HYBRIDGE

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

WP1: Identification and modelling of uncertain hybrid systems

A Multi-Aircraft Model for Conflict Detection and Resolution Algorithm Evaluation

W. Glover[1] and J. Lygeros[2]

February 18, 2004

Version:	*1.3*
Task number:	*1.3*
Deliverable number:	*D1.3*
Contract:	*IST-2001-32460 of European Commission*

[1] Department of Engineering, University of Cambridge, U.K.
[2] Department of Electrical and Computer Engineering, University of Patras, Greece.

DOCUMENT CONTROL SHEET

Title of document: *A multi-aircraft model for conflict detection and resolution algorithm evaluation*

Authors of document: *W. Glover and J. Lygeros*

Deliverable number: *D1.3*

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

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

DOCUMENT CHANGE LOG

Version #	Issue Date	Sections affected	Relevant information
0.1	4/22/2003	All	First draft
0.2	1/6/2003	All	Second draft
1.0	23/7/2003	All	Submitted to the consortium
1.2	10/12/2003	All	Revised based on reviewers comments
1.3	18/2/2004	All	Revised based on reviewers comments

Version 1.0		Organisation	Signature/Date
Authors	W. Glover	UCAM	
	J. Lygeros	U. Patras	
Internal reviewers	H. Blom	NLR	
	M. Prandini	UniBs	

HYBRIDGE, IST-2001-32460
Work Package WP1, Deliverable D1.3

A Multi-Aircraft Model for Conflict Detection and Resolution Algorithm Evaluation

Prepared by:
W. Glover[*] and J. Lygeros[†]

Abstract

A method for modelling the evolution of multiple flights from the point of view of an air traffic controller is developed. The model consists of many instances of flights, each with different aircraft dynamics, flight plan and flight management system. The motions of different flights are coupled through the effect of the wind, which is modelled as a random field. Estimates of the statistical properties of the wind field (variance and spatio-temporal correlation structure) are extracted from publicly available weather data. The model is coded in Java, so that it can be simulated to generate realistic data for evaluating conflict detection and resolution algorithms.

[*] Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK, Tel. +44122333260, Fax. +441223332662, wg214@eng.cam.ac.uk

[†] Department of Electrical and Computer Engineering, University of Patras, Rio, Patras, GR-26500, GREECE, Tel. +30 2610 996 458, Fax. +30 2610 991 812, lygeros@ee.upatras.gr

Contents

List of Figures

List of Tables

List of Acronyms

ATC	Air Traffic Controller
BADA	Base of Aircraft Data
CAS	Calibrated Airspeed
CFMU	Central Flow Management Unit
FMS	Flight Management System
GPS	Global Positioning System
NCEP	National Center for environmental Prediction
NOAA	National Oceanic and Atmosphere Administration
ROCD	Rate of Climb/Descent
RUC	Rapid Update Cycle
PMM	Point Mass Model
TEM	Total Energy Model
ESF	Energy Share Factor
TRACON	Terminal Radar Approach Control (U.S. equivalent of TMA)
VOR	VHF Omni-directional radio Ranging
JRE	Java Runtime Environment
GUI	Graphical User Interface
CTAS	Center-TRACON Automation System

Notation

Flight Plan

$O(i)$	Coordinates of way point i (from CFMU, $O(i) \in \mathbb{R}^3$)
$t(i)$	Expected time of arrival at way point i (from CFMU)
$\Psi(i)$	Angle i^{th} segment of nominal path makes with $X-$axis (computed from $O(i)$)

Aircraft continuous states and inputs

x_1	$X-$position in global coordinates
x_2	$Y-$position in global coordinates
x_3	Altitude (also h)
x_4	True airspeed (also V)
x_5	Heading angle (also ψ)
x_6	Aircraft mass (also m)
u_1	Engine thrust (also T)
u_2	Roll angle (also ϕ)
u_3	Pitch angle (also γ)
$\hat{u}_3$	Energy Share Factor (also $f(M)$)
w	Three dimensional wind speed (disturbance)

Aircraft continuous parameters

C_L	Aerodynamic lift coefficient
C_D	Aerodynamic drag coefficient (from BADA)
S	Wing surface area (from BADA)
ρ	Air density (from BADA)
g	Gravitational acceleration, $g = 9.81ms^{-2}$

Aircraft discrete parameters

Aircraft_Type
Engine_Type

FMS discrete states

FL	Flight Level
WP	Way point index
FP	Flight Phase
AM	Acceleration mode
CM	Climb Mode
TrM	Troposphere mode
SHM	Speed Hold Mode
RPM	Reduced Power Mode

FMS continuous parameters

ϕ_{nom}	Nominal bank angle for turning (currently $\phi_{nom} = 35°$)
V_{nom}	Nominal speed, either CAS or Mach (from BADA)
k_1, k_2	Bank angle controller gains
δ	Cross track deviation from the reference path
θ	Heading error
$\overline{\phi}$	Bank angle saturation limit (currently $\overline{\phi} = 35°$)
$\overline{\psi}$	Heading angle limit for bank angle controller (currently $\overline{\psi} = 60°$)

Logic formulae

$\wedge$	AND
$\vee$	OR

1 Introduction

1.1 Aim and Scope

The main aim of Deliverable D1.3 of HYBRIDGE is summarised in Task 1.3 of WP1:

> Identify a detailed multi-aircraft model, to be used as a "real world" for validating subsequent algorithms. Because of the interaction between stochastic and hybrid dynamics, conventional system identification methods are insufficient for this task. In the process, we will therefore develop new system identification results for dealing with this class of stochastic hybrid systems

Our main aim is to develop a model that does not necessarily reproduce exactly the systems used in commercial aircraft, but adequately simulates their behaviour from the point of view of an Air Traffic Controller (ATC), while maintaining a workable degree of simplicity. The purpose of the model is to be used primarily as a basis for numerical experiments for the evaluation of conflict detection and resolution algorithms. While it may be possible to also use the model as a basis for conflict detection methods (e.g. by Monte-Carlo simulation) the use of the model in this capacity is limited because

1. Simulation of the model is computationally intensive.
2. The model does not include any assumptions about forecasts of the nominal wind. As will become apparent later, our model treats wind as an unknown, stochastic quantity with known statistics. Effectively this is the same as assuming that the predictable, nominal wind is zero. While it should be possible to add nominal wind information in the model (e.g. by including it in the mean of the stochastic wind) this direction is not pursued here.

The model is framed in the context of stochastic hybrid processes presented in the Deliverable D1.2 of the HYBRIDGE project [1]. In particular, the model comprises

- Continuous dynamics, arising from the physical motion of the aircraft.
- Discrete dynamics, arising mainly because of the flight plan and the logic embedded in the Flight Management System (FMS).
- Stochastic dynamics, arising primarily because of the effect of wind on the aircraft tracks.

The model also allows one to include uncertainty about some of the actions of the air traffic controller, for example the time at which they order an aircraft to begin its final descent. This delay by the air traffic controller is not modelled in further detail; work on this topic may follow in WP7 of Hybridge. Finally, our model also allows one to capture parametric uncertainty, for example, uncertainty about the initial mass of the aircraft. The long term aim is to enable evaluation of algorithms that estimate such parameters, based for example on adaptive control and system identification methods [7]. The model has been coded in Java and can be used to generate "real-world" data for evaluation of the performance of conflict detection and resolution algorithms.

