we will work with FSMs as in the above definition. We denote a transition $(q, \sigma, q') \in E$ of FSM $M$ by $q \xrightarrow[E]{\sigma} q'$.

A state run of M is a (possibly infinite) sequence of transitions:

$$q_0 \xrightarrow[E]{\sigma_1} q_1 \xrightarrow[E]{\sigma_2} \dots \tag{2.4}$$

with $q_0 \in Q$ and $(q_i, \sigma_{i+1}, q_{i+1}) \in E$. An output run is a (possibly infinite) sequence $\{\psi_i\}_{i \in \mathbb{N}_0}$ such that there exists a state run of the form (2.4) with $\psi_i = \eta(q_i)$, $i \in \mathbb{N}_0$.

**Definition 2.2.4.** *[30] Let $\mathcal{L}(M)$ be the language generated by M, or equivalently, the collection of output runs of FSM M.*

When $q_0$ is skipped from the tuple in (2.3) any state in $Q$ is assumed to be an initial state. By definition of $E$, a transition of the form $q \xrightarrow[E]{\varnothing} q'$ is allowed. Such a transition is viewed as private or internal to $M$. Throughout this thesis we refer to an input $\sigma = \varnothing$ as internal, and an input $\sigma \neq \varnothing$ as external to $M$. Analogously, for a state $q \in Q$, $\eta(q) = \varnothing$ is allowed, meaning that state $q$ is not visible from the external environment.

## 2.3 Hybrid Systems

Hybrid systems provide a powerful mathematical framework to describe many application domains of interest characterized by the interaction of discrete and continuous variables. Discrete dynamics of hybrid systems are typically modeled by means of FSMs, while continuous dynamics by means of nonlinear dynamical control systems. Interaction between discrete and continuous variables are captured by invariant, guard, and reset conditions.

The following definition of hybrid systems is inspired by the classical model proposed in [29].

**Definition 2.3.1** (Hybrid system)**.** *A hybrid system is a tuple*

$$\mathcal{H} = (Q \times X, \{q_0\} \times X_0, U, Y, \mathcal{E}, \Sigma, E, \Psi, \eta, Inv, G, R), \tag{2.5}$$

*where:*

- $Q \times X$ *is the hybrid state space, where:*
  - $Q$ *is a finite set of* $N$ *discrete states;*
  - $X \subseteq \mathbb{R}^n$ *is the continuous state space.*
- $\{q_0\} \times X_0 \subseteq Q \times X$ *is the set of initial discrete and continuous conditions.*
- $U \subseteq \mathbb{R}^m, Y \subseteq \mathbb{R}^p$ *are the sets of continuous control inputs and outputs.*
- $\{\mathcal{E}(q)\}_{q \in Q}$ *associates to each discrete state* $q \in Q$ *the continuous time–invariant dynamics*

  $$\mathcal{E}(q)\colon \dot{x} = f_q(x, u),$$

  *and the output map* $y = g_q(x)$*. Given an initial condition* $x_0$ *at time* $t_0$ *and a control input* $u|_{t_0}^t\colon [t_0, t] \to U$*, we denote the solution at time* $t$ *according to* $f_q$ *by*

  $$x(t) = x_q(t, x_0, u|_{t_0}^t).$$

  *The solution of the above differential equation exists and it is unique, provided that* $f_q$ *is Lipschitz continuous with respect to its arguments.*
- $\Sigma$ *is the set of discrete input symbols. It includes the empty string* $\varepsilon$*, that corresponds to the null input.*
- $E \subseteq Q \times 2^{\Sigma} \times Q$ *is a collection of edges.*
- $\Psi$ *is the finite set of discrete output symbols. It includes the empty string* $\varepsilon$*, that corresponds to unobservable output.*
- $\eta\colon Q \to 2^{\Psi}$ *is the output function, that associates to each edge a discrete output symbol.*

- $\{Inv_q\}_{q\in Q}$ *associates to each discrete state* $q \in Q$ *an invariant set* $Inv_q \subseteq X$.
- $\{G_e\}_{e\in E}$ *associates to each edge* $e = (q, \sigma, q') \in E$ *a guard set* $G_e \subseteq Inv_{q'}$.
- $\{R_e\}_{e\in E}$ *associates to each edge* $e = (q, \sigma, q') \in E$ *a reset map* $R_e \colon Inv_{q'} \to 2^{Inv_{q'}}$.

Referring to [29], we recall the definition of *hybrid time basis.*

**Definition 2.3.2** (Hybrid time basis)**.** *A hybrid time basis*

$$\tau \triangleq \{I_k\}_{0\leq k\leq |\tau|}$$

*is a finite or infinite sequence of intervals* $I_k = [t_k, t'_k]$. *The length* $t'_k - t_k$ *of every interval* $I_k$ *denotes the dwelling time in a discrete state, while the extremes* $t_k, t'_k$ *specify the switching instants of the hybrid flow. The number of such intervals is* $|\tau| + 1$, *where* $|\tau|$ *is the cardinality of the time basis. Furthermore, the following conditions hold:*

1. $t_k \leq t'_k$ for $k > 0$, and $t'_{k-1} = t_k$ for $k > 1$;
2. If the sequence is infinite, i.e. $|\tau| = \infty$, then $I_k$ is closed for all $k$;
3. If the sequence is finite, i.e. $|\tau| < \infty$, then the last interval $I_{|\tau|}$ might be right–open.

We can now formally introduce the semantic of hybrid systems which is given by the notion of execution.

**Definition 2.3.3** (Hybrid execution)**.** *A hybrid execution is a tuple*

$$\chi = (\tau, \sigma, u, q, x, y, \eta),$$

*where:*

- $\tau$ *is a hybrid time basis;*
- $\sigma : \tau \to 2^{\Sigma}$ *is the discrete input;*
- $u$ *is the continuous input;*
- $q$ *describe the evolution of the discrete state by means of function* $q \colon \tau \to Q$*;*
- $x$ *describes the evolution of the continuous state by means of functions* $x \colon \mathbb{R}^+ \to X$ *so that for any* $t \in I_j$ *and* $I_j \in \tau$, $x(t)$ *is the unique solution of the differential equation with initial time* $t_j$, *initial condition* $x(t_j)$ *and control* $u|_{I_j}$*;*
- $y$ *is the continuous output;*
- $\eta$ *is the output function.*

Executions can be classified in the following three categories, according to the time basis on which they are defined:

- Finite, if $\tau$ is a finite sequence and the last interval in $\tau$ is closed;
- Infinite, if $\tau$ is an infinite sequence, or $\|\tau\| = \infty$;
- Zeno, if it is infinite but $\|\tau\| < \infty$.

In the sequel we will need the notion of non-blocking hybrid systems, as recalled in the following definition.

**Definition 2.3.4** (Non-blocking Hybrid System)**.** *A non-deterministic hybrid system $\mathcal{H}$ is called non-blocking if for all initial states $(\widehat{q}, \widehat{x}) \in \{q_0\} \times X_0$ there exists an infinite execution starting at $(\widehat{q}, \widehat{x})$.*

It is readily seen that by extracting entities $Q, q_0, \Sigma, \Psi, \eta$ and $E$ from the tuple in 2.5, defining a hybrid system, we obtain a finite state machine in the sense of definition 2.2.3.

## 2.4 Hybrid Automata

In this section we introduce some subclasses of hybrid systems: hybrid automata, rectangular automata, multi-rate automata and timed automata.

**Definition 2.4.1** (Hybrid automaton)**.** *A hybrid automaton is a hybrid system*

$$\mathcal{H} = (Q \times X, \{q_0\} \times X_0, U, Y, \mathcal{E}, \Sigma, E, \Psi, \eta, \text{Inv}, G, R)$$

*where $U = \varnothing$. In the further developments we will remove the symbol $U$ in the tuple and refer to a hybrid automaton by means of the tuple:*

$$\mathcal{H} = (Q \times X, \{q_0\} \times X_0, Y, \mathcal{E}, \Sigma, E, \Psi, \eta, \text{Inv}, G, R)$$

A hybrid automaton may be non deterministic. Hybrid automata are very important because the most advanced results in the formal analysis of hybrid systems have been obtained for this class of systems.
We now introduce a subclass of hybrid automata which will be useful in the sequel to model different agents acting in Air Traffic Management systems: the class of rectangular automata. A rectangular automaton is a hybrid automaton where each of the sets involved is a rectangular set. More formally:

**Definition 2.4.2** (Rectangular automaton)**.** *A rectangular automaton is a hybrid automaton*

$$\mathcal{H} = (Q \times X, \{q_0\} \times X_0, Y, \mathcal{E}, \Sigma, E, \Psi, \eta, \text{Inv}, G, R)$$

*where:*

- *The set $X_0$ is rectangular;*
- *For every $q \in Q$, the set $Inv$ is rectangular;*
- *For every $q \in Q$, there is a rectangular set $B^q$ such that*

$$\mathcal{E}_q : \dot{x} \in B^q;$$

- *For every edge $e \in E$, the set $G_e$ is rectangular;*
- *For every edge $e \in E$, the set $R_e$ is rectangular.*

*A rectangular automaton is said to be* initialized *if the following condition holds: if the bounds on the derivative of a continuous variable $x_i \in \mathbb{R}$ change after a discrete transition $e$, then the continuous variable must be nondeterministically reset within a rectangular interval $I$, i.e. $R_e(x_i) = I, \forall x_i \in \mathbb{R}$.*

*Multi-rate automata* can be seen as a particular case of rectangular automata:

**Definition 2.4.3** (Multi-Rate automaton)**.** *A multi-rate automaton is a rectangular automaton that satisfies the following constraints:*

- *$X_0$ is a singleton set;*
- *For every $q \in Q$, the set $B^q$ is a singleton state;*
- *For every edge $e \in E$, the set $R_e$ is a singleton set.*

In a multi-rate automaton, each variable follows constant, rational slope, which may be different in different locations. The simplest hybrid automaton is the *timed automaton*:

**Definition 2.4.4** (Timed automaton)**.** *A timed automaton is a multi-rate automaton such that for every $q \in Q, B^q = \{(1, 1, \ldots, 1)\}$.*

Timed automata can be used to encode timing constraints and their variables can be seen as clocks associated to the time that the state spends in a discrete state. There are strong results for the verification of properties of timed automata. Moreover, automatic model checking tools for timed automata are available, e.g. *UPPAAL* [10].
The expressive power of rectangular automata is in general greater than the one of multirate automata, which in turn is greater than the one of timed automata. However, special classes of rectangular automata and timed automata are equivalent in the sense of bisimulation equivalence [43],[15]. We do not provide here formal statements of equivalence between the two models. We only mention that initialized rectangular automata can be translated into timed automata. A comprehensive exposition of such equivalence among the classes of systems introduced in this section, can be found in [53].

UPPAAL [10] is a toolbox for the verification of real-time systems jointly developed by Uppsala University and Aalborg University. The tool is designed to verify systems that can be modeled as networks of timed automata extended with integer variables, structured data types, and channel synchronization. This model-checker is based on the theory of timed automata and its modeling language offers additional features such as bounded integer variables and urgency. The query language of UPPAAL, used to specify properties to be checked, is a subset of CTL (Computation Tree Logic, [52]).

### The Modeling Language

In UPPAAL a system is modeled as a network of several timed automata in parallel. The model is further extended with discrete variables that are part of the state. These variables are used as in programming languages: they are read, written, and are subject to common arithmetic operations.

A state of the system is defined by the locations of all automata, the clock constraints, and the values of the discrete variables. Every automaton may fire an edge separately or synchronize with another automaton, which leads to a new state. We give the basic definitions of the syntax and semantics for timed automata: C

is a set of clocks and B(C) is the set of conjunctions over simple conditions of the form $x \bowtie c$ or $x - y \bowtie c$, where $x, y \in C$, $c \in \mathbf{N}$ and $\bowtie \in \{<, \leq, =, >, \geq\}$.

The UPPAAL modeling language extends timed automata with the following additional features:

- *Templates*: automata are defined with a set of parameters that can be of any type;
- *Binary synchronization*: channels are declared as `chan c`. An edge labeled with `c!` synchronizes with another labeled `c?` ;
- *Broadcast channels*: are declared as `broadcast chan c`. In a broadcast synchronization one sender `c!` can synchronize with an arbitrary number of receivers `c?`; any receiver that can synchronize in the current state must do so. If there are no receivers, then the sender can still execute the `c!` action, i.e. broadcast sending is never blocking;

Expressions in UPPAAL range over clocks and integer variables. Expressions are used with the following labels:

- *Guard*: is a particular expression satisfying the following conditions: it is side-effect free; it evaluates to a boolean; only clocks, integer variables, and constants are referenced (or arrays of these types); clocks and clock differences are only compared to integer expressions; guards over clocks are essentially conjunctions.
- *Synchronization*: a synchronization label is either on the form *Expression!* or *Expression?* or is an empty label. The expression must be side-effect free, evaluate to a channel, and only refer to integers, constants and channels;
- *Assignment*: an assignment label is a comma separated list of expressions with a side-effect; expressions must only refer to clocks, integer variables, and constants and only assigns integer values to clocks;
- *Invariant*: an invariant is an expression that satisfies the following conditions: it is side-effect free; only clock, integer variables, and constants are referenced; it is a conjunction of conditions of the form $x < e$ or $x \leq e$, where $x$ is a clock reference and e evaluates to an integer.

**The Query Language**

The main purpose of a model checker is to verify the model with respect to a requirement specification. Like the model, the requirement specification must be expressed in a formally well-defined and machine readable language. Several such logics exist in the scientific literature, and UPPAAL uses a simplified version of CTL [52].

Like in CTL, the query language consists of path formulae and state formulae. State formulae describe individual states, whereas path formulae quantify over paths or traces of the model. Path formulae can be classified into reachability, safety and liveness. A state formula is an expression that can be evaluated for a state without looking at the behavior of the model. For instance, this could be a simple expression, like $i == 7$, that is true in a state whenever $i$ equals 7. The syntax of state formulae is a superset of that of guards, i.e., a state formula is a side-effect free expression, but in contrast to guards, the use of disjunctions is not restricted. It is also possible to test whether a particular process is in a given location using an expression on the form P.$q_1$, where P is a process and $q_1$ is a location.

- **Reachability Properties**: Reachability properties are the simplest class of properties. They ask whether a given state formula $\varphi$, *possibly* can be satisfied by any reachable state. *Does a path exist, starting from the initial state, such that $\varphi$ is eventually satisfied?* Reachability properties are often used while designing a model to perform safety checks. We express that some state satisfying $\varphi$ should be reachable using the path formula $E \diamond \varphi$, and in UPPAAL we write this property using the syntax $E <> \varphi$.

- **Safety Properties**: Safety properties are of the form *something bad will never happen.* A variation of this property is that *something will possibly never happen.* For instance, when playing a game, a safe state is one in which we can still win the game, hence we will possibly not loose. In UPPAAL these properties are formulated positively, e.g. *something good is invariantly true.* Let $\varphi$ be a state formula, we express that $\varphi$ should be true in all reachable states with the path formulae $A[]\ \varphi$, whereas $E[]\ \varphi$ means that there should exist a maximal[1] path such that $\varphi$ is always true.

- **Liveness Properties**: Liveness properties are of the form *something will eventually happen*, e.g. in a model of a communication protocol we may require that any message that has been sent should eventually be received. Liveness is expressed with the path formula $A <> \varphi$, meaning that $\varphi$ is eventually satisfied. The most useful form is the *leads to or response property*, which can be expressed as $\varphi \rightarrow \psi$, namely whenever $\varphi$ is satisfied, then eventually $\psi$ will be satisfied; in the communication protocol example, whenever a message is sent then eventually it will be received.

[1] A maximal path is a path that is either infinite or where the last state has no outgoing transitions.

Chapter 3

# Equivalence Notions and Composition

In this thesis we focus on the notion of bisimulation equivalence that is widely used as an effective tool to mitigate complexity of verification and control design of large scale complex systems. The chapter is organized as follows. In Section 3.2 we recall some equivalence notions. In Section 3.3 we focus on composition by defining the notion of arenas of finite state machine and arenas of hybrid systems. In Section 3.4 we introduce a novel class of equivalence for arenas of finite state machine, termed compositional bisimulation.

## 3.1 Equivalence notions of Finite State Machine

Several notions of equivalence have been proposed for the class of finite state machines as for example isomorphism, bisimulation, 2-nested simulation and language equivalence. The interested reader is referred to [39] where a detailed description of these notions together with a formal analysis of the relationships among them are reported.
Consider a pair of FSMs $M_i = (Q_i, q_{0,i}, \Sigma_i, \Psi_i, \eta_i, E_i)$ $(i = 1, 2)$. We start by recalling the notion of isomorphism.

**Definition 3.1.1.** *The FSMs* $M_1 = (Q_1, q_{0,1}, \Sigma_1, \Psi_1, \eta_1, E_1)$ *and* $M_2 = (Q_2, q_{0,2}, \Sigma_2, \Psi_2, \eta_2, E_2)$ *are isomorphic, denoted* $M_1 \cong^{\text{iso}} M_2$*, if there exists a bijective function*

$$\mathcal{T} : Q_1 \to Q_2$$

*such that:*

*(i)* $q_{0,2} = \mathcal{T}(q_{0,1})$.

*(ii)* $\eta_1(q_1) = \eta_2(\mathcal{T}(q_1))$ *for any* $q_1 \in Q_1$.

*(iii)* $q_1 \xrightarrow[E_1]{\sigma} q_1'$ *if and only if* $\mathcal{T}(q_1) \xrightarrow[E_2]{\sigma} \mathcal{T}(q_1')$.

The notion of isomorphism is an equivalence relation on the class of FSMs. The notion of simulation relation is reported hereafter.

**Definition 3.1.2.** *Given a pair of FSMs* $M_1 = (Q_1, q_{0,1}, \Sigma_1, \Psi_1, \eta_1, E_1)$ *and* $M_2 = (Q_2, q_{0,2}, \Sigma_2, \Psi_2, \eta_2, E_2)$*, a set*

$$R \subseteq Q_1 \times Q_2$$

*is a simulation relation from* $M_1$ *to* $M_2$ *if for any* $(q_1, q_2) \in R$ *the following conditions are satisfied:*

*(i)* $\eta_1(q_1) = \eta_2(q_2)$*;*

*(ii) existence of* $q_1 \xrightarrow[E_1]{\sigma_1} q_1'$ *implies existence of* $q_2 \xrightarrow[E_2]{\sigma_2} q_2'$ *such that* $\sigma_1 = \sigma_2$ *and* $(q_1', q_2') \in R$*;*

*(iii)* $(q_{0,1}, q_{0,2}) \in R$*.*

*The FSM* $M_1$ *is simulated by the FSM* $M_2$*, or equivalently* $M_2$ *simulates* $M_1$*, denoted* $M_1 \preceq M_2$*, if there exists a simulation relation from* $M_1$ *to* $M_2$*.*

We now recall hereafter the definition of 2-nested equivalence and language equivalence [39].

**Definition 3.1.3** (2-Nested equivalence)**.** *Two FSMs* $M_1$ *and* $M_2$ *are 2-Nested equivalent if* $\Sigma_1 \preceq \Sigma_2$ *and* $\Sigma_2 \preceq \Sigma_1$*.*

**Definition 3.1.4** (Language equivalence)**.** *Two FSMs* $M_1$ *and* $M_2$ *are language equivalent if* $\mathcal{L}(M_1) = \mathcal{L}(M_2)$*.*

We finally introduce the notion of bisimulation equivalence.

**Definition 3.1.5.** *Given a pair of FSMs* $M_1 = (Q_1, q_{0,1}, \Sigma_1, \Psi_1, \eta_1, E_1)$ *and* $M_2 = (Q_2, q_{0,2}, \Sigma_2, \Psi_2, \eta_2, E_2)$*, a set*

$$R \subseteq Q_1 \times Q_2,$$

*is a bisimulation relation between* $M_1$ *and* $M_2$ *if:*

*(i)* $R$ *is a simulation relation from* $M_1$ *to* $M_2$*;*

*(ii)* $R^{-1}$ *is a simulation relation from* $M_2$ *to* $M_1$*.*

*FSMs* $M_1$ *and* $M_2$ *are bisimilar, denoted* $M_1 \cong M_2$*, if there exists a bisimulation relation between* $M_1$ *and* $M_2$*.*

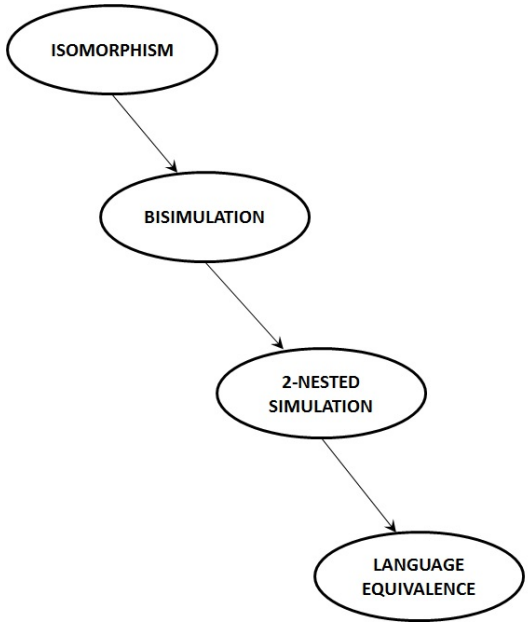



**Figure 3.1:** Isomorphism, Bisimulation, 2-Nested Simulation and Language equivalence.

Relationships among the above notions of equivalence are reported in Figure 5.34.

In this thesis we focus on the notion of bisimulation equivalence that is widely used as an effective tool to mitigate complexity of verification and control design of large scale complex systems. Basic facts about bisimulation are reported hereafter. For a detailed description of these concepts we refer to [5, 43, 15].

**Proposition 3.1.6.** *Bisimulation equivalence is an equivalence relation on the class of FSMs, i.e. it satisfies the following properties:*

*(i) (Reflexivity)* $M \cong M$.

*(ii) (Symmetry)* $M_1 \cong M_2$ *implies* $M_2 \cong M_1$.

*(iii) (Transitivity)* $M_1 \cong M_2$ *and* $M_2 \cong M_3$ *imply* $M_1 \cong M_3$.

**Definition 3.1.7.** *The maximal bisimulation relation between FSMs* $M_1$ *and* $M_2$ *is a bisimulation relation* $R^*(M_1, M_2)$ *such that* $R \subseteq R^*(M_1, M_2)$ *for any bisimulation relation* $R$ *between* $M_1$ *and* $M_2$.

**Proposition 3.1.8.** *The maximal bisimulation relation exists and is unique.*

**Proposition 3.1.9.** *Given an FSM* $M$ *the set* $R^*(M, M)$ *is an equivalence relation on the set of states of* $M$*, i.e. it satisfies the following properties:*

*(i) (Reflexivity)* $(q, q) \in R^*(M, M)$.

*(ii) (Symmetry) If* $(q_1, q_2) \in R^*(M, M)$ *then* $(q_2, q_1) \in R^*(M, M)$.

*(iii) (Transitivity) If* $(q_1, q_2) \in R^*(M, M)$ *and* $(q_2, q_3) \in R^*(M, M)$ *then* $(q_1, q_3) \in R^*(M, M)$.

We next recall hereafter the notion of quotient [5] of an FSM $M$ induced by the equivalence relation $R^*(M, M)$.

**Definition 3.1.10.** *The quotient of an FSM* $M = (Q, q_0, \Sigma, \Psi, \eta, E)$ *induced by* $R^*(M, M)$ *is the FSM*

$$M^* = (Q^*, q_0^*, \Sigma^*, \Psi^*, \eta^*, E^*),$$

*where:*

- $Q^* = \{C_1, C_2, ..., C_N\}$, *where* $C_i$ *are the equivalence classes induced by* $R^*(M, M)$ *on* $X$.
- $q_0^* = \{q^0\}$.
- $\Sigma^* = \Sigma$.
- $\Psi^* = \Psi$.
- $\eta^* : Q^* \to 2^{\Psi^*}$ *is defined by* $\eta^*(C_i) = \psi$, *if* $\eta(q) = \psi$ *for any* $q \in C_i$;
- $E^* \subseteq Q^* \times 2^{\Sigma^*} \times Q^*$ *is defined by* $C_i \xrightarrow[E^*]{\sigma} C_j$, *if* $q \xrightarrow[E]{\sigma} q'$ *for any* $q \in C_i$ *and* $q' \in C_j$.

**Proposition 3.1.11.** *[5] The quotient of* $M$ *induced by* $R^*(M, M)$, *denoted* $\mathbf{M}_{\min}(M)$, *is the FSM bisimilar to* $M$ *with the minimal number of states.*

**Proposition 3.1.12.** *FSM* $\mathbf{M}_{\min}(M)$ *exists and is unique up to isomorphisms.*

The following technical result will be used in the sequel.

**Lemma 3.1.13.** *If* $\mathbf{M}_{\min}(M_1) \cong \mathbf{M}_{\min}(M_2)$ *then* $M_1 \cong^{\text{iso}} M_2$.

*Proof.* Let $Q_i$ be the set of states of $M_i$. Minimality of $\mathbf{M}_{\min}(M_1)$ and $\mathbf{M}_{\min}(M_2)$ implies that the maximal bisimulation relation $R^*$ between $\mathbf{M}_{\min}(M_1)$ and $\mathbf{M}_{\min}(M_2)$ is such that for any $q_1 \in Q_1$ and $q_2 \in Q_2$, sets

$$R^*(q_1) = \{q_2 \in Q_2 | (q_1, q_2) \in R^*\}$$

and

$$(R^*)^{-1}(q_2) = \{q_1 \in Q_1 | (q_1, q_2) \in R^*\}$$

are singletons. Hence, define function $\mathcal{T} : Q_1 \to Q_2$ by $\mathcal{T}(q_1) = q_2$ when $R^*(q_1) = \{q_2\}$. It is easy to see that function $\mathcal{T}$ satisfies the properties that are required in Definition 3.1.1. □

We conclude this section by recalling space and time complexity in checking bisimulation equivalence between FSMs.

**Proposition 3.1.14.** *[44] Space complexity in checking $M_1 \cong M_2$ is*

$$O(|Q_1| + |E_1| + |Q_2| + |E_2|).$$

**Proposition 3.1.15.** *[44] Time complexity in checking $M_1 \cong M_2$ is*

$$O((|E_1| + |E_2|)\ln(|Q_1| + |Q_2|)).$$

## 3.2 Equivalence notions of Hybrid Systems

In the following we introduce some equivalence notions for the class of hybrid systems [64].

**Definition 3.2.1.** *Given two hybrid systems*

$$\mathcal{H}_i = (Q_i \times X_i, \{q_{0,i}\} \times X_{0,i}, U_i, Y_i, \mathcal{E}_i, \Sigma_i, E_i, \Psi_i, \eta_i)$$

*, $i = 1, 2$, such that $U_1 = U_2$. A hybrid bisimulation between $\mathcal{H}_1$ and $\mathcal{H}_2$ is a subset*

$$R \subseteq (Q_1 \times X_1) \times (Q_2 \times X_2)$$

*satisfying the following property. Take any $((q_{0,1}, x_{0,1}), (q_{0,2}, x_{0,2})) \in R$ and any input $u_1 = u_2$, then for any execution $\chi_1 = (\tau_1, \sigma_1, u_1, q_1, x_1, y_1, \eta_1)$ of $\mathcal{H}_1$, there should exist an execution $\chi_2 = (\tau_2, \sigma_2, u_2, q_2, x_2, y_2, \eta_2)$ of $\mathcal{H}_2$ satisfying the following conditions:*

- $((q_1(j), x_1(t, j)), (q_2(j), x_2(t, j))) \in R$,
- $(\psi_1(j), y_1(t, j)) = (\psi_2(j), y_2(t, j))$,

*for all $t \in I_j$ and $I_j \in \tau_1 = \tau_2$.*

**Definition 3.2.2.** *Two hybrid system $\mathcal{H}_1$ and $\mathcal{H}_2$ are bisimilar, denoted $\mathcal{H}_1 \cong \mathcal{H}_2$, if there exists a hybrid bisimulation $R$ such that the projection of $R$ on each hybrid state space equals this hybrid space:*

$$\pi|_{Q_i \times X_i}(R) = Q_i \times X_i, i = 1, 2$$

The proposed definition is mainly inspired by the classical notions given for concurrent processes ([65],[66],[67],[13]) and by the definitions introduced in [64].

## 3.3 Composition

### 3.3.1 Arenas of finite state machines

Given $N$ FSMs $M_i = (Q_i, q_{0,i}, \Sigma_i, \Psi_i, \eta_i, E_i)$ $(i = 1, 2, ..., N)$, the interaction among these FSMs can be captured by the classical notion of parallel composition [5]. The parallel composition of $N$ FSMs $M_i$, denoted by

$$M_1||M_2||\ldots||M_N$$

is the FSM

$$M = (Q, q_0, \Sigma, \Psi, \eta, E),$$

where:

- $Q = Q_1 \times Q_2 \times \ldots \times Q_N$;
- $q_0 = (q_{0,1}, q_{0,2}, \ldots, q_{0,N})$;
- $\Sigma = \Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_N$;
- $\Psi = \Psi_1 \times \Psi_2 \times \ldots \times \Psi_N$;
- $\eta((q_1, q_2, ..., q_N)) = (\eta(q_1), \eta(q_2), \ldots, \eta(q_N))$;
- $E \subseteq Q \times \Sigma \times Q$ is such that

$$(q_1, q_2, ..., q_N) \xrightarrow[E]{\sigma} (q'_1, q'_2, ..., q'_N), \tag{3.1}$$

whenever $q_i \xrightarrow[E_i]{\sigma_i} q'_i$ is a transition of $M_i$ for some $\sigma_i$ $(i = 1, 2, ..., N)$ and

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N).$$

We now introduce a notion of parallel composition in which a communication network is placed; this communication network describes the communication channel through which FSMs can communicate. We introduce this novel notion through the use of the concept of non–flat systems [31, 32]. *Arenas of Finite State Machines* (AFSMs) are a new class of non–flat systems, which can be roughly described as a collections of FSMs that interact concurrently through a communication network. The syntax of an AFSM is specified by a directed graph:

$$\mathbb{A} = (\mathbb{V}, \mathbb{E}),$$

where:

- $\mathbb{V}$ is a collection of $N$ FSMs $M_i = (Q_i, q_{0,i}, \Sigma_i, \Psi_i, \eta_i, E_i)$ $(i = 1, 2, ..., N)$;
- $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ describes the communication network of the FSMs $M_i$.

In the definition of $\mathbb{E}$ self loops $(M_i, M_i) \in \mathbb{E}$ would model communication of $M_i$ with itself, which is tautological. For this reason in the sequel we assume $(M_i, M_i) \notin \mathbb{E}$. By expanding each vertex $M_i \in \mathbb{V}$ of $\mathbb{A}$ an ordinary FSM is obtained, which is defined by:

$$\mathbb{M}(\mathbb{A}) = (Q, q_0, \Sigma, \Psi, \eta, E),$$

where:

- $Q = Q_1 \times Q_2 \times ... \times Q_N$;
- $q_0 = (q_{0,1}, q_{0,2}, ..., q_{0,N})$;
- $\Sigma = \bigcup_{M_i \in \mathbb{V}} \Sigma_i$;
- $\Psi = \bigcup_{M_i \in \mathbb{V}} \Psi_i$;
- $\eta((q_1, q_2, ..., q_N)) = \bigcup_{M_i \in \mathbb{V}} \eta_i(q_i)$;
- $E \subseteq Q \times 2^{\Sigma} \times Q$ is such that

$$(q_1, q_2, ..., q_N) \xrightarrow[E]{\sigma} (q'_1, q'_2, ..., q'_N), \tag{3.2}$$

whenever $q_i \xrightarrow[E_i]{\sigma_i} q'_i$ is a transition of $M_i$ for some $\sigma_i$ $(i = 1, 2, ..., N)$ and

$$\sigma = \bigcup_{M_i \in \mathbb{V}} (\sigma_i \backslash ( \bigcup_{M_j \in \mathrm{Pre}(\mathbb{A}, M_i)} \eta_j(q_j))), \tag{3.3}$$

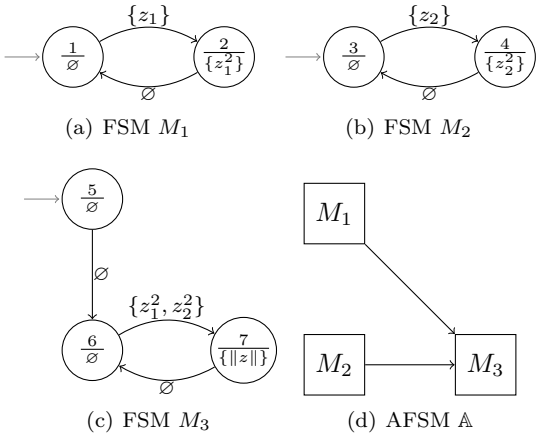
where $\mathrm{Pre}(\mathbb{A}, M_i) = \{M_j \in \mathbb{V} \,|\, (M_j, M_i) \in \mathbb{E}\}$.

**Proposition 3.3.1.** *Given an AFSM $\mathbb{A}$, the FSM $\mathbb{M}(\mathbb{A})$ is unique.*

*Proof.* Entities $Q$, $q_0$, $\Sigma$, $\Psi$ and $\eta$ in $\mathbb{M}(\mathbb{A})$ are uniquely determined from $\mathbb{A}$. For any collection of $N$ transitions $q_i \xrightarrow[E_i]{\sigma_i} q'_i$ in $M_i$ there exists one and only one transition in $\mathbb{M}(\mathbb{A})$ of the form (3.2) with $\sigma$ uniquely specified by (3.3). □

FSM $\mathbb{M}(\mathbb{A})$ specifies the semantics of the AFSM $\mathbb{A}$. Such a semantic is implicitly given through a composition of FSMs that can be regarded as a notion of parallel composition [5] that respects the topology of the AFSM communication network. The following simple example illustrates syntax and semantics of AFSMs.

**Example 3.3.1.** Consider a distributed system composed of three computers $C_1$, $C_2$ and $C_3$, whose goal is to compute the Euclidean norm $\|z\| = \sqrt{z_1^2 + z_2^2}$ of a vector $z = (z_1, z_2) \in \mathbb{R}^2$ in a distributed fashion. While $C_1$ and $C_2$ are delegated to compute respectively $z_1^2$ and $z_2^2$, $C_3$ takes as inputs the computations of $C_1$ and $C_2$ and outputs $\|z\|$. This simple distributed system can be modeled as the AFSM $\mathbb{A} = (\mathbb{V}, \mathbb{E})$ where $\mathbb{V} = \{M_1, M_2, M_3\}$ and $\mathbb{E} = \{(M_1, M_3), (M_2, M_3)\}$.