Our multi-aircraft model contains a number of parameters, such as the masses of aircraft, their aerodynamic coefficients, the gains of the controllers used to model the FMS, the variance and spatio-temporal correlation of the wind, etc. We obtain typical values for many of these parameters (aircraft masses, aerodynamic coefficients) from the Base of Aircraft Data (BADA) database [2]. For the remaining parameters (FMS gains, wind statistics), one would ideally like to determine values through formal system identification experiments. However, for the reasons highlighted in Deliverable D1.2 (unavailability of data, multiple time scales, etc.), this approach may be unrealistic within the scope of the HYBRIDGE project. To circumvent the problem we first estimate the wind statistics from publicly available weather data. We then run Monte-Carlo simulations of our multi-aircraft model driven by random wind with the computed statistics, compare the results to earlier studies on the deviation of aircraft from their flight plan, and tune the values of the FMS gains to get the results of the simulations to match the conclusions of these studies.

1.2 Model Overview

Our model allows one to capture many flights taking place at the same time. In the simulation, each flight is represented by an instance of the class *Aircraft*. With each *Aircraft* we associate the following model components:

- The flight plan.
- The aircraft dynamics.
- The flight management system.

A separate instance of each of these components is generated for each *Aircraft*. In the notation used in this report, however, we will drop the dependence of these components on *Aircraft* for simplicity.

The evolution of flights is also affected by the weather. The only element of weather present in our model is wind speed. We model the wind speed as a stochastic quantity, correlated in space and time. Therefore, the evolutions of different flights are coupled to one another through

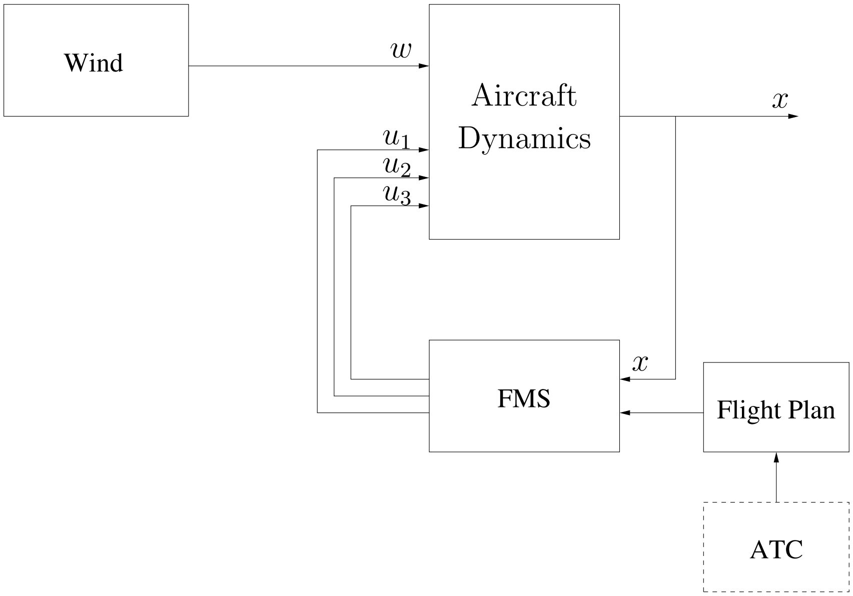


Figure 1: Block diagram of multi-aircraft model components

- The wind model.

The relations between these components are summarised in Figure 1. The component labelled ATC in the figure is not part of our model. Our model only provides a few functions to allow one to capture some rudimentary aspects of controller behaviour, such as uncertainty about the timing of commands. The aircraft velocity is affected by both the wind and the flight management system action, with the latter depending on the value taken by the state of the aircraft dynamics block.

1.3 Overview of D1.3

The rest of this report provides the details of the models developed for each one of the components listed in Figure 1 and clarifies the nature of the interactions between them. The flight plan is discussed in Section 2, the model for the aircraft dynamics in Section 3, the model for the FMS in Section 4 and the basics of the model for the wind in Section 5. Section 6 discusses the issues that had to be resolved to code the model in Java. Section 7 presents the tuning of the model parameters, based on publicly available weather data and Monte-Carlo simulation of the model itself. A summary of the assumptions made during the development of the model and of the improvements being pursued is given in Section 8.

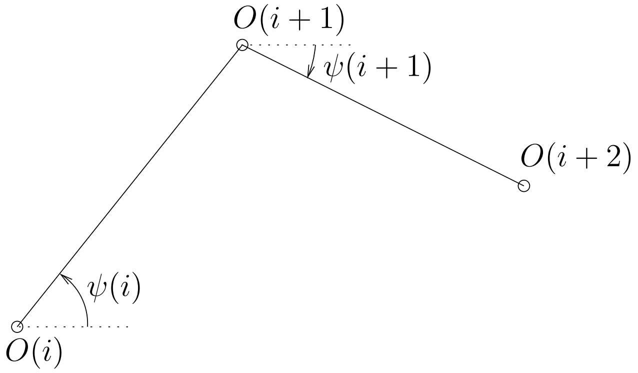


Figure 2: Typical flight plan showing way-points and angles

2 Flight Plan

The flight plan consists of a sequence of way-points, $\{O(i)\}_{i=0}^{M}$ in three dimensions, $O(i) \in \mathbb{R}^3$. Each way-point is time stamped with an expected time of arrival, $\{t(i)\}_{i=0}^{M}$. The way point data used in our simulations comes from Eurocontrol's Central Flow Management Unit (CFMU).

For this paragraph we consider just the first two components of the way points, i.e. the projection of the way points onto the horizontal plane. The sequence of the way points defines a sequence of straight lines joining each way point to the next. We refer to this sequence of straight lines as the *reference path*. For each way point, $O(i)$, we also define the *reference course*, $\Psi(i)$, as the angle that the line segment joining $O(i)$ to $O(i+1)$ makes with the X-axis of the frame in which the way point coordinates are given (Figure 2). If $O(i) = (x(i), y(i), z(i))$ it is easy to see that

$$\Psi(i) = \tan^{-1}\left(\frac{y(i+1) - y(i)}{x(i+1) - x(i)}\right).$$

In our implementation the expected times of arrival are ignored. Instead we assume that the airspeed when flying between two way-points is dictated by the speed profiles provided by BADA (see Section 4.3). This approach is sometimes referred to as a *3D way-point model*. An alternative to our approach would be a *4D way-point model*. This would require one to add a system for either assigning different speeds depending on the arrival times, or assigning different climb and descent profiles, e.g. staying longer at high altitude and then descending faster to produce a faster but less efficient flight path. [13] provides an example of trajectory synthesis using the 4D approach.

3 Aircraft Dynamics

From the point of view of ATC an aircraft can be adequately modelled using a Point Mass Model (PMM). The states of the model are the horizontal position (X and Y) and altitude (h) of the aircraft, the true airspeed (V), the flight path angle (γ) and the heading angle (ψ). The control inputs to the model are the engine thrust (T), the angle of attack (α) and the bank angle (ϕ). The movement of the aircraft is also affected by the wind, which acts as a disturbance. We will model the wind through its speed $W = (w_1, w_2, w_3) \in \mathbb{R}^3$. We ignore the effect of inputs such as spoilers, leading edge slats, landing gear, etc.

True airspeed is the actual speed of the aircraft relative to the surrounding air. Calibrated airspeed is the airspeed that is measured by an aircraft before it has been corrected for air density, so at high altitudes the calibrated airspeed will be less than the true airspeed because of the lower air density.

Assuming a point mass model, simple Newtonian dynamics without wind lead to the following equations of motion (also used in [12] and in a modified form in [3]).

$$\begin{aligned}
\dot{X} &= V\cos(\psi)\cos(\gamma) + w_1\\
\dot{Y} &= V\sin(\psi)\cos(\gamma) + w_2\\
\dot{h} &= V\sin(\gamma) + w_3\\
\dot{V} &= \frac{1}{m}\left[(T\cos(\alpha) - D) - mg\sin(\gamma)\right]\\
\dot{\psi} &= \frac{1}{mV}(L\sin(\phi) + T\sin(\alpha)\sin(\phi))\\
\dot{\gamma} &= \frac{1}{mV}[(L + T\sin(\alpha))\cos(\phi) - mg\cos(\gamma)].
\end{aligned}$$

m is the mass of the aircraft and g the gravitational acceleration. L and D denote respectively the lift and drag forces, which are functions of the state and angle of attack. Typically

$$\begin{aligned}
L &= \frac{C_L S\rho}{2}(1 + c\alpha)V^2,\\
D &= \frac{C_D S\rho}{2}(1 + b_1\alpha + b_2\alpha^2)V^2,
\end{aligned}$$

where S is the surface area of the wings, ρ is the air density (which depends on altitude) and C_D, C_L, c, b_1, b_2 are aerodynamic lift and drag coefficients whose values generally depend on the phase of the flight (whether the flaps are extended, the landing gear down, etc.). See section 3.2.

Note that the model assumes that the airspeed is not affected by the rate of change of the wind velocity. For example if the aircraft is flying with no wind and then suddenly encounters a head wind then in our model the airspeed will stay the same producing a sudden change in the ground speed, whereas in reality the effect of the headwind will produce a more gradual change in the ground speed caused by the increased drag on the aircraft. This is not entirely accurate, but is an easily implementable approximation in the first iteration of the model. It will generally be a good approximation assuming that the wind velocity does not change too quickly in time, or space (see section 7.1).

As we are dealing mainly with commercial airliners we can simplify the model further by assuming that the aircraft always operates near trimmed flight conditions ($\alpha \approx 0$) and treat the flight path angle as an input instead of a state. The result is what is referred in the BADA manual as a Total Energy Model (TEM). The TEM effectively equates the rate of work done by the forces acting on the aircraft to the rate of increase in its total (potential plus kinetic) energy. Figure 3 shows the forces acting on the aircraft under the assumptions imposed so far. BADA also introduces an additional state to keep track of changes in the aircraft mass, which could be substantial especially over longer flights.

The aircraft motion can now be captured by a control system with six states ($x = (x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{R}^6$), three inputs ($u = (u_1, u_2, u_3) \in \mathbb{R}^3$) and three disturbances ($w = (w_1, w_2, w_3) \in \mathbb{R}^3$). We set $x_1 = X$, $x_2 = Y$, $x_3 = h$, $x_4 = V$, $x_5 = \psi$, $x_6 = m$, $u_1 = T$, $u_2 = \phi$, $u_3 = \gamma$. The equations of motion become

$$\dot{x} = \begin{bmatrix} x_4 \cos(x_5) \cos(u_3) + w_1 \\ x_4 \sin(x_5) \cos(u_3) + w_2 \\ x_4 \sin(u_3) + w_3 \\ -\frac{C_D S \rho}{2} \frac{x_4^2}{x_6} - g \sin(u_3) + \frac{1}{x_6} u_1 \\ \frac{C_L S \rho}{2} \frac{x_4}{x_6} \sin(u_2) \\ -\eta u_1 \end{bmatrix}. \tag{1}$$

The state and inputs are subject to constraints: $x_3 > 0, x_4 \in [V_{min}, V_{max}]$, $x_6 \in [m_{min}, m_{max}]$, $u_1 \in [T_{min}, T_{max}]$, $u_2 \in [\phi_{min}, \phi_{max}]$, $u_3 \in [\gamma_{min}, \gamma_{max}]$. The values of the state and input bounds and the values of the parameters C_D, S and ρ can be obtained from the BADA database. C_L is discussed in section 3.2. We always use $g = 9.81ms^{-2}$.

The equations of motion can also be written using what is known as the Energy Share Factor (ESF). The ESF, which is denoted in BADA by $f(M)$, is a function of the Mach number, M, and determines the ratio of the available power which is allocated to climbing (increasing potential energy) versus accelerating (increasing kinetic energy) while following a selected speed profile during climb or descent[1]. In our context the ESF can be thought of as an alternative input $\hat{u}_3 = f(M)$ that replaces the flight path angle (u_3) in the equations. The two input parameterisations are related by

$$\left(u_1 - \frac{C_D S \rho}{2} x_4^2 \right) \hat{u}_3 = g x_6 \sin(u_3). \tag{2}$$

[1] During descent the power devoted to climb is of course negative.

Using conservation of energy the equations of motion (1) using the ESF instead of the flight path angle as input become:

$$\dot{x} = \begin{bmatrix} x_4 \cos(x_5)\cos(u_3) + w_1 \\ x_4 \sin(x_5)\cos(u_3) + w_2 \\ x_4 \sin(u_3) + w_3 \\ \frac{1}{x_6}\left(u_1 - \frac{C_D S \rho}{2} x_4^2\right)(1 - \hat{u}_3) \\ \frac{C_L S \rho}{2} \frac{x_4}{x_6} \sin(u_2) \\ -\eta u_1 \end{bmatrix}. \tag{3}$$

For much of the discussion below the ESF model will be adopted. This is because it is extensively used in the BADA documentation (and apparently also in practice). From a mathematical point of view the ESF is useful because it is a function of just the mach number. Notice that for a given thrust setting u_1 each value of the ESF can be related to a unique value of the flight path angle (at least in the range of interest $[-\pi/2, \pi/2]$) using equation (2), provided that $|\dot{x}_3/x_4| < 1$. The same is not true the other way around however: there is a singularity when $u_1 = \frac{C_D S \rho}{2} x_4^2$, i.e. when the thrust exactly matches the drag. In this case the aircraft is not accelerating, all available power goes to climbing and the ESF becomes infinite. A better energy share type parametrization may be the ratio of power devoted to climbing over the total available power; this parametrization would still have a singularity whenever the increase in potential energy is exactly matched by a decrease in kinetic energy, or vice versa.

Notice that the model of equations (1) and (3) ignores the use of inputs such as spoilers, leading edge slats, trailing edge flaps, landing gear, etc. The effect of some of these inputs is reflected indirectly however, by changes in the aerodynamic parameters discussed in Section 3.2.

3.1 Discrete Parameters

Each *Aircraft* has associated with it a discrete aircraft type (e.g. Airbus A330, Boeing 737, etc.) We use a discrete parameter *Aircraft_Type* to store this; the (many) possible values that the parameter *Aircraft_Type* can take are listed in the BADA documentation. In our simulator *Aircraft_Type* is used to retrieve values from the BADA database for parameters such as drag coefficients, bounds on the mass, bounds on speed, etc.

BADA also provides an engine type for each *Aircraft_Type.* This is used by the flight management system to set the maximum climb thrust. To simplify the discussion, we use a separate discrete parameter *Engine_Type* to denote this information. *Engine_Type* takes one of three values, *Jet, Turboprop* and *Piston.*

3.2 Aerodynamic Parameters

The two aerodynamic coefficients, C_L and C_D are set following the procedure suggested in the BADA documentation. Since the model reflects trimmed flight conditions, C_L is set to ensure that the vertical component of the lift exactly balances the weight of the aircraft, including a correction term for changes in the bank angle.

$$C_L = \frac{2x_6 g}{\rho x_4^2 S \cos u_2}$$

Notice that in general the lift coefficient depends on the state of the system, as well as the surface area of the wing, S, and the air density ρ. The expression for ρ is a function of altitude and is determined by the standard atmosphere equations described in detail in [2], Chapter 3.

The C_L is also used in the computation of the drag coefficient C_D. C_D depends on the phase of the flight (whether the flaps are extended, the landing gear down, etc.) The dependence is shown in Table 1. *FP* and *CM* are part of the discrete state of the FMS discussed in Section 4.2. The parameters $C_{D0,CR}$, $C_{D2,CR}$, etc. depend on *Aircraft_Type* and can be obtained from the BADA database.