(a) FSM $M_1$ (b) FSM $M_2$

(c) FSM $M_3$ (d) AFSM $\mathbb{A}$

**Figure 3.2:** AFSM $\mathbb{A}$ in Example 3.3.1.

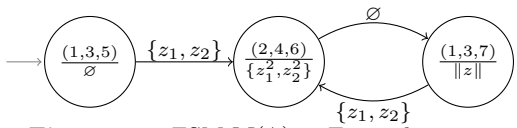



**Figure 3.3:** FSM $\mathbb{M}(\mathbb{A})$ in Example 3.3.1.

FSMs $M_i$, each one modeling computers $C_i$, are illustrated in Figures[1] 3.2(a)(b)(c), while AFSM $\mathbb{A}$, modeling the computers' network, is depicted in Figure 3.2(d).

By expanding $\mathbb{A}$, the FSM $\mathbb{M}(\mathbb{A})$ is obtained, whose accessible part [2] is depicted in Figure 3.3. Starting from $(1, 3, 5)$, when receiving the input $\{z_1, z_2\}$, FSM $\mathbb{M}(\mathbb{A})$ outputs in state $(2, 4, 6)$ the set $\{z_1^2, z_2^2\}$ and finally in state $(1, 3, 7)$ the requested output $\{\|z\|\}$. For illustrating the construction of FSM $\mathbb{M}(\mathbb{A})$, we describe in detail the construction of the transition $(2, 4, 6) \xrightarrow[\Delta]{u} (1, 3, 7)$. By applying the compositional rules defining the semantics of AFSMs, one gets: $2 \xrightarrow{\varnothing} 1$ is in $M_1$, $4 \xrightarrow{\varnothing} 3$ is in $M_2$, and $6 \xrightarrow{\{z_1^2, z_2^2\}} 7$ is in $M_3$. Moreover, one first note that $\mathrm{Pre}(\mathbb{A}, M_1) = \mathrm{Pre}(\mathbb{A}, M_2) = \varnothing$ and $\mathrm{Pre}(\mathbb{A}, M_3) = \{M_1, M_2\}$, from which $u = \varnothing$. The resulting transition $(2, 4, 6) \xrightarrow{\varnothing} (1, 3, 7)$ is indeed in $\mathbb{M}(\mathbb{A})$, as shown in Figure 3.3.

[1] Each circle denotes a state and each edge a transition. In each circle, upper symbol denotes the state and lower symbol the output set associated with the state; symbols labeling edges denote the input sets associated with the transitions.

[2] The accessible part of the FSM $M$ in (2.3) is the unique sub–finite state machine extracted from $M$, containing all and only the states of $M$ that are reachable (or equivalently, accessible) in a finite number of transitions from its initial state $x^0$, see e.g. [30].

### 3.3.2 Arenas of hybrid systems

Interaction among different hybrid systems can be captured by an appropriate notion of composition that we now introduce. Consider $N \geq 1$ hybrid systems:

$$\mathcal{H}_i = (Q_i \times X_i, q_{0,i} \times X_{0,i}, U_i, Y_i, \mathcal{E}_i, \Sigma_i, E_i, \Psi_i, \eta_i).$$

The evolution of each hybrid system $\mathcal{H}_i$ depends on the information that $\mathcal{H}_i$ has from all hybrid systems $\mathcal{H}_j$ sharing information with it.

The notion of composition that we focus here is obtained by adapting the notion of AFSM, described in the previous section. An *Arena of Hybrid Systems* (AHS) is described by a directed graph

$$\mathbb{A}_h = (\mathbb{V}, \mathbb{E})$$

where:

- $\mathbb{V} = \{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_N\}$ is the set of vertices.
- $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is the set of edges, where $(\mathcal{H}_i, \mathcal{H}_j) \in \mathbb{E}$, if $\mathcal{H}_i$ interacts with $\mathcal{H}_j$.

By expanding each vertex $\mathcal{H}_i \in \mathbb{V}$ of $\mathbb{A}_h$ an ordinary hybrid system is obtained, which is defined by:

$$\mathbb{H}(\mathbb{A}_h) = (Q \times X, \{q_0\} \times X_0, U, Y, \mathcal{E}, \Sigma, E, \Psi, \eta),$$

where:

- $Q = Q_1 \times Q_2 \times \ldots \times Q_N$.
- $X = X_1 \times X_2 \times \ldots \times X_N$.
- $q_0 = (q_{0,1}, q_{0,2}, \ldots, q_{0,N})$.
- $X_0 = X_{0,1} \times X_{0,2} \times \ldots \times X_{0,N}$.
- $U = U_1 \times U_2 \times \ldots \times U_N$.
- $Y = Y_1 \times Y_2 \times \ldots \times Y_N$.
- $\mathcal{E}$ associates to each discrete state $(q_1, q_2, ..., q_N) \in Q$ the continuous dynamics

  $$\dot{x} = (f_{1,q_1}(x_1, u_1), f_{2,q_2}(x_2, u_2), ..., f_{N,q_N}(x_N, u_N)),$$

  with output $y = (g_{1,q_1}(x_1), g_{2,q_2}(x_2), ..., g_{N,q_N}(x_N))$.
- $\Sigma = \bigcup_{H_i \in \mathbb{V}} \Sigma_i$;
- $\Psi = \bigcup_{H_i \in \mathbb{V}} \Psi_i$;
- $\eta((q_1, q_2, ..., q_N)) = \bigcup_{H_i \in \mathbb{V}} \eta_i(q_i)$;

- $E \subseteq Q \times 2^{\Sigma} \times Q$ is such that

$$(q_1, q_2, ..., q_N) \xrightarrow[E]{\sigma} (q_1', q_2', ..., q_N'),$$

whenever $q_i \xrightarrow[E_i]{\sigma_i} q_i'$ is a transition of $H_i$ for some $\sigma_i$ $(i = 1, 2, ..., N)$ and

$$\sigma = \bigcup_{H_i \in \mathbb{V}} (\sigma_i \backslash ( \bigcup_{H_j \in \mathrm{Pre}(\mathbb{A}_h, H_i)} \eta_j(q_j))),$$

where $\mathrm{Pre}(\mathbb{A}_h, H_i) = \{H_j \in \mathbb{V} \,|\, (H_j, H_i) \in \mathbb{E}\}$.

We stress that the above definition captures interactions between discrete variables and not between continuous variables. This choice is motivated by the application domain we are interested in, where interaction among agents can be naturally represented by an exchange of discrete signals (and not of continuous signals) in the hybrid systems that model the agents.
While the notion of AFSM considers interaction of FSMs, the notion of AHS considers interaction of hybrid systems. A mixed model in which FSMs interact with hybrid systems can be easily obtained by adding to FSMs fictitious dynamics as $\dot{x} = 0$ and by thus modeling the interaction among FSMs and hybrid systems by the notion of AHSs. This mixed interaction among FSMs and hybrid systems arises for example in the mathematical model of air traffic management systems.

## 3.4 Compositional Bisimulation

### 3.4.1 Compositional Bisimulation of AFSMs

In this section we introduce a novel class of equivalence for AFSMs, termed compositional bisimulation, that is based on the communication network governing the interaction mechanism among the FSMs. The compositional bisimulation equivalence between AFSMs implies bisimulation equivalence between the corresponding expanded FSMs. This result is important because it implies that all properties preserved by bisimulation equivalence, e.g. linear temporal logic properties [5], are also preserved by compositional bisimulation. Therefore, it can be of help in the formal verification and control design of complex systems modeled by AFSMs that admit compositional bisimulation.

A naïve approach to check bisimulation equivalence of two AFSMs $\mathbb{A}^1$ and $\mathbb{A}^2$ consists in first expanding them to FSMs $\mathbb{M}(\mathbb{A}^1)$ and $\mathbb{M}(\mathbb{A}^2)$ and then apply standard bisimulation algorithms (see e.g. [44, 46, 47]). The main practical limitation of this approach resides in the well–known state explosion problem, see e.g. [38, 37]. This is the key reason for us to propose an alternative approach to check bisimulation equivalence of AFSMs which is centered on the notion of *compositional bisimulation* that is introduced hereafter.

**Definition 3.4.1.** *Given a pair of Arenas* $\mathbb{A}^j = (\mathbb{V}^j, \mathbb{E}^j)$ *of FSMs* $M_1^j$, $M_2^j$, ..., $M_{N_j}^j$ *(*$j = 1, 2$*), a set* $\mathbb{R} \subseteq \mathbb{V}^1 \times \mathbb{V}^2$*, is a compositional bisimulation relation between* $\mathbb{A}^1$ *and* $\mathbb{A}^2$ *if for any* $(M_i^1, M_j^2) \in \mathbb{R}$ *the following conditions are satisfied:*

- $M_i^1 \cong M_j^2$*;*
- *existence of* $(M_i^1, M_{i'}^1) \in \mathbb{E}^1$ *implies existence of* $(M_j^2, M_{j'}^2) \in \mathbb{E}^2$ *such that* $(M_{i'}^1, M_{j'}^2) \in \mathbb{R}$*;*
- *existence of* $(M_j^1, M_{j'}^2) \in \mathbb{E}^2$ *implies existence of* $(M_i^1, M_{i'}^1) \in \mathbb{E}^1$ *such that* $(M_{i'}^1, M_{j'}^2) \in \mathbb{R}$*.*

*The AFSMs* $\mathbb{A}^1$ *and* $\mathbb{A}^2$ *are compositionally bisimilar, denoted* $\mathbb{A}^1 \cong_c \mathbb{A}^2$*, if there exists a total compositional bisimulation relation between* $\mathbb{A}^1$ *and* $\mathbb{A}^2$*.*

Basic facts on bisimulation equivalence extends to compositional bisimulation as follows:

**Proposition 3.4.2.** *The notion of compositional bisimulation is an equivalence relation on the class of AFSMs.*

**Definition 3.4.3.** *The maximal compositional bisimulation relation between AFSMs* $\mathbb{A}^1$ *and* $\mathbb{A}^2$ *is a compositional bisimulation relation* $\mathbb{R}^*(\mathbb{A}^1, \mathbb{A}^2)$ *such that* $\mathbb{R} \subseteq \mathbb{R}^*(\mathbb{A}^1, \mathbb{A}^2)$ *for any compositional bisimulation relation* $\mathbb{R}$*.*

**Proposition 3.4.4.** *The maximal compositional bisimulation exists and is unique.*

**Proposition 3.4.5.** *The set* $\mathbb{R}^*(\mathbb{A}, \mathbb{A})$ *is an equivalence relation on the collection of FSMs in* $\mathbb{A}$*.*

**Proposition 3.4.6.** *The quotient of* $\mathbb{A}$ *induced by* $\mathbb{R}^*(\mathbb{A}, \mathbb{A})$ *is the minimal (in terms of the number of the FSMs involved) compositionally bisimilar AFSM of* $\mathbb{A}$*.*

**Proposition 3.4.7.** *The minimal AFSM of an AFSM* $\mathbb{A}$*, denoted* $\mathbf{A}_{\min}(\mathbb{A})$*, exists and is unique, up to isomorphisms.*

Checking compositional bisimulation equivalence of AFSMs is equivalent to checking bisimulation equivalence of appropriate FSMs, as discussed hereafter. Consider a pair of AFSMs $\mathbb{A}^j = (\mathbb{V}^j, \mathbb{E}^j)$ $(j = 1, 2)$. Since bisimulation is an equivalence relation on the set $\mathbb{V}_1 \cup \mathbb{V}_2$ of FSMs, it induces a partition of $\mathbb{V}_1 \cup \mathbb{V}_2$ in $K$ equivalence classes $C_1, C_2, ..., C_K$ where $M_i, M_j \in C_k$ if and only if $M_i \cong M_j$. Note that $\{C_k\}_{k \in K}$ is a finite set. Define the tuple:

$$M_{\mathbb{A}^j} = (Q_{\mathbb{A}^j}, \Sigma_{\mathbb{A}^j}, \Psi_{\mathbb{A}^j}, \eta_{\mathbb{A}^j}, E_{\mathbb{A}^j}), \tag{3.4}$$

where

- $Q_{\mathbb{A}^j} = \mathbb{V}^j$;
- $\Sigma_{\mathbb{A}^j} = \varnothing$;
- $\Psi_{\mathbb{A}^j} = \{C_k\}_{k \in K}$;
- $\eta_{\mathbb{A}^j} : Q_{\mathbb{A}^j} \to 2^{\Psi_{\mathbb{A}^j}}$ is defined by $\eta_{\mathbb{A}^j}(M_i^j) = \{C_k\}$ if $M_i^j \in C_k$,
- $E_{\mathbb{A}^j} \subseteq Q_{\mathbb{A}^j} \times \varnothing \times Q_{\mathbb{A}^j}$ is such that $M_i^j \xrightarrow[E_{\mathbb{A}^j}]{\varnothing} M_{i'}^j$ when $(M_i^j, M_{i'}^j) \in \mathbb{E}^j$. By definition of $H_{\mathbb{A}^j}$, $H_{\mathbb{A}^j}(M_i^j) = H_{\mathbb{A}^{j'}}(M_{i'}^{j'})$ if and only if $M_i^j \cong M_{i'}^{j'}$.

The syntax of the tuple in (3.4) is the same as the one of FSMs from which, the following result holds.

**Proposition 3.4.8.** $\mathbb{A}^1 \cong_c \mathbb{A}^2$ *if and only if* $M_{\mathbb{A}^1} \cong M_{\mathbb{A}^2}$.

*Proof.* By Definitions 3.1.2 and 3.4.1, it is readily seen that $\mathbb{A}^1 \cong_c \mathbb{A}^2$ if and only if the set $\mathbb{R}^*(\mathbb{A}^1, \mathbb{A}^2)$ is a total bisimulation relation between $M_{\mathbb{A}^1}$ and $M_{\mathbb{A}^2}$. □

We are now ready to present the main result of this section, that shows that the notion of compositional bisimulation of AFSMs is consistent with the notion of bisimulation of the corresponding expanded FSMs.

**Theorem 3.4.9.** *[48] If* $\mathbb{A}^1 \cong_c \mathbb{A}^2$ *then* $\mathbb{M}(\mathbb{A}^1) \cong \mathbb{M}(\mathbb{A}^2)$.

Theorem 3.4.9 can be used to reduce the size of AFSMs through compositional bisimulation, as follows.

**Corollary 3.4.10.** *[48]* $\mathbf{M}_{\min}(\mathbb{M}(\mathbb{A})) \cong^{\text{iso}} \mathbf{M}_{\min}(\mathbb{M}(\mathbf{A}_{\min}(\mathbb{A})))$.

The above result suggests a method to use compositional bisimulation for complexity reduction of AFSMs, as summarized in the following algorithm:

- Compute the relation $\mathbb{R}^*(\mathbb{A}, \mathbb{A})$.
- Compute the quotient $\mathbf{A}_{\min}(\mathbb{A})$.
- Expand the AFSM $\mathbf{A}_{\min}(\mathbb{A})$ to the FSM $\mathbb{M}(\mathbf{A}_{\min}(\mathbb{A}))$.
- Compute the relation $R^*(\mathbb{M}(\mathbf{A}_{\min}(\mathbb{A})), \mathbb{M}(\mathbf{A}_{\min}(\mathbb{A})))$.
- Compute the quotient $\mathbf{M}_{\min}(\mathbb{M}(\mathbf{A}_{\min}(\mathbb{A})))$.

### 3.4.2 Compositional Bisimulation of AHSs

In this section we generalize the theory of compositional bisimulation from AFSMs to AHSs. A naïve approach to check bisimulation equivalence of two AHSs $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$ consists in first expanding them to the corresponding hybrid systems $\mathbb{H}(\mathbb{A}_h^1)$ and $\mathbb{H}(\mathbb{A}_h^2)$ and then apply bisimulation algorithms developed in [64]. The main practical limitation of this approach resides in the well–known state explosion problem, see e.g. [38, 37]. This is the key reason for us to propose an alternative approach to check bisimulation equivalence of AHSs which is centered on the notion of *compositional bisimulation* that is introduced hereafter.

**Definition 3.4.11.** *Given a pair of Arenas $\mathbb{A}_h^j = (\mathbb{V}^j, \mathbb{E}^j)$ of hybrid systems $\mathcal{H}_1^j$, $\mathcal{H}_2^j$, ..., $\mathcal{H}_{N_j}^j$ $(j = 1, 2)$, a set $\mathbb{R} \subseteq \mathbb{V}^1 \times \mathbb{V}^2$, is a compositional bisimulation relation between $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$ if for any $(\mathcal{H}_i^1, \mathcal{H}_j^2) \in \mathbb{R}$ the following conditions are satisfied:*

- *$\mathcal{H}_i^1 \cong \mathcal{H}_j^2$ in the sense of definition 3.2.1;*
- *existence of $(\mathcal{H}_i^1, \mathcal{H}_{i'}^1) \in \mathbb{E}^1$ implies existence of $(\mathcal{H}_j^2, \mathcal{H}_{j'}^2) \in \mathbb{E}^2$ such that $(\mathcal{H}_{i'}^1, \mathcal{H}_{j'}^2) \in \mathbb{R}$;*
- *existence of $(\mathcal{H}_j^2, \mathcal{H}_{j'}^2) \in \mathbb{E}^2$ implies existence of $(\mathcal{H}_i^1, \mathcal{H}_{i'}^1) \in \mathbb{E}^1$ such that $(\mathcal{H}_{i'}^1, \mathcal{H}_{j'}^2) \in \mathbb{R}$.*

*The AHSs $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$ are compositionally bisimilar, denoted $\mathbb{A}_h^1 \cong_c \mathbb{A}_h^2$, if there exists a total compositional bisimulation relation between $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$.*

Basic facts about compositional bisimulation on AFSMs naturally extend to compositional bisimulation of AHSs, as shown in the following.

**Definition 3.4.12.** *The notion of compositional bisimulation is an equivalence relation on the class of AHSs.*

**Definition 3.4.13.** *The maximal compositional bisimulation relation between AHSs $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$ is a compositional bisimulation relation $\mathbb{R}^*(\mathbb{A}_h^1, \mathbb{A}_h^2)$ such that $\mathbb{R} \subseteq \mathbb{R}^*(\mathbb{A}_h^1, \mathbb{A}_h^2)$ for any compositional bisimulation relation $\mathbb{R}$.*

**Proposition 3.4.14.** *The maximal compositional bisimulation exists and is unique.*

**Proposition 3.4.15.** *The set $\mathbb{R}^*(\mathbb{A}_h, \mathbb{A}_h)$ is an equivalence relation on the collection of hybrid systems in $\mathbb{A}_h$.*

**Proposition 3.4.16.** *The quotient of $\mathbb{A}_h$ induced by $\mathbb{R}^*(\mathbb{A}_h, \mathbb{A}_h)$ is the minimal (in terms of the number of the hybrid systems involved) compositionally bisimilar AHS of $\mathbb{A}_h$.*

**Proposition 3.4.17.** *The minimal AHS of an AHS $\mathbb{A}_h$, denoted $\mathbf{A}_{\min}(\mathbb{A}_h)$, exists and is unique, up to isomorphisms.*

We are now ready to present the main result of this section, that shows that the notion of compositional bisimulation of AHSs is consistent with the notion of bisimulation of the corresponding expanded hybrid systems.

**Theorem 3.4.18.** *If* $\mathbb{A}_h^1 \cong_c \mathbb{A}_h^2$ *then* $\mathbb{H}(\mathbb{A}^1) \cong \mathbb{H}(\mathbb{A}^2)$.

The proof of this result can be given along the lines of the proof of Theorem 3.4.9 and is therefore omitted.

## 3.5 Complexity Analysis

In this section we compare computational complexity in checking compositional bisimulation equivalence between AFSMs and bisimulation equivalence between the corresponding expanded FSMs. Similar results can be derived for the notion of compositional bisimulation of AHSs.
Consider a pair of AFSMs $\mathbb{A}^i = (\mathbb{V}^i, \mathbb{E}^i)$ composed of $N_i$ FSMs and set $\mathbb{M}(\mathbb{A}^i) = (Q^i, q_{0,i}, \Sigma^i, \Psi^i, \eta^i, E^i)$ $(i = 1, 2)$. As common practice in the analysis of non–flat systems, e.g. [38, 37], in the sequel we evaluate how computational complexity scales with the number $N_i$ of FSMs in AFSMs $\mathbb{A}^i$. We start by evaluating the computational complexity in checking bisimulation equivalence of the flattened systems $\mathbb{M}(\mathbb{A}^1)$ and $\mathbb{M}(\mathbb{A}^2)$. As a direct application of Propositions 3.1.14 and 3.1.15, one gets the following results.

**Corollary 3.5.1.** *Space complexity in checking* $\mathbb{M}(\mathbb{A}^1) \cong \mathbb{M}(\mathbb{A}^2)$ *is*

$$O(2^{N_1} + 2^{N_2}).$$

**Corollary 3.5.2.** *Time complexity in checking* $\mathbb{M}(\mathbb{A}^1) \cong \mathbb{M}(\mathbb{A}^2)$ *is*

$$O((2^{N_1} + 2^{N_2}) \ln(2^{N_1} + 2^{N_2})).$$

The above result quantifies the aforementioned state explosion problem [38, 37] in the class of AFSMs. We now discuss computational complexity in checking compositional bisimulation.

**Proposition 3.5.3.** *Space complexity in checking* $\mathbb{A}^1 \cong_c \mathbb{A}^2$ *is*

$$O(N_1^2 + N_2^2).$$

*Proof.* Direct consequence of Propositions 3.1.14 and 3.4.8. □

**Proposition 3.5.4.** *Time complexity in checking* $\mathbb{A}^1 \cong_c \mathbb{A}^2$ *is*

$$O((N_1^2 + N_2^2) \ln(N_1 + N_2)).$$

*Proof.* By Proposition 3.4.8, checking $\mathbb{A}^1 \cong_c \mathbb{A}^2$ reduces to:

(1) construct FSMs $M_{\mathbb{A}^1}$, and $M_{\mathbb{A}^2}$; and

(2) check if $M_{\mathbb{A}^1} \cong M_{\mathbb{A}^2}$.

Regarding (1), time complexity effort reduces to the one of defining functions $H_{\mathbb{A}^1}$ and $H_{\mathbb{A}^2}$ which amounts to $O((N_1 + N_2)^2)$. Regarding (2), by Proposition 3.1.15, time complexity in checking $M_{\mathbb{A}^1} \cong M_{\mathbb{A}^2}$ is given by $O((N_1^2 + N_2^2)\ln(N_1 + N_2))$. Since the last term is dominant over $O((N_1 + N_2)^2)$, the result follows. □

Chapter 4

# Analysis of Critical Observability

In this chapter we first review the notion of critical observability as introduced in [7]. We then propose some theoretical results towards the computation complexity reduction in checking critical observability of large scale complex systems. The chapter is organized as follows. In Section 4.1 we introduce the notion of critical observability and the definition of observer. In Section 4.2 and 4.3 we extend the concept of compositional bisimulation to AFSMs and AHSs, by defining the notion of critical compositional bisimulation.

## 4.1 Critical Observer

Given a hybrid system $\mathcal{H}$, let $\mathcal{R} \subset Q$ be the set of *critical states* of $\mathcal{H}$, i.e. the set of discrete states associated to unsafe or unallowed behaviors of $\mathcal{H}$. We say that $\mathcal{H}$ is $\mathcal{R}$–critically observable if it is possible to construct a system that is able to detect whether the current discrete state of $\mathcal{H}$ belongs to $\mathcal{R}$ or not on the basis of the observations. Formally:

**Definition 4.1.1.** *Given a hybrid system $\mathcal{H}$, an observer of the critical set $\mathcal{R}$ is a system $\mathcal{O}_{\mathcal{R}}$ whose input is the discrete output of $\mathcal{H}$ and whose output $\hat{y}(t)$ is such that*[1]*:*

$$\forall\, k \geq 0, \forall\, t \in [t_k, t'_k), \qquad \hat{y}(t) = \begin{cases} 1 & \text{if } \; q(I_k) \in \mathcal{R} \\ 0 & \text{if } \; q(I_k) \in Q \setminus \mathcal{R}. \end{cases}$$

*System $\mathcal{H}$ is said to be $\mathcal{R}$–critically observable if an observer $\mathcal{O}_{\mathcal{R}}$ exists. Moreover, if $\mathcal{O}_{\mathcal{R}}$ exists it is said to be a $\mathcal{R}$–critical observer for $\mathcal{H}$.*

We report hereafter the definition of observer in case of FSMs.

**Definition 4.1.2.** *Given an FSM $M$ and a critical relation $\mathfrak{R}_c$, an $\mathfrak{R}_c$–critical observer is an observer $\mathcal{O}_{\mathfrak{R}_c}$ whose input is the output of $M$ and whose output*

[1] The entities $t_k$, $t'_k$, $I_k$ and $q(.)$ have been introduced in Section 2.3.

*function* $\hat{H}$ *is such that*

$$\hat{\eta}(q) = \begin{cases} 1, & \text{if } q \in \mathfrak{R}_c, \\ 0, & \text{if } q \notin \mathfrak{R}_c. \end{cases}$$

*FSM* $M$ *is said to be* $\mathfrak{R}_c$*–critically observable if an* $\mathfrak{R}_c$*–critical observer* $\mathcal{O}_{\mathfrak{R}_c}$ *exists.*

Given a hybrid system $\mathcal{H}$, we refer to a critical observer of $\mathcal{H}$ as the FSM:

$$\mathcal{O} = (\hat{Q}, \hat{Q}_0, \hat{\Sigma}, \hat{\Psi}, \hat{\eta}, \hat{E}),$$

where:

- $\hat{Q} \subseteq 2^Q$ is a set of states.
- $\hat{Q}_0 \subseteq \hat{Q}$ is the set of initial states.
- $\hat{\Sigma}$ is the set of inputs which coincides with the set of discrete outputs $\Psi$ of $\mathcal{H}$.
- $\hat{\Psi}$ is the set of outputs which coincides with $\hat{Q}$.
- $\hat{\eta} : \hat{Q} \to \hat{\Psi}$ is the output function which coincides with identity function.
- $\hat{E} \subseteq Q \times 2^{\hat{\Sigma}} \times Q$ is the transition relation. We denote a transition $(q, \sigma, q') \in \hat{E}$ of $\mathcal{O}$ by $q \xrightarrow[\hat{E}]{\sigma} q'$.

The construction of such observers is rather standard in the literature on discrete event systems. From the above definition it is readily seen that the space complexity of $\mathcal{O}$ is $O(|2^Q|)$, i.e. the size of the set of states $\hat{Q}$ of $\mathcal{O}$ grows exponentially with the size of the set of discrete states $Q$ of the hybrid system $\mathcal{H}$.

If a hybrid system $\mathcal{H}$ is not critically observable, information coming from the continuous dynamics can be used to generate additional discrete signals that provide extra information to discriminate the discrete states, as proposed in [17]. When using information coming from the continuous dynamics, some time is required in the generation of additional discrete signals. This implies a non–instantaneous detection of critical states. However in many cases a bounded delay in the detection of such critical states is acceptable[2] and this motivates the definition of [11] reported hereafter:

**Definition 4.1.3.** *Given a hybrid system* $\mathcal{H}$*, an observer with delay* $\delta > 0$ *of the critical set* $\mathcal{R}$ *is a system* $\mathcal{O}^{\delta}_{\mathcal{R}}$ *whose input is the discrete output of* $\mathcal{H}$ *and whose output* $\hat{y}(t)$ *is such that:*

$$\forall\, k \geq 0, \forall\, t \in [t_k + \delta, t'_k), \qquad \hat{y}(t) = \begin{cases} 1 & \text{if } \; q(I_k) \in \mathcal{R} \\ 0 & \text{if } \; q(I_k) \notin \mathcal{R}. \end{cases}$$

[2]Bounded delay in the detection of critical states is acceptable for example for hybrid systems with positive dwell time, i.e. hybrid systems in which the dwelling time in each discrete state is greater than a positive real, called dwell time.

*System $\mathcal{H}$ is said to be $\mathcal{R}$–critically observable with delay $\delta$ if an observer $\mathcal{O}_{\mathcal{R}}^{\delta}$ exists. Moreover if $\mathcal{O}_{\mathcal{R}}^{\delta}$ exists it is said to be a $\mathcal{R}$–critical observer with delay $\delta$ for $\mathcal{H}$.*

An algorithm to check critical observability with delay can be found in [11].

## 4.2 Critical Compositional bisimulation of AFSMs

The notion of critical observability of FSMs introduced in the previous section naturally extends to AFSMs by appropriately defining a critical relation that extends the set of critical states to a collection of FSMs in an AFSM. Given an AFSM $\mathbb{A} = (\mathbb{V}, \mathbb{E})$, consider the following tuple:

$$\mathfrak{R}_c = (\mathfrak{R}_c^1, \mathfrak{R}_c^2, ..., \mathfrak{R}_c^N), \tag{4.1}$$

where:

- $\mathfrak{R}_c^1$ is the collection of sets $\mathfrak{R}_{i_1} \subseteq Q_{i_1}$ $(i_1 = 1, 2, ..., N)$ of critical states for $M_{i_1}$.
- $\mathfrak{R}_c^2$ is the collection of sets $\mathfrak{R}_{i_1,i_2} \subseteq Q_{i_1} \times Q_{i_2}$ $(i_1, i_2 = 1, 2, ..., N)$ of critical states arising from the interaction of $M_{i_1}$ and $M_{i_2}$.
  ...
- $\mathfrak{R}_c^N$ is the collection of sets $\mathfrak{R}_{i_1,i_2,...,i_N} \subseteq Q_{i_1} \times Q_{i_2} \times \ldots \times Q_{i_N}$ $(i_1, i_2, ..., i_N = 1, 2, ..., N)$ of critical states arising from the interaction of $M_{i_j}$ with $j = 1, 2, ..., N$.

The above critical relation involving states of FSMs naturally induces suitable critical relations on the corresponding FSMs, as follows:

$$\mathcal{R}_c = (\mathcal{R}_c^2, ..., \mathcal{R}_c^N), \tag{4.2}$$

where:

- $\mathcal{R}_c^2 \subseteq \mathbb{V} \times \mathbb{V}$ is such that $(M_{i_1}, M_{i_2}) \in \mathcal{R}_c^2$ if $\mathfrak{R}_{i_1,i_2} \neq \varnothing$;
  ...
- $\mathcal{R}_c^N \subseteq \mathbb{V} \times ... \times \mathbb{V}$ is such that $(M_{i_1}, M_{i_2}, ..., M_{i_N}) \in \mathcal{R}_c^N$ if $\mathfrak{R}_{i_1,i_2,...,i_N} \neq \varnothing$.

Compositional bisimulation provides an efficient method to the complexity reduction of large scale complex AFSMs. However, the notion of compositional bisimulation in Definition 3.4.1 does not capture safety criticality interaction among the FSMs in the AFSM. For this reason we now adapt the notion of compositional bisimulation so that it also respects the aforementioned safety criticality interaction.

**Definition 4.2.1.** *Consider a pair of arenas $\mathbb{A}^j = (\mathbb{V}^j, \mathbb{E}^j)$ of FSMs $M_1^j$, $M_2^j$, ..., $M_{N^j}^j$ $(j = 1, 2)$ and a pair of critical relations*

$$\mathcal{R}_{cj} = (\mathcal{R}_{cj}^2, ..., \mathcal{R}_{cj}^{N^1}), \;\; j = 1, 2,$$

*each one being related to $\mathbb{A}^j$ and of the form in (4.6). A relation*

$$\mathbb{R} \subseteq \mathbb{V}^1 \times \mathbb{V}^2$$

*is a $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–critical compositional simulation relation of $\mathbb{A}^1$ by $\mathbb{A}^2$ if for any $(M_{i_1}^1, M_{j_1}^2) \in \mathbb{R}$ the following conditions are satisfied:*

*(i)* $M^1_{i_1} \cong M^2_{j_1}$*;*

*(ii) existence of* $(M^1_{i_1}, M^{1,+}_{i_1}) \in \mathbb{E}^1$ *implies existence of* $(M^2_{j_1}, M^{2,+}_{j_1}) \in \mathbb{E}^2$ *so that* $(M^{1,+}_{i_1}, M^{2,+}_{j_1}) \in \mathbb{R}$*;*

*(iii) The following* $N$ *conditions hold:*

*(iii,1) for any* $M^1_{i_2} \in \mathbb{V}^1$ *such that* $(M^1_{i_1}, M^1_{i_2}) \in \mathcal{R}^2_{c1}$*, there exists* $M^2_{j_2} \in \mathbb{V}^2$ *such that:*

- $(M^2_{j_1}, M^2_{j_2}) \in \mathcal{R}^2_{c2}$*;*
- $(M^1_{i_2}, M^2_{j_2}) \in \mathbb{R}$*;*

*(iii,2) for any* $M^1_{i_2}, M^1_{i_3} \in \mathbb{V}^1$ *such that* $(M^1_{i_1}, M^1_{i_2}, M^1_{i_3}) \in \mathcal{R}^3_{c1}$*, there exist* $M^2_{j_2}, M^2_{j_3} \in \mathbb{V}^2$ *such that:*

- $(M^2_{j_1}, M^2_{j_2}, M^2_{j_3}) \in \mathcal{R}^3_{c2}$*;*
- $(M^1_{i_2}, M^2_{j_2}) \in \mathbb{R}$*,* $(M^1_{i_3}, M^2_{j_3}) \in \mathbb{R}$*;*

...

*(iii,N) for any* $M^1_{i_2}, M^1_{i_3}, \ldots, M^1_{i_{N^1}} \in \mathbb{V}^1$ *such that* $(M^1_{i_1}, M^1_{i_2}, \ldots, M^1_{i_{N^1}}) \in \mathcal{R}^{N^1}_{c1}$*, there exist* $M^2_{j_2}, M^2_{j_3}, \ldots, M^2_{j_{N^2}} \in \mathbb{V}^2$ *such that:*

- $(M^2_{j_1}, M^2_{j_2}, \ldots, M^2_{j_N}) \in \mathcal{R}^{N^2}_{c2}$*;*
- $(M^1_{i_k}, M^2_{j_k}) \in \mathbb{R}$ *for any* $k = 1, ..., \max\{N^1, N^2\}$*.*

*Relation* $\mathbb{R}$ *is a* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation between* $A^1$ *and* $A^2$ *if:*

*(i)* $\mathbb{R}$ *is a* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional simulation relation from* $\mathbb{A}_1$ *to* $\mathbb{A}_2$*;*

*(ii)* $\mathbb{R}^{-1}$ *is a* $(\mathcal{R}_{c2}, \mathcal{R}_{c1})$*–compositional simulation relation from* $\mathbb{A}_2$ *to* $\mathbb{A}_1$*.*

*AFSMs* $\mathbb{A}^1$ *and* $\mathbb{A}^2$ *are* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*-compositionally bisimilar, denoted*

$$\mathbb{A}^1 \cong_{(\mathcal{R}_{c1}, \mathcal{R}_{c2})} \mathbb{A}^2,$$

*if there exists a* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation between* $\mathbb{A}^1$ *and* $\mathbb{A}^2$*.*

Basic facts about compositional bisimulation naturally extends to $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositional bisimulation, as follows.

**Proposition 4.2.2.** *The notion of* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation is an equivalence relation on the class of AFSMs.*

**Definition 4.2.3.** *The maximal* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation between AFSMs* $\mathbb{A}^1$ *and* $\mathbb{A}^2$ *is an* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation* $\mathbb{R}^*(\mathbb{A}^1, \mathbb{A}^2)$ *such that* $\mathbb{R} \subseteq \mathbb{R}^*(\mathbb{A}^1, \mathbb{A}^2)$ *for any* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation* $\mathbb{R}$*.*

**Proposition 4.2.4.** *The maximal $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositional bisimulation exists and is unique.*

**Proposition 4.2.5.** *The set $\mathbb{R}^*(\mathbb{A}, \mathbb{A})$ is an equivalence relation on the collection of FSMs in $\mathbb{A}$.*

**Proposition 4.2.6.** *The quotient of $\mathbb{A}$ induced by $\mathbb{R}^*(\mathbb{A}, \mathbb{A})$ is the minimal (in terms of the number of the FSMs involved) $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositionally bisimilar AFSM of $\mathbb{A}$.*

**Proposition 4.2.7.** *The minimal AFSM of an AFSM $\mathbb{A}$, denoted $\mathbf{A}_{\min}(\mathbb{A})$, exists and is unique, up to isomorphisms.*

By generalizing the results reported in [48], standard bisimulation algorithms can be appropriately adapted to compute $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositional bisimulations between AFSMs.

**Bisimulation Algorithm**: Given a pair of arenas $\mathbb{A}^j = (\mathbb{V}^j, \mathbb{E}^j)$ of FSMs $M_1^j$, $M_2^j$, ..., $M_N^j$ $(j = 1, 2)$, we define a sequence of relations

$$\mathbb{R}_0, \mathbb{R}_1, \ldots, \mathbb{R}_N \subseteq \mathbb{V} \times \mathbb{V},$$

as follows:

1. $\mathbb{R}_0$ is composed by all pairs $(M_{i_1}^1, M_{j_1}^2)$ so that $M_{i_1}^1 \cong M_{j_1}^2$.
2. $(M_{i_1}^1, M_{j_1}^2) \in \mathbb{R}_{N+1}$ if and only if
   - $(M_{i_1}^1, M_{j_1}^2) \in \mathbb{R}_N$;
   - existence of $(M_{i_1}^1, M_{i_1}^{1,+}) \in \mathbb{E}^1$ implies existence of $(M_{j_1}^2, M_{j_1}^{2,+}) \in \mathbb{E}^2$ such that $(M_{i_1}^{1,+}, M_{j_1}^{2,+}) \in \mathbb{R}_N$;
   - existence of $(M_{j_1}^2, M_{j_1}^{2,+}) \in \mathbb{E}^2$ implies existence of $(M_{i_1}^1, M_{i_1}^{1,+}) \in \mathbb{E}^1$ such that $(M_{i_1}^{1,+}, M_{j_1}^{2,+}) \in \mathbb{R}_N$;
   - for any $M_{i_2}^1 \in \mathbb{V}^1$ such that $(M_{i_1}^1, M_{i_2}^1) \in \mathbb{R}_c^2$, exists $M_{j_2}^2 \in \mathbb{V}^2$ such that $(M_{j_1}^2, M_{j_2}^2) \in \mathbb{R}_c^2$ and $(M_{i_2}^1, M_{2_2}^2) \in \mathcal{R}_c^2$;

     $\vdots$

     for any $M_{i_2}^1, M_{i_3}^1, \ldots, M_{i_N}^1 \in \mathbb{V}^1$ such that $(M_{i_1}^1, M_{i_2}^1, \ldots, M_{i_N}^1) \in \mathbb{R}_c^N$, exists $M_{j_2}^2, M_{j_3}^2, \ldots, M_{j_N}^2 \in \mathbb{V}^2$ such that $(M_{j_1}^2, M_{j_2}^2, \ldots, M_{j_N}^2) \in \mathbb{R}_c^N$ and $(M_{i_2}^1, M_{i_3}^1, \ldots, M_{i_N}^1, M_{j_2}^2, M_{j_3}^2, \ldots, M_{j_N}^2) \in \mathcal{R}_c^N$;
   - for any $M_{j_2}^2 \in \mathbb{V}^2$ such that $(M_{j_1}^2, M_{j_2}^2) \in \mathbb{R}_c^2$, exists $M_{i_2}^1 \in \mathbb{V}^1$ such that $(M_{i_1}^1, M_{i_2}^1) \in \mathbb{R}_c^2$ and $(M_{j_2}^2, M_{j_3}^2, \ldots, M_{j_N}^2, M_{i_2}^1, M_{i_3}^1, \ldots, M_{i_N}^1) \in \mathcal{R}_c^N$;;

     $\vdots$

     for any $M_{j_2}^2, M_{j_3}^2, \ldots, M_{j_N}^2 \in \mathbb{V}^2$ such that $(M_{j_1}^2, M_{j_2}^2, \ldots, M_{j_N}^2) \in \mathbb{R}_c^N$, exist $M_{i_2}^1, M_{i_3}^1, \ldots, M_{i_N}^1 \in \mathbb{V}^2$ such that $(M_{i_1}^1, M_{i_2}^1, \ldots, M_{i_N}^1) \in \mathbb{R}_c^N$.

We now have all the ingredients to present the following result.

**Theorem 4.2.8.** *Consider two AFSMs* $\mathbb{A}^1$ *and* $\mathbb{A}^2$ *and a pair of critical relations* $\mathfrak{R}_{c1}$ *and* $\mathfrak{R}_{c2}$ *for* $\mathbb{M}(\mathbb{A}^1)$ *and* $\mathbb{M}(\mathbb{A}^2)$*. Denote by* $\mathcal{R}_{c1}$ *and* $\mathcal{R}_{c2}$ *the sets of critical relations naturally induced by* $\mathfrak{R}_{c1}$ *and* $\mathfrak{R}_{c2}$ *on* $\mathbb{A}^1$ *and* $\mathbb{A}^2$*. If* $\mathbb{A}^1 \cong_{(\mathcal{R}_{c1},\mathcal{R}_{c2})} \mathbb{A}^2$ *then* $\mathbb{M}(\mathbb{A}^1)$ *is* $\mathfrak{R}_{c1}$*-critically observable if and only if* $\mathbb{M}(\mathbb{A}^2)$ *is* $\mathfrak{R}_{c2}$*-critically observable.*

*Proof.* (Sketch.) Let be $\mathbb{M}(\mathbb{A}^i) = (Q^i, Q^i_0, \Sigma^i, \Psi^i, \eta^i, E^i)$ $(i = 1, 2)$. Let $\mathcal{O}_{\mathfrak{R}_{c1}} = (\hat{Q}_1, \hat{Q}_{0,1}, \hat{\Sigma}_1, \hat{\Psi}_1, \hat{\eta}_1, \hat{E}_1)$ be a $\mathfrak{R}_{c1}$–critical observer for $\mathbb{M}(\mathbb{A}^1)$ and define:

$$\mathcal{O}_{\mathfrak{R}_{c2}} = (\hat{Q}_2, \hat{Q}_{0,2}, \hat{\Sigma}_2, \hat{\Psi}_2, \hat{\eta}_2, \hat{E}_2),$$

where $\hat{Q}_2 \subseteq 2^{Q^2}$ is a set of states, $\hat{Q}^2_0 \subseteq 2^{Q^2}$ is the set of initial states, $\hat{\Sigma}_2$ is the set of inputs which coincides with the set of outputs $\Psi^2$ of $\mathbb{M}(\mathbb{A}^2)$, $\hat{\Psi}_2$ is the set of outputs which coincides with $\hat{Q}_2$. The output function $\hat{\eta}_2 : \hat{Q}_2 \to \hat{\Psi}_2$ is defined as follows. Consider any state $\mathbf{Q}^2$ of $\mathcal{O}_{\mathfrak{R}_{c2}}$. Pick any $q^2 = (q^2_1, q^2_2, ..., q^2_{N^2}) \in \mathbf{Q}^2$ and consider $q^1 = (q^1_1, q^1_2, ..., q^1_{N^1}) \in \mathbf{Q}^1$ such that $(q^1_i, q^2_j) \in R^*(M^1_i, M^2_j)$ and $(M^1_i, M^2_j) \in \mathbb{R}$ for any $i \in [1; N^1]$ and $j \in [1; N^2]$. Then define $\hat{H}_2(\mathbf{Q}^2) = \hat{\eta}_1(\mathbf{Q}^1)$. Consider

$$\mathbf{Q}^1 \xrightarrow[\hat{E}^1]{\sigma} \mathbf{Q}^{1,+}. \tag{4.3}$$

For any state $q^{1,+} = (q^{1,+}_1, q^{1,+}_2, ..., q^{1,+}_{N^1}) \in \mathbf{Q}^{1,+}$ there exists a state $q^{2,+} = (q^{2,+}_2, q^{2,+}_2, ..., q^{2,+}_{N^2}) \in \mathbf{Q}^{2,+}$ such that $(q^{1,+}_i, q^{2,+}_j) \in R^*(M^1_i, M^2_j)$ and $(M^1_i, M^2_j) \in \mathbb{R}$ for any $i \in [1; N^1]$ and $j \in [1; N^2]$. Define then the transition

$$\mathbf{Q}^2 \xrightarrow[\hat{E}^2]{\sigma} \mathbf{Q}^{2,+}, \tag{4.4}$$

in $\mathcal{O}_{\mathfrak{R}_{c2}}$. The result then holds as a direct consequence of Definition 4.3.1. □

## 4.3 Critical Compositional bisimulation of AHSs

The notion of critical observability of hybrid system introduced in the previous section naturally extends to AHSs by appropriately defining a critical relation that extends the set of critical states to a collection of hybrid systems in an AHS. Given an AHS $\mathbb{A}_h = (\mathbb{V}, \mathbb{E})$, consider the following tuple:

$$\mathfrak{R}_c = (\mathfrak{R}_c^1, \mathfrak{R}_c^2, ..., \mathfrak{R}_c^N), \tag{4.5}$$

where:

- $\mathfrak{R}_c^1$ is the collection of sets $\mathfrak{R}_{i_1} \subseteq Q_{i_1}$ $(i_1 = 1, 2, ..., N)$ of critical states for $\mathcal{H}_{i_1}$.
- $\mathfrak{R}_c^2$ is the collection of sets $\mathfrak{R}_{i_1,i_2} \subseteq Q_{i_1} \times Q_{i_2}$ $(i_1, i_2 = 1, 2, ..., N)$ of critical states arising from the interaction of $\mathcal{H}_{i_1}$ and $\mathcal{H}_{i_2}$.
  ...
- $\mathfrak{R}_c^N$ is the collection of sets $\mathfrak{R}_{i_1,i_2,...,i_N} \subseteq Q_{i_1} \times Q_{i_2} \times \ldots \times Q_{i_N}$ $(i_1, i_2, ..., i_N = 1, 2, ..., N)$ of critical states arising from the interaction of $\mathcal{H}_{i_j}$ with $j = 1, 2, ..., N$.

The above critical relation involving discrete states of hybris systems naturally induces suitable critical relations on the corresponding hybrid systems, as follows:

$$\mathcal{R}_c = (\mathcal{R}_c^2, ..., \mathcal{R}_c^N), \tag{4.6}$$

where:

- $\mathcal{R}_c^2 \subseteq \mathbb{V} \times \mathbb{V}$ is such that $(\mathcal{H}_{i_1}, \mathcal{H}_{i_2}) \in \mathcal{R}_c^2$ if $\mathfrak{R}_{i_1,i_2} \neq \varnothing$;
  ...
- $\mathcal{R}_c^N \subseteq \mathbb{V} \times ... \times \mathbb{V}$ is such that $(\mathcal{H}_{i_1}, \mathcal{H}_{i_2}, ..., \mathcal{H}_{i_N}) \in \mathcal{R}_c^N$ if $\mathfrak{R}_{i_1,i_2,...,i_N} \neq \varnothing$.

Compositional bisimulation provides an efficient method to the complexity reduction of large scale complex AHSs. However, the notion of compositional bisimulation in Definition 3.4.1 does not capture safety criticality interaction among the hybrid systems in the AHS. For this reason we now adapt the notion of compositional bisimulation so that it also respects the aforementioned safety criticality interaction.

**Definition 4.3.1.** *Consider a pair of arenas $\mathbb{A}_h^j = (\mathbb{V}^j, \mathbb{E}^j)$ of hybrid systems $\mathcal{H}_1^j$, $\mathcal{H}_2^j$, ..., $\mathcal{H}_{N^j}^j$ $(j = 1, 2)$ and a pair of critical relations*

$$\mathcal{R}_{cj} = (\mathcal{R}_{cj}^2, ..., \mathcal{R}_{cj}^{N^1}), \;\; j = 1, 2,$$

*each one being related to $\mathbb{A}_h^j$ and of the form in (4.6). A relation*

$$\mathbb{R} \subseteq \mathbb{V}^1 \times \mathbb{V}^2$$

*is a $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–critical compositional simulation relation of $\mathbb{A}_h^1$ by $\mathbb{A}_h^2$ if for any $(\mathcal{H}_{i_1}^1, \mathcal{H}_{j_1}^2) \in \mathbb{R}$ the following conditions are satisfied:*

*(i)* $\mathcal{H}^1_{i_1} \cong \mathcal{H}^2_{j_1}$ *in the sense of Definition 3.2.1;*

*(ii) existence of* $(\mathcal{H}^1_{i_1}, \mathcal{H}^{1,+}_{i_1}) \in \mathbb{E}^1$ *implies existence of* $(\mathcal{H}^2_{j_1}, \mathcal{H}^{2,+}_{j_1}) \in \mathbb{E}^2$ *so that* $(\mathcal{H}^{1,+}_{i_1}, \mathcal{H}^{2,+}_{j_1}) \in \mathbb{R}$*;*

*(iii) The following N conditions hold:*

*(iii,1) for any* $\mathcal{H}^1_{i_2} \in \mathbb{V}^1$ *such that* $(\mathcal{H}^1_{i_1}, \mathcal{H}^1_{i_2}) \in \mathcal{R}^2_{c1}$*, there exists* $\mathcal{H}^2_{j_2} \in \mathbb{V}^2$ *such that:*

- $(\mathcal{H}^2_{j_1}, \mathcal{H}^2_{j_2}) \in \mathcal{R}^2_{c2}$*;*
- $(\mathcal{H}^1_{i_2}, \mathcal{H}^2_{j_2}) \in \mathbb{R}$*;*

*(iii,2) for any* $\mathcal{H}^1_{i_2}, \mathcal{H}^1_{i_3} \in \mathbb{V}^1$ *such that* $(\mathcal{H}^1_{i_1}, \mathcal{H}^1_{i_2}, \mathcal{H}^1_{i_3}) \in \mathcal{R}^3_{c1}$*, there exist* $\mathcal{H}^2_{j_2}, \mathcal{H}^2_{j_3} \in \mathbb{V}^2$ *such that:*

- $(\mathcal{H}^2_{j_1}, \mathcal{H}^2_{j_2}, \mathcal{H}^2_{j_3}) \in \mathcal{R}^3_{c2}$*;*
- $(\mathcal{H}^1_{i_2}, \mathcal{H}^2_{j_2}) \in \mathbb{R}$*,* $(\mathcal{H}^1_{i_3}, \mathcal{H}^2_{j_3}) \in \mathbb{R}$*;*

...

*(iii,N) for any* $\mathcal{H}^1_{i_2}, \mathcal{H}^1_{i_3}, \ldots, \mathcal{H}^1_{i_{\mathcal{H}^1}} \in \mathbb{V}^1$ *such that* $(\mathcal{H}^1_{i_1}, \mathcal{H}^1_{i_2}, \ldots, \mathcal{H}^1_{i_{N^1}}) \in \mathcal{R}^{N^1}_{c1}$*, there exist* $\mathcal{H}^2_{j_2}, \mathcal{H}^2_{j_3}, \ldots, \mathcal{H}^2_{j_{N^2}} \in \mathbb{V}^2$ *such that:*

- $(\mathcal{H}^2_{j_1}, \mathcal{H}^2_{j_2}, \ldots, \mathcal{H}^2_{j_N}) \in \mathcal{R}^{N^2}_{c2}$*;*
- $(\mathcal{H}^1_{i_k}, \mathcal{H}^2_{j_k}) \in \mathbb{R}$ *for any* $k = 1, ..., \max\{N^1, N^2\}$*.*

*Relation* $\mathbb{R}$ *is a* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation between* $\mathbb{A}^1_h$ *and* $\mathbb{A}^2_h$ *if:*

*(i)* $\mathbb{R}$ *is a* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional simulation relation from* $\mathbb{A}^1_h$ *to* $\mathbb{A}^2_h$*;*

*(ii)* $\mathbb{R}^{-1}$ *is a* $(\mathcal{R}_{c2}, \mathcal{R}_{c1})$*–compositional simulation relation from* $\mathbb{A}^2_h$ *to* $\mathbb{A}^1_h$*.*

*AHSs* $\mathbb{A}^1_h$ *and* $\mathbb{A}^2_h$ *are* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*-compositionally bisimilar, denoted*

$$\mathbb{A}^1_h \cong_{(\mathcal{R}_{c1}, \mathcal{R}_{c2})} \mathbb{A}^2_h,$$

*if there exists a* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation between* $\mathbb{A}^1_h$ *and* $\mathbb{A}^2_h$*.*

Basic facts about compositional bisimulation naturally extends to $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositional bisimulation, as follows.

**Proposition 4.3.2.** *The notion of* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation is an equivalence relation on the class of AHSs.*

**Definition 4.3.3.** *The maximal* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation between AHSs* $\mathbb{A}^1_h$ *and* $\mathbb{A}^2_h$ *is an* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation* $\mathbb{R}^*(\mathbb{A}^1_h, \mathbb{A}^2_h)$ *such that* $\mathbb{R} \subseteq \mathbb{R}^*(\mathbb{A}^1_h, \mathbb{A}^2_h)$ *for any* $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$*–compositional bisimulation relation* $\mathbb{R}$*.*

**Proposition 4.3.4.** *The maximal $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositional bisimulation exists and is unique.*

**Proposition 4.3.5.** *The set $\mathbb{R}^*(\mathbb{A}_h, \mathbb{A}_h)$ is an equivalence relation on the collection of hybrid systems in $\mathbb{A}_h$.*

**Proposition 4.3.6.** *The quotient of $\mathbb{A}_h$ induced by $\mathbb{R}^*(\mathbb{A}_h, \mathbb{A}_h)$ is the minimal (in terms of the number of the hybrid systems involved) $(\mathcal{R}_{c1}, \mathcal{R}_{c2})$–compositionally bisimilar AHS of $\mathbb{A}_h$.*

**Proposition 4.3.7.** *The minimal AHS of an AHS $\mathbb{A}_h$, denoted $\mathbf{A}_{\min}(\mathbb{A}_h)$, exists and is unique, up to isomorphisms.*

We now have all the ingredients to present the following result.

**Theorem 4.3.8.** *Consider two AHS $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$ and a pair of critical relations $\mathfrak{R}_{c1}$ and $\mathfrak{R}_{c2}$ for $\mathbb{H}(\mathbb{A}_h^1)$ and $\mathbb{H}(\mathbb{A}_h^2)$. Denote by $\mathcal{R}_{c1}$ and $\mathcal{R}_{c2}$ the sets of critical relations naturally induced by $\mathfrak{R}_{c1}$ and $\mathfrak{R}_{c2}$ on $\mathbb{A}_h^1$ and $\mathbb{A}_h^2$. If $\mathbb{A}_h^1 \cong_{(\mathcal{R}_{c1}, \mathcal{R}_{c2})} \mathbb{A}_h^2$ then $\mathbb{H}(\mathbb{A}_h^1)$ is $\mathfrak{R}_{c1}$-critically observable if and only if $\mathbb{H}(\mathbb{A}_h^2)$ is $\mathfrak{R}_{c2}$-critically observable.*

In the sequel we present some more results to check critical observability that are complementary to the ones presented above.

**Proposition 4.3.9.** *Consider a hybrid system $\mathcal{H}$ and a set of critical states $\mathcal{R}$. Suppose that $\mathcal{R} = \mathcal{R}^1 \cup \mathcal{R}^2$. Then $\mathcal{H}$ is $\mathcal{R}$–critically observable if $\mathcal{H}$ is $\mathcal{R}^1$–critically observable and $\mathcal{R}^2$–critically observable.*

*Proof.* If $\mathcal{H}$ is $\mathcal{R}^1$–critically observable and $\mathcal{R}^2$–critically observable there exist a pair of observers $\mathcal{O}_1$ and $\mathcal{O}_2$ which are able to detect whether the discrete state of $\mathcal{H}$ is in $\mathcal{R}^1$ and $\mathcal{R}^2$ or not. Define the hybrid observer $\mathcal{O}$ as the shuffle product $\mathcal{O}_1 \times \mathcal{O}_2$ of the observers $\mathcal{O}_1$ and $\mathcal{O}_2$ and with output $\hat{y}$ defined by $\hat{y}(t) = [\hat{y}_1(t) \ \vee \ \hat{y}_2(t)]$. Suppose that $q(I_k) \in \mathcal{R}$ at time $t$. Then either $q(I_k) \in \mathcal{R}^1$ or $q(I_k) \in \mathcal{R}^2$, which corresponds to $\hat{y}_1(t) = 1$ or $\hat{y}_2(t) = 1$, from which $\hat{y}(t) = 1$. Suppose now that $q(I_k) \in Q \backslash \mathcal{R}$ at time $t$. Then $q(I_k) \in Q \backslash \mathcal{R}^1$ and $q(I_k) \in Q \backslash \mathcal{R}^2$, which corresponds to $\hat{y}_1(t) = 0$ and $\hat{y}_2(t) = 0$, from which $\hat{y}(t) = 0$. Thus $\mathcal{O}$ is a $\mathcal{R}$–critical observer for $\mathcal{H}$ and hence $\mathcal{H}$ is $\mathcal{R}$–critically observable. □

**Proposition 4.3.10.** *Consider an AHS $\mathbb{A}_h = (\mathbb{V}, \mathbb{E})$, with $\mathbb{V} = \{\mathcal{H}_1, \mathcal{H}_2\}$, $\mathbb{E} = \mathbb{E}|_{\{(\mathcal{H}_1, \mathcal{H}_2), (\mathcal{H}_2, \mathcal{H}_1)\}}$ , and a set of critical states $\mathcal{R}^1 \times \mathcal{R}^2 \subseteq Q_1 \times Q_2$. Then $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}^1 \times \mathcal{R}^2$–critically observable if $\mathcal{H}_1$ is $\mathcal{R}^1$–critically observable and $\mathcal{H}_2$ is $\mathcal{R}^2$–critically observable.*

*Proof.* For $i = 1, 2$ let $\mathcal{O}_i$ be a $\mathcal{R}_i$–critical observer for $\mathcal{H}_i$ and denote by $\hat{y}_i$ the output of $\mathcal{O}_i$. Define the hybrid observer $\mathcal{O}$ as the shuffle product $\mathcal{O}_1 \times \mathcal{O}_2$ of the observers $\mathcal{O}_1$ and $\mathcal{O}_2$ and with output $\hat{y}$ defined by $\hat{y}(t) = [\hat{y}_1(t) \ \wedge \ \hat{y}_2(t)]$. We now show that $\mathcal{O}$ is a $\mathcal{R}^1 \times \mathcal{R}^2$–critical observer for $\mathbb{H}(\mathbb{A}_h)$. Suppose that

$q(I_k) = (q_1(I_k), q_2(I_k)) \in \mathcal{R}^1 \times \mathcal{R}^2$ at time $t$. Then $q_i(I_k) \in \mathcal{R}^i$ which implies $\hat{y}_i(t) = 1$; thus $\hat{y}(t) = 1$. Suppose now that $q(I_k) \notin \mathcal{R}^1 \times \mathcal{R}^2$ at time $t$. By using similar arguments it is easy to show that $\hat{y}(t) = 0$. Thus $\mathcal{O}$ is a $\mathcal{R}^1 \times \mathcal{R}^2$–critical observer for $\mathbb{H}(\mathbb{A}_h)$ and hence $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}^1 \times \mathcal{R}^2$–critically observable. □

We can now give the main result of this section.

**Theorem 4.3.11.** *Consider an AHS $\mathbb{A}_h = (\mathbb{V}, \mathbb{E})$ of $N$ hybrid systems $\mathcal{H}_1$, $\mathcal{H}_2$, ..., $\mathcal{H}_N$. Let $\mathcal{R} \subseteq Q_1 \times Q_2 \times ... \times Q_N$ be a critical relation for $\mathbb{H}(\mathbb{A}_h)$. Then $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}$–critically observable if and only if the following conditions are satisfied:*

- *$\mathcal{H}_{i_1}$ is $\mathcal{R}_{i_1}$–critically observable for any $i_1 = 1, 2, ..., N$;*
- *Given $\mathbb{A}_h^{'} = (\mathbb{V}^{'}, \mathbb{E}^{'})$, with $\mathbb{V}^{'} = \{\mathcal{H}_{i_1}, \mathcal{H}_{i_2}\}$ and $\mathbb{E}^{'} = \mathbb{E}|_{\{(\mathcal{H}_{i_1}, \mathcal{H}_{i_2}), (\mathcal{H}_{i_2}, \mathcal{H}_{i_1})\}}$. $\mathbb{H}(\mathbb{A}_h^{'})$ is $\mathcal{R}_{i_1,i_2}$–critically observable for any $i_1, i_2 = 1, 2, \ldots, N$;*
  $\vdots$
- *Given $\mathbb{A}_h^{''} = (\mathbb{V}^{''}, \mathbb{E}^{''})$, with $\mathbb{V}^{''} = \{\mathcal{H}_{i_1}, \mathcal{H}_{i_2}, \ldots, \mathcal{H}_{i_N}\}$ and $\mathbb{E}^{''} = \mathbb{E}|_{\{(\mathcal{H}_{i_1}, \mathcal{H}_{i_2}, ..., \mathcal{H}_{i_N}), (\mathcal{H}_{i_2}, \mathcal{H}_{i_1}, ...}$ $\mathbb{H}(\mathbb{A}_h^{''})$ is $\mathcal{R}_{i_1,i_2,...,i_N}$–critically observable for any $i_1, i_2, \ldots, i_N = 1, 2, \ldots, N$.*

*Proof.* (Necessity) Obvious. (Sufficiency) By Proposition 4.3.9, $\mathcal{H}$ is $\mathcal{R}$–critically observable if $\mathcal{H}$ is critical observable w.r.t. the critical relations in Definition 4.3.1. Now consider the set $\mathcal{R}'_{i_1}$. It is readily seen that such set can be rewritten as

$$\mathcal{R}'_{i_1} = Q_1 \times Q_2 \times ... \times Q_{i_1-1} \times \mathcal{R}_{i_1} \times Q_{i_1+1} \times ... \times Q_N.$$

Given the AHSs $\mathbb{A}_h^{'} = (\mathbb{V}^{'}, \mathbb{E}^{'})$ with $\mathbb{V}^{'} = \{\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_{i_1-1}\}$ and $\mathbb{A}_h^{''} = (\mathbb{V}^{''}, \mathbb{E}^{''})$ with $\mathbb{V} = \{\mathcal{H}_{i_1+1}, \mathcal{H}_{i_1+2}, ..., \mathcal{H}_N\}$, by Proposition 4.3.10 AHS $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}'_{i_1}$–critically observable if:

- $\mathbb{H}(\mathbb{A}_h^{'})$ is $Q_1 \times Q_2 \times ... \times Q_{i_1-1}$–critically observable;
- $\mathcal{H}_{i_1}$ is $\mathcal{R}_{i_1}$–critically observable;
- $\mathbb{H}(\mathbb{A}_h^{''})$ is $Q_{i_1+1} \times Q_{i_1+2} \times ... \times Q_N$–critically observable.

From the definition of critical observability it is clear that the first and third conditions are always satisfied. Indeed, the discrete state of hybrid system $\mathbb{H}(\mathbb{A}_h^{'})$ always evolves in its state space $Q_1 \times Q_2 \times \ldots \times Q_{i_1-1}$: hence the first condition is satisfied. The same ratio applies to the third condition. From this discussion we get that $\mathcal{H}$ is $\mathcal{R}'_{i_1}$–critically observable if $\mathcal{H}_{i_1}$ is $\mathcal{R}_{i_1}$–critically observable. By applying the same reasoning to other critical relations appearing in Definition 4.3.1, the result follows. □

Chapter 5

# Air Traffic Management Procedures

In this chapter we provide a mathematical modeling and analysis of some ATM scenarios studied within the ATM community: ASEP-ITP, Lateral Crossing, $A^3$ ConOps, TMA T1 Scenario, ACRA. The analysis of critical observability is reported in detail for the ASEP-ITP procedure and TMA T1 scenario. In Section 5.1 we report the description of the TMA T1 scenario, the mathematical model and the analysis of critical observability. We have the same for ASEP-ITP in Section 5.2. In Section 5.3 we discuss about the ASAS lateral crossing: description of the procedure, mathematical model and a few consideration on the analysis of critical observability. The Section 5.4 is organized as the previous one and refers to $A^3$ ConOps.

## 5.1 Terminal Manoeuvring Area T1 Scenario

The aim of the SESAR 2020 Programme is to improve efficiency in future European Air Traffic Management. A central notion in the SESAR 2020 Concept of Operation is the one of Reference Business Trajectory (RBT). The RBTs allows pilots to follow their assigned trajectories with a sensible reduction of the controller interventions. In a busy Terminal Manoeuvring Area (TMA), the minimum spacing between (the centerlines of) the route structures and RBTs in the TMA must be reduced to 5 NM. T1 refers to the reduction of separation minima in the TMA; this reduction allows significant capacity increase for complex or constrained TMAs. In this TMA, routes are typically Standard Instrument Departure (SID) routes, Standard Terminal Arrival Routes (STAR) and also cruise routes at a lower flight level. The TMA T1 concept aims at reducing the nominal lateral distances between these routes to 5 NM.

### 5.1.1 Description of TMA T1 Scenario

#### Operational context

In this scenario it is assumed that Medium Jet aircraft will represent about 72% of the total fleet mixes, 20% of them are ATM-1 capable and 80% are ATM-3 capable.

These ATM capability levels have been introduced within the SESAR concept with the aim of describing the on-going deployment of progressively more advanced ATM systems for aircraft, ground systems and airports. We recall from [50] that SESAR defined six levels that will be progressively deployed. Capability levels are associated with stakeholder systems, procedures, human resources, etc. Upgrading a stakeholder to a higher capability level consists in deploying new enablers. Service levels are associated with operational services that are offered by a service provider and consumed by a service user. Upgrading a service to a higher service level consists in deploying operational improvement steps. Delivering a service at a given service level X requires that both the service provider and the service user have at least evolved to capability level X.

Backward compatibility is also required: each system with a given capability level should also be able to provide and receive services at a lower service level. This ensures interoperability between systems of different capability levels. For example:

- Aircraft at capability Level 3 is flying into a capability Level 2 airport. Aircraft will use service Level 2. The performance benefits are those associated with service Level 2.
- Aircraft at capability Level 1 is flying into a capability Level 2 airport. Aircraft will use service Level 1 that is included in service Level 2. The performance benefits are those associated with service Level 1.
- For a service to be used it is necessary that both the service provider and the service user possess the required capability, but not necessarily all the capabilities of a particular level.

Aircraft equipped with capabilities required by the key SESAR target date of 2020 are referred as ATM Capability Level 3 (ATM-3).
In the TMA T1 operation one high density TMA is considered which accommodate several airports. In this TMA it is possible to define various closely spaced SIDs and STARs and several cruise routes at a lower flight level. An example of route situation is depicted in Figure 2.1 (Taken from [45]), where the STARs are denoted in red, the SIDs in green, and the cruise routes in blue. Moreover, white rectangles identify some types of encounters that are briefly described hereafter:

- **Encounter type 1**. Independent departures from different airports following parallel SID routes, which are spaced laterally by 5 NM, see Figure 5.2 (Taken from [45]).
- **Encounter type 2**. One arrival aircraft to and one departure aircraft from different airports following, respectively, a STAR and a SID, which are spaced laterally by 5 NM, see Figure 5.3 (Taken from [45]).
- **Encounter type 3**. A flight in cruise (en-route) flying at a lower flight level and a flight on a STAR on paths which are spaced laterally by 5 NM, see Figure 5.4 (Taken from [45]).

For more details we refer to [51]. The minimum distance between the SIDs/STARs and en route lanes is $S_{min} = 5NM$(see Figures 2.2, 2.3 and 2.4). This is a reduction with respect to current route separation minima, which would allow significant capacity increase for complex or constrained TMAs. The TMA-T1 concept does not pose new constraints on the radar separation minimum, hence this is assumed to be equal to the current minimum $S_{radar} = 3NM$.