Flight Phase	Drag Coefficient
$(CM = D) \wedge (FP = AP)$	$C_D = C_{D0,AP} + C_{D2,AP} C_L^2$
$(CM = D) \wedge (FP = LA)$	$C_D = C_{D0,LDG} + C_{D0,\Delta LDG} + D_{D2,AP} C_L^2$
else	$C_D = C_{D0,CR} + C_{D2,CR} C_L^2$

Table 1: Equations for drag coefficient, where *FP* denotes Flight Phase, *AP* is approach, *LA* is landing, *CM* is Climb Mode and *D* is descent.

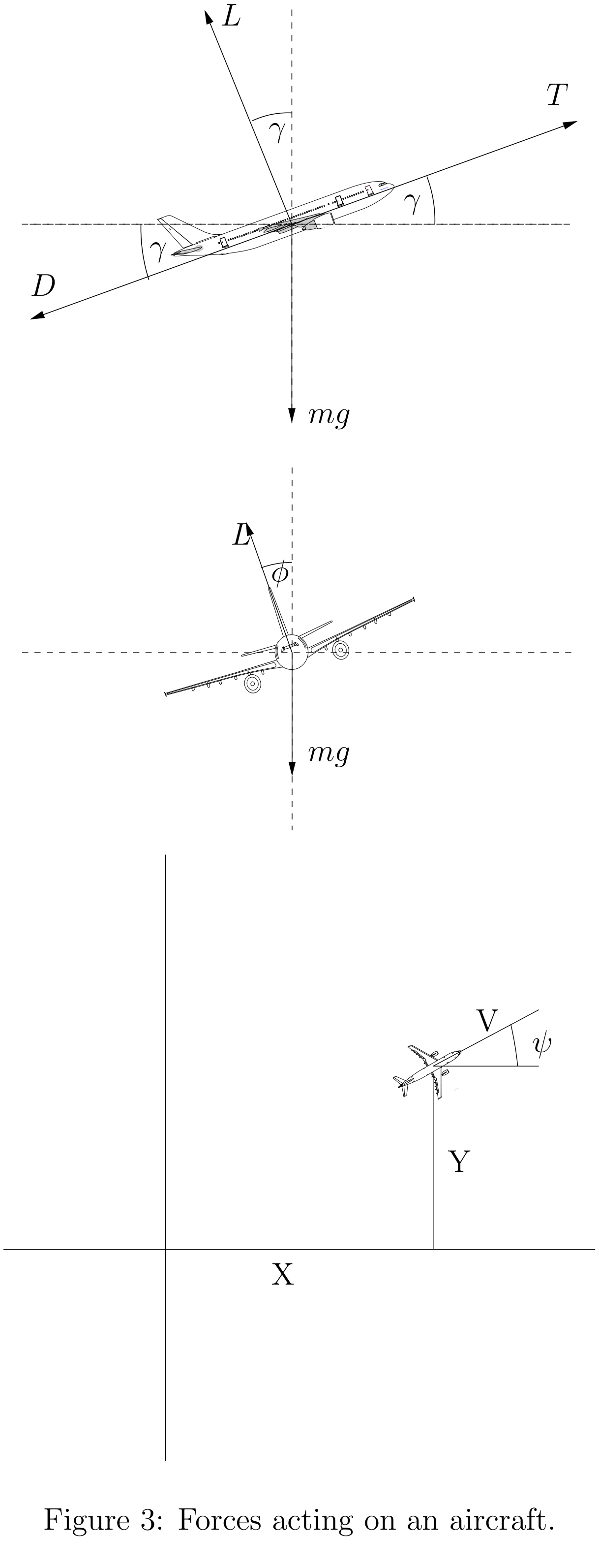


Figure 3: Forces acting on an aircraft.

4 Flight Management System

4.1 Inputs and Outputs

The FMS can be thought of as a controller that measures the state of the aircraft dynamics, x, and uses it together with the flight plan information to determine the values for the inputs, u.

The thrust u_1 and flight path angle u_3 are used to set the speed and the Rate of Climb/Descent (ROCD). At the moment, our model assumes that the FMS always tries to track a desired speed, V_{nom}; the procedure for setting V_{nom} is discussed in Section 4.3. When cruising at a constant altitude, the FMS sets the flight path angle to zero, hence achieving zero ROCD. The thrust is then used to control the speed through the equation

$$\dot{x}_4 = -\frac{C_D S \rho}{2} \frac{x_4^2}{x_6} + \frac{1}{x_6} u_1. \tag{4}$$

When climbing or descending, on the other hand, the thrust is set to a fixed value. The flight path angle (or the ESF) is then adjusted to control the speed through

$$\dot{x}_4 = -\frac{C_D S \rho}{2} \frac{x_4^2}{x_6} - g \sin(u_3) + \frac{1}{x_6} u_1. \tag{5}$$

The FMS accepts whatever ROCD is obtained from

$$\dot{x}_3 = x_4 \sin(u_3) + w_3.$$

This procedure for setting the thrust and flight path angle appears to be the most commonly used, because it allows for efficient climbs and descents. An alternative procedure where ROCD is explicitly controlled is usually only invoked by the instruction of ATC. For landing controlled ROCD may have to be used to make sure the aircraft hits the runway.

We assume that aircraft control their horizontal position using exclusively the bank angle input (u_2). This is done by first controlling the heading angle (x_5) through the equation

$$\dot{x}_5 = \frac{C_L S \rho}{2} \frac{x_4}{x_6} \sin(u_2).$$

x_5 can then be used to control the horizontal position of the aircraft (x_1, x_2) through the equations

$$\begin{aligned}\dot{x}_1 &= x_4 \cos(x_5) \cos(u_3) + w_1\\ \dot{x}_2 &= x_4 \sin(x_5) \cos(u_3) + w_2.\end{aligned}$$

u_2 is set in our model to correct cross-track deviations from the reference path; the procedure for doing this is discussed in Section 4.6.

Notice that the other two inputs (thrust u_1 and flight path angle u_3) also affect horizontal position, either directly (u_3 also appears in the above equations) or indirectly (through the speed, x_4). The direct effect of u_3 is clearly small: it enters through $\cos(u_3)$ which is second order in u_3 and positive for realistic values of u_3. The effect of x_4 can be substantial; it appears, however, that with 3D FMS (the current standard, see also Section 2) x_4 is set independently of the aircraft's horizontal position. This is clearly something that will have to be changed in our model if 4D FMS become more common in the future.

The values of the inputs u are determined to some extent by a conventional, continuous controller. However, the parameters and set points of this controller depend on a fairly complicated, logic based, decision making process. We refer to the discrete quantities used in this decision making as the *discrete state* of the FMS. We start by describing the dynamics (possible values and transitions) of this discrete state then proceed to show how it is used to determine the inputs and parameters of the FMS.

4.2 Discrete State

The discrete state of the FMS can be represented by 8 discrete variables: flight level (*FL*), way-point index (*WP*), acceleration mode (*AM*), climb mode (*CM*), speed hold mode (*SHM*), flight phase (*FP*), reduced power mode (*RPM*) and troposphere mode (*TrM*).

Flight Level. The discrete variable representing the flight level takes on values representing the following altitude discretisation

$$\{0ft, 500ft, 1000ft, 1500ft, 2000ft, 3000ft, 6000ft, 10000ft, 14000ft\}$$

We use $FL = 0, 1, 2, \ldots, 8$ to denote these discretisation levels. The value of FL is updated based exclusively on the continuous state x_3, the altitude. The dynamics of FL (which should be obvious) are shown in Figure 4.

FL is used primarily to determine the nominal speed, V_{nom}, that the aircraft should use. This is done based on the airline procedure tables discussed in Section 4.3. The above choice of discrete levels is dictated by the airline procedure models contained in BADA.