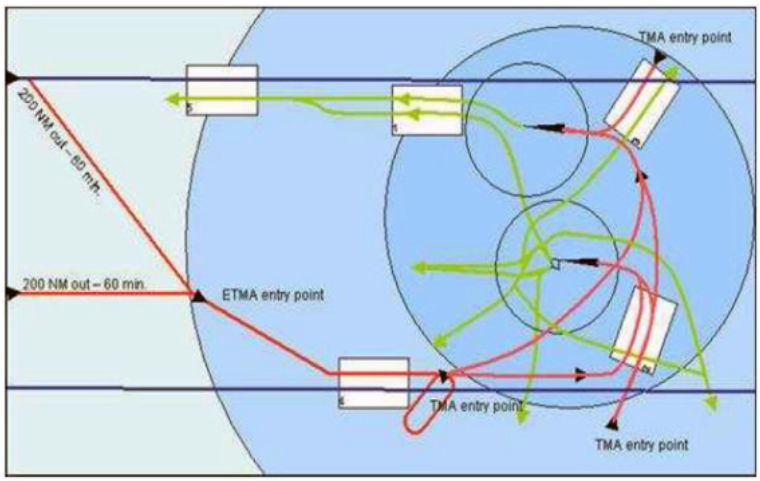



**Figure 5.1:** Example of extended TMA (ETMA) and TMA with entry points and several SIDs (green), STARs (red) and cruise routes (blue).

### Human roles

In the following we describe the human roles and responsibilities that are of key importance for the TMA T1 operation:

- **Planning Controllers**. They are responsible for establishing the non conflicting reference trajectory for each arrival and departure, by using the TMA airspace to the required time horizon and associated stability, consistency and accuracy requirements.
- **Executive Controllers**. Their responsibility concerns the execution of each reference trajectory in the TMA within the context of the applicable airspace rules, with the aim of assuring separation, avoiding collisions, and sequencing aircraft where required.
- **The Flight Crew (Pilots)**. They are responsible for the safe operation of the flight, in the respect of ATC (Air Traffic Control) instructions.

Moreover, additional human roles are

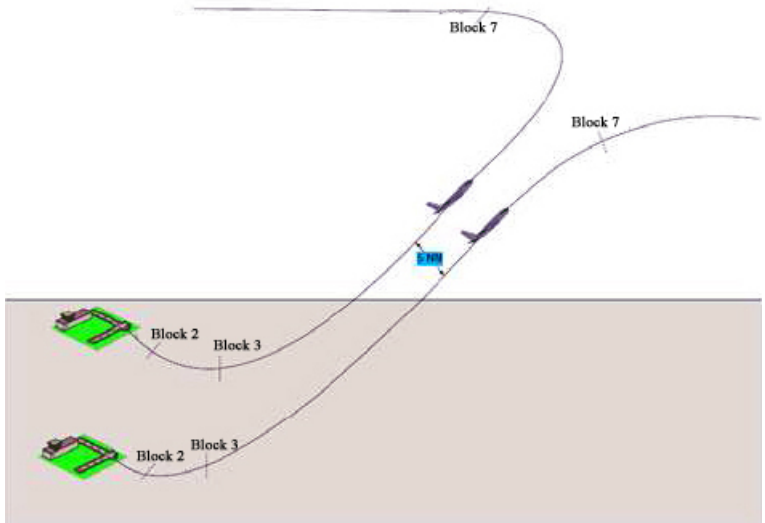



**Figure 5.2:** Encounter type 1. The distance between the route lanes is 5 NM and the radar separation minimum is 3 NM.

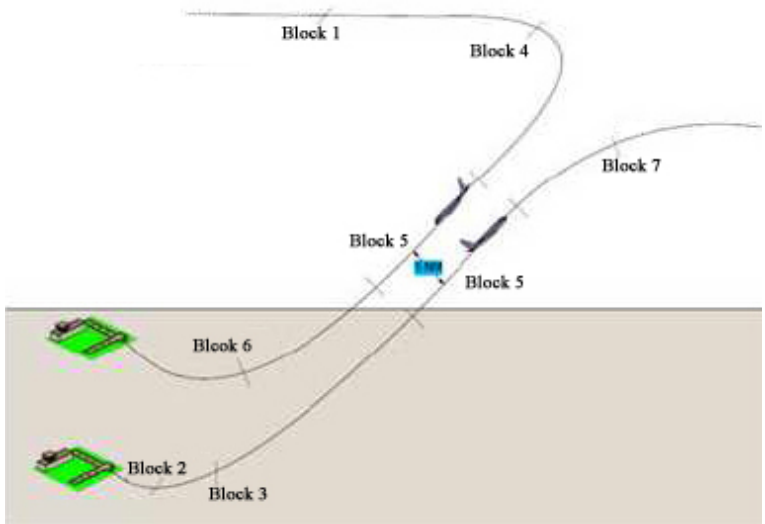



**Figure 5.3:** Encounter type 2. The distance between the route lanes is 5 NM and the radar separation minimum is 3 NM.

- **Air Flow and Capacity Management staff**. General responsibilities concern the regulation of traffic where demand at times exceeds the declared

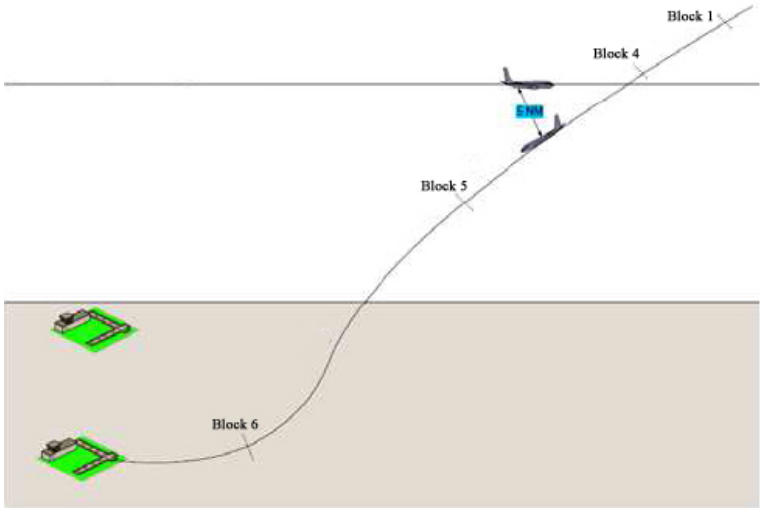



**Figure 5.4:** Encounter type 3. The distance between the route lanes is 5 NM and the radar separation minimum is 3 NM.

capacity of the air traffic control services.

- **High level, managerial, organizational and decision layers of ANSPs (Air Navigation Service Providers)**: ATM Staffing and Recruitment, ATM Training and Development and ATM Competence and Licensing.

### Procedures

The procedures followed by the Executive Controller and the Pilot–Flying in this scenario are similar to those of todays. Executive Controllers keep their situation awareness through their traffic situation displays, by communicating with other controllers and pilots. Pilots keep their situation awareness through their navigation displays, by communicating with ATC, and by communications with each other (i.e. between pilot-flying and pilot-not-flying), and the windows. The main difference between the TMA T1 concept and the todays one is mostly concerned with the trajectory planning. In the following, we focus on this difference. Before an aircraft departure, the Airline downlinks a Systemized Business Trajectory (SBT). Before take-off, Airline and ATM agree on the SBT, which is further registered as a Reference Business Trajectory (RBT), and finally distributed through the System Wide Information Management (SWIM). After take-off, the RBT is updated and down linked by the pilot to ATC, through the Air Ground Data Link (AGDL). When the RBT is accepted by pilot and ATC, it is registered as an Update in the SWIM. From then on, it is referred to as a registered RBT. Every stakeholder can access the RBTs by the SWIM. In some situations it is possible to update the RBT,

for example if a change or delay occur during the flight. In the end the updates must converge.

Before an aircraft to pass top of descent, the Planning Controller uplinks, through the Controller Pilot Data Link Communication (CPDLC), the request to that aircraft to downlink its current RBT. Then the pilot-not-flying downlinks, through AGDL, the RBT that is in the Flight Management System (FMS). Next the RBT, received by the ATC system, is compared with the RBTs of the other aircraft. Finally the Planning Controller sends the RBTs to the Arrival Manager (AMAN) controller, who sets up a sequencing, leading to an arrival time over waypoint. The AMAN controller sends back this time to the Planning Controller. If there is a mismatch in this comparison, the Planning Controller sends a constraint to the aircraft through the CPDLC. This constraint will be a Target Time of Arrival (TTA), that is a planned time over an exit waypoint. The Planning Controller en-route needs to be sure that the Executive Controller gets a plan that is in line with TTA. The pilot-not-flying works together with his FMS for evaluating the constraint on feasibility. If the constraint is feasible, the pilot-not-flying uses CPDLC to accept the constraint, and updates the trajectory planning as current. If not, the pilot uses R/T to report back to the Executive Controller. Then, the Executive Controller reports to the Planning Controller, that is requested to provide a new constraint. If the aircraft has successfully modified its RBT so that it respects the ATC requested constraints, then this RBT is registered via data link (AGDL) into the SWIM. Before starting the Continuous Descent Approach (CDA), the Planning Controller requests through an acknowledging trigger from Monitoring Aids (MONA), the aircraft to submit a calculated 4D trajectory to the ground. MONA uses as inputs, short term intent information (ADS-B) and radar information. The request is sent to the FMS through data link. The aircraft FMS calculates the 4D trajectory information from the RBT information.

MONA regularly monitors if the 4D trajectory is realized or not by the aircraft. If mismatches are found, the Executive Controller can be informed. Medium Term Conflict Detection (MTCD) regularly verifies possible conflicting situations arising in the 4D trajectories planned by the various aircraft. In principle, MTCD is a Planning Controller tool. For conflict-free planning (MTCD), the RBT is not used. The controller will always want to do "what-if modeling" in order to see the effect of alternative planned trajectories. The "what-if modeling" is a part of the MTCD system. If MTCD or MONA reveals a conflict or a deviation, then one of the involved aircraft is requested to modify its RBT. A planning conflict may become a tactical conflict, especially when the aircraft reaches a sector. Note that conflicts (MTCD) have more priority than delays (MONA).If the conflict becomes tactical, the Executive Controller can take any action that is considered necessary to maintain separation. A planning action given by a Planning Controller is requested to take care of recovery. This recovery includes creation of a valid RBT and if possible to readmit the aircraft in the landing sequence. The role of Short Term Conflict Alert (STCA) in this advanced concept is as today, i.e., a last resort conflict detection tool for the Executive Controller.

The Executive Controller communicates directly with the pilot. Most communication is done through R/T. If communication is not urgent, CPDLC can be used. Information from ATC system to FMS which do not involve humans are restricted to e.g. weather information. Conversely, information from FMS to ATC system which do not involve humans may include more, e.g. flight status information, updated estimates over waypoints, local weather, etc. The Executive Controller can make use of the RBT, although he has no responsibility for it. The Planning Controller makes use of the RBT or other plans, and can change the RBT (with pilot in the loop). The Planning Controller requests changes to the RBT and the pilot agrees. In dense airspace, each RBT will include either a SID or a STAR. Aircraft on a SID and aircraft on a STAR spaced at 5 NM are generally controlled by two different controllers (if the SID and the STAR considered are to/from two different airports), i.e., the Arrival controller and the Departure controller will be two different persons.

**Technical systems**

In this section we describe the main technical systems available to the ATC and to the pilots.

ATC Technical systems:

- **Traffic situation display**. This is a graphical representation of the controller area of interest, e.g., airport, sector, and etc. It displays the position of the aircraft within the area of interest/responsibility.
- **Radio communication**. They include Air-Ground, Ground-Ground voice communications via radio, telephone, etc.
- **System wide information manager (SWIM)**. SWIM provides data sharing of ATM system information which include equipment operational status, Notice to Airmen (NOTAM), status of navigational aids and airspace restrictions.
- **Controller pilot data link communications (CPDLC)**. This is a data link application allowing direct exchange of text-based messages between a controller and a pilot.
- **Arrival manager (AMAN)**. This is a system aid for ATC at airports, that calculates a planned Arrival flow with the goal of maintaining an optimal throughput at the runway, of reducing arrival queuing and of distributing the information to various stakeholders.
- **Monitoring aids (MONA)**. This is a flight plan conformance monitoring (FPCM) device that supports controllers in monitoring flights under their control. This facilitates detection of actual or predicted deviations from the system trajectory, i.e. route deviation and level bust. In addition, MONA may provide controllers with reminders regarding planned actions.

- **Medium term conflict detection (MTCD)**. This is a planning tool, assisting the controller through the detection of potential conflicts, which include conflicts between aircraft or a penetration of an aircraft into a defined airspace.
- **Short term conflict alert (STCA)**. It checks possible conflicting trajectories in a time horizon of about less than 2 minutes and alerts the controller before lost of separation.
- **Automatic dependent surveillance broadcast (ADS-B) ground station**. It is a cooperative surveillance technique for air traffic control and related applications. An ADS-B-equipped aircraft periodically broadcasts its position and other relevant information to ground stations and other aircraft equipped with ADS-B.

Aircraft technical systems

- **Advanced flight management system (FMS)**. FMS derives and holds the RBT and allows the pilot to modify the RBT as required in flight. An FMS uses various sensors to determine the aircraft position. Given the position and the flight plan, the FMS calculates a trajectory based on the RBT and uses this information to guide the aircraft through autopilot/auto throttle. The FMS is normally controlled through a display and a keyboard, and sends the RBT for display on the aircraft Multi Function Display (MFD).
- **Precision Area Navigation (P-RNAV).** This system allows a required track-keeping accuracy of 1 NM for at least 95% of the flight time. This level of navigation accuracy can be achieved with the support of DME/DME, GPS or VOR/DME. It can also be maintained for short periods using Inertial Reference System (IRS); the length of time within which a particular IRS can be used to maintain P-RNAV accuracy without external update, is determined at the time of certification.
- **Radio communication**. They include Air-Ground, voice communications via radio, telephone, etc.
- **Controller pilot data link communications (CPDLC)**. This is a data link application that allows direct exchange of text-based messages between a controller and a pilot.
- **Mode-A/C/S transponder**. It provides a data downlink of flight parameters via Secondary Surveillance Radars which allows radar processing systems and hence, controllers, to monitor various data on a flight.
- **Automatic dependent surveillance broadcast (ADS-B) aircraft transmitter**. This is a cooperative surveillance technique for air traffic control and related applications. An ADS-B-equipped aircraft periodically broadcasts its

position and other relevant information to potential ground stations and to other aircraft with ADS-B-in equipment.

- **Airborne Collision Avoidance System (ACAS)**. This aircraft collision avoidance system is designed to reduce the incidence of mid-air collisions between aircraft. It monitors the airspace around an aircraft for other aircraft equipped with a Mode-S transponder and warns pilots in case of the presence of other transponder-equipped aircraft that may cause potential mid–air collisions.

### Non-nominal Conditions

When designing a new ATM advanced concept, it is fundamental to proceed with a safety analysis from the early beginning of the design process. Deliverable [51] provides an exhaustive hazard and qualitative scenario analysis for the TMA T1 operation, which include several non-nominal encounter scenarios. Building upon the analysis of [51], the work [45] provides complementary insight into some specific encounter types TMA T1. The TMA T1 operation allows a better sharing of information between agents and hence improves shared situation awareness between agents and the ability to anticipate the need and preferences of other agents. The way RBT intent information is used in this procedure is novel with respect to what currently done in practice. In this section we describe the non–nominal conditions defined in [45].
For the TMA T1 concept, six types of agents have been identified:

- Weather, particularly wind affecting the flight path of the aircraft.
- ATC system, including all technical equipment of the air traffic controller, such as traffic situation display, MONA, systems providing surveillance data, and short term conflict alert (STCA), but also including communication means such as CPDLC and R/T communication. Other elements considered in this agent are the Airspace structure and the RBTs available to ATC of all aircraft.
- Executive Controller, including his/her actions, performance and situation awareness.
- Guidance Navigation and Control system for each aircraft. This includes the auto flight system, the navigation system (FMS), control panel, aircraft performance data and a deviation alerting system. It also includes communication systems such as CPDLC and R/T communication, and the RBT as available in the FMS.
- Aircraft crew, including their actions, performance and situation awareness.
- Aircraft, including aircraft type and the actually flown flight path (referred to as aircraft evolution). In addition, this agent includes an entrance list of

aircraft to the routes (i.e., each aircraft has a particular time and location of entry into the sector).

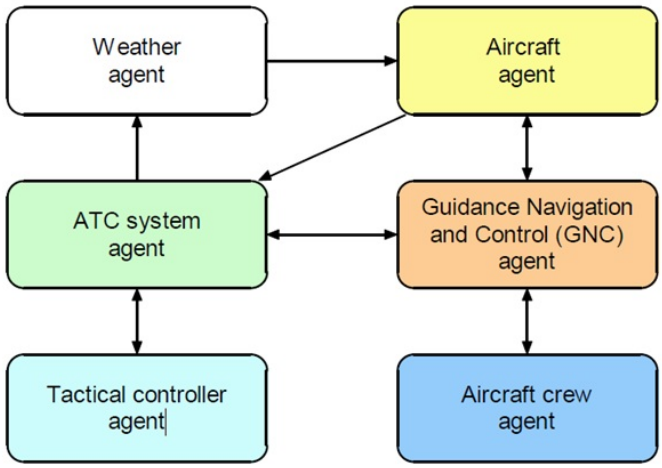



**Figure 5.5:** Agents for TMA T1

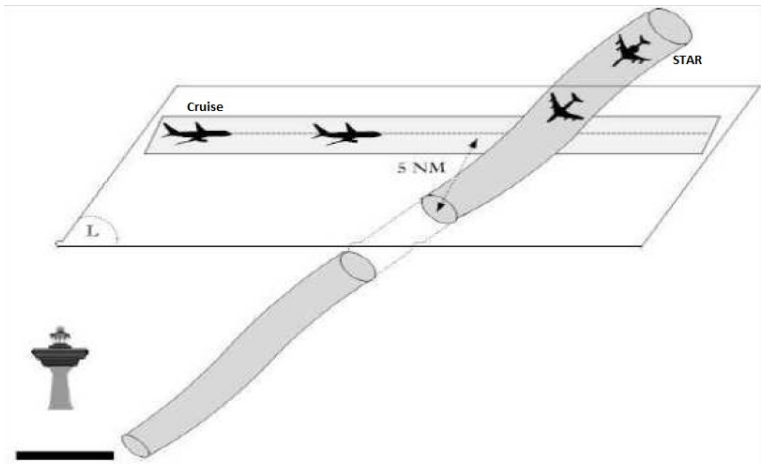



**Figure 5.6:** Encounter type considered with aircraft on a STAR and aircraft en-route

As a case study, we consider here an Encounter type 3 scenario as previously described and depicted in Figure 5.6. In Section 2.3 we recalled that the documented intent of an aircraft is its RBT, which is shared through the SWIM. However,

it may happen that due to unpredictable events, failures and etc., a mismatch occurs between the intent available in the airborne system (the advanced Flight Management System) and in the ATC system. Moreover, the situation awareness (SA) of the Executive Controller concerning the intent of an aircraft may be different. Similarly, the situation awareness of the Flight Crew may be different. For each of the aforementioned four agents, two possibilities regarding the intent situation awareness, have been identified:

- **FMS / Guidance Navigation and Control system of a particular aircraft**. The intent (i.e. the RBT-based flight plan) of the aircraft as provided by the FMS system can be either that the aircraft should make a turn away from its current course (i.e. on a SID, STAR, or en route lane), or that the aircraft should continue flying in a straight line.
- **Flight crew**. The intent situation awareness of the flight crew can be: either the own aircraft makes a turn away from its current course or, it continues to fly in a straight line.
- **ATC system**. The intent of a particular aircraft provided by the ATC system can be: either the aircraft makes a turn away from its current course or, it continues to fly in a straight line.
- **Executive Controller**. The intent situation awareness of the controller regarding a particular aircraft can be: either it makes a turn away from its current course or, it continues to fly in a straight line.

The combination of the above reported intent situation awareness, results in 16 possible configurations, as reported in Table 5.1.

In the sequel we suppose that:

**A1.** If an aircraft flies on its FMS (e.g. autopilot) then:

- If the FMS has the intent situation awareness to make a turn, the aircraft will make the turn.
- If the FMS has the intent situation awareness to fly in a straight line, the aircraft will fly in a straight line.

**A2.** If the aircraft flies on pilot control panel (i.e. decoupled from FMS), then:

- If the pilot has the intent situation awareness to make a turn, the aircraft will make the turn.
- If the pilot has the intent situation awareness to fly in a straight line, the aircraft will fly in a straight line.

and we can identifies three non-nominal encounter conditions:

| nr. | FMS | Flight Crew | ATC system | ATCo |
|---|---|---|---|---|
| **C01** | straight | straight | straight | straight |
| **C02** | straight | straight | straight | turn |
| **C03** | straight | straight | turn | straight |
| **C04** | straight | straight | turn | turn |
| **C05** | straight | turn | straight | straight |
| **C06** | straight | turn | straight | turn |
| **C07** | straight | turn | turn | straight |
| **C08** | straight | turn | turn | turn |
| **C09** | turn | straight | straight | straight |
| **C10** | turn | straight | straight | turn |
| **C11** | turn | straight | turn | straight |
| **C12** | turn | straight | turn | turn |
| **C13** | turn | turn | straight | straight |
| **C14** | turn | turn | straight | turn |
| **C15** | turn | turn | turn | straight |
| **C16** | turn | turn | turn | turn |

**Table 5.1:** Sixteen combinations of intent situation awareness options for four agents.

- **No ATC**. We consider a situation in which the Controller does not, or is not able to give a conflict recovery instruction to an aircraft in conflict. In this situation, we assume that the aircraft on the en route lane makes a turn away from its lane; the aircraft on the STAR maintains a straight line. However, no instruction is given to resolve any conflict. This situation occur due to failing communication or failing ground surveillance equipment.

- **STCA only**. We consider a situation in which the aircraft en route makes a turn and the aircraft on the STAR maintains a straight line. The Controller and the ATC system have the intent situation awareness that the turn can be safely made. This means that the aircraft on approach has an RBT according to the STAR, and the aircraft en route has an RBT that is making a turn away from the en route lane. The Controller is monitoring the positions and velocities of all aircraft that are available to him through surveillance equipment and the traffic situation display. Before separation is less than 5NM, about two minutes before a conflict occurs, the short term conflict alert system warns the Controller of a conflict. Then the controller uses R/T to give the flight crew of one of the aircraft an avoidance instruction. In particular the controller gives to the aircraft on the STAR an instruction to level off, thus ensuring

vertical separation. Another option is to send the aircraft en route back to the en route lane.

- **FPCM and/or STCA**. We consider a situation in which the aircraft makes a turn, and the Controller has the situation awareness that the turn can be safely made. The ATC system has the intent situation awareness that the aircraft should continue to fly in a straight line. Then the aircraft on approach has an RBT according to the STAR, and the aircraft en route has an RBT according to the en route lane. We assume that the aircraft en route makes this turn, the aircraft on the STAR maintains a straight line. The Controller is monitoring the positions and velocities of all aircraft that are available to him through surveillance equipment and the traffic situation display. The flight plan conformance monitoring (MONA) detects that the aircraft is making a turn away from its intent RBT. The flight plan conformance monitoring detects that the aircraft is making a turn away from its intent RBT. The Controller is alerted to this deviation, and he uses R/T to give the flight crew of one of the aircraft a recovering instruction. The controller give the aircraft on the STAR an instruction to level off, thus ensuring vertical separation. Another option is to send the aircraft en route back to the en route lane.

### 5.1.2 Mathematical model of TMA T1 Scenario

We now introduce an AHS that properly models this scenario. We start by providing a detailed description of a simplified scenario in which only one aircraft crew agent, one tactical controller agent, one ATC system agent and one guidance navigation and control agent operate. In the next section we consider a more realistic scenario in which a larger number of agents operate. Define the AHS $\mathbb{A} = (\mathbb{V}, \mathbb{E})$, where:

- $\mathbb{V}$=$\{M_{airc}, M_{atc}, M_{sys}, M_{gnc}\}$, where:

  $M_{crew}$ represents the Aircraft crew agent.

  $M_{atc}$ the Tactical controller agent.

  $M_{sys}$ the ATC system agent.

  $M_{gnc}$ the Guidance navigation and control agent.

- $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ as depicted in Figure 5.7.

We now describe in detail each FSM or hybrid system composing the AHS $\mathbb{A}$. The hybrid system associated to the Aircraft agent is described by:

$$M_{crew} = (Q_{crew} \times X_{crew}, \{q_{0,crew}\} \times X_{0,crew}, U_{crew}, Y_{crew}, \mathcal{E}_{crew}, \Sigma_{crew}, E_{crew}, \Psi_{crew}, \eta_{crew}),$$

where:

- $Q_{crew} = \{q_{1,crew}, \ldots, q_{9,crew}\}$ where:

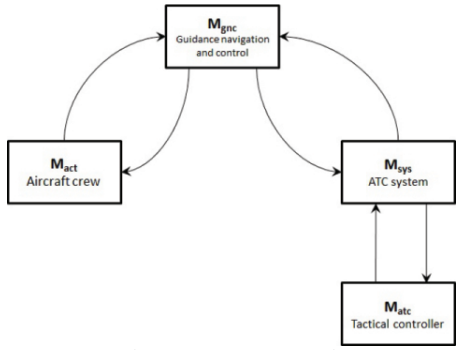



**Figure 5.7:** AFSM for the TMA T1 Scenario

- $q_{1,crew}$ is the Monitoring of flight according to RBT state.
- $q_{2,crew}$ is the Detection of conflict or deviation state.
- $q_{3,crew}$ is the Conflict reported by the controller.
- $q_{4,crew}$ is the Flight-plan deviation resolution manoeuvre state.
- $q_{5,crew}$ is the Conflict avoidance manoeuvre state.
- $q_{6,crew}$ is the Execution of manoeuvre state.
- $q_{7,crew}$ is the Detection of deviation from flightplan state.
- $q_{8,crew}$ is the Flight trajectory data updated state.
- $q_{9,crew}$ is the Controller instruction state.

- $X_{crew} \subset \mathbb{R}^6$ is the continuous state space with $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in X_{crew}$, where:
  - $x_1$ and $x_2$ indicate the horizontal position.
  - $x_3$ is the altitude.
  - $x_4$ is the true airspeed.
  - $x_5$ is the heading angle.
  - $x_6$ is the flight path angle.

- $q_{0,crew} = q_{1,crew}$ and $X_{0,crew} = \{(x_1, x_2, x_3, x_4, x_5, x_6)\}$.
- $U_{crew} \subset \mathbb{R}^3$ with $u = (u_1, u_2, u_3) \in U_{airc}$, where:
  - $u_1$ is the engine thrust.
  - $u_2$ is the bank angle.
  - $u_3$ is the flight path angle.

- $Y_{airc} = X_{airc}$.
- $\{\mathcal{E}_{crew,q}\}_{q \in Q_{crew}}$ associates to each discrete state $q \in Q_{airc}$ the continuous dynamics $\dot{x} = f_q(x,u)$ and $y = x$, where $f_q(x,u)$ is given[1] by:

$$f_q(x,u) = \begin{cases} \dot{x_1} = x_4 \cos(x_5)\cos(x_6) \\ \dot{x_2} = x_4 \sin(x_5)\cos(x_6) \\ \dot{x_3} = x_4 \sin(\alpha) \\ \dot{x_4} = \frac{1}{m}\left[u_1 \cos(\alpha) - D - mg\sin(x_6)\right] \\ \dot{x_5} = \frac{1}{mx_4}\left[L\sin(u_2) + u_1 \sin(\alpha)\sin(u_2)\right] \\ \dot{x_6} = \frac{1}{mx_4}\left[(L + u_1\sin(\alpha))\cos(u_2) - mg\cos(u_3))\right] \end{cases}$$

  for each $i = 1, 2, ..., 9$, where $L$ is the lift force, $D$ the drag force, $\alpha$ the angle of attack, $g$ gravitational acceleration.

- $\Sigma_{crew} = \{\sigma_{1,crew}, \ldots, \sigma_{12,crew}\}$ where:
  - $\sigma_{1,crew} = \psi_{9,gnc}$ communication of an avoidance manoeuvre by the controller.
  - $\sigma_{2,crew} = \sigma_{3,crew} = \psi_{14,gnc}$ resolution manoeuvre.
  - $\sigma_{4,crew} = \varnothing$ execution of the manoeuvre (internal).
  - $\sigma_{5,crew} = \psi_{6,gnc}$ transfer of data to the devices.
  - $\sigma_{7,crew} = \psi_{8,gnc}$ display messages from controller.
  - $\sigma_{8,crew} = \psi_{9,gnc}$ avoidance manoeuvre.
  - $\sigma_{9,crew} = \sigma_{10,airc} = \sigma_{11,airc} = \varnothing$ internal inputs.
  - $\sigma_{12,crew}$ flight data mismatch.
- $E_{crew} \subseteq X_{crew} \times 2^{U_{crew}} \times X_{crew}$ is depicted in Figure 5.8.
- $\Psi_{airc} = \{\psi_{1,airc}, \ldots, \psi_{7,airc}\}$ where:
  - $\psi_{1,crew}$ represents cruise.
  - $\psi_{2,crew}$ choice of a manoeuvre.
  - $\psi_{3,crew}$ execution of the manoeuvre.
  - $\psi_{4,crew} = \psi_{5,crew} = \psi_{6,crew} = \psi_{7,crew} = \varnothing$ non measurable outputs.
- $\eta_{crew} : Q_{crew} \to 2^{\Psi_{crew}}$.

The FSM associated to the tactical controller agent is described by:

$$M_{atco} = (Q_{atco}, q_{0,atco}, \Sigma_{atco}, \Psi_{atco}, \eta_{atco}, E_{atco}),$$

where:

[1] The proposed model has been taken from [25].

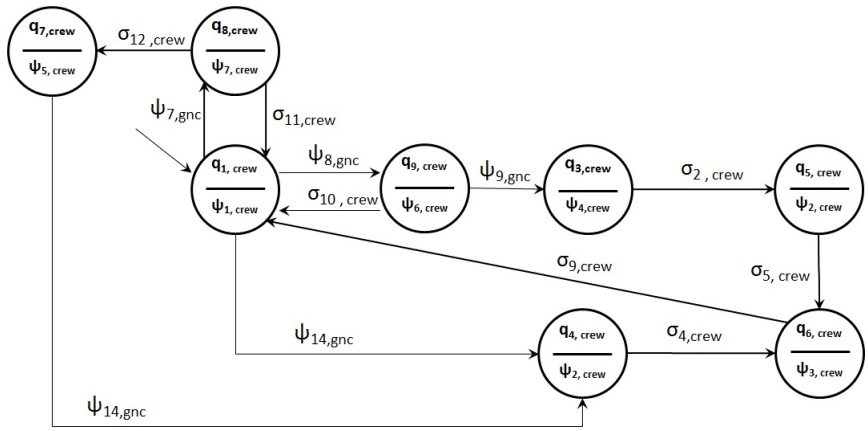



**Figure 5.8:** FSM for aircraft crew agent

- $Q_{atco} = \{q_{1,atco}, \ldots, q_{6,atco}\}$ where:
  - $q_{1,atco}$ is the Monitoring state.
  - $q_{2,atco}$ is the Conflict resolution state.
  - $q_{3,atco}$ is the Avoidance manoeuvre state.
  - $q_{4,atco}$ is the Resolution manoeuvre state.
  - $q_{5,atco}$ is the Termination manoeuvre state.
  - $q_{6,atco}$ is the ATCo messages.
- $q_{0,atco} = q_{1,atco}$.
- $\Sigma_{atco} = \{\sigma_{1,atco}, \ldots, \sigma_{9,atco}\}$ where:
  - $\sigma_{1,atco}$ represents monitor of a conflict.
  - $\sigma_{2,atco} = \psi_{4,sys}$ detection of a conflict.
  - $\sigma_{3,atco}$ manoeuvre resolution (internal).
  - $\sigma_{4,atco}$ manoeuvre avoidance (internal).
  - $\sigma_{5,atco}$ communication of flight information to the pilot.
  - $\sigma_{6,atco} = \sigma_{7,atco} = \sigma_{8,atco} = \sigma_{9,atco} = \varnothing$ internal inputs.
- $\Psi_{atco} = \{\psi_{1,atco}, \ldots, \psi_{6,atco}\}$ where:
  - $\psi_{1,atco}$ represents resolution manoeuvre searched.
  - $\psi_{2,atco}$ avoidance manoeuvre.
  - $\psi_{3,atco}$ resolution manoeuvre chosen.

- $\psi_{4,atco}$ the send of a flight data to the aircraft crew.
- $\psi_{5,atco} = \psi_{6,atco} = \varnothing$ non measurable outputs.

- $\eta_{atco} : Q_{atco} \to 2^{\Psi_{atco}}$ is depicted in Figure 5.9.
- $E_{atco} \subseteq Q_{atco} \times 2^{\Sigma_{atco}} \times Q_{atco}$ is depicted in Figure 5.9.

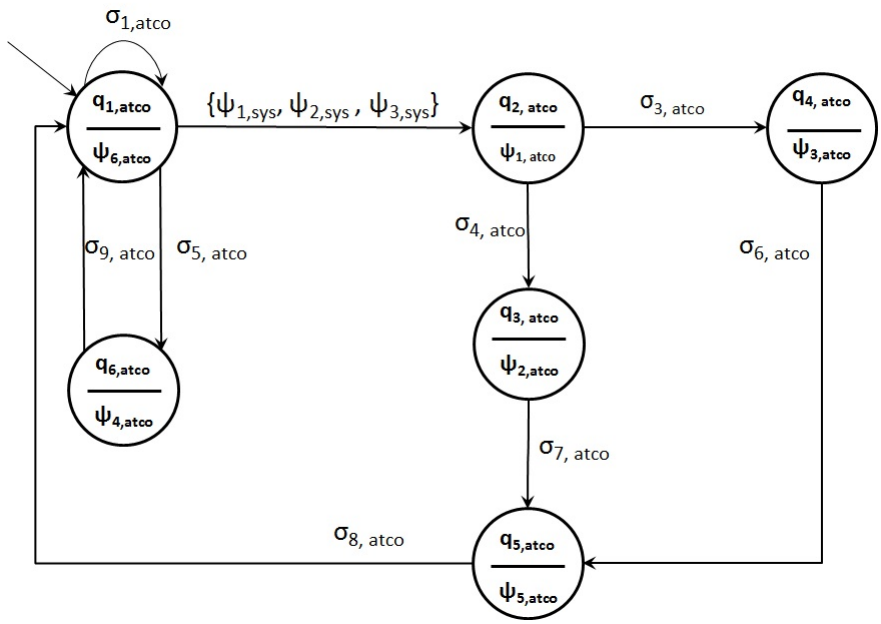



**Figure 5.9:** FSM for tactical control agent

The FSM associated to the ATC system agent is described by:

$$M_{sys} = (Q_{sys}, q_{0,sys}, \Sigma_{sys}, \Psi_{sys}, \eta_{sys}, E_{sys}),$$

where:

- $Q_{sys} = \{q_{1,sys}, \ldots, q_{8,sys}\}$ where:
  - $q_{1,sys}$ is the Monitoring trajectories state.
  - $q_{2,sys}$ is the Trajectory conflict state.
  - $q_{3,sys}$ is Vertical error state.
  - $q_{4,sys}$ is the Transversal error state.
  - $q_{5,sys}$ is Controller Pilot Data Link Communication state.
  - $q_{6,sys}$ is Radio/Telecommunication device state.

  - $q_{7,sys}$ is Flight path state.
  - $q_{8,sys}$ is Conflict detection state.

- $q_{0,sys} = q_{1,sys}$.
- $\Sigma_{sys} = \{\sigma_{1,sys}, \ldots, \sigma_{15,sys}\}$ where:
  - $\sigma_{1,sys}$ represents monitor of a conflict by the information coming from input sensor.
  - $\sigma_{2,sys}$ occurrence of a transversal error.
  - $\sigma_{3,sys}$ occurrence of a vertical error.
  - $\sigma_{4,sys}$ occurrence of a trajectory conflict.
  - $\sigma_{6,sys} = \{\psi_{4,atc}, \psi_{7,gnc}\}$ the send of a flight data to the aircraft crew or the receive of data from the aircraft crew.
  - $\sigma_{7,sys} = \{\psi_{2,atc}, \psi_{3,atc}\}$ avoidance manoeuvre or resolution manoeuvre.
  - $\sigma_{5,sys} = \sigma_{8,sys} = \sigma_{9,sys} = \sigma_{10,sys} = \sigma_{11,sys} = \sigma_{13,sys} = \sigma_{15,sys} = \varnothing$.
  - $\sigma_{12,sys}$ the comparison of position/velocity of two aircraft.
  - $\sigma_{14,sys}$ the comparison of position/velocity of aircraft with respect to own RBT.
- $\Psi_{sys} = \{\psi_{1,sys}, \ldots, \psi_{8,sys}\}$ where:
  - $\psi_{1,sys}$ represents generation of an FPCM alarm due to a transversal error.
  - $\psi_{2,sys}$ generation of an FPCM alarm due to a vertical error.
  - $\psi_{3,sys}$ generation of an STCA alarm due to a trajectory conflict.
  - $\psi_{4,sys} = \psi_{7,sys} = \psi_{8,sys}\varnothing$ a non measurable output.
  - $\psi_{5,sys}$ transfer of messages to the aircrew.
  - $\psi_{6,sys}$ instruction for the aircrew.
- $E_{sys} : Q_{sys} \rightarrow 2^{\Psi_{sys}}$.
- $E_{sys} \subseteq Q_{sys} \times 2^{\Sigma_{sys}} \times Q_{sys}$ is depicted in Figure 5.10.

The FSM associated to the Guidance Navigation and Control agent is described by:

$$M_{gnc} = (Q_{gnc}, q_{0,gnc}, \Sigma_{gnc}, \Psi_{gnc}, \eta_{gnc}, E_{gnc}),$$

where:

- $Q_{gnc} = \{q_{1,gnc}, \ldots, q_{14,gnc}\}$ where:
  - $q_{1,gnc}$ is the Monitoring state.
  - $q_{2,gnc}$ is the Generation target state.

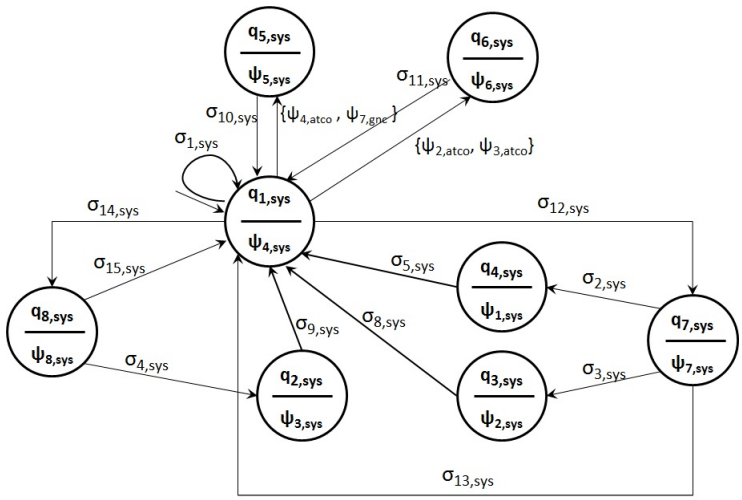



**Figure 5.10:** FSM for ATC system agent

- $q_{3,gnc}$ is the Receive information state.
- $q_{4,gnc}$ is the Waypoint state.
- $q_{5,gnc}$ is the Data navigation sensor state.
- $q_{6,gnc}$ is the Display next waypoint state.
- $q_{7,gnc}$ is the Autopilot state.
- $q_{8,gnc}$ is the Determine position and velocity state.
- $q_{9,gnc}$ is the Navigation data state.
- $q_{10,gnc}$ is Transfer to aircraft state.
- $q_{11,gnc}$ is Display data state.
- $q_{12,gnc}$ is CPDLC display state.
- $q_{13,gnc}$ is TCAS monitoring state.
- $q_{14,gnc}$ is TCAS alert state.

- $q_{0,gnc} = q_{1,gnc}$.
- $\Sigma_{gnc} = \{\sigma_{1,gnc}, \ldots, \sigma_{20,gnc}\}$ where:
  - $\sigma_{1,gnc}$ represents generation of a new aircraft target.
  - $\sigma_{2,gnc}$ information coming from the sensor of the aircraft.
  - $\sigma_{3,gnc}$ information coming from the waypoint.
  - $\sigma_{4,gnc}$ generation of a target to the display.

- $\sigma_{5,gnc}$ generation of a target to autopilot.
- $\sigma_{6,gnc}$ setting of parameters to determine position and velocity.
- $\sigma_{7,gnc}$ setting of parameters to determine navigation data.
- $\sigma_{8,gnc} = y_{5,sys}$ transfer of messages to the aircrew.
- $\sigma_{9,gnc} = \sigma_{10,gnc} = \sigma_{11,gnc} = \sigma_{12,gnc} = \sigma_{13,gnc} = \sigma_{14,gnc} = \sigma_{15,gnc} = \sigma_{16,gnc} = \sigma_{17,gnc} = \sigma_{19,gnc} = \sigma_{20,gnc} = \varnothing$ internal inputs.
- $\sigma_{18,gnc}$ acquiring position and velocity of aircrafts.

- $\Psi_{gnc} = \{\psi_{1,gnc}, \ldots, \psi_{14,gnc}\}$ where:
  - $\psi_{1,gnc}$ represents downloading of information from technical systems.
  - $\psi_{2,gnc}$ uploading of data coming from the aircraft sensors.
  - $\psi_{3,gnc}$ uploading of position and velocity to technical systems.
  - $\psi_{4,gnc}$ transfer of waypoint to the aircraft system.
  - $\psi_{5,gnc}$ transfer of navigation data to the aircraft system.
  - $\psi_{6,gnc}$ transfer of the new target to the aircraft system.
  - $\psi_{7,gnc}$ transfer of data to the devices.
  - $\psi_{8,gnc}$ display of data.
  - $\psi_{9,gnc}$ conflict resolution instruction.
  - $\psi_{10,gnc} = \psi_{11,gnc} = \psi_{12,gnc} = \psi_{13,gnc} = \varnothing$ non measurable outputs.
  - $\psi_{14,gnc}$ resolution manoeuver to be executed.
- $\eta_{gnc} : Q_{gnc} \to 2^{\Psi_{gnc}}$.
- $E_{gnc} \subseteq Q_{gnc} \times 2^{\Sigma_{gnc}} \times Q_{gnc}$ is depicted in Figure 5.11.

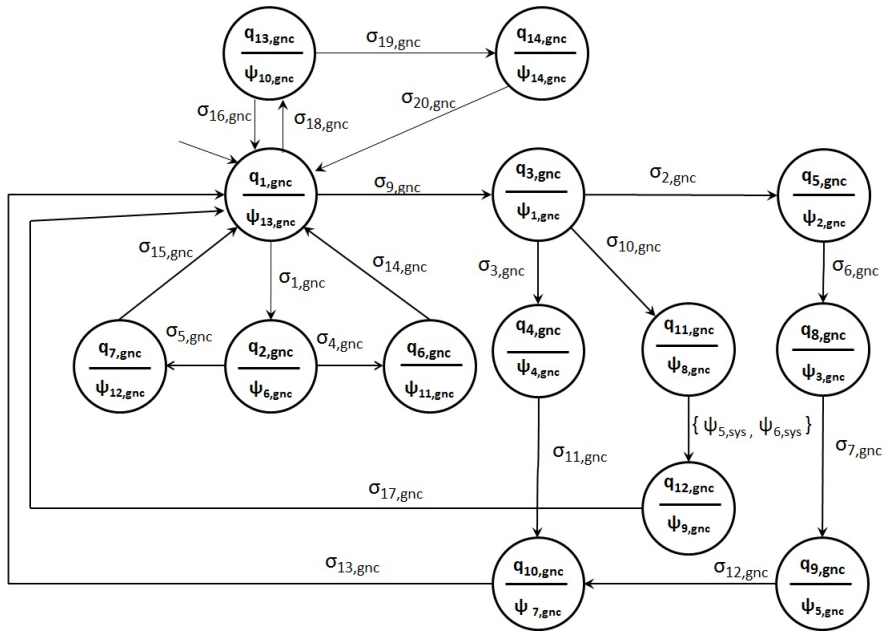



**Figure 5.11:** FSM for guidance navigation and control agent

### 5.1.3 Analysis of critical observability of TMA T1 Scenario

Consider a scenario in which 4 SID aircraft, 4 STAR aircraft, 4 cruise routes aircraft and one ATC operate. The communication scheme that models exchange of information among the agents involved can be described by the AHS $\mathbb{A} = (\mathbb{V}, \mathbb{E})$ shown in Figure 5.12, where:

- $\mathbb{V} = \{M_1, \ldots, M_{26}\}$ is a collection of:
  - 12 hybrid systems $M_i, i = 1, \ldots, 12$ representing the aircraft-crew (SIDs in green, STARs in red and cruise routes in blue).
  - 12 FSMs $M_i, i = 13, \ldots, 24$ representing the guidance-navigation and control of each aircraft.
  - 1 FSM $M_{25}$ representing the ATC System.
  - 1 FSM $M_{26}$ the Controller.

  The models of each hybrid system and FSM have been detailed in the previous section.
- $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ describes the interaction of hybrid systems and FSMs as shown in Figure 5.12.

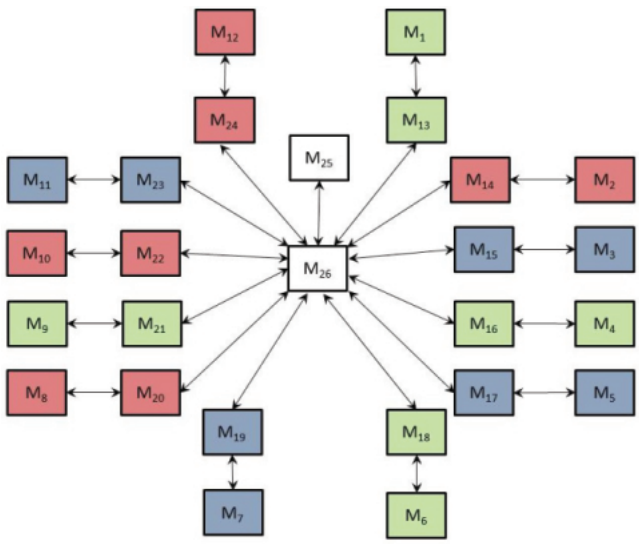



**Figure 5.12:** AFSM with 12 aircraft and one ATC

Whenever two aircraft are closer than 3NM apart in horizontal direction while being closer than 1000ft apart in vertical direction, they are said to be in conflict.

This translates in considering the hybrid systems that model these aircraft as belonging to a certain critical relation. In the sequel we consider the following critical relation:

$$\mathcal{R}_c = \mathcal{R}_c^2 \cup \mathcal{R}_c^3,$$

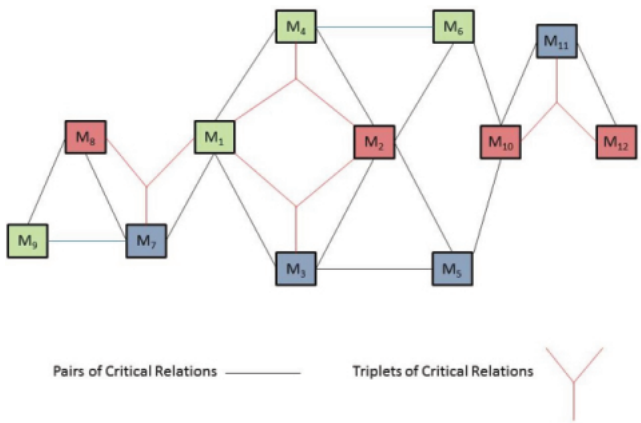



**Figure 5.13:** Critical relation $\mathcal{R}_c$

where

- $\mathcal{R}_c^2 = \{(M_4, M_6), (M_2, M_6), (M_2, M_5), (M_3, M_5), (M_6, M_{10}), (M_5, M_{10}), (M_8, M_9), (M_7, M_9), (M_1, M_4), (M_2, M_4), (M_1, M_3), (M_{10}, M_{11}), (M_{11}, M_{12}), (M_7, M_8), (M_1, M_7), (M_2, M_3)\}$;
- $\mathcal{R}_c^3 = \{(M_1, M_7, M_8), (M_1, M_2, M_3), (M_1, M_2, M_4), (M_{10}, M_{11}, M_{12}), (M_1, M_4, M_2), (M_{12}, M_{11}, M_{10})\}$.

This critical relation is shown in Figure 5.13. The goal is to check critical observability of this system. Hence, we first need to construct a critical observer for $\mathbb{M}(\mathbb{A})$. For the construction of the observer we need to define the critical relation $\mathfrak{R}_c$ (on the states) as in (4.5) that is obtained by detailing the critical relation $\mathcal{R}_c$ (on the FSMs) as in (4.6). We obtained:

$$\mathfrak{R}_c = \bigcup_{i \in [2;3]} \mathfrak{R}_c^i,$$

where

- $\mathfrak{R}_c^2 = \{\mathfrak{R}_{i_1,i_2}, i_1, i_2 = 1, 2, \ldots, 12\}$, where $\mathfrak{R}_{i_1,i_2} = \{(q_6^{i_1}, q_6^{i_2})\}$;
- $\mathfrak{R}_c^3 = \{\mathfrak{R}_{i_1,i_2,i_3}, i_1, i_2, i_3 = 1, 2, \ldots, 12\}$, where $\mathfrak{R}_{i_1,i_2,i_3} = \{(q_6^{i_1}, q_6^{i_2}, q_6^{i_3})\}$.

The size of the $\mathbb{M}(\mathbb{A})$ is very large and the construction of $\mathcal{O}_{\mathfrak{R}_c}$ is therefore rather demanding from the computational complexity point of view. More precisely, since the cardinality of the set of states of $\mathbb{M}(\mathbb{A})$ is about $2.36 \cdot 10^{26}$, the computational complexity in the construction of the corresponding critical observer is given by $O(2^{2.36 \cdot 10^{26}})$. By applying the results reported in Section 3 we computed the quotient of $\mathbb{A}$ induced by the maximal critical compositional bisimulation $\mathbb{R}^*$. The AHS obtained, denoted $\widehat{\mathbb{A}}$, and depicted in Figure 5.14, is $(\mathcal{R}_c, \widehat{\mathcal{R}}_c)$-bisimilar to $\mathbb{A}$ with $\widehat{\mathcal{R}}_c$ described by:

$$\widehat{\mathcal{R}}_c = \bigcup_{i \in [2;3]} \widehat{\mathcal{R}}_c^i,$$

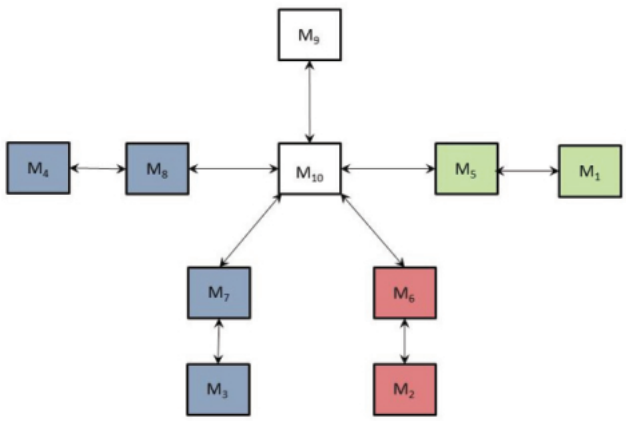



**Figure 5.14:** AFSM with 4 aircraft and one ATC

where:

- $\widehat{\mathcal{R}}_c^2 = \{(M_2, M_4), (M_3, M_4), (M_1, M_3), (M_2, M_3)\}$;
- $\widehat{\mathcal{R}}_c^3 = \{(M_1, M_2, M_3)\}$.

This critical relation is shown in Figure 5.15. For the construction of the observer we need to define the critical relation $\mathfrak{R}_c$ (on the states) as in (4.5) that is obtained by detailing the critical relation $\mathcal{R}_c$ (on the FSMs) as in (4.6). We obtain:

$$\widehat{\mathfrak{R}}_c = \bigcup_{i \in [2;3]} \widehat{\mathfrak{R}}_c^i,$$

where:

- $\widehat{\mathfrak{R}}_c^2 = \{\mathfrak{R}_{i_1,i_2}, i_1, i_2 = 1, 2, \ldots, 4\}$, where $\mathfrak{R}_{i_1,i_2} = \{(q_6^{i_1}, q_6^{i_2})\}$;

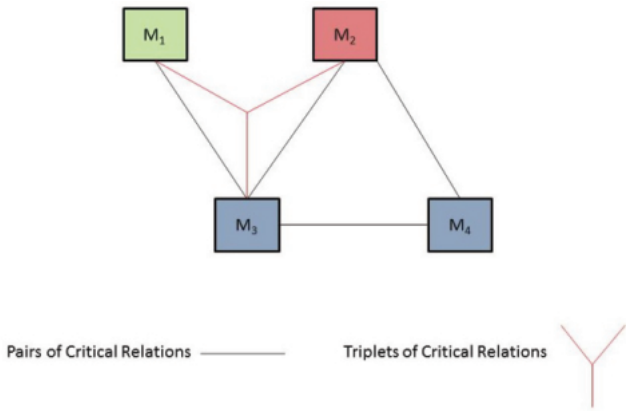



**Figure 5.15:** Critical relation $\widehat{\mathcal{R}}_c$

| | Computational Complexity |
|---|---|
| Observer for $\mathbb{A}$ | $O(2^{2.36\cdot 10^{26}})$ |
| Observer for $\widehat{\mathbb{A}}$ | $O(2^{6.74\cdot 10^{9}})$ |

**Table 5.2:** Computational complexity analysis.

- $\widehat{\mathfrak{R}}_c^3 = \{\mathfrak{R}_{i_1,i_2,i_3}, i_1, i_2, i_3 = 1, 2, \ldots, 4\}$, where $\mathfrak{R}_{i_1,i_2,i_3} = \{(q_6^{i_1}, q_6^{i_2}, q_6^{i_3})\}$.

The cardinality of the set of states of the reduced FSM $\mathbb{M}(\hat{\mathbb{A}})$ is about $6.74 \cdot 10^9$, and the computational complexity in the construction of the corresponding critical observer is therefore given by $O(2^{6.74\cdot 10^9})$. By applying the results reported in [16] it is possible to show that the FSM $\mathbb{M}(\hat{\mathbb{A}})$ is critically observable. Since $\mathbb{A} \cong_{(\mathcal{R}_c, \widehat{\mathcal{R}}_c)} \widehat{\mathbb{A}}$, by Theorem 4.3 we conclude that the original FSM $\mathbb{M}(\mathbb{A})$ is critically observable. The computational complexity reduction achieved by using the approach presented in Section 3 is summarized in Table 5.3.

## 5.2 Airborne Separation In Trail Procedure

The In Trail Procedure (ITP) is part of the Airborne Separation Assistance Systems (ASAS). ASAS embraces the goal of improving flight management by introducing a stronger interaction between pilots and controllers. The In Trail Procedure is seen as an Airborne Separation (ASEP) Application which is one of the four ASAS application categories. ASEPñITP applications involve the transfer of responsibilities for the separation from the controller to the flight crew during the execution of the procedure. This can happen when the flight crew have more appropriate surveillance equipments, i.e. ADS-B , and is therefore able to monitor separation and act, if necessary.

### 5.2.1 Description of the In Trail Procedure

The Airborne Separation In Trail Procedure (ASEP-ITP) [13, 49] described hereafter is a procedure that aims at improving flight efficiency along oceanic routes where procedural control is performed, and is an extension of the Airborne Traffic Situational Awareness In Trail Procedure (ATSA-ITP).

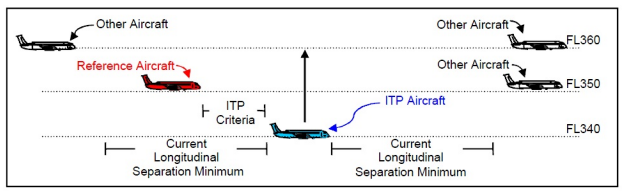



**Figure 5.16:** Example of ITP geometry

The ASEP-ITP allows climb or descend through only one flight level for a maximum of 2000 feet in RVSM airspace (and 4000 feet in non-RVSM) and the ITP speed/distance criteria are designed so that under nominal conditions the proposed 5NM separation minimum is preserved throughout the ITP manoeuvre. The proposed ITP speed/distance criteria are the following:

- initiation ITP distance of no less than 10 NM and positive ground speed differential of no more than 20 kts, or

- ITP distance of no less than 15 NM and positive ground speed differential of no more than 30 kts.

The ITP encompasses a set of six vertical geometries: leading climb (as shown in Figure 5.16), leading descend, following climb, following descend, combined leading-following climb and combined leading-following descend. These geometries

are designed on the basis of the relative position of the ITP aircraft and one or two reference aircraft.

The ITP aircraft must maintain a minimum 300 ft/min of climb or descend and constant cruise Mach number throughout the ITP manoeuvre. The reference aircraft must be non-manoeuvring and it is not expected to manoeuvre during the ITP. Given these conditions, it can be shown that a 4000 ft flight level change would result in a reduction in the initial distance of 4.5 NM assuming a positive ground speed differential of 20 kts. To ensure that the ITP separation minimum of 5NM will be guaranteed during the flight level change under these conditions, the initial distance between the aircraft must exceed 9.5 NM. So using 10 NM of initial distance the separation minimum is guaranteed. In the same way it could be proved that with positive ground speed differential of more than 20 but less than 30 kts, an initial distance of 15 NM ensures that ITP separation minimum is respected.

A compact view of the ASEP-ITP phases is illustrated in Figure 5.17, and is now described.

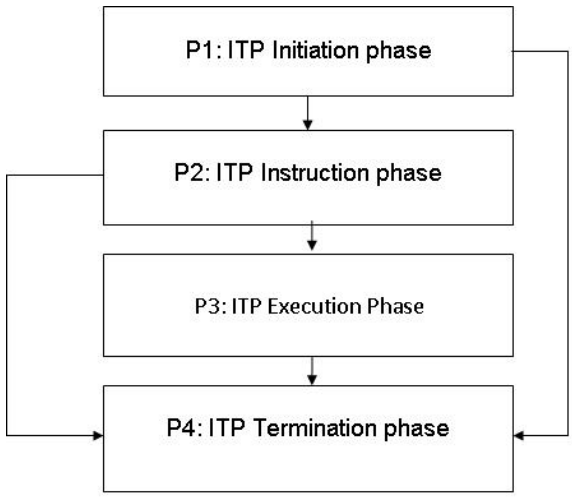



**Figure 5.17:** ASEP-ITP phases diagram

### ASEP-ITP phases

#### ITP Initiation phase

The decision to request an ITP rather than a standard flight level change will typically be based on a number of factors outside the scope of the ITP application, such as crew preference and judgment, the magnitude of the desired flight level change, and any other information available to the crew about the flight's progress and proximate traffic situation.

Once the flight crew has decided to consider requesting an ITP, the flight crew proceeds through the following steps to formulate and initiate the request:

1. Identification of ITP flight levels

    - The crew identifies a requested flight level, which is a flight level above (for a climb) or below (for a descend) one flight level and that is no more than 4000 ft from the initial flight level.

2. Checking ITP aircraft Performance by the crew:

    - The ITP aircraft is capable of performing a rate of climb or descend of at least 300 fpm at the assigned Mach number to the requested flight level.
    - The ITP aircraft is not expected to manoeuvre except for a climb or descend or a change of course to remain on their clearance.

3. Identification of reference aircraft. The crew selects as reference aircraft up to two potentially blocking aircraft which meet the following criteria:

    - The ITP aircraft has the same direction with potentially blocking aircraft.
    - Qualified ADS-B data are available from potentially blocking aircraft.
    - The ITP speed/distance criteria are met with potentially blocking aircraft.

4. ITP Request

    - If the ITP criteria are met, the ITP aircraft crew requests the ITP, using the required ITP phraseology which provides the controller with the requested ITP flight level change geometry (i.e., leading or following), the ITP distance and the flight ID of reference aircraft.

**ITP Instruction Phase**

1. Issue of ITP Clearance by ATCo controller depends if standard separation will be met with all aircraft at the requested flight level and at all flight levels between the ITP aircraft's initial flight level and requested flight level. If so, a standard (non-ITP) flight level change clearance can be issued. *If not,*

    - Determine whether the ITP request message format is correct and that the flight crew has correctly identified the reference aircraft at the intervening flight level.
    - Determine whether standard separation will be met with other aircraft (i.e., all but the reference aircraft) at the requested flight level and at all flight levels between the ITP aircraft's initial Flight Level and requested flight level.

- Determine whether the ITP aircraft is not a reference aircraft in another ITP clearance.
- Determine whether the ITP aircraft and the reference aircraft are on the same track.
- Determine whether the reference aircraft are non-manoeuvring and not expected to manoeuvre during the ITP. The controller will not issue an ITP clearance if a reference aircraft is starting a manoeuvre or expected to manoeuvre.
- Determine whether the positive Mach differential is no greater than 0.03 Mach.

Based on the ITP aircraft's request and the controller's determination of the previous six conditions, the controller would issue the ITP clearance.

2. ITP Crew Re-Assessment

   - After the ITP clearance is issued, the flight crew of the ITP aircraft must again determine whether the ITP criteria continue to be met with respect to the reference aircraft immediately before initiating the climb or descend. If the ITP criteria are no longer met, the crew refuses the clearance and remains at the initial flight level.

**ITP Execution Phase**

1. ITP Aircraft Crew Tasks during the ITP Manoeuvre

   - As after a standard climb or descend clearance, the crew must initiate the ITP without delay after receipt of the clearance. Note that the crew re-assessment should not cause an undue delay in the initiation of this manoeuvre.
   - The crew must maintain the original cruise Mach number during the climb or descend.
   - The ITP aircraft must maintain a minimum 300 fpm climb or descend rate, or the minimum rate required by regulation, whichever greater, throughout the ITP manoeuvre.
   - The ITP aircraft crew shall monitor the ITP distance to the reference aircraft during the climb or descend. The crew monitors the ASAS equipment indicating the range of the blocking aircraft. If the separation minimum is predicted to be violated a temporary speed change is allowed.
   - The ITP flight crew reports the establishment at the new flight level.
   - If the ITP cannot be successfully completed as cleared once the climb or descend has been initiated, an abnormal termination occurs. ATCo must be notified immediately when this condition occurs.

2. Controller Tasks during the ITP Manoeuvre

    - The controller will not issue any manoeuvre clearance to the reference aircraft until the ITP Aircraft reports establishment at the new flight level or the ITP is abnormally terminated.

**ITP Termination Phase**

1. The ITP is completed when the ITP flight crew reports established at the new flight level.

2. If the ITP aircraft cannot successfully complete the ITP once the climb or descend has been initiated, an abnormal termination occurs.

### 5.2.2 Mathematical model of ASEP-ITP Procedure

**Pilot flying of ITP Aircraft Agent**

The hybrid model of the agent Pilot Flying of ITP Aircraft is given by:

$$\mathcal{H}_p = (Q_p \times X_p, Q_{p,0} \times X_{p,0}, U_p, Y_p, \mathcal{E}_p, \Sigma_p, E_p, \Psi_p, \eta_p) \tag{5.1}$$

where:

- $Q_p = \{q_{p,i}, i = 1, 2, ..., 13\}$ is the set of discrete states where:
  - $q_{p,1}$ is the Normal cruise state;
  - $q_{p,2}$ is the ITP Aborted state;
  - $q_{p,3}$ is the ITP Initiation state;
  - $q_{p,4}$ is the ITP Instruction state;
  - $q_{p,5}$ is the ITP Rejected state;
  - $q_{p,6}$ is the ITP Denied state;
  - $q_{p,7}$ is the ITP Standard execution state;
  - $q_{p,8}$ is the Non ITP criteria compliant execution state;
  - $q_{p,9}$ is the Wrong execution state;
  - $q_{p,10}$ is the Wrong termination state;
  - $q_{p,11}$ is the Abnormal termination state;
  - $q_{p,12}$ is the ITP Termination state;
  - $q_{p,13}$ is the Execution after ASAS Conflict detection state.
- $X_p \subset \mathbb{R}^6$ is the continuous state space with $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in X_p$, where $x_1$ and $x_2$ indicate the horizontal position, $x_3$ is the altitude, $x_4$ is the true airspeed, $x_5$ is the heading angle and $x_6$ is the flight path angle.
- $Q_{p,0} = \{q_1\}$ and $X_{p,0} = \{(x_0, z_i, v_{x0}, 0)\}$ is the set of initial states.
- $U_p \subset \mathbb{R}^3$ with $u = (u_1, u_2, u_3) \in U_p$, where $u_1$ is the engine thrust, $u_2$ is the bank angle and $u_3$ is the flight path angle.
- $Y_p = X_p$.

- $\{\mathcal{E}_{p,q}\}_{q\in Q}$ associates to each discrete state $q \in Q$ the continuous dynamics $\dot{x} = f_q(x)$ and $y = x$, where $f_q(x)$ is given[2] by:

$$f_q(x) = \begin{cases} \dot{x_1} = x_4 \cos(x_5)\cos(x_6) \\ \dot{x_2} = x_4 \sin(x_5)\cos(x_6) \\ \dot{x_3} = x_4 \sin(\alpha) \\ \dot{x_4} = \frac{1}{m}\left[u_1 \cos(\alpha) - D - mg\sin(x_6)\right] \\ \dot{x_5} = \frac{1}{mx_4}\left[L\sin(u_2) + u_1 \sin(\alpha)\sin(u_2)\right] \\ \dot{x_6} = \frac{1}{mx_4}\left[(L + u_1\sin(\alpha))\cos(u_2) - \right. \\ \qquad \left. -mg\cos(u_3))\right] \end{cases}$$

  for each $i = 1, 2, ..., 13$, where $L$ is the lift force, $D$ the drag force, $\alpha$ the angle of attack, $g$ gravitational acceleration.

- $\Sigma_p = \{\sigma_{p,i}, i = 1, 2, ..., 9\} \bigcup \{\varepsilon\}$ is the set of discrete inputs, where:
  - $\sigma_{p,1}$ represents the verification of ITP pre-conditions;
  - $\sigma_{p,2}$ the reassessment failed after a clearance reception;
  - $\sigma_{p,3}$ the ITP criteria are not verified;
  - $\sigma_{p,4}$ the ITP criteria verified;
  - $\sigma_{p,5}$ the clearance denied;
  - $\sigma_{p,6}$ the clearance issued;
  - $\sigma_{p,7}$ detection of an abnormal event
  - $\sigma_{p,8}$ a situational awareness inconsistency;
  - $\sigma_{p,9}$ an ASAS conflict detection communication;
  - $\varepsilon$ is an internal event.
- $E_p$ is the set of transitions given by the graph depicted in Figure 5.18.
- $\Psi_p = \{\psi_{p,i}, i = 1, 2, ..., 7\} \cup \{\varepsilon\}$ is the set of discrete outputs, where:
  - $\psi_{p,1}$ represents the clearance rejected by the crew;
  - $\psi_{p,2}$ the clearance request;
  - $\psi_{p,3}$ the setting of flight parameters for the climb;
  - $\psi_{p,4}$ the abnormal termination communication by the crew to the controller;
  - $\psi_{p,5}$ the report established at the new flight level;
  - $\psi_{p,6}$ the reversion to cruise operation;
  - $\psi_{p,7}$ the setting of flight parameters to solve an ASAS conflict detection;
  - $\varepsilon$ an unobservable transition.
- $\eta_p$ is the output function defined in the graph depicted in Figure 5.18.

[2]The proposed model has been taken from [25].

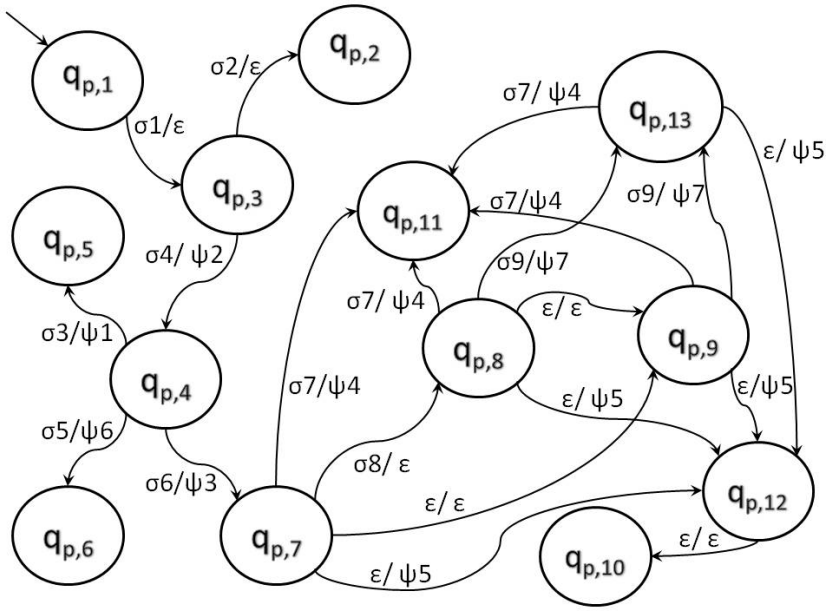



**Figure 5.18:** Directed graph of pilot flying of ITP aircraft agent.

The paths of the graph in Figure 5.18 identify all possible steps of the procedure from the viewpoint of the pilot, for a detailed description see [8].