Way Point Index. The discrete variable representing the way point index takes integer values reflecting the number of way points in the flight plan,

$$WP \in \{0, 1, \ldots, M\}.$$

$WP = i$ implies that the aircraft is on its way between the i^{th} way-point at position $O(i) \in \mathbb{R}^3$ and the $(i+1)^{st}$ way point at position $O(i+1) \in \mathbb{R}^3$.

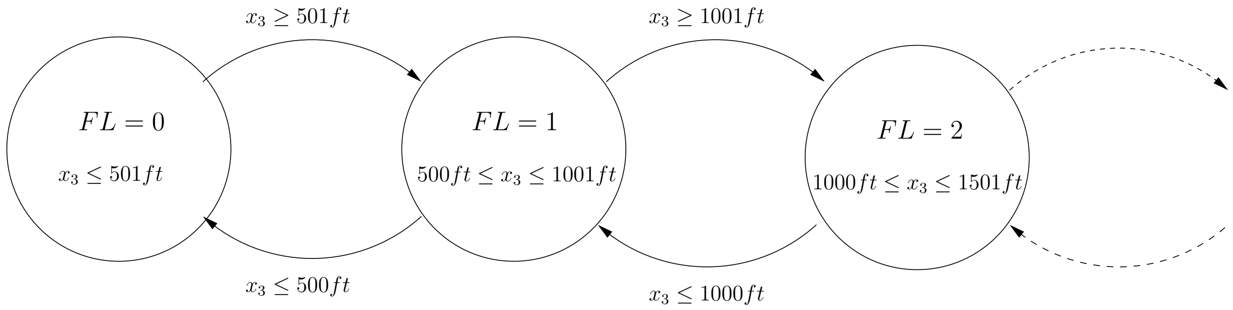


Figure 4: Finite State Machine for FL.

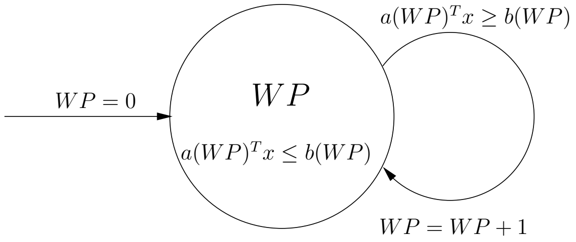


Figure 5: Finite State Machine for WP.

The value of the way point is updated based on the horizontal position of the aircraft, x_1 and x_2. The dynamics are summarised in Figure 5. To determine when the aircraft switches from one way point to the next we adopt a simple model for the way aircraft perform turns. In practice aircraft execute turns from one way point to the next using one of two methods.

- **Fly-past**: The aircraft starts turning before it reaches the next way point and "cuts the corner". This appears to be the preferred method for most modern aircraft.
- **Fly-over**: The aircraft gets all the way to the next way-point before turning. Fly-over typically results in longer trajectories and was apparently common before the Global Positioning System (GPS) became standard, when aircraft depended more heavily on VHF Omni-directional radio Ranging (VOR) points for navigation.

Here we adopt the fly past alternative and assume that turns are nominally executed along arcs of circles. It should be noted that in the presence of wind the actual path relative to the ground will not be the arc of a circle, this assumption is only made to compute the starting point of the turn. The turn geometry in this case is shown in Figure 6. We assume that the aircraft switches to the new way point at the beginning of the turn. To determine when this happens we need to determine the values of all the parameters listed in the figure (the radius r, distance d, etc.) and in particular the equations of the dotted line $a(i)^T x = b(i)$.

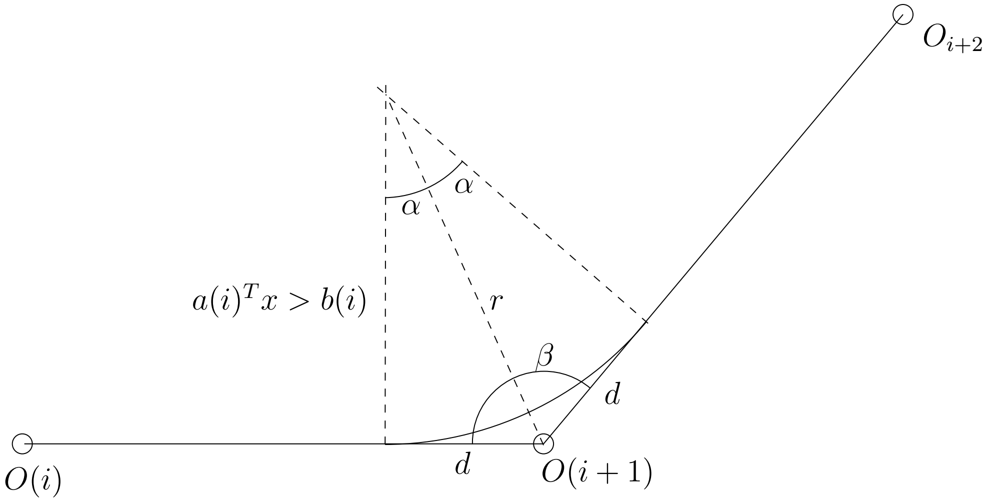


Figure 6: Geometry of turn between way-points assuming zero wind.

We start by computing the turn radius. To do this we assume that turns are executed at a fixed bank angle $\pm\phi_{nom}$. Under the simplifying assumptions introduced in Section 3 the turn rate is then governed by the equation

$$\dot{x}_5 = \frac{C_L S \rho}{2} \frac{x_4}{x_6} \sin(\phi_{nom}). \tag{6}$$

Based on the BADA documentation we take $\phi_{nom} = 35°$. Assuming that the aircraft remains level throughout the turn we equate the component of the lift in the vertical direction to the weight of the aircraft

$$\frac{C_L S \rho}{2} x_4^2 \cos(\phi_{nom}) = g x_6.$$

Substituting into equation (6) leads to

$$\dot{x}_5 = \frac{g \tan(\phi_{nom})}{x_4}.$$

Assuming that the aircraft starts the turn at time 0 with a heading $x_5(0) = \Psi(i)$ and maintains constant speed ($x_4 = V$) throughout the turn, at time t the heading angle is

$$x_5(t) = t \frac{\tan(\phi_{nom}) g}{V} + \Psi(i)$$

while the distance travelled by the aircraft is

$$r(x_5(t) - \Psi(i)) = tV.$$

Dividing the two equations leads to

$$r = \frac{V^2}{g \tan(\phi_{nom})}. \tag{7}$$

It is easy to see from the figure that

$$\beta = \pi - |\Psi(i) - \Psi(i+1)| \text{ and } \alpha = \frac{\pi - \beta}{2} = \frac{|\Psi(i) - \Psi(i+1)|}{2}.$$

The distance, d, from way-point $O(i+1)$ at which the aircraft should begin its turn is therefore given by

$$d = \frac{V^2}{g \tan(\phi_{nom})} \tan\left(\frac{|\Psi(i) - \Psi(i+1)|}{2}\right).$$

We assume that the aircraft begins its turn as soon as it reaches a vertical plane perpendicular to the reference path and passing through the dotted line as shown in Figure 6. The above calculation provides all the information necessary to determine the parameters $a(i) \in \mathbb{R}^6$ and $b(i) \in \mathbb{R}$ that define the equation of that plane

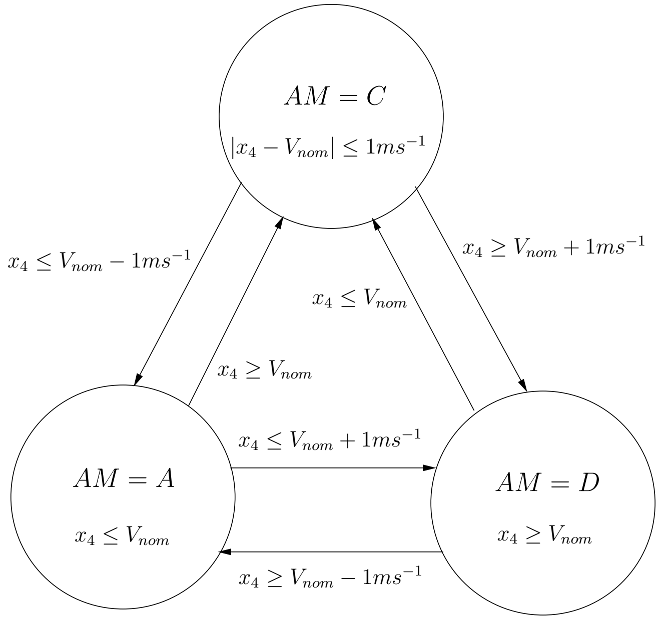
$$a(i)^T x = b(i).$$

Notice that only the first two coordinates of x (the X and Y positions) play a role in determining when the turn should begin, therefore all entries of $a(i)$ will be zero except possibly the first two.

Before moving on, we highlight some subtle features of the calculations and assumptions introduced in this section. One problem with the fly-past model adopted here is that for large turns the aircraft may cut the corner too much. This could be resolved by having the aircraft fly over the way-point when the turn angle is greater than a certain amount. In the current implementation, we prevent aircraft from cutting the corner too much by restricting the distance d in Figure 6 to be at most twice the distance r. This results in the aircraft overshooting the next flight segment if the turn angle is too great. The controller used to set the bank angle in Section 4.6 can easily correct for this deviation.

An advantage of the adopted approach is that the nominal turn trajectory with zero wind is easy to compute and is composed of straight lines and arcs of circles. This is desirable because straight line and circular paths both correspond to trimmed flight conditions [5], which commercial aircraft tend to use in their maneuvers. With a non-zero wind the turn trajectory relative to he ground will no longer be the arc of a circle.

A useful consequence of the turning procedure assumed here is that in many cases cross track errors will tend to be reduced during the turn. Aircraft flying closer to the centre of the turn and turning using the nominal bank angle ϕ_{nom} will exit the turn faster, having turned through a smaller angle. Similarly aircraft further away from the centre of the turn will turn for longer. In either case, aircraft will end up closer to the reference trajectory at the end of the turn than they were at the beginning of the turn. Note that this observation is only valid for small deviations and small turning angles. For example, in a turn of 90 degrees with the aircraft starting on the outside of the reference path, the aircraft will turn for too long, cross the reference path and end up on the other side facing away from the reference path.



$AM = C$
$|x_4 - V_{nom}| \leq 1ms^{-1}$
$x_4 \leq V_{nom} - 1ms^{-1}$
$x_4 \geq V_{nom} + 1ms^{-1}$
$x_4 \leq V_{nom}$
$x_4 \geq V_{nom}$
$x_4 \leq V_{nom} + 1ms^{-1}$
$AM = A$
$x_4 \leq V_{nom}$
$AM = D$
$x_4 \geq V_{nom}$
$x_4 \geq V_{nom} - 1ms^{-1}$

Figure 7: Finite State Machine for AM.

Acceleration Mode. The discrete variable for the acceleration mode reflects whether the aircraft is accelerating, decelerating or cruising at constant speed,

$$AM \in \{A, D, C\}.$$

The value is reset whenever the desired speed of the aircraft changes. This may be the case, for example, when the aircraft changes flight level, or the climb mode state changes. The state returns to C when the aircraft reaches the desired speed. A buffer of $1ms^{-1}$ is introduced to prevent chattering.

Climb Mode. The climb mode reflects whether the aircraft is climbing, descending or flying level

$$CM \in \{C, D, L\}.$$

The value is reset whenever the aircraft starts a new segment of the reference path. If $WP = i$ and $O(i+1) = (x, y, z)$ the difference between the present altitude, x_3 and z is used to determine whether to climb ($CM = C$) or descent ($CM = D$). The state returns to L when the aircraft reaches its desired altitude, z. A buffer of $1m$ is introduced to prevent chattering.

Speed Hold Mode. The speed hold mode represents whether the aircraft is above or below the transition altitude, which is the altitude at which the aircraft changes from

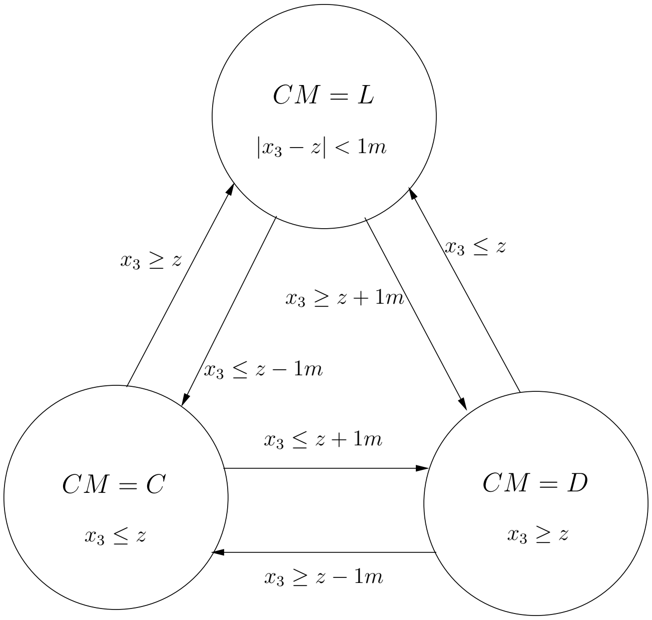


Figure 8: Finite State Machine for CM.

holding constant Calibrated Airspeed (CAS) to holding constant Mach (M). The transition altitude is the altitude where the desired true airspeed of the aircraft determined by CAS is equal to the desired true airspeed determined by the M number specified in the BADA database. This is to ensure that the mach number essentially imposes an upper limit on the aircraft speed.

The speed hold mode takes on two values

$$SHM \in \{C, M\}.$$

The value is exclusively determined by the altitude of the aircraft: $SHM = C$ if x_3 is less than the transition altitude, $SHM = M$ if x_3 is above the transition altitude. The speed hold mode determines the energy share factor in certain climb and acceleration mode combinations (in particular, when $AM = C$ and $CM \in \{C, D\}$).

Troposphere Mode. The troposphere mode represents whether the aircraft is above or below the tropopause. This affects some variables in the atmosphere model such as air density. The tropopause is the boundary between the troposphere and the stratosphere. The troposphere mode takes on two values

$$TrM = \{L, H\}.$$

$TrM = L$ if x_3 is below 11000 feet and $TrM = H$ if x_3 is above 11000 feet.

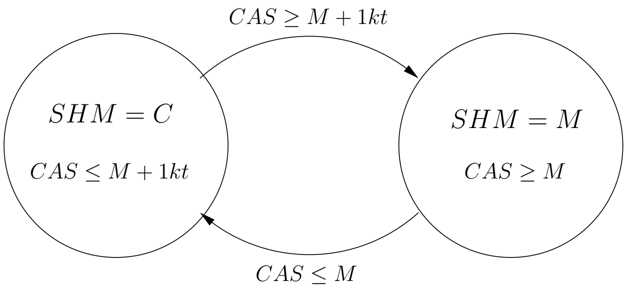


Figure 9: Finite State Machine for SHM. CAS represents the nominal airspeed based on the calibrated airspeed schedule. M represents the airspeed based on the Mach number

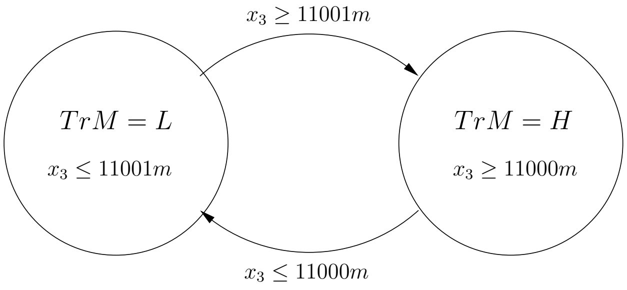


Figure 10: Finite State Machine for TrM.