### Air Traffic Controller

The hybrid model of the air traffic controller is given by the hybrid system $\mathcal{H}_{atc}$ consisting in the tuple:

$$\mathcal{H}_{atc} = (Q_{atc} \times X_{atc}, Q_{atc,0} \times X_{atc,0}, U_{atc} \times Y_{atc}, \mathcal{E}_{atc}, \Sigma_{atc}, E_{atc}, \Psi_{atc}, \eta_{atc}) \quad (5.2)$$

where:

- $Q_{atc} = \{q_{atc,i}, i = 1, 2, ..., 5\}$ is the set of discrete states, where:
  - $q_{atc,1}$ is the monitoring of the airspace;
  - $q_{atc,2}$ the clearance issued;
  - $q_{atc,3}$ the wrong clearance issued;
  - $q_{atc,4}$ the abnormal termination;
  - $q_{atc,5}$ the clearance refused.
- $X_{atc} = \varnothing$.
- $Q_{atc,0} = \{q_{atc,1}\}$ and $X_{atc,0} = \varnothing$.

- $U_{atc} = \varnothing$ and $Y_{atc} = \varnothing$.
- $\mathcal{E}_{atc} = \varnothing$.
- $\Sigma_{atc} = \{\sigma_{atc,i}, i = 1, 2, ..., 5\}$ is the set of discrete inputs, where:
  - $\sigma_{atc,1}$ represents the request of an ITP;
  - $\sigma_{atc,2}$ the abnormal termination communication;
  - $\sigma_{atc,3}$ a situational awareness inconsistency;
  - $\sigma_{atc,4}$ the communication by the crew of the establishment at the new flight level;
  - $\sigma_{atc,5}$ is the message of rejection of the clearance by the aircrew.
- $E_{atc}$ is the set of transitions given by the graph depicted in Figure 5.19.
- $\Psi_{atc} = \{\psi_{atc,1}, i = 1, 2, ..., 5\} \cup \{\varepsilon\}$ is the set of discrete outputs where:
  - $\psi_{atc,1}$ represents the clearance issued
  - $\psi_{atc,2}$ the ITP request denied,
  - $\psi_{atc,3}$ the communication to the aircrew of the abnormal termination message reception
  - $\psi_{atc,4}$ the confirmation of the reception of a standard ITP termination message,
  - $\psi_{atc,5}$ the confirmation of the reception of the rejection of the clearance by the aircrew,
  - $\varepsilon$ is associated with an unobservable transition.
- $\eta_{atc} : E_{atc} \to \Psi_{atc}$ is the discrete output function defined in the graph depicted in Figure 5.19.

The paths of the graph in Figure 5.19 identify all possible steps of the procedure from the viewpoint of the air traffic controller, for a detailed description see [8]. The above hybrid model is characterized by no continuous variables and in fact its state space $X_{atc}$ is empty. In ATM systems one air traffic controller is responsible for more than one clearance aircraft flying in his designed sky area. A hybrid system modeling one air traffic controller, responsible for $N$ clearance aircraft can be obtained by composing the hybrid model $\mathcal{H}_{atc}$ with $N - 1$ copies of it.

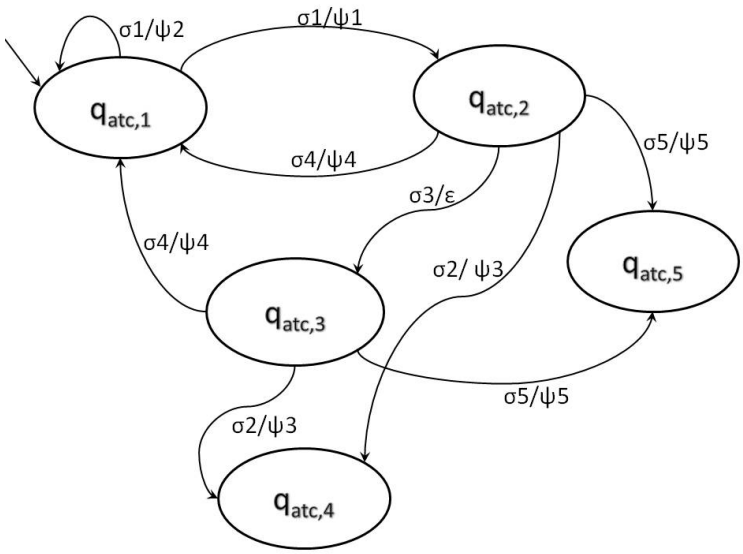



**Figure 5.19:** Directed graph of the Air Traffic Controller.

### 5.2.3 Analysis of Critical Observability of the ASEP-ITP

Consider a scenario in which 4 ITP aircraft $\mathcal{H}_p^1, \mathcal{H}_p^2, \mathcal{H}_p^3, \mathcal{H}_p^4$ and one ATC $\mathcal{H}_{atc}$ operate. As stressed in the previous section, one ATC interacting with 4 ITP aircraft can be modeled by means of the composition of 4 hybrid systems $\mathcal{H}_{atc}^1, \mathcal{H}_{atc}^2, \mathcal{H}_{atc}^3, \mathcal{H}_{atc}^4$. Hybrid models of $\mathcal{H}_p^i$ and $\mathcal{H}_{atc}^i$ coincide with the ones in (5.3) and (5.2), respectively. In the further developments we refer to state $q_{p,j}$ of $\mathcal{H}_p^i$ by $q_{p,j}^i$ and to the state $q_{atc,j}$ of $\mathcal{H}_{atc}^i$ by $q_{atc,j}^i$.

By applying the compositional rules introduced in Section 3.3.2 the AHS modeling the interaction of the agents $\mathcal{H}_p^i$ and $\mathcal{H}_{atc}^i$ can be defined, and resulting in:

$$\mathbb{A}_h = (\mathbb{V}, \mathbb{E}),$$

where:

- $\mathbb{V}$ is a collection of 8 hybrid systems, $\mathcal{H}_p^i$, $i = 1, \ldots, 4$ represent the pilots and $\mathcal{H}_{atc}^j, j = 1, \ldots, 4$ represent the air traffic controllers;

- $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ describes the communication network of the hybrid systems.

The next step in the analysis of the ASEP–ITP is the definition of the critical

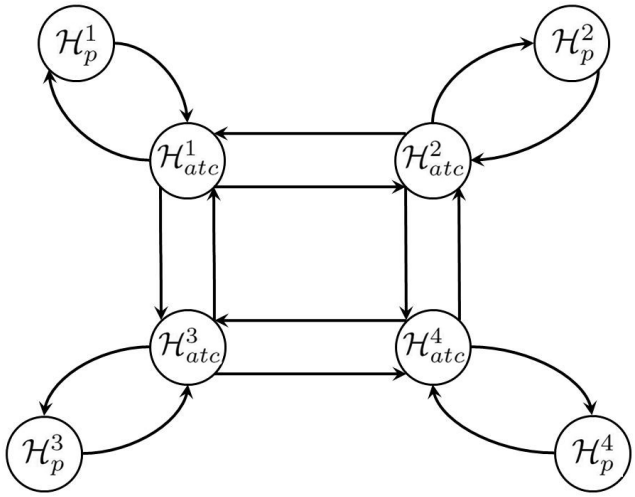



**Figure 5.20:** Communication scheme of 5 agents acting in the ASEP–ITP.

relation $\mathcal{R}$, resulting in:

$$\begin{aligned}\mathcal{R} \;=\; & (\bigcup_{p_i} \mathcal{R}'_{p_i})\cup \\ & (\bigcup_{p_i,p_j,atc_i,atc_j} \mathcal{R}'_{p_i,p_j,atc_i,atc_j})\cup \\ & (\bigcup_{p_i,p_j,p_k,atc_i,atc_j,atc_k} \mathcal{R}'_{p_i,p_j,p_k,atc_i,atc_j,atc_k})\cup \\ & \mathcal{R}'_{p_1,p_2,p_3,p_4,atc_1,atc_2,atc_3,atc_4},\end{aligned}$$

where:

- $\mathcal{R}_{p_i} = \{q^i_{p,8}, q^i_{p,9}, q^i_{p,10}\}$.
- $\mathcal{R}_{p_i,p_j,atc_i,atc_j} = \{q^i_{p,7}, q^j_{p,7}, q^i_{atc,3}, q^j_{atc,3}\}$.
- $\mathcal{R}_{p_i,p_j,p_k,atc_i,atc_j,atc_k} =$ $\{q^i_{p,7}, q^j_{p,7}, q^k_{p,7}, q^i_{atc,3}, q^j_{atc,3}, q^k_{atc,3}\}$.
- $\mathcal{R}_{p_1,p_2,p_3,p_4,atc_1,atc_2,atc_3,atc_4} =$ $\{q^1_{p,7}, q^2_{p,7}, q^3_{p,7}, q^4_{p,7}, q^1_{atc,3}, q^2_{atc,3}, q^3_{atc,3}, q^4_{atc,3}\}$.

Second, third and fourth critical relations model the situation in which the ATC asks at the same time to more than one aircraft to execute the ASEP–ITP and this can result in being safety critical.

**Step 0.** By applying the techniques shown in Section 4.1 a critical observer $\mathcal{O}$ can be constructed to check critical observability of $\mathcal{H}$. However, the cardinality of

the state space of the obtained observer may be intractable from the computational point of view. In fact, the cardinality $|Q|$ of the set $Q$ of discrete states of $\mathcal{H}$ is given by:

$$|Q| = \prod_{i=1,2,\ldots,4} |Q^i_{atc}| \cdot \prod_{i=1,2,\ldots,4} |Q^i_p| = 5^4 \cdot 13^4 \simeq 1.78 \cdot 10^7.$$

Remember from previous sections that the cardinality of the set of discrete states of the critical observer $\mathcal{O}$ for $\mathcal{H}$ grows exponentially with $|Q|$ possibly amounting to $2^{|Q|} \simeq 2^{1.78 \cdot 10^7} \simeq 1.03 \cdot 10^{5358034}$ in the worst case. It is clear that the construction of such an observer can be very demanding from the computational point of view. Thus we approach the analysis of critical observability by using the complexity reduction techniques illustrated in Section 4.3, as follows:

**Step 1.** Since

$$\mathcal{R}'_{p_i,p_j,p_k,atc_i,atc_j,atc_k} \subset \mathcal{R}'_{p_i,p_j,atc_i,atc_j}$$
$$\mathcal{R}'_{p_1,p_2,p_3,p_4,atc_1,atc_2,atc_3,atc_4} \subset \mathcal{R}'_{p_i,p_j,atc_i,atc_j}$$

by applying Proposition 4.3.9, the hybrid system $\mathcal{H}$ is $\mathcal{R}$–critically observable if and only if it is critically observable w.r.t. the critical relation:

$$\mathcal{R} = (\bigcup_{p_i} \mathcal{R}'_{p_i}) \cup (\bigcup_{p_i,p_j,atc_i,atc_j} \mathcal{R}'_{p_i,p_j,atc_i,atc_j}).$$

By applying Theorem 4.3.11 the hybrid system $\mathcal{H}$ is $\mathcal{R}$–critically observable if and only if:

(C1) $\mathcal{H}^i_p$ is $\mathcal{R}_{p_i}$–critically observable.

(C2) $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}_{p_i,p_j,atc_i,atc_j}$–critically observable.

Since $|Q_p| = 13$ and the number of aircraft involved is 4, the computational complexity in checking condition (C1) is $O(4 \cdot 2^{13}) = O(32768)$; regarding condition $(C2)$ the cardinality of $|Q^i_p \times Q^j_p \times Q^i_{atc} \times Q^j_{atc}| = 13^2 \cdot 5^2 = 4225$ and the computational complexity in the construction of the critical observer is therefore given by $O(|2^{Q^i_p \times Q^j_p \times Q^i_{atc} \times Q^j_{atc}}|) \simeq O(2^{4225}) \simeq O(6.4210^{1271})$. Since we have to consider all possible combinations of the agents involved, resulting in 6 combinations, the overall computational complexity in checking condition (C2) yields $O(6.4210^{1271} \cdot 6) \simeq O(3.85 \cdot 10^{1272})$, which added to the computational complexity of condition (C1) finally amounts to $O(4225 + 6.4210^{1271} \cdot 6) \simeq O(3.85 \cdot 10^{1272})$.

**Step 2.** Condition $(C1)$ involves the study of critical observability for each of the 4 agents $\mathcal{H}^i_p$ with respect to their critical relations $\mathcal{R}_{p_i}$. Since the hybrid models $\mathcal{H}^i_p$ coincide one each other and the critical relations $\mathcal{R}_{p_i}$ coincide one each other, it is sufficient to analyze critical observability of only one aircraft. Hence, the computational complexity in checking condition $(C1)$ becomes $O(2^{13}) \simeq O(8192)$.

By using similar arguments, the computational complexity in checking condition $(C2)$ becomes $O(6.42 \cdot 10^{1271})$. The overall computational complexity in checking conditions (C1) and (C2) amounts to $O(8192 + 6.42 \cdot 10^{1271}) \simeq O(6.42 \cdot 10^{1271})$.

**Step 3.** We now proceed with a further step by considering condition $(C2)$. By applying Proposition 4.3.10, $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}_{p_i,p_j,atc_i,atc_j}$–critically observable if and only if $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}_{p_i,atc_i}$–critically observable and $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}_{p_j,atc_j}$–critically observable. The overall computational complexity in checking this condition is $O(2^{13 \cdot 5} \cdot 4) \simeq O(1.47 \cdot 10^{20})$, which added to the computational complexity in checking condition $(C1)$ yields an overall complexity equal to $O(2^{13} + (2^{13 \cdot 5} \cdot 4)) \simeq O(1.47 \cdot 10^{20})$.

**Step 4.** Since hybrid models of $\mathcal{H}_p^i$,$\mathcal{H}_{atc}^i$ and $\mathcal{H}_p^j$,$\mathcal{H}_{atc}^j$ are the same and critical relations $\mathcal{R}_{p_i,atc_i}$ and $\mathcal{R}_{p_j,atc_j}$ are the same we need to only analyze critical observability of $\mathbb{H}(\mathbb{A}_h)$ with respect to $\mathcal{R}_{p_i,atc_i}$. The overall computational complexity in checking this condition is $O(2^{13 \cdot 5}) \simeq O(3.68 \cdot 10^{19})$, which added to the computational complexity in checking condition $(C1)$ yields an overall computational complexity equal to $O(3.68 \cdot 10^{19})$.

**Step 5.** By applying Proposition 4.3.10 $\mathbb{H}(\mathbb{A}_h)$ is $\mathcal{R}_{p_i,atc_i}$–critically observable if and only if $\mathcal{H}_p$ is $\{q_{p,7}\}$–critically observable and $\mathcal{H}_{atc}$ is $\{q_{atc,3}\}$–critically observable. The overall computational complexity in checking this condition is $O(2^{13} + 2^5) \simeq O(8224)$, which added to the computational complexity in checking condition $(C1)$ yields an overall computational complexity equal to $O(8192 + 8224) \simeq O(16416)$.

**Step 6.** Finally the conditions outlined in Step 5 reduce to the following ones:

(C3) $\mathcal{H}_p$ is $\mathcal{R}_p$–critically observable and $\{q_{p,7}\}$–critically observable.

(C4) $\mathcal{H}_{atc}$ is $\{q_{atc,3}\}$–critically observable.

The improvement obtained in Step 6 w.r.t. Step 5 is due to the fact that while checking conditions in Step 5 requires the construction of 3 observers, 2 for the agent pilot and 1 for the agent air traffic controller, checking conditions in Step 6 require the construction of 2 observers, 1 for the agent pilot and 1 for the agent air traffic controller. The overall computational complexity required in checking conditions (C3) and (C4) is $O(2^{13} + 2^5) \simeq O(8224)$. The computational complexity reduction achieved by the procedure shown above is summarized in Table 5.3.

The above procedure reduces the analysis of critical observability of the ASEP–ITP to the analysis of critical observability in conditions (C3) and (C4). We start by considering condition $(C3)$. For doing so we need to construct an observer for $\mathcal{H}_p$. By using the results recalled in Section 3.2 the following observer is obtained:

$$O_p = (\hat{Q}_p, \hat{Q}_{0,p}, \hat{\Sigma}_p, \hat{\Psi}_p, \hat{E}_p, \hat{\eta}_p)$$

where:

| | Computational Complexity |
|---|---|
| Step 0 | $O(1.03 \cdot 10^{5358034})$ |
| Step 1 | $O(3.85 \cdot 10^{1272})$ |
| Step 2 | $O(6.42 \cdot 10^{1271})$ |
| Step 3 | $O(1.47 \cdot 10^{20})$ |
| Step 4 | $O(3.68 \cdot 10^{19})$ |
| Step 5 | $O(16416)$ |
| Step 6 | $O(8224)$ |

**Table 5.3:** Computational complexity reduction analysis.

- $\hat{Q}_p = \{\{q_{p,1}, q_{p,2}, q_{p,3}\}, \{q_{p,4}\}, \{q_{p,5}\}, \{q_{p,6}\}, \{q_{p,7}, q_{p,8}, q_{p,9}\}, \{q_{p,11}\}, \{q_{p,10}, q_{p,12}\}\}$.
- $\hat{Q}_{0,p} = \{q_{p,1}, q_{p,2}, q_{p,3}\}$.
- $\hat{\Sigma}_p = \Psi_{p_i}$.
- $\hat{\Psi}_p = \hat{Q}_{p_i}$.
- $\hat{E}_p$ is depicted in Figure 5.21.
- $\hat{\eta}_p(\hat{q}) = \hat{q}$ for any $\hat{q} \in \hat{Q}_{p_i}$.

We start by checking the first part of condition (C3); the obtained observer $\mathcal{O}_p$ illustrated in Figure 5.21, shows that $\mathcal{H}_p$ is not $\mathcal{R}_p$–critically observable. Indeed when the state of $\mathcal{O}_p$ is in $\{q_{p,7}, q_{p,8}, q_{p,9}\}$ it is not possible to distinguish the critical states $q_{p,8}, q_{p,9}$ from the noncritical state $q_{p,7}$. Analogously when the state of $\mathcal{O}_p$ is in $\{q_{p,10}, q_{p,12}\}$, it is not possible to distinguish the critical state $q_{p,10}$ from the noncritical state $q_{p,12}$.

In order to render the hybrid model $\mathcal{H}_p$, $\mathcal{R}_p$–critically observable, extra discrete–outputs are needed, and can be designed as follows. We define a partial function $h_p : Q_p \to \Psi_p$ that associates to each state $q \in Q_p$ an additional discrete output symbol $h(q) \in \Psi_p$ in order to detect when the execution reaches one of the critical discrete states $q_{p,8}$, $q_{p,9}$ or $q_{p,10}$. The extra output $h(q_{p,8})$ might be generated using an alarm that detects a failure in the surveillance system. The extra output $h(q_{p,9})$ might be generated using measurements of position and velocity of the aircraft. The extra output $h(q_{p,10})$ might be obtained by adding to the procedure a communication from the oceanic controller to the pilot, after the Aircraft Status Report at the next waypoint. The generation of these extra outputs requires a time delay. Construction of critical observers with time delay has been studied in [11].

The observer with delay associated with agent $\mathcal{H}_p$ and critical relation $\mathcal{R}_p$ is illustrated in Figure 5.22. The obtained observer is now critical in the sense that it is possible to detect when the discrete state reaches the set of critical states after the bounded time delay needed for the generation of the extra outputs.

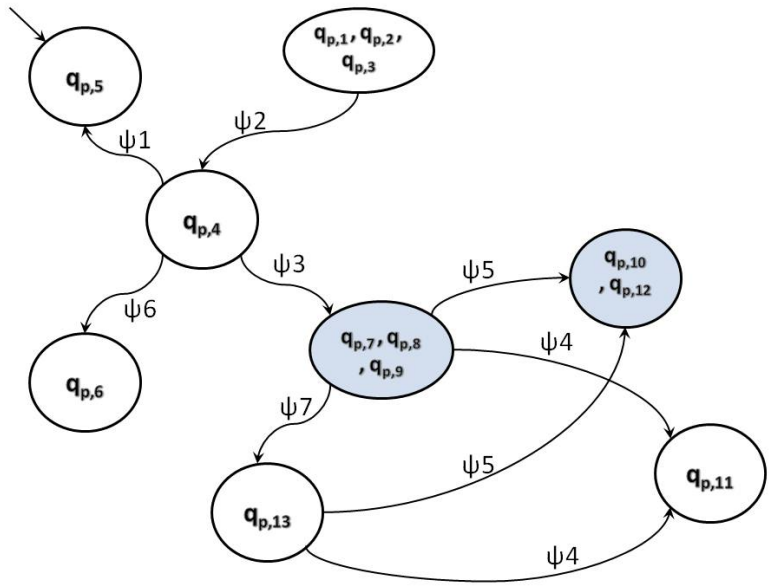



**Figure 5.21:** $\mathcal{R}_p$–critical observer for hybrid system $\mathcal{H}_p$.

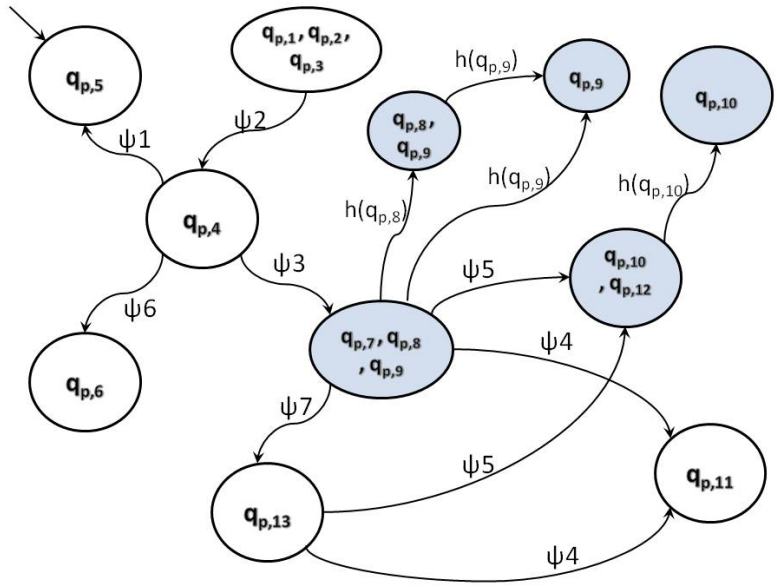



**Figure 5.22:** $\mathcal{R}_p$–critical observer with delay for hybrid system $\mathcal{H}_p$.

We proceed with a further step by checking the second part of condition (C3). The obtained observer $\mathcal{O}_p$ illustrated in Figure 5.23, shows that $\mathcal{H}_p$ is not critically observable with respect to the set of critical states $\{q_{p,7}\}$. Indeed when the state of the critical observer $\mathcal{O}_p$ is in $\{q_{p,7}, q_{p,8}, q_{p,9}\}$ it is not possible to distinguish the critical state $q_{p,7}$ from the noncritical state $q_{p,8}, q_{p,9}$. In order to render the hybrid model $\mathcal{H}_p$ critically observable the extra discrete output $h(q_{p,7})$ is needed to be designed; this can be done by using an alarm generated from ground surveillance systems. The obtained critical observer with delay is depicted in Figure 5.24.

We conclude by checking condition (C4). The following observer is obtained:

$$\mathcal{O}_{atc} = (\hat{Q}_{atc}, \hat{q}_{0,atc}, \hat{\Sigma}_{atc}, \hat{\Psi}_{atc}, \hat{E}_{atc}, \hat{\eta}_{atc}),$$

where:

- $\hat{Q}_{atc} = \{\{q_{atc,1}\}, \{q_{atc,2}, q_{atc,3}\}, \{q_{atc,4}\}, \{q_{atc,5}\}\}$.
- $\hat{q}_{0,atc} = \{q_{atc,1}\}$.
- $\hat{\Sigma}_{atc} = \Psi_{atc}$.
- $\hat{\Psi}_{atc} = \hat{Q}_{atc}$.
- $\hat{E}_{atc}$ is depicted in Figure 5.25.

- $\hat{\eta}_{atc}(\hat{q}) = \hat{q}$, for any $\hat{q} \in \hat{Q}_{atc}$.

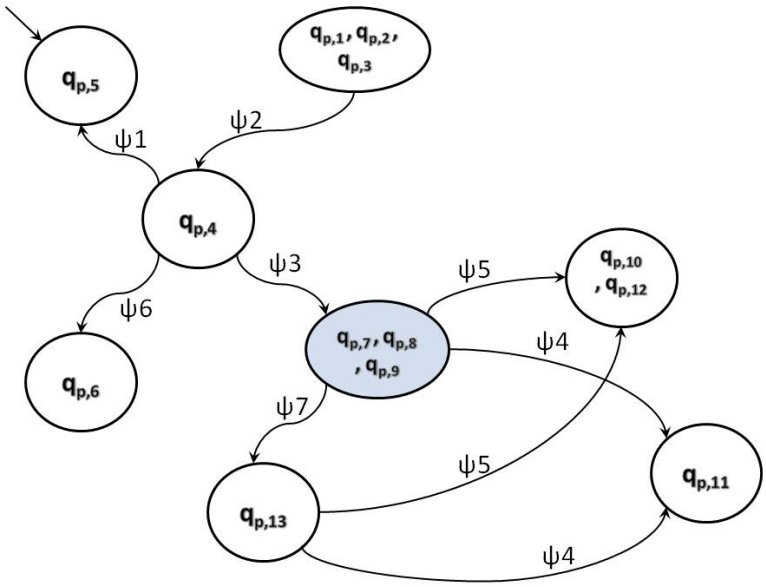



**Figure 5.23:** $\{q_{p,7}\}$–critical observer for $\mathcal{H}_p$

The observer $\mathcal{O}_{atc}$ illustrated in Figure 5.25, shows that $\mathcal{H}_{atc}$ is not critically observable with respect to the set of critical states $\{q_3^{atc}\}$ because it fails in distinguishing between the critical state $q_{atc,3}$ and the noncritical state $q_{atc,2}$. By proceeding as in the previous cases it is possible to render the hybrid system $\mathcal{H}_{atc}$ critically observable with respect to $\{q_{atc,3}\}$ by introducing an extra discrete–output $h(q_{atc,3})$; such extra output can be generated by the technical instrumentations. The obtained critical observer with delay is illustrated in Figure 5.26.

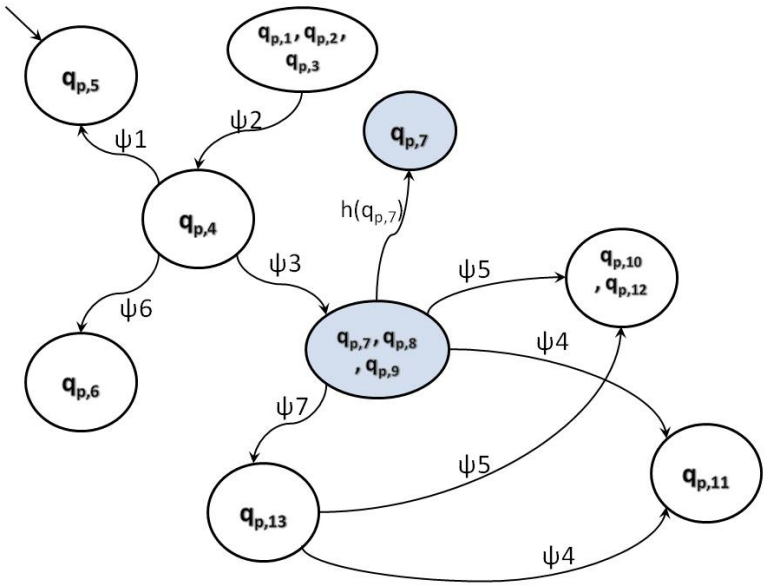



**Figure 5.24:** $\{q_{p,7}\}$–critical observer with delay for $\mathcal{H}_p$.

The analysis that we performed highlights that the ASEP-ITP is not critically observable in the sense that not all unsafe and/or unallowed operations by the agents can be detected. However, provided that additional signals can be generated, as detailed in the above analysis, the procedure can be made critically observable. Although the analysis of the ASEP–ITP has been performed in a scenario with 4 aircraft and 1 air traffic controller, this analysis can be easily extended to a scenario in which an arbitrary large number of agents operate.

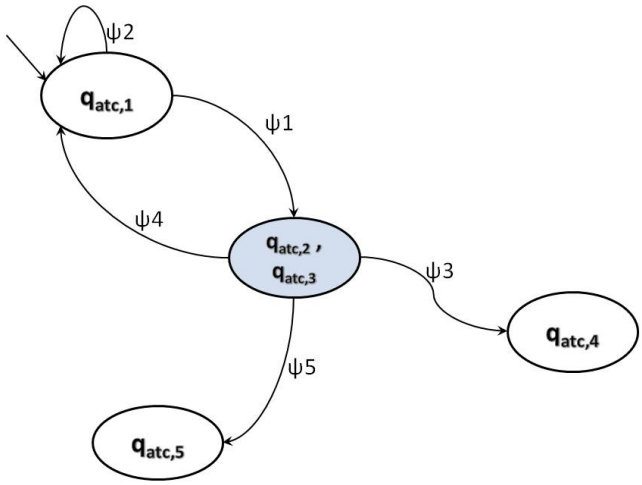



**Figure 5.25:** $\{q_{atc,3}\}$–critical observer for $\mathcal{H}_{atc}$.

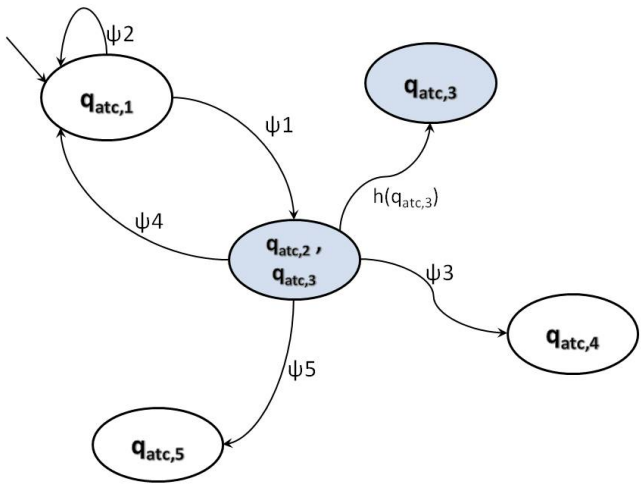



**Figure 5.26:** $\{q_{atc,3}\}$–critical observer with delay for $\mathcal{H}_{atc}$

### 5.2.4 UPPALL model of ASEP-ITP Procedure

We can represent the hybrid model of the ASEP-ITP as a timed automaton, so that we can verify of the properties. It is possible to translate a rectangular automaton into

a timed automaton, obtaining an equivalent system that preserves all the temporal properties of the original system. This is due to the fact that both rectangular automata and timed automata admit a finite bisimulation [53]. This implies that, given any rectangular automaton, it is possible to construct a bisimilar (and thus equivalent) timed automaton. The main issue is translating the dynamics and guards of the rectangular automaton into clocks and guards of the timed automaton. However, since rectangular automata are characterized by very simple dynamics, such computation can be performed in a closed form, as will be illustrated in the next section, in the definition of the timed automaton that models the Pilot Flying of ITP Aircraft. For instance, we stress that the translation from rectangular automata to timed automata also preserves observability properties, that is the rectangular automaton is observable if and only if the timed automaton is observable. The implication is in fact symmetric, because of the equivalence of the two systems.

**Pilot flying of ITP aircraft Agent**

The hybrid system of the pilot can be defined like a timed automaton in the following way:

$$Pilot = (L, l_0, C, A, E, I)$$

where:

- $L$ is the set of following locations:
  - *Cruise*;
  - *ITP Aborted*;
  - *ITP Initiation*;
  - *Wait*;
  - *ITP Instruction*;
  - *ITP Rejected*;
  - *ITP Denied*;
  - *ITP Standard Execution*;
  - *Abnormal Termination*;
  - *NITPC CExe*(Non ITP Criteria compliant execution);
  - *ITP Termination*;
  - *Wrong Execution*;
  - *Exe ASAS Conf*(Execution after ASAS conflict detection);
  - *Wrong Termination.*
- $l_0 = \{Cruise\}$ is the initial location;

- $C = \{t_1, t_2\}$ is the set of clocks and it is used in the guard of some locations. In particular we have that:
  - $t_1 = \frac{z_{fin} - z_{in}}{\dot{z}_{min}} = \frac{40000ft - 36000ft}{300fpm} = 13$ minutes
  - $t_2 = \frac{z_{fin} - z_{in}}{\dot{z}_{max}} = \frac{40000ft - 36000ft}{1000fpm} = 4$ minutes

  supposing that the aircrafts carry out a change of the level of flight passing from 36000 ft to 40000 ft with a minimal speed of 300 fpm and a maximum of 1000 fpm;
- $A$ is a set of actions and co-actions, defined in the Figure 5.27;
- $E$ is a set of edges between locations, defined in Figure 5.27;
- $I = \{x \leq 10\}$ assigns invariants to locations, in particular to ITP Instruction.