Flight Phase. From the point of view of the flight management system the descent portion of flight is divided into 4 phases: upper descent (*UD*), lower descent (*LD*), approach (*AP*) and landing (*LA*).

$$FP \in \{UD, LD, AP, LA\}.$$

The rules for changing the values of *FP* are summarised in Figure 11. $V_{minapproach}$, $V_{mincruise}$ and *hdes* are parameters that depend on *Aircraft_Type* and can be obtained from the BADA database.

FP is used to set the thrust input. As discussed in Section 3.2, it also has an effect on the dynamics of the aircraft. For example during approach and landing the aircraft will have a high lift configuration which will also increase drag. When the aircraft is not descending, i.e. when $CM \in \{C, L\}$, *FP* has no effect on the dynamics.

Reduced Power. The reduced power is a state that represents whether the aircraft should be climbing with reduced power or not. We simply set

$$RPM \in \{ON, OFF\}$$

The transitions of *RPM* are summarised in Figure 12. This state is dependent only on altitude. The reduced climb power has been introduced to allow climbs using less than

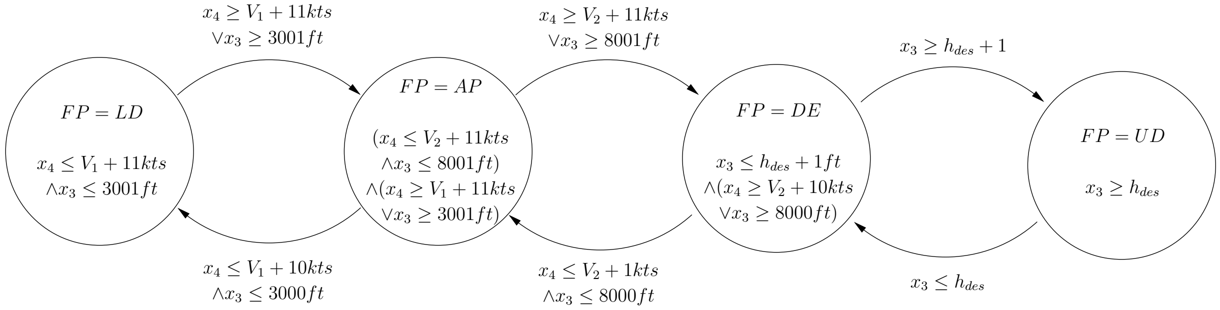


Figure 11: Finite State Machine for FP. V_1, V_2 represent $V_{min,approach}$ and $V_{min,cruise}$ which are parameters from BADA.

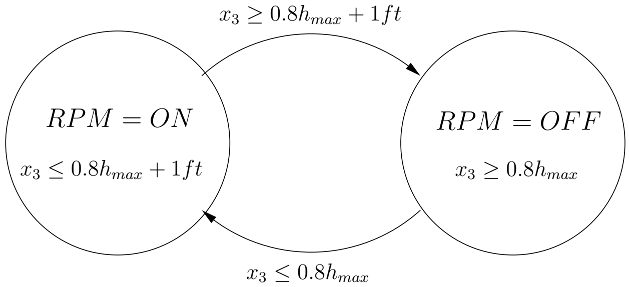


Figure 12: Finite State Machine for RPM.

the maximum climb setting. In day-to-day operation, many aircraft use a reduced setting during climb to extend engine life and save cost. See [2], Section 3.8 for more details.

4.3 Desired Speed Settings

The desired true airspeed of the aircraft, V_{nom}, is set based on the airline procedure models provided in the BADA documentation. The value of V_{nom} depends on the discrete parameter *Aircraft_Type* and on the discrete states *FL*, *CM* and *SHM*. If *SHM* $= M$, V_{nom} is set equal to a mach number which depends on *Aircraft_Type* and whether the aircraft is climbing or descending (*CM*). If *SHM* $= C$, V_{nom} is set equal to a calibrated airspeed which depends on *Aircraft_Type*, *FL* and*CM*. The values of V_{nom} for the different values of *FL*, *CM* and *SHM* and all aircraft types supported by BADA are obtained from the BADA database.

4.4 Thrust Settings

The setting of the thrust depends on the discrete states *AM*, *CM*, *FP* and on the discrete parameter *Engine_Type*. First, *Engine_Type* is used to set the value of a parameter

$T_{MaxClimb}$ which depends on the aircraft altitude and speed.

$$T_{MaxClimb} = \begin{cases} C_{Tc1}\left(1 - \frac{x_3}{C_{Tc2}} + C_{Tc3}x_3^2\right) & \text{if } \textit{Engine\_Type}\text{=Jet} \\ \frac{C_{Tc1}}{x_4}\left(1 - \frac{x_3}{C_{Tc2}}\right) + C_{Tc3} & \text{if } \textit{Engine\_Type}\text{=Turboprop} \\ C_{Tc1}\left(1 - \frac{x_3}{C_{Tc2}}\right) + \frac{C_{Tc3}}{x_4} & \text{if } \textit{Engine\_Type}\text{=Piston} \end{cases}$$

The values for the parameters C_{Tc1}, C_{Tc2} and C_{Tc3} depend on *Aircraft_Type* and are obtained from the BADA database.

$T_{MaxClimb}$ is then used to set the thrust input u_1 based on the discrete states *CM*, *FP*, *RPM* and *AM*

$$u_1 = \begin{cases} T_{MaxClimb} & \text{if } (CM = C) \wedge (RPM = OFF) \\ T_{MaxClimb}C_{powred} - \frac{C_D S \rho}{2} x_4^2 (1 - C_{powred}) & \text{if } (CM = C) \wedge (RPM = ON) \\ C_{Tdes}(FP)\, T_{MaxClimb} & \text{if } [CM = D] \vee [(CM = L) \wedge (AM = D)] \\ 0.95 T_{MaxClimb} & \text{if } (CM = L) \wedge (AM = A) \\ \frac{C_D S \rho}{2} x_4^2 & \text{if } (CM = L) \wedge (AM = C) \end{cases}$$

The values for the parameters $C_{Tdes}(FP)$ for the different values of *FP* depend on *Aircraft_Type* and are obtained from the BADA database.

4.5 Flight path angle/Energy Share factor settings

Following BADA we use the ESF, $\hat{u}_3$, instead of the flight path angle, u_3, to control vertical motion. For the most part the ESF is set to a constant value, as shown on Table 2. When $AM = C$ and $CM = C$ or D the ESF is set to ensure that the aircraft speed x_4 tracks the nominal speed V_{nom}. The ESF used in this case depends on the discrete states *FL*, *SHM* and *TrM*. The dependence is omitted since it can be found in the BADA documentation.