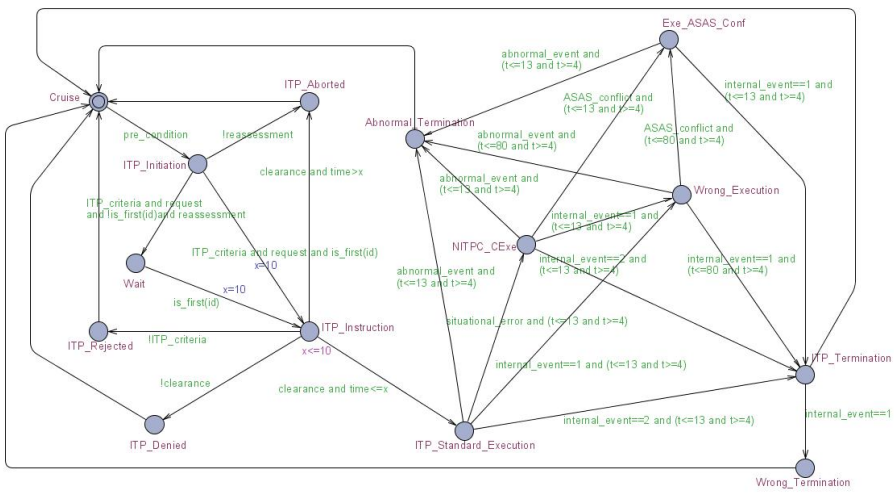



**Figure 5.27:** Model of the pilot under UPPAAL

### Oceanic controller Agent

Regarding the Oceanic controller we have the following timed automaton:

$$Controller = (L, l_0, C, A, E, I)$$

where:

- $L$ is the set of following locations:
  - *Monitoring*;

- *ITP Request*;
- *ITP Clearance Issued*;
- *ITP Clearance Refused*;
- *ITP Clearance Denied*;
- *Wrong Clearance Issued*;
- *Abnormal Termination.*

- $l_0 = \{Monitoring\}$ is the initial location;
- C={y} is the set of clocks;
- $A$ is a set of actions and co-actions, defined from the Figure 8;
- $E$ is a set of edges between locations, defined in Figure 8;
- $I = \{y \leq 10\}$ assigns invariants to locations, in particular to ITP Clearance Issued.

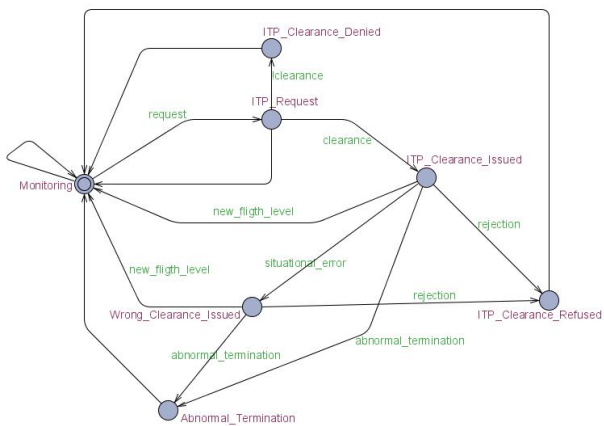



**Figure 5.28:** Model of the controller under UPPAAL

### Scenario: Wrong Execution of ITP with ASAS conflict detection and Wrong Termination

We apply our verification procedure to a specific scenario of the ITP, that models an error in the procedure. Initially, the flight is in its cruise phase and the pilot consider the will to execute an ASEP-ITP. If the pre-conditions are verified we enter in the ITP Initiation phase. In this phase the pilot checks the ITP Criteria. The ITP speed distance criteria are checked using the data $v_{G0}, v_{Gr0}$ and $x_0 - x_{r0}$. Basing

on his situational awareness the crew verifies that all the criteria are satisfied. Thus he communicates the request of an ITP clearance to the controller. The controller also verifies the ITP Criteria and gives to the aircrew the clearance to execute the manoeuvre. The pilot starts the manoeuvre respecting the performances envelope (Standard ITP Execution) but during the execution of the manoeuvre the pilot does not keep the Mach number constant, exceeding the bound of 0.01 Mach error (Wrong Execution). The ASAS technical system is functioning properly so it detects the possibility of a conflict and generates an alert. The technical system also assists the pilot providing information on how to resume the standard execution. In this phase (Execution after ASAS conflict detection) the pilot can temporarily change the vertical speed or the Mach number to solve the possible conflict. The pilot must revert to the initial Mach number after the conflict alarm is solved. We assume that the pilot forgets to revert the speed of the aircraft to the initial Mach number. When the aircraft stabilizes in the requested flight level (ITP termination) the pilot communicates to the ATC the establishment at that flight level. Without having any information the flight is in an hazardous situation. In fact, the immediate verification of a guard condition which checks if the Mach has changed, brings to a (Wrong Execution). The graph of Figure 5.29 enhances this path.

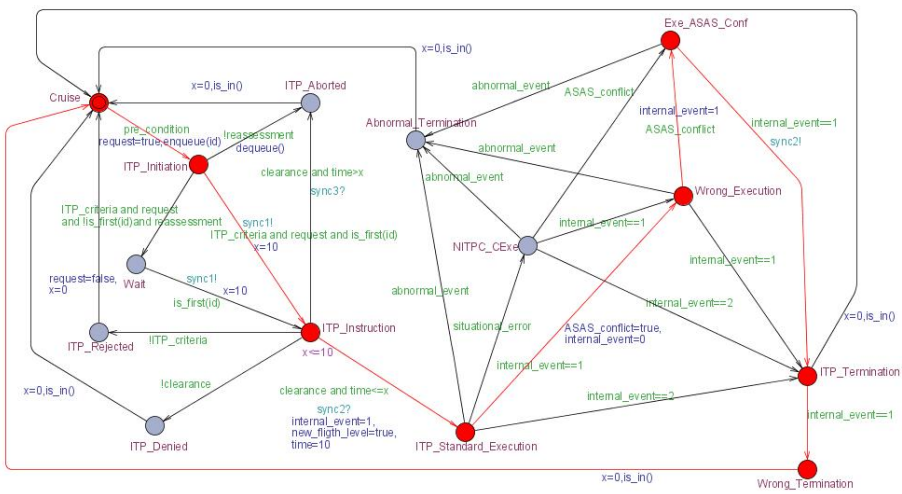



**Figure 5.29:** Wrong Execution of ITP with ASAS conflict detection

Once described this scenario, we can analyze the Reachability properties of the pilot. In order to perform automatic verification with UPPAAL, we need to formalize these properties using the appropriate syntax:

- *E <> ASEP_Pilot.Cruise*: does there exist a path starting from the initial state, such that Cruise is eventually satisfied along that path? **Yes**, the property is satisfied;

- *E[ ] ASEP_Pilot.Cruise*: does there exist a maximal path such that Cruise is always satisfied? **Yes**, the property is satisfied;
- *E <> ASEP_Pilot.ITPInstruction*: does there exist a path starting at the initial state, such that ITP Instruction is eventually satisfied along that path? **Yes**, the property is satisfied;
- *E[ ] ASEP_Pilot.ITPInstruction*: does there exist a maximal path such that ITP Instruction is always satisfied? **No**, the property is not satisfied.

The property *E <> ASEP_Pilot.Cruise* means that the location Cruise can be reached, namely such that the node Cruise will eventually be touched by a path. While the property *E[ ] ASEP_Pilot.Cruise* means that the system can reach a safe state without passing through dangerous situations. This type of property leads to determine whether a safe state can be reached without passing through unsafe states. The same reasoning applies to the other properties for ITP Instruction.

## 5.3 ASAS Lateral Crossing Procedure

In this section we model and analyze the ASAS Lateral Crossing Procedure.

### 5.3.1 Description of the ASAS Lateral Crossing Procedure

The purpose of the ASAS Lateral Crossing procedure is to provide a new set of air traffic control clearances, allowing one aircraft to cross or pass a target aircraft through the use of ASAS. The controller gives the responsibility for the separation to the flight crew of the clearance aircraft with respect to a specific single other aircraft. Except in these limited specific circumstances where the flight crew takes responsibility for separation, ATCo retains all other separation responsibility.

The separation task is delegated to the flight crew in order to support an increase in controller availability, leading to gains in efficiency, and potential capacity within the considered sectors, whilst maintaining or raising current safety levels. The ASAS Lateral Crossing procedure is a procedure in which the qualified flight crew of suitably equipped aircraft maintain safe separation when crossing one aircraft designated by ATCo, in compliance with the separation minima to be applied during the ASAS Lateral Crossing procedure, i.e. Airborne separation minima.

#### Roles and responsibilities

The separation assurance related tasks are delegated to flight crews, upon controller initiative who decides to delegate if appropriate and helpful. The controller delegates separation responsibility to one aircraft and transfers the corresponding separation tasks to the flight crew. The separation responsibility delegated to the flight crew is limited to a unique designated aircraft and is limited in time (duration of the lateral crossing) space (manoeuvre envelope) and scope (maintain separation with target aircraft).

The transfer of responsibility starts as soon as the clearance aircraft has accepted the clearance. The transfer of responsibility back to the controller occurs when the clearance aircraft has passed the clear of traffic (COT) point and the flight crew reports this event to the ground. During the execution of the ASAS Lateral Crossing procedure the flight crew of the clearance aircraft is in charge of maintaining separation from the target aircraft. Throughout the procedure, the air traffic controller remains responsible for maintaining separation between the clearance aircraft and all other aircraft in the sector.

#### Operating principles

**ATCo perspective.** The ASAS Lateral Crossing procedure can only be initiated by the controller. There is no obligation for the controller to use the ASAS Lateral Crossing procedure. The controller should ensure that the target will maintain its track and speed. This could be done by checking the flight plan or by giving an

explicit instruction. It is not foreseen that ATCo will have to specifically inform the flight crew of the target aircraft. Then a manoeuvring envelope is defined by:

- a maximum track alteration; within the ASSTAR project, a maximum value of 45 degrees for track alteration is envisaged;
- a maximum along track distance TKmax, after which the delegation should end and the responsibility for separation would revert to the controller. By default, this distance corresponds to the along track distance between the current position of the clearance aircraft and the crossing point between the target aircraft track and the own aircraft track;
- a maximum cross track deviation, XTKmax (e.g. 8 NM).

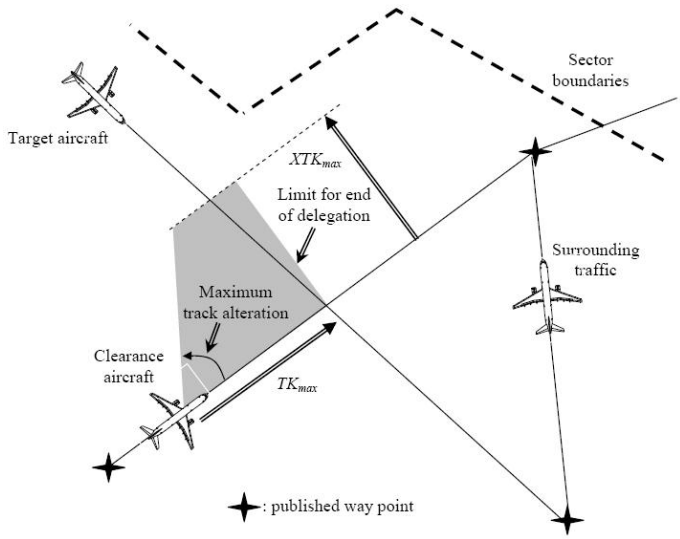



**Figure 5.30:** ASAS Lateral crossing: ATCo perspective.

When the clearance aircraft is clear of traffic (COT), the flight crew reports to the controller and the ASAS Lateral Crossing procedure is completed: the separation task reverts to the controller.

**Airborne perspective**. The flight crew performs the ASAS Lateral Crossing manoeuvre and the corresponding separation task using onboard ASAS functions. Prior to the acceptance of the ASAS Lateral Crossing procedure, positive identification of the target aircraft is required by the clearance aircraft. It is neither envisaged that the ASAS Lateral Crossing separation advisories are directly coupled to the aircraft flight control system without any check by the flight crew.

Indeed, the operational implementation of the ASAS separation advisories is envisioned through a pilot in the loop process. The foreseen implementation sequence is:

- the ASAS algorithms provide ASAS separation advisories on a specific display; this should enable the flight crew to anticipate the duration and the shape of the deviation.
- the flight crew analysis the ASAS separation advisories; ASAS separation advisories such as an offset route or a turning point route will be examined.
- If the flight crew is satisfied with the ASAS separation advisories, then the appropriate manual action will be undertaken by the flight crew to modify the aircraft navigation.

The flight crew will be responsible for reporting information about their navigation change back to the controller. Once the flight crew has determined that the aircraft is clear of traffic, the flight crew reports this to the controller and then resumes its own navigation. The lateral crossing procedure ends when the controller acknowledges the COT report and resumes responsibility for separation. The COT point is computed such that the resuming navigation does not put the clearance aircraft and the target aircraft on converging tracks.

The Clear of Traffic (COT) point with respect to the target aircraft is generated when:

- Target and clearance aircraft are diverging laterally and the current distance between aircraft is equal or greater than the value of the applicable lateral separation.
- The resume manoeuvre anticipated onboard the clearance aircraft will not generate a conflict with the target aircraft.

It is anticipated that in some cases, no deviation from the current navigation may be required. This would result in a better flight efficiency.

**Phases of the ASAS Lateral Crossing Procedure**

The ASAS Lateral Crossing procedure can be divided into the following phases:

- **Phase 1**: set up phase
- **Phase 2**: identification phase
- **Phase 3**: clearance phase
- **Phase 4**: execution phase
- **Phase 5**: termination phase

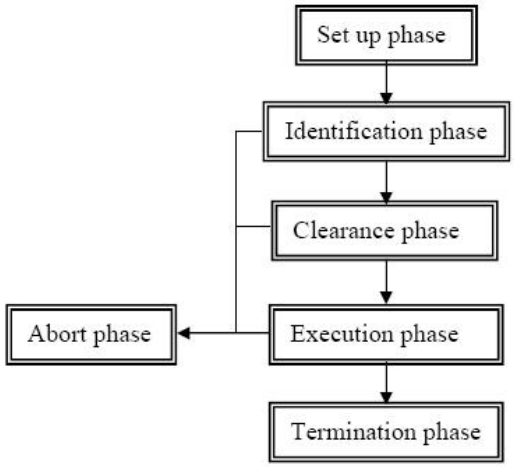



**Figure 5.31:** Phase diagram for ASAS Lateral Crossing procedure.

- **Phase 6**: abort phase

**Set up phase**. During this phase, the controller makes a decision whether to initiate the ASAS Lateral Crossing procedure. The controller checks that the following applicability conditions are satisfied:

- A conflict between the clearance aircraft and the target aircraft is anticipated by the air traffic controller;
- The angle of convergence between initial tracks is between 45 degrees and 135 degrees (that is the ICAO definition of crossing tracks).
- Appropriate ADS-B capabilities for the target aircraft.
- The target aircraft is in steady flight conditions: the controller shall ensure that the target will maintain its track and speed. This could be by checking the flight plan or by giving an explicit instruction.
- Appropriate ASAS lateral crossing capabilities for clearance aircraft.
- ASAS lateral crossing capabilities can only be used when there is sufficient time for the various stages to be performed.
- Confirmation of absence of other conflicting aircraft by checking that the distance from surrounding traffic (other than the target aircraft) to the clearance aircraft is compatible with the lateral crossing envelope manoeuvre (i.e. maximum track alteration, maximum along track distance TKmax, maximum cross track deviation, XTKmax).

If the applicability conditions are not satisfied the controller engages an ATCo based conflict resolution. Otherwise, he may initiate the identification phase. There is no requirement for the ATCo to inform the target aircraft about the set up phase of ASAS Lateral Crossing procedure.

**Identification phase**. The controller nominates a target aircraft to the clearance aircraft using the target aircraft identification. The clearance aircraft confirms reception of the identification message to the controller. Then, the flight crew identifies the target aircraft on the on-board traffic display. Finally, the flight crew communicates the result of the target acquisition process to the controller. If the target aircraft is not positively identified, the controller engages an ATCo based conflict resolution.

**Clearance phase**. The controller passes an ASAS Lateral Crossing clearance. This message includes: clearance aircraft and target aircraft identification and details the specific manoeuvre to be carried out (pass behind or pass in front) No agreement is required from the flight crew of the target aircraft. Nevertheless, it may be required that ATCo instructs the target aircraft to maintain a heading or track so as to ensure that any unexpected manoeuvre of the target aircraft will not thwart the ASAS Lateral Crossing procedure.

The clearance aircraft flight crew initiates the onboard ASAS crossing function (ASAS Logic) according to the received clearance. The ASAS crossing function provides an ASAS separation advisory which consists in a suggestion for new navigation. The suggested new navigation enables the flight crew to anticipate the duration and the shape of the whole lateral crossing manoeuvre. Then the flight crew assesses the feasibility of the ASAS separation advisory.

If the flight crew reports that the ASAS Lateral Crossing manoeuvre is not achievable, the controller engages an ATCo based conflict resolution. Indeed, as far as air traffic controller must maintain separation between the surrounding traffic and both aircraft involved in the procedure, the lateral crossing envelope manoeuvre is a way to give some visibility of the airborne solution to the controller.

If the flight crew feels that the ASAS Lateral Crossing manoeuvre is achievable, he reports to the controller that the execution phase of the ASAS lateral crossing manoeuvre is engaged. Then, the controller monitors the separation between surrounding traffic, but does not monitor separation between the clearance aircraft and the target aircraft.

**Execution phase**. The execution phase deals with the implementation of the ASAS separation advisory and the monitoring of the lateral crossing manoeuvre. It is anticipated that in some cases, the trajectory suggested by the ASAS separation advisory is the same as the current navigation, so that no deviation from the current navigation will occur.

As far as the clear of traffic (COT) point is passed, the clearance aircraft reports to the controller and the termination phase is engaged. In case of inconsistent ASAS

lateral crossing advisory, the clearance aircraft flight crew reports to the controller, who engages the abort phase. The manoeuvre induced by the ASAS lateral crossing advisory should not trigger short term conflict alerts (STCA). Nevertheless if such event occurs (e.g. the flight crew does not precisely follow the ASAS lateral crossing advisory), the procedure is immediately aborted by the air traffic controller.

**Termination phase**. Once the flight crew has determined that the aircraft is clear of traffic, the flight crew reports this to the controller and then resumes its own navigation. The lateral crossing procedure ends when the controller acknowledges the COT report and resumes responsibility for separation.

**Abort phase**. If the flight crew of the clearance aircraft becomes unable to maintain separation with the target aircraft, he must report to the air traffic controller, and a contingency procedure is used. The contingency procedure will in particular address the conditions under which the separation management task could be reverted to the controller. The controller may also initiate the termination of the ASAS Lateral Crossing procedure at any of the stages. In that case, the separation management task reverts immediately to the controller.

### 5.3.2 Analysis of critical observability of ASAS Lateral Crossing

The Lateral Crossing Procedure is characterized by the following agents:

- Clearance Aircraft
- Reference Aircraft
- Air Traffic Controller

We do not provide the model of the reference aircraft because the flight crew of the reference aircraft does not have the awareness of existence of a lateral crossing manoeuvre in which it is involved.

The hybrid model of the clearance aircraft is given by:

$$\mathcal{H}_p = (Q_p \times X_p, Q_{p,0} \times X_{p,0}, U_p \times Y_p, \mathcal{E}_p, \Sigma_p, E_p, \Psi_p, \eta_p)$$

where:

- $Q_p = \{q_i, i = 1, 2, ..., 15\}$ is the set of discrete states as detailed in Figure 5.32, where
  - $q_1$ is the Navigation state;
  - $q_2$ is the Identification state;
  - $q_3$ is the Instruction state;
  - $q_4$ is the Execution state;

- $q_5$ is the Termination state;
- $q_6$ is the Identification phase after conflict detection state;
- $q_7$ is the Instruction phase after conflict detection state;
- $q_{81}$ is the Wrong execution from the pilot for manoeuvre generated correctly state;
- $q_{82}$ is the Wrong execution for manoeuvre generated in wrong way from system ASAS state;
- $q_{83}$ is the Total or partial loss of onboard information state;
- $q_{84}$ is the Wrong execution for unexpected behavior of the target state;
- $q_{85}$ is the Wrong instructions from the controller state;
- $q_9$ is the Aborted manoeuvre state;
- $q_{10}$ is the Wrong termination state;
- $q_{11}$ is the Aborted procedure state.

- $X_p \subset \mathbb{R}^6$ is the continuous state space with $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in X_p$, where $x_1 = X$ and $x_2 = Y$ indicate the horizontal position, $x_3 = h$ is the altitude, $x_4 = V$ is the true airspeed, $x_5 = \psi$ is the heading angle, $x_6 = \gamma$ is the flight path angle.
- $q_{p,0} = \{q_1\}$ and $X_{p,0} = \{(x_{1_0}, x_{2_0}, x_{3_0}, x_{4_0}, x_{5_0}, x_{6_0})\}$ is the set of initial states.
- $U_p \subset \mathbb{R}^3$ with $u = (u_1, u_2, u_3) \in U_p$, where $u_1 = T$ is the engine trust, $u_2 = \phi$ is the bank angle, $u_3 = \gamma$ is the flight path angle.
- $Y_p = X_p$.
- $\{\mathcal{E}_{p,q}\}_{q \in Q}$ associates to each discrete state $q \in Q$ the continuous dynamics $\dot{x} = f_q(x)$ and $y = x$, where $f_q(x)$ is given[3] by:

$$f_{q_i}(x) = \begin{cases} \dot{X} = V\cos(\psi)\cos(\gamma) \\ \dot{Y} = V\sin(\psi)\cos(\gamma) \\ \dot{h} = V\sin(\alpha) \\ \dot{V} = \frac{1}{m}\left[T\cos(\alpha) - D - mg\sin(\gamma)\right] \\ \dot{\psi} = \frac{1}{mV}\left[L\sin(\phi) + T\sin(\alpha)\sin(\phi)\right] \\ \dot{\gamma} = \frac{1}{mV}\left[(L + T\sin(\alpha))\cos(\phi) - mg\cos(\gamma))\right] \end{cases}$$

  for each $i = 1, 2, ..., 15$, where $L$ is the lift force, $D$ the drag force, $\alpha$ the angle of attack, $g$ gravitational acceleration.
- $\Sigma_p = \{\sigma_i, i = 1, 2, ..., 14\} \cup \{\varepsilon\}$ is the set of discrete inputs, where:

[3]The proposed model has been taken from [25].

- $\sigma_1$ represents the communication from the controller of target selected for the procedure to execute;
- $\sigma_2$ represents the communication from the controller of target correctly identified;
- $\sigma_3$ represents the acknowledgement of feasible manoeuvre;
- $\sigma_4$ represents the acknowledgement of COT point passed;
- $\sigma_5$ represents the acknowledgement from the controller that he has received communication on the COT passed point;
- $\sigma_6$ represents the target not identified onboard (conflict detection);
- $\sigma_7$ represents the order from the controller to abort the procedure due to uncorrect identification of the target;
- $\sigma_8$ represents the communication from the controller of target not correctly identified;
- $\sigma_9$ indicates that the manoeuvre cannot be executed (conflict detection);
- $\sigma_{10}$ represents the order from the controller to undertake the procedure of back–up for wrong execution;
- $\sigma_{11}$ represents the order from the controller to undertake the procedure of back–up for dangerous situation;
- $\sigma_{12}$ represents the order from the controller to undertake the procedure of back–up for loss of onboard information;
- $\sigma_{13}$ represents the order from the controller to undertake the procedure of back–up for unexpected behavior of the target;
- $\sigma_{14}$ represents the order from the controller to undertake the procedure of back–up due to wrong orders sent by the controller.

- $E_p$ is the set of transitions as shown in Figure 5.32.
- $\Psi_p = \{\psi_i, i = 1, 2, ..., 11\} \cup \{\varepsilon\}$ is the set of discrete outputs, where:
  - $\psi_1$ represents the communication to the controller of the possibility to execute the manoeuvre;
  - $\psi_2$ represents the communication to the controller that the CTO point was passed;
  - $\psi_3$ represents the communication to the controller to abort the procedure;
  - $\psi_4$ represents the communication to the controller of conflict detection (target not identified);
  - $\psi_5$ represents the communication to the controller of conflict detection (not feasible manoeuvre);
  - $\psi_6$ represents the communication to the controller to abort the procedure for not feasible manoeuvre;

- $\psi_7$ represents the message of confirmation to the controller of received order to undertake the procedure of back–up for wrong execution;
- $\psi_8$ represents message of confirmation to the controller of received order to undertake the procedure of back–up for dangerous situation;
- $\psi_9$ represents the message of confirmation to the controller of received order to undertake the procedure of back–up for loss of onboard information;
- $\psi_{10}$ represents the message of confirmation to the controller of received order to undertake the procedure of back–up for unexpected behavior of the target;
- $\psi_{11}$ represents the message of confirmation to the controller of received order to undertake the procedure of back–up for wrong orders.

- $\eta_p$ is the output function as shown in Figure 5.32.

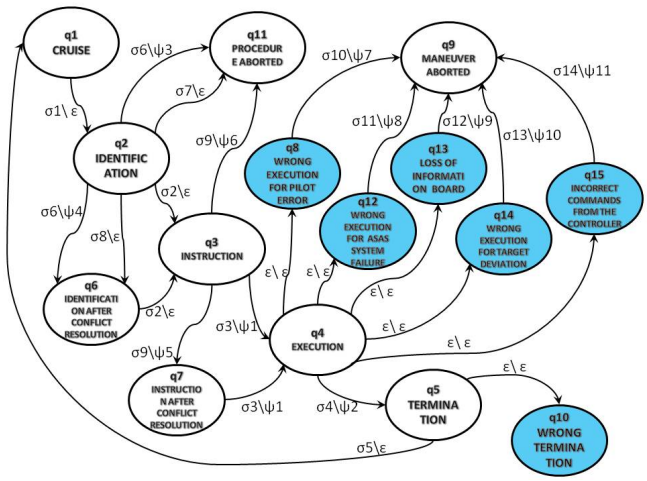



**Figure 5.32:** Hybrid system of the clearance aircraft.

For the air traffic controller we have the following model:

$$\mathcal{H}_{atc} = (Q_{atc} \times X_{atc}, q_{atc,0} \times X_{atc,0}, U_{atc} \times Y_{atc}, \mathcal{E}_{atc}, \Sigma_{atc}, E_{atc}, \Psi_{atc}, \eta_{atc})$$

where:

- $Q_{atc} = \{q_i, ..., i = 1, 2, ..., 9\}$ is the set of discrete state, as detailed in Figure 5.33, where:
  - $q_1$ is the Monitoring state;
  - $q_2$ is the Set-up state;

- $q_3$ is the Identification state;
- $q_4$ is the Instruction state;
- $q_5$ is the Execution state;
- $q_6$ is the Termination state;
- $q_7$ is the Conflict resolution state;
- $q_9$ is the Manoeuvre performed to incorrect command sent by the controller state;
- $q_{10}$ is the Operation aborted state;
- $q_{11}$ is the Procedure aborted state.

and $X_{atc} = \varnothing$.

- $q_{atc,0} = \{q_1\}$ and $X_{atc,0} = \varnothing$.
- $U_{atc} = \varnothing$ and $Y_{atc} = \varnothing$.
- $\mathcal{E}_{atc} = \varnothing$.
- $\Sigma_{atc} = \{\sigma_i, i = 1, 2, ..., 16\} \cup \{\varepsilon\}$ is the set of discrete inputs, where:
  - $\sigma_1$ represents the decision to undertake the LC(lateral crossing) procedure;
  - $\sigma_2$ represents the acknowledgement of satisfied conditions for the procedure to start;
  - $\sigma_3$ represents the target aircraft correctly identified;
  - $\sigma_4$ represents the communication from the clearance aircraft of executable manoeuvre;
  - $\sigma_5$ represents the communication from the clearance aircraft of COT point passed;
  - $\sigma_6$ represents the resumption of the responsibilities for the control of the separation;
  - $\sigma_7$ represents the conflict detection (conditions for the applicability of the procedure are not satisfied);
  - $\sigma_8$ represents the conflict resolved in phase of set up;
  - $\sigma_9$ represents the communication from the clearance aircraft of unidentified target on board (conflict detection);
  - $\sigma_{10}$ represents the communication from the clearance aircraft of decision to undertake the procedure of back up for an unidentified target on board;
  - $\sigma_{11}$ represents the target aircraft not correctly identified;
  - $\sigma_{12}$ represents the communication from the clearance of not executable instruction (conflict detection);

  - $\sigma_{13}$ represents the communication from the clearance aircraft of decision to undertake the procedure of back–up for not executable manoeuvre;
  - $\sigma_{14}$ represents the communication from the clearance aircraft of decision to undertake the procedure of back–up for dangerous situation;
  - $\sigma_{15}$ represents the conflict resolved in identification phase;
  - $\sigma_{16}$ represents the conflict resolved instruction phase.

- $E_{atc}$ is the set of transitions as shown in Figure 5.33.
- $\Psi_{atc} = \{\psi_i, i = 1, 2, ..., 5\} \cup \{\varepsilon\}$ is the set of discrete outputs, where:
  - $\psi_1$ represents the communication to the clearance aircraft of target aircraft candidate to the manoeuvre;
  - $\psi_2$ the communication to the clearance of target aircraft correctly identified;
  - $\psi_3$ the confirmation to the clearance aircraft of reception of the message of CTO passed;
  - $\psi_4$ the communication to the clearance of target not correctly identified;
  - $\psi_5$ the order for the clearance aircraft of execution of the procedure of back–up for target not correctly identified.
- $\eta_{atc} : E_{atc} \to \Psi_{atc}$ is the discrete output function as shown in Figure 5.33.

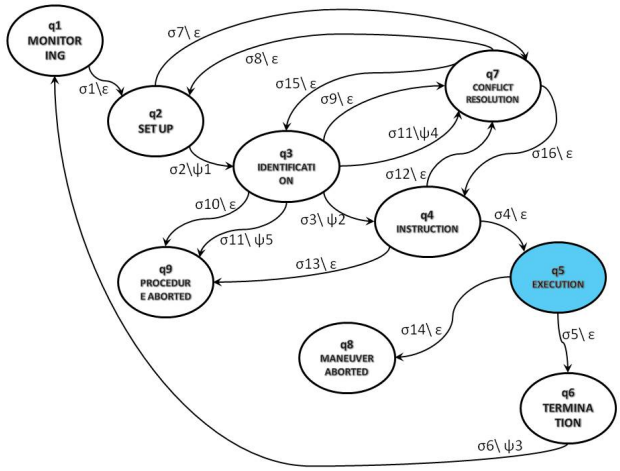



**Figure 5.33:** Hybrid system of the air traffic controller.

Consider a scenario in which $N$ clearance aircraft $\mathcal{H}_p^1, \mathcal{H}_p^2, ..., \mathcal{H}_p^N$ and one ATC $\mathcal{H}_{atc}$ operate. We can apply the same methodology, seen in the previous sections, to analyze the critical observability of this procedure. This analysis highlights that the Lateral Crossing Procedure is not critically observable in the sense that not all unsafe and/or unallowed operations by the agents can be detected. However, provided that additional signals can be generated, as detailed in the above analysis, the procedure can be made critically observable. We stress that the above analysis has been carried out for a scenario in which an arbitrary large number of agents operate.

## 5.4 Autonomous Aircraft Advanced ($A^3$) ConOps

In this section we model and analyze a scenario in the Autonomous Aircraft Advanced ($A^3$) ConOps framework.

### 5.4.1 Description of the $A^3$ ConOps scenario

Autonomous Aircraft Advanced ($A^3$) Concepts of Operations (ConOps) is a description of a future airborne self separation operation in the enroute phase of flight. The flight crews of this aircrafts will be able to ensure separation from surrounding traffic and other obstacles, without the assistance of Air Traffic Controller. The $A^3$ airspace, see Figure 5.34, is divided into 3 classes:

- **Managed Airspace**: high density zones like TMA Areas.
- **Unmanaged Airspace**: all airspace where ATC services cannot be provided.
- **Self Separating Airspace**: all airspace whose boundaries are defined in time and space by the dynamic allocation of Managed and Unmanaged airspace.

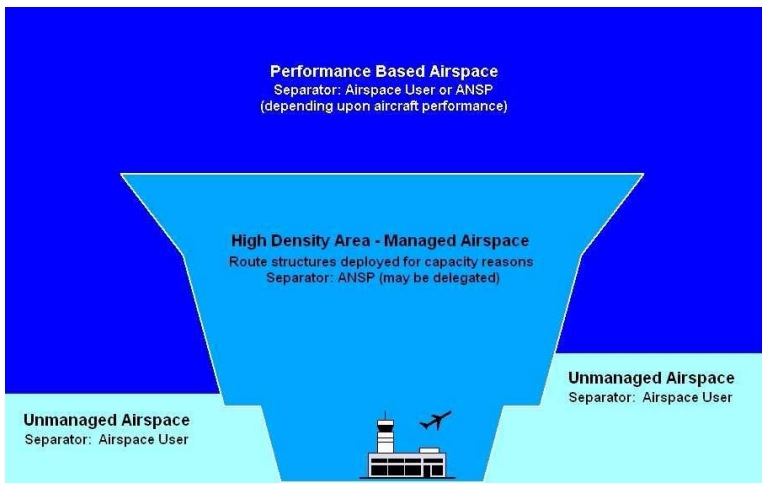



**Figure 5.34:** $A^3$ ConOps Airspace.

Note that in Managed Airspace all flights are subject to ATC clearances, while in Self Separating Airspace aircraft are responsible for separation, in accordance with Autonomous Flight Rules. In this zone aircraft are allowed to take whatever climbing/descent profile they may prefer, with the only limitations being the requirement of self separation, and the safety and comfort of the manoeuvres. The scope of the $A^3$ ConOps focuses on en-route operations in which all aircraft are self-separating, without the participation of the ATC. An $A^3$ flight is defined as the

flight between a departing TMA exit point and an arriving TMA entry point. Along this flight, the aircraft follows its route while maintaining separation from all other aircraft and other conflict elements.

### Autonomous Flight Operations

The Autonomous Flight Operations are operations in which the flight crew has responsibility for self separation and is required to operate according to specific autonomous flight rules. In this context it is possible to consider the following operations:

- **Normal Operations**: all equipment is functioning nominally and the flight crew is able to perform their ATM functions as required.
- **Non-normal Operations**: there is a degradation in any, several or all on-board equipment performance, flight crew performance, SWIM network performance, aircraft performance, but self separation operations can be maintained.
- **Emergency Operations**: there is a degradation in any, several or all on-board equipment performance, flight crew performance, SWIM network performance, aircraft performance, but the continuation of operations under the $A^3$ ConOps is not allowed.

The flight crew is responsible for the safe, efficient and on-time operation of the flight, and it is responsible for separation with all other aircraft. Note that a conflict does not imply that loss of separation has already occurred but it implies that a loss of separation will probably occur if no action is taken.

### Autonomous Flight Rules

The aircraft that operate in Self Separating Airspace must observe some rules. We report from [23] such rules:

- Autonomous aircraft are responsible for maintaining separation with all other aircraft.
- Autonomous aircraft are required to maintain separation from designated areas and no-fly zones.
- Autonomous aircraft are required to adhere to flow management constraints. Renegotiation will have to take place if these constraints can not be met.
- Lower priority autonomous aircraft involved in a medium term Intent based conflict ruled by priority are required to manoeuvre to solve it sufficiently in advance, so that the conflict does not continue until the conflict resolution becomes a short term cooperative conflict.

- Autonomous aircraft shall not manoeuvre in a way that creates a short term (3 to 5 minutes) cooperative conflict.
- The trajectory of autonomous aircraft shall at no time place the aircraft in a 2 minutes state vector conflict (blunder protection).
- Autonomous aircraft shall not enter Managed Airspace without the approval of the controlling entity of that airspace.

One of the difficulties of an autonomous aircraft concept is the limited availability of information about the surrounding traffic. Then in $A^3$ ConOps each aircraft will provide this kind of information:

- **State data**: current position, velocity vector, priority level, etc. broadcasted independently through data link, like ADS-B, an on-board instrumentation.
- **Intent data**: trajectory change and conformance monitoring data broadcasted through data link and also provided to SWIM.
- **Reference Business Trajectory**: the own route of aircraft is not used by other airborne systems.

All this information is required for the realization of the concepts of $A^3$ ConOps environment.

Intent related non–nominal conditions identified in D7.1b [19] (some of which caused by situation awareness inconsistencies) are considered in the modeling of the scenario. In this scenario two aircraft operate and their RBTs may intersect; in this case a conflict occurs and procedures for conflict resolution have to be designed. In particular, we focus on a mid–term conflict. Then a conflict resolution procedure is engaged and based on priority rules associated with the aircraft. Priorities are assigned to each aircraft so that when a conflict takes place, the aircraft with lower priority has to solve the conflict by generating a closed manoeuvre, i.e. a conflict solution provided in the form of a consistent RBT update; we recall also that closed manoeuvres contrast open manoeuvres which solve a detected conflict situation but a consistent continuation of the flight after the maneuver is not considered. In the following we describe the scenario from a single aircraft perspective. This scenario covers the situation when own aircraft is flying its RBT and a mid-term conflict is detected, i.e., the RTTL (Remaining Time To Loss of separation) for closed solution is more than STT (Short Term time Threshold). The conflict is assumed to be solved through a closed manoeuvre. Actions taken by own aircraft depend on the priorities of the aircraft involved. If own aircraft has higher priority than the other aircraft then own aircraft continues flying its RBT. The only action required on own aircraft is an enhanced monitoring of the conflicting aircraft. The other aircraft is requested instead, to solve the conflict. If the other aircraft starts to broadcast and fly a new trajectory, which does not cause other conflicts, no further actions are required on own aircraft. If TTL<STT and the other aircraft still has not

| N. | Description |
|---|---|
| 1 | Own a/c intent is not conflict free and nobody is aware |
| 2 | Another a/c intent is not conflict free and nobody is aware |
| 3 | Another a/c intent intentionally not conflict free; others are not aware |
| 4 | Own a/c intent intentionally is not conflict free; others are not aware |
| 5 | Intent of ownship aircraft not broadcasted |
| 6 | Intent of one other aircraft not received |
| 7 | New intents of multiple a/c not received and crew does not know |
| 8 | Own crew has SA difference for another a/c |
| 9 | Ownship state/intent is not properly perceived by encountering crew |
| 10 | Intent exchange does not work well and nobody is aware |

**Table 5.4:** Intent related non–nominal conditions identified in D7.1b.

broadcasted information on the resolution of the conflict, own aircraft is requested to solve the conflict through an open manoeuvre. An open manoeuvre solves the conflict situation but does not guarantee a consistent continuation of the flight. The system of the other aircraft which is requested to solve the conflict situation should suggest several possible solutions. The flight crew may select one solution and approve it or may require modifications or even suggest its own solution. As soon as the flight crew accepts one of the solutions and executes the manoeuvre, the new intent is broadcasted.
The correct working of this procedure relies upon many factors, one of which is a correct situation awareness of the agents involved. In particular, each agent needs to have a correct awareness of its situation and of the surrounding agents situations, as well. However, in many cases agents lack in having such correct situation awareness. Deliverable 7.1b [19] identified ten intent related (non-nominal) conditions, eight of which are caused by situation awareness inconsistencies of the agents involved. Table 5.4, taken from Deliverable 7.1b summarizes such conditions.

### 5.4.2 Analysis of critical observability of $A^3$ ConOps

By using hybrid systems, we can model aircraft and pilots operating in the $A^3$ scenario. The resulting model is described by the tuple:

$$\mathcal{H}_p = (Q_p \times X_p, Q_{p,0} \times X_{p,0}, U_p, Y_p, \mathcal{E}_p, \Sigma_p, E_p, \Psi_p, \eta_p),$$

where:

- $Q_p = \{q_i, i = 1, 2, ..., 16\}$ is the set of discrete states, where:

  $q_1$ is the Regular flight state.

  $q_2$ is the Monitoring state.

$q_3$ is the Conflict detection state.

$q_4$ is the Enhanced monitoring state.

$q_5$ is the No conflict free RBT state, due to condition TTL $<$ STT.

$q_6$ is the Generation of open manoeuvre state.

$q_7$ is the Execution state.

$q_8$ is the Generation of new RBT state.

$q_9$ is the Generation of a closed manoeuvre state.

$q_{10}$ is the Situation in which the solution is accepted. state

$q_{11}$ is the Situation in which no solution is accepted state.

$q_{12}$ is the Solution given by flight crew state.

$q_{13}$ is the Uploading of new trajectories state.

$q_{14}$ is the Situation in which the solution is modified state.

$q_{15}$ is the Situation in which there is no detection of conflict state.

$q_{16}$ is the Detection of a non–existent conflict state.

- $X_p \subset \mathbb{R}^6$ is the continuous state space with

$$x = (x_1, x_2, x_3, x_4, x_5, x_6) \in X_p,$$

where:

- $x_1 = X$ and $x_2 = Y$ indicate the horizontal position.
- $x_3 = h$ is the altitude.
- $x_4 = V$ is the true airspeed.
- $x_5 = \psi$ is the heading angle.
- $x_6 = \gamma$ is the flight path angle.

- $Q_{p,0} = \{q_1\}$ and $X_{p,0} = \{(x_1^0, x_2^0, x_3^0, x_4^0, x_5^0, x_6^0)\}$ is the set of initial states.
- $U_p \subset \mathbb{R}^3$ with $u = (u_1, u_2, u_3) \in U_p$, where:

- $u_1 = T$ is the engine thrust.
- $u_2 = \phi$ is the bank angle.
- $u_3 = \gamma$ is the flight path angle.

- $Y_p = X_p$.

- $\{\mathcal{E}_p(q)\}_{q\in Q}$ associates to each discrete state $q \in Q$ the continuous dynamics $\dot{x} = f_{q_i}(x)$ and $y = x$, where $f_{q_i}(x)$ is given by[4]:

$$\begin{cases} \dot{X} = V\cos(\psi)\cos(\gamma) \\ \dot{Y} = V\sin(\psi)\cos(\gamma) \\ \dot{h} = V\sin(\alpha) \\ \dot{V} = \frac{1}{m}\left[T\cos(\alpha) - D - mg\sin(\gamma)\right] \\ \dot{\psi} = \frac{1}{mV}\left[L\sin(\phi) + T\sin(\alpha)\sin(\phi)\right] \\ \dot{\gamma} = \frac{1}{mV}\left[(L + T\sin(\alpha))\cos(\phi) - mg\cos(\gamma))\right] \end{cases}$$

  for each $i = 1, 2, ..., 16$, where $L$ is the lift force, $D$ the drag force, $\alpha$ the angle of attack, $g$ gravitational acceleration.

- $\Sigma_p = \{\sigma_i, i = 1, 2, ..., 27\} \cup \{\varepsilon\}$ is the set of discrete inputs, where:

  - $\sigma_1$ represents the monitoring phase.
  - $\sigma_2$ represents the detection of a medium term conflict with another aircraft.
  - $\sigma_3$ represents the monitoring phase.
  - $\sigma_4$ represents an open manoeuvre initiation for the resolution of a ST conflict.
  - $\sigma_5$ represents the execution of an open manoeuvre for the resolution of a ST conflict.
  - $\sigma_6$ represents the RBT update.
  - $\sigma_7$ represents the normal cruise.
  - $\sigma_8$ represents the generation of a closed manoeuvre with lower priority.
  - $\sigma_9$ represents the initiation of analysis and the refusal of the CR solution.
  - $\sigma_{10}$ represents the generation of a new CR solution.
  - $\sigma_{11}$ represents the initiation of an open manoeuvre for the resolution of a ST conflict.
  - $\sigma_{12}$ represents the generation of a new CR solution.
  - $\sigma_{13}$ represents the analysis and the acceptance of the CR solution.
  - $\sigma_{14}$ represents the change in the CR solution.
  - $\sigma_{15}$ represents the acceptance of the CR solution.
  - $\sigma_{16}$ represents the execution of a closed manoeuvre.
  - $\sigma_{17}$ represents the conflict not solved by the other aircraft.

[4]The proposed model has been taken from [25].

- $\sigma_{18}$ represents the detection of a medium term conflict from the other aircraft.
- $\sigma_{19}$ represents the conflict solved by the other aircraft.
- $\sigma_{20}$ represents the conflict not solved by the other aircraft.
- $\sigma_{21}$ represents the execution of an open manoeuvre from the other aircraft.
- $\sigma_{22}$ represents a non–existent medium term conflict.
- $\sigma_{23}$ represents not received data.
- $\sigma_{24}$ represents the continuation of monitoring.
- $\sigma_{25}$ represents the non–reception of data by the other aircraft.
- $\sigma_{26}$ represents a problem on on–board system.
- $\sigma_{27}$ represents intent not received.

- $E_p$ is the set of transitions as shown in Figure 5.35.
- $\Psi_p = \{\Psi_i, i = 1, 2, ..., 12\} \cup \{\varepsilon\}$ is the set of discrete outputs, where:
  - $\Psi_1$ represents information on surrounding traffic.
  - $\Psi_2$ represents the presence of a medium term conflict with the other aircraft.
  - $\Psi_3$ represents the highest priority in the CR.
  - $\Psi_4$ represents the continuation of monitoring.
  - $\Psi_5$ represents the start of an open manoeuvre.
  - $\Psi_6$ represents the execution of an open manoeuvre to solve a medium term conflict.
  - $\Psi_7$ represents the update of RBT.
  - $\Psi_8$ represents the resolution of the conflict.
  - $\Psi_9$ represents the broadcast of information to the other aircraft regarding the existence of a conflict.
  - $\Psi_{10}$ represents the broadcast of a new RBT and the return to regular flight.
  - $\Psi_{11}$ represents the conflict resolution.
  - $\Psi_{12}$ represents the order to the other aircraft to quit the CR through a closed manoeuvre.
- $\eta_p$ is the output function as shown in Figure 5.35.

The above hybrid system, also depicted in Figure 5.35, correctly models aircraft behaviors in nominal condition of operation. However, if non–nominal conditions **C1**–**C10** take place, such model is not adequate and needs to be generalized. Such generalization can be done by adding suitable discrete states to the hybrid model, as follows:

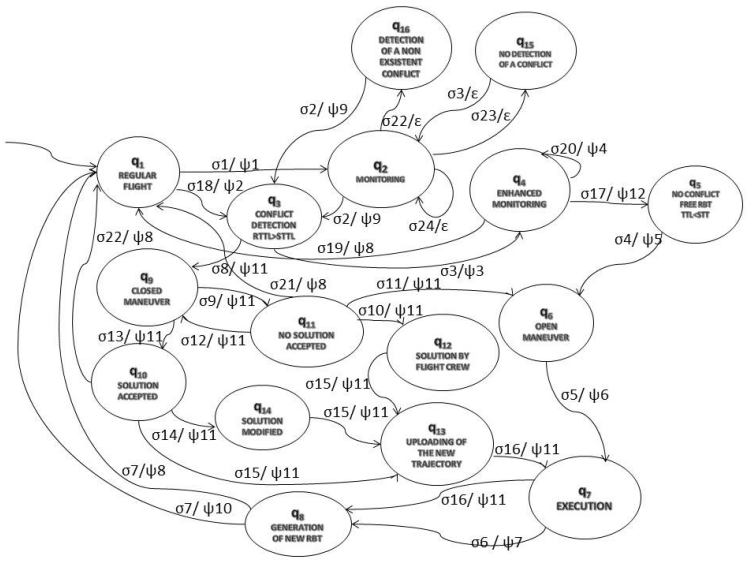



**Figure 5.35:** Hybrid system of the aircraft in nominal conditions.

- $q_{17}$ represents the situation of no detection of CR, due to onboard system failure.
- $q_{18}$ represents the situation of no detection of CR, due to lack of transmission.
- $q_{19}$ represents the situation of no detection of a conflict, due to onboard system failure.
- $q_{20}$ represents the situation of no detection of a conflict, due to lack of transmission.
- $q_{21}$ represents the situation of data not broadcasted.
- $q_{22}$ represents the situation of intent not received.

In the sequel we refer to these states as *critical states*, because they can lead to unsafe or even catastrophic events. The resulting hybrid system is depicted in Figure 5.36. The aforementioned critical states correctly model non–nominal conditions **C1**–**C10**, as discussed hereafter:

**C1.** This condition is modeled by means of two or more hybrid systems $\mathcal{H}_p^i$ in which one hybrid system, say $\mathcal{H}_p^1$, is in state $q_{15}$ and the remaining ones are either in state $q_{19}$ or in state $q_{20}$.

**C2.** This condition can be modeled by following the same reasoning as in the previous situation.

**C3.** This condition is modeled by means of two or more hybrid systems $\mathcal{H}_p^i$ in which one hybrid system, say $\mathcal{H}_p^1$, is in state $q_3$ and the remaining ones are either in state $q_{19}$ or in state $q_{20}$.

**C4.** This condition can be modeled by following the same reasoning as in the previous situation.

**C5.** This condition is modeled by means of state $q_{21}$ in the hybrid system $\mathcal{H}_p$.

**C6.** This condition can be modeled by following the same reasoning as in the previous situation.

**C7.** This condition is modeled by means of state $q_{22}$ in the hybrid system $\mathcal{H}_p$.

**C8.** The specialization of this condition to non proper detection of conflict situations has been modeled as in case C3.

**C9.** This condition is modeled by means of states $q_{17}, q_{18}, q_{19}, q_{20}$ and $q_{22}$ in the hybrid system $\mathcal{H}_p$.

**C10.** This condition is modeled by means of state $q_{22}$ in the hybrid system $\mathcal{H}_p$.

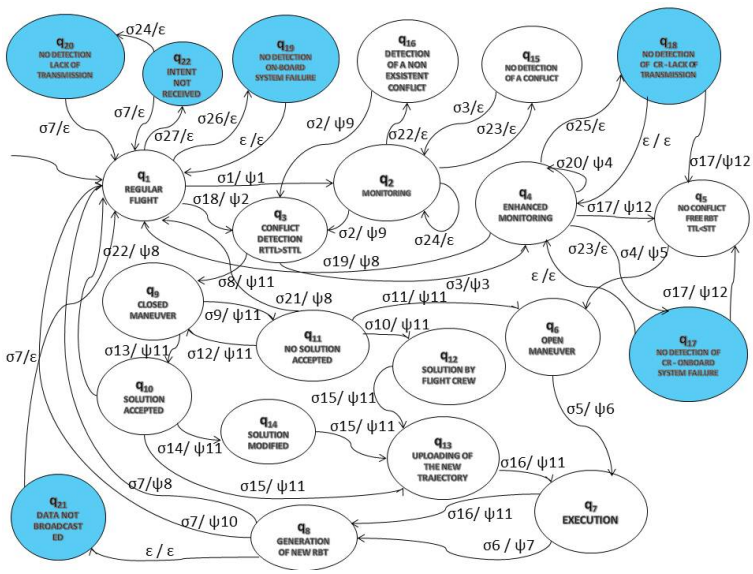



**Figure 5.36:** Hybrid system of the aircraft in nominal and non–nominal conditions.

The hybrid model in Figure 5.36 correctly describes the evolution of agents in the $A^3$ ConOps scenario, both in nominal and non–nominal conditions.

As we have seen in the previous sections, the formal analysis of the $A^3$ ConOps demonstrated that this scenario is not critically observable. This means that not all situation awareness inconsistencies can be always detected during the evolution of the ATM scenario. In particular this happens because:

- It is not possible to distinguish critical states $q_{17}$ and $q_{18}$ from the noncritical state $q_4$.
- It is not possible to distinguish critical state $q_{21}$ from the noncritical state $q_8$.
- It is not possible to distinguish critical states $q_{19}$, $q_{20}$, $q_{22}$ from the noncritical state $q_1$.
- It is not possible to distinguish critical states $q_{15}$ and $q_{16}$ from the noncritical state $q_2$.

An analysis of the mitigation means of potential unsafe events due to no detection of the aforementioned critical states has been performed, and reported hereafter:

- Critical states $q_{20}$, $q_{18}$, $q_{22}$ related to the absence of transmission. This type of failure is detectable for onboard system. According to [21] the update rates are required both for state and intent ADS-B messages. If information is not refreshed within the specified time period, information is marked as degraded and alternative information sources (e.g. SWIM, point-to-point data links) are used to get recent data. Furthermore, for the degraded intent information the trajectory prediction used in CD is reduced to shorter look-ahead time. Also there is onboard conformance monitoring function, continuously comparing the received state data with the available intent information and again reducing the look-ahead time when a deviation is detected. Furthermore, an independent CD functions working only with state data is required within ASAS equipment.
- Critical states $q_{17}$ and $q_{19}$ related to the failure of onboard (ASAS) equipment. The main mitigation mean for this type of failure are built–in test functions which inform flight crew about a failure of the system. Another backup is the situation awareness of the flight crew maintained through CDTI. However, this type of CD may be feasible only for short term time horizon (e.g., ATCo today considers about 5 minutes look ahead time only).
- Critical states $q_{15}$ and $q_{16}$ related to the general failure of CD function. The main mitigation of the impact (effect) for this type of problems is the short-term CR with implicit coordination ensuring that the other conflicting aircraft will solve potential conflict even without the manoeuvring of own aircraft. Considering the prevention of this hazard, the flight crew situation awareness and training remain the main mitigation means.
- Critical state $q_{21}$ not affecting own onboard functions. This failure is difficult to detect onboard own aircraft. In addition to built-in test function in transponder,

it is assumed that within the SWIM there will be a conformance monitoring function [21] detecting if there is no deviation between the known RBT and actual state information and will potentially inform surrounding aircraft.

Chapter 6

# Conclusions

In this thesis we used Hybrid Systems and Arenas of Finite State Machines mathematical formalism to model and analyze complex ATM scenarios. Multi–Agent Situation Awareness (MASA) inconsistencies have been modelled by a set of critical states. We defined a set of critical states that correspond to situation awareness inconsistencies. The possibility of detecting those critical states depends on the so–called critical observability property of the system: if the hybrid model or the AFSM is critically observable, our algorithms allow the detection of errors, on the basis of the information available. If the hybrid model or the AFSM is not critically observable, then our proposed approach is able to identify potential extra information that could be of help in obtaining critical observability.

We developed a compositional framework to model and analyze a complex multi-agent ATM scenario. We addressed critical observability of AHSs and AFSMs. We first proposed a definition of composition for AHSs, based on the exchange of discrete data among the systems involved. Moreover for the AFSMs we proposed the notion of critical compositional bisimulation. Then, we investigated compositional properties for critically observable subsystems. We proposed a method for separately analyzing the single agents instead of analyzing directly their composition, which usually generates an explosion of the computational complexity of the system. We proved that a safety critical observer for the overall system can be derived from the critical observers designed for each of the subsystems.

We considered four different procedures involving an arbitrary number of agents: the Terminal Manoeuvring Area T1 scenario, the Airborne Separation In Trail Procedure, the ASAS Lateral Crossing procedure and a scenario within the Autonomous Aircraft Advanced ($A^3$) ConOps. The analysis of observability of critical states arising in the composition of the agents involved in those procedures presents a particular interest from the situation awareness inconsistencies among agents point of view. Often the mathematical model of each agent is not enough to define critical states that reflect an inconsistent situation awareness of the agent with respect to the other agents. Using our compositional framework, situation awareness inconsistencies among agents can be easily modeled by defining a suitable critical

relation, that corresponds to inconsistencies of inter-agent situation awareness. The analysis that we performed showed that the aforementioned four case studies are not critically observable and therefore not all unsafe and/or unallowed operations can be detected. Possible solutions to render those procedures critically observable have been discussed and based on the generation of extra (alarm) signals, that detect the occurrence of such events.

We summarize hereafter the results achieved in this thesis:

- We provided a sound mathematical paradigm that appropriately models agents acting in ATM procedures in both nominal and non-nominal operating modes;
- We provided a compositional framework that appropriately models the interaction among the agents involved in ATM procedures;
- We provided a formal methodology to analyze Multi–Agent Situation Awareness (MASA) inconsistencies arising in the evolution of ATM procedures, which may lead to unsafe and/or catastrophic events;
- We provided efficient algorithms for the reduction of the computational complexity arising in the analysis of MASA inconsistencies in realistic ATM scenarios, where a large number of agents operate;

Appendix A

# Acronyms

| | |
|---|---|
| ACAS | Airborne Collision Avoidance System |
| ADS-B | Airborne Dependant Surveillance Broadcast |
| AMAN | Arrival manager |
| ASAS | Airborne Separation Assistance System |
| ASEP | Airborne Separation |
| ASSTAR | Advanced Safe Separation Technologies and Algorithms |
| ATC | Air Traffic Controller |
| ATM | Air Traffic Management |
| ATSA | Airborne Traffic Situational Awareness |
| CD | Conflict Detection |
| CDTI | Cockpit Display of Traffic Information |
| CPDLC | Controller-Pilot Datalink Communication |
| CR | Conflict Resolution |
| FMS | Flight management system |
| fpm | Feet per minute |
| ITP | In-Trail Procedure |
| NM | Nautical Miles |
| MASA | Multi–Agent Situation Awareness |
| MONA | Monitoring aids |
| MTCD | Medium term conflict detection |
| P-RNAV | Precision Area Navigation |
| RBT | Reference Business Trajectory |
| RVSM | Reduced Vertical Separation Minima |
| STCA | Short term conflict alert |
| SWIM | System Wide Information Management |
| TMA | Terminal Manoeuvring Area |

# Bibliography


[1] R. Alur and M. Yannakakis. *Model checking of hierarchical state machines.* ACM Transactions on Programming Languages and Systems, 23(3):273-303, 2001.

[2] R. Alur, S. Kannan, and M. Yannakakis. *Communicat ing hierarchical state machines.* In Computer Science Automata, Languages and Programming, volume 1644 of Lecture Notes in Computer Science, pages 169-178, Springer Verlag, 1999.

[3] A. Ames, A. Abate and S. Sastry. *Sufficient Conditions for the Existence of Zeno Behavior in Hybrid Systems.* Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, December 2005

[4] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, September 1999, ISBN 0-7923-8609-4.

[5] E.M. Clarke, O. Grumberg and D.A. Peled. *Model Checking.* The MIT Press, Cambridge, Massachusetts, 2002.

[6] M. Colageo and A. Di Francesco. *Hybrid System Framework for the Safety Modelling of the In Trail Procedure.* ICRAT 2008 - 3rd International Conference on Research in Air Transportation, Fairfax, Virginia, USA, June 2008, 01-04.

[7] E. De Santis, M. D. Di Benedetto,S. Di Gennaro, A. D'Innocenzo and G. Pola. *Critical Observability of a Class of Hybrid Systems and Application to Air Traffic Management.* Book Chapter of Lecture Notes on Control and Information Sciences, Springer Verlag, 2005.

[8] M.D. Di Benedetto, A. Petriccone and G. Pola. *Report on Observability Properties of Hybrid-System Composition.* January 2011, Deliverable 4.2, iFly.

[9] E. De Santis, M.D. Di Benedetto, A. Petriccone and G. Pola. *A Compositional Hybrid System Approach to the Analysis of Air Traffic Management Systems.* Proc. of the 8th Innovative Research Workshop & Exhibition, EUROCONTROL, Paris, France, December 2009.

[10] K. G. Larsen, P. Pettersson and W. Yi. *Uppaal in a nutshell.* International Journal on Software Tools for Technology Transfer, volume 1(1), pages 134–152, December 1997.

[11] M. D. Di Benedetto, S. Di Gennaro and A. D'Innocenzo. *Discrete State Observability of Hybrid Systems.* International Journal of Robust and Nonlinear Control, Special Issue on Observability and Observer Design for Hybrid Systems, volume 19(14), pages 1564–1580, 2008.

[12] M.D. Di Benedetto,Petriccone and G. Pola. *Review of SESAR 2020 Conops.* MAREA Project, October 2011.

[13] C. Montijn,G. Graniero and B. K. Obbink. *Qualitative Risk Assessment for ASEP-ITP.* D6.1b ASSTAR Projects, v.1.0, 01 February 2007.

[14] S. Stroeve, H.A.P. Blom and M. Van der Park. *Multi–Agent Situation Awareness Error Evolution in Accident Risk Modelling.* FAA–Eurocontrol, ATM2003, 2003, http://atm2003.eurocontrol.fr/.

[15] D.M.R. Park. *Concurrency and automata on infinite sequences.* volume 104 of Lecture Notes in Computer Science, pages 167-183, 1981.

[16] A. Petriccone,G. Pola, M.D. Di Benedetto and E. De Santis, *A Complexity Reduction Approach to the Detection of Safety Critical Situations in Air Traffic Management Systems.* Proceedings of the $49^{th}$ Conference on Decision and Control, Atlanta, USA, 2081–2085, December 2010.

[17] A. Balluchi, L. Benvenuti, M.D. Di Benedetto,and A.L. Sangiovanni-Vincentelli. *Design of Observers for Hybrid Systems.* Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, volume 2289, pages 76–89, Springer Verlag, 2002.

[18] M.D. Di Benedetto, A. D' Innocenzo, A. Petriccone and G. Pola. *Intermediate Report on Compositionality Properties of Critical Observability.* 12 November 2010, Deliverable 4.2i, iFly.

[19] H.A.P. Blom, G.J. Bakker, M.B. Klompstra and F.J.L. Bussink. *Hazard Identification and Initial Hazard Analysis of $A^3$ ConOps based operation.* August 2009 Deliverable 7.1b, iFly.

[20] Petr Casek and Eva Gelnarova. *Operational Services and Environment Description (OSED) of Airborne Self-Separation Procedure (SSEP).* 2009, Deliverable 9.1, iFly.

[21] P. Casek and P. $MejzlA - k$. *Operational Performance Assessment (OPA).* 2010, Deliverable 9.3, iFly.

[22] M. Colageo, M.D. Di Benedetto and A. D'Innocenzo. *Report on hybrid models and critical observer synthesis for multi-agent situation awareness.* 22 May 2007, Deliverable 4.1, iFly.

[23] G. Cuevas, I. Echegoyen, J. Garcia, P. Casek, C. Keinrath, R. Weber, P. Gotthard, F. Bussink and A. Luuk. *Autonomous Aircraft Advanced $A^3$ ConOps*. 22 August 2008, Deliverable 1.3, iFly.

[24] E. De Santis, M.D. Di Benedetto and G. Pola. *Observability of Internal Variables in Interconnected Switching Systems*. Proceedings of the 45th IEEE Conference on Decision and Control, pages 4121-4126, San Diego, CA, USA, December 2006, 13-15.

[25] W. Glover and J. Lygeros. *A Multi-Aircraft Model for Conflict Detection and Resolution Algorithm Evaluation*. Project IST-2001-32460 HYBRIDGE, Deliverable 1.3, 18 February 2004.

[26] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison–Wesley, 1979.

[27] J.M. Loscos, T. Miquel, B. Hasquenoph, B. Gayraud, S. Chabert, B. Raynaud. *Specific and detailed conditions of use for applicability to radar airspace*. D1.3 ASSTAR Projects, 17 April 2007.

[28] J. Lygeros. *Lecture Notes on Hybrid Systems*. ENSIETA, 2-6/2, 2004.

[29] J. Lygeros, C. Tomlin and S. Sastry. *Controllers for reachability specications for hybrid systems*. Automatica, Special Issue on Hybrid Systems, volume 35, 1999.

[30] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[31] R. Alur and M. Yannakakis. *Model checking of hierarchical state machines*. ACM Transactions on Programming Languages and Systems. volume 23, number 3, pages 273-303, 2001.

[32] R. Alur, S. Kannan and M. Yannakakis. *Communicating Hierarchical State Machines*. Computer Science Automata, Languages and Programming, Lecture Notes in Computer Science, volume 1644, pages 169-178, Springer Verlag, 1999.

[33] D. Harel. *Statecharts: A visual formalism for complex systems*. Science of Computer Programming, volume 8, pages 231-274, 1987.

[34] R. Alur, M. Benedikt, K. Etessami, P. Godefroid, T. Reps and M. Yannakakis. *Analysis of Recursive State Machines*. ACM Transactions on Programming Languages and Systems, volume 27, number 4, pages 786-818, July 2005.

[35] K. Etessami. *Analysis of recursive game graphs using data flow equations*. 5th International Conference on Verification, Model Checking, and Abstract Interpretation, Lecture Notes in Computer Science, volume 2937, pages 282-296, Springer Verlag, 2004.

[36] T. Cachat. *Symbolic Strategy Synthesis for Games on Pushdown Graph.* Computer Science Automata, Languages and Programming, Lecture Notes in Computer Science, volume 2380, pages 704-715, Springer Verlag, 2002.

[37] Z. Sawa and P. Janfçar. *Hardness of equivalence checking for composed finite-state systems.* Acta Informatica, volume 76, number 3, pages 169-191, 2009.

[38] F. Laroussinie and P. Schnoebelen. *The State Explosion Problem from Trace to Bisimulation Equivalence.* Foundations of Software Science and Computation Structures, Lecture Notes in Computer Science, volume 1784, pages 192-207, Springer Verlag, 2000.

[39] R.J. van Glabbeek. *The linear time–branching time spectrum*, CONCUR '90 Theories of Concurrency: Unification and Extension, Lecture Notes in Computer Science, volume 458, pages 278-297, Springer Verlag 1990.

[40] A. Rabinovich, *Complexity of equivalence problems for concurrent systems of finite agents.* Information and Computation, volume 139, number 2, pages 111-129, 1997.

[41] E.F. Moore. *Gedanken–experiments on sequential machines.* Annals of Mathematics Studies, Automata Studies, volume 34, pages 129-153, Princeton University Press, Princeton, NJ, 1956.

[42] D. Bustan and O. Grumberg. *Modular Minimization of Deterministic Finite–State Machines.* 6th International Workshop on Formal Methods for Industrial Critical Systems, volume 6, pages 163-178, Paris, France, December 2001.

[43] Milner *Communication and Concurrency.* Prentice Hall, 1989.

[44] R. Paige and R.E. Tarjan. *Three Partition Refinement Algorithms.* SIAM Journal on Computing, volume 16, number 6, pages 987-989, 1987.

[45] M.H.C. Everdij, H Zmarrou, G.J. Bakker and H.A.P. Blom. *D7.4 Preliminary Safety Case – Part 2 TMA T1.* November 2010, http://reset.aena.es/start/frames.html.

[46] A. Dovier, C. Piazza and A. Policriti. *An Efficient Algorithm for Computing Bisimulation.* Theoretical Computer Science, volume 311, number 1-3, pages 221-256, 2004.

[47] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing Company, 1979.

[48] Giordano Pola, Maria D. Di Benedetto and E. De Santis IFAC World Congress, Milan, Italy, August 2011.

[49] *In-Trail Procedure in Procedural Airspace (ATSA-ITP) Application Description.* ASSTAR Projects June 2007, v.8.0.

[50] *SESAR European Air Traffic Management Master Plan.* Edition 1, March 2009.

[51] *D7.3 Qualitative Hazard Report – Part 2 – TMA Route Separation Reduction.* March 2010, http://reset.aena.es/start/frames.html.

[52] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* The MIT Press, Cambridge, Massachusetts, 2002.

[53] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas. *Discrete abstractions of hybrid systems.* Proccedings of the IEEE, 88(2):971ñ984, July 2000.

[54] R.J. van Glabbeek. *The linear time-branching time spectrum* CONCURí90, 1990.

[55] E. Hollnagel,D.D.Woods and N. Leveson *Resilience engineering: Concepts and precept.* Ashgate, Aldershot, England 2006.

[56] E. Hollnagel and C.P. Nemeth. *Resilience Engineering Perspectives: Remaining Sensitive to the Possibility of Failure.* volume 1, Ashgate, England, 2008.

[57] C.P. Nemeth, E. Hollnagel and S. Dekker *Resilience Engineering Perspectives, Preparation and restoration.* volume 2, Ashgate, England, 2009.

[58] M.D. Di Benedetto, A. D'Innocenzo and A. Petriccone. *Automatic Verification of Temporal Properties of Air Traffic Management Procedures Using Hybrid Systems.* EUROCONTROL Innovative ATM Research Workshop & Exhibition, December, 2008.

[59] A. Arnold. *Finite transition systems and semantics of communicating systems.* Prentice-Hall, 1994.

[60] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, third edition, 1996.

[61] R.E. Kalman, P.L. Falb, and M.A. Arbib. *Topics in Mathematical Systems Theory.* International series in pure and applied mathematics, McGraw-Hill, New York, 1969.

[62] R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors. *Hybrid Systems.* Lecture Notes in Computer Science, volume 736, Springer-Verlag, 1993.

[63] A. Pneuli and J. Sifakis (Eds.). *HSpecial issue on hybrid systems.* Theoretical Computer Science 138. 1995.

[64] G. Pola, A.J. der Schaft, and M.D. Di Benedetto. *Equivalence of switching linear systems by bisimulation.* International Journal of Control, volume 79, pages 74-92, 2006.