	AM=A	*AM*=C	*AM*=D
CM=C	0.3	Track V_{nom}	0.7
CM=L	0	0	0
CM=D	0.7	Track V_{nom}	0.3

Table 2: Settings for the energy share factor

4.6 Bank Angle Control

Our model assumes that the FMS sets the bank angle based on the heading error and the cross track deviation from the reference path. The controller operates in continuous

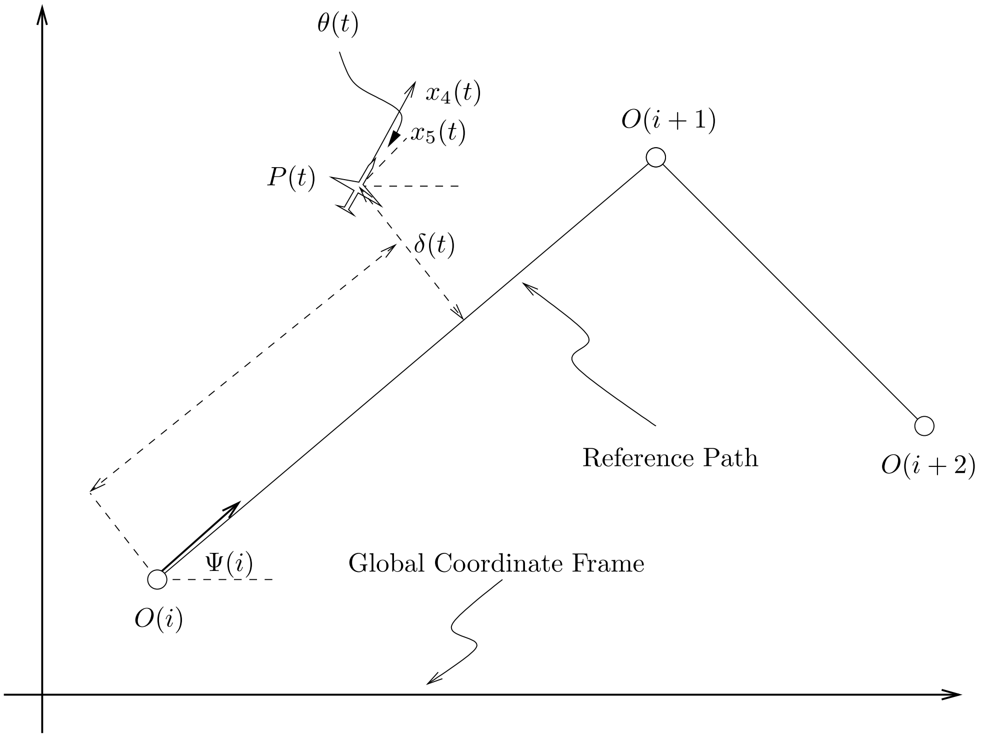


Figure 13: Geometry for the calculation of cross track and heading errors.

time and consists of a linear feedback part, followed by non-linearities to ensure that the behaviour is reasonable even with extreme inputs. Such extreme inputs may arise in sharp turns dictated by the flight plan, especially in the interactive version of the simulator, where the operator can move the way points at will. Figure 13 shows the geometry of the situation.

At time t, let $P(t) = (x_1(t), x_2(t), x_3(t))$ denote the position of the aircraft and assume that the aircraft is moving between way-point i and way-point $i+1$, i.e. $WP = i$. Then the course error is

$$\theta(t) = \Psi(i) - x_5(t) \tag{8}$$

and the cross-track deviation from the reference path is

$$\delta(t) = [\ -\sin(\Psi(i)) \quad cos(\Psi(i)) \quad 0\ ](P(t) - O(i)). \tag{9}$$

The linear part of the controller is given by

$$\phi_1(t) = k_1\delta(t) + k_2\theta(t).$$

The gains k_1 and k_2 are arbitrary at this stage; methods for setting their values are discussed in subsequent sections.

The linear controller may command unrealistically large bank angles when faced with large deviations from the reference path. To prevent this we introduce a saturation on ϕ_1

at some angle $\overline{\phi} \geq 0$.

$$\phi_2(t) = \begin{cases} -\overline{\phi} & \text{if } \phi_1(t) \leq -\overline{\phi} \\ \phi_1(t) & \text{if } -\overline{\phi} \leq \phi_1(t) \leq \overline{\phi} \\ \overline{\phi} & \text{if } \phi_1(t) \geq \overline{\phi}. \end{cases} \tag{10}$$

At the moment, $\overline{\phi}$ is taken the same as the nominal turning angle ϕ_{nom} used to compute the nominal turning radius in Section 4.2, i.e. $\overline{\phi} = \phi_{nom} = 35°$.

Even with the saturation, due to the non-linear nature of the system the aircraft may not converge to the reference path if it deviates too far; instead it may travel in circles. To prevent this we introduce a further limit on the bank angle as a function of the heading error. This leads to the final setting for the bank angle

$$u_2(t) = \begin{cases} \min\{\phi_2(t), 0\} & \text{if } \pi/2 \geq x_5(t) \geq \overline{\psi} \\ \max\{\phi_2(t), 0\} & \text{if } -\pi/2 \leq x_5(t) \leq -\overline{\psi} \end{cases} \tag{11}$$

Notice that the same controller is used to track the straight segments of the reference path and for turning. When the aircraft approaches the next way point, the value of *WP* changes as discussed in Section 4.2. As a consequence the value of i changes in equations (8) and (9) which leads to a step change in the inputs $\delta(t)$ and $\theta(t)$ of the controller. The controller then adjusts u_2 to track the new straight segment of the reference path.

Stability A minimum requirement to ensure that the bank angle controller developed above produces realistic results is that it is stable. A full stability study of this controller is complicated because

- Both the aircraft model and the controller are nonlinear.
- The settings for the thrust and flight path angle affect the performance of the bank angle controller but are not controlled by it and therefore have to be treated as disturbances.
- The system is hybrid, due to the discontinuities δ and θ when the value of *WP* changes.
- The system is stochastic because of the uncertainty in the wind speed w.

Such a full blown stability analysis is beyond the scope of D1.3. Here we investigate the stability of the controller under a number of simplifying assumptions:

- Level flight, i.e. $u_3(t) = 0$.
- Constant speed, i.e. $x_4(t) = V$.

- Straight reference path, i.e. WP is constant.
- Constant wind.

We use linearisation to determine bounds on the gains k_1 and k_2 to ensure local asymptotic stability.

Without loss of generality we assume that the reference path is aligned with the x_1 direction. In this case only two states, the lateral position of the aircraft, x_2, and the heading angle, x_5, are important in our analysis. They are governed by the equations:

$$\dot{x}_2 = V\sin(x_5) + w_2 \text{ and } \dot{x}_5 = \frac{C_L S\rho}{2}\frac{x_4}{x_6}\sin(u_2). \tag{12}$$

Setting C_L according to the discussion in Section 3.2 and substituting the linear part of the bank angle controller, the second equation becomes

$$\dot{x}_5 = \frac{g}{V}\tan(k_1 x_2 + k_2 x_5).$$

We find the equilibrium point $(\overline{x}_2, \overline{x}_5)$ by setting $\dot{x}_2 = \dot{x}_5 = 0$. This leads to

$$\overline{x}_5 = -\sin^{-1}\left(\frac{w_y}{v}\right) \text{ and } \overline{x}_2 = \frac{k_2}{k_1}\sin^{-1}\left(\frac{w_y}{v}\right).$$

Taking partial derivatives of equations (12) about this equilibrium leads to the linearisation

$$\begin{bmatrix} 0 & V\cos\left(\sin\left(\frac{w_y}{v}\right)\right) \\ \frac{g}{V}k_1 & \frac{g}{V}k_2 \end{bmatrix}.$$

For stability we need the eigenvalues of this matrix to be negative. This means that we have to ensure that

$$k_1 < 0 \text{ and } k_2 < 0.$$

Notice that unless $w_2 = 0$ the system will converge to an equilibrium with $x_2 \neq 0$ and $x_5 \neq 0$. This means that the aircraft will converge exactly to the reference path only if the wind is parallel to the reference path. If, however, the wind has a (constant) non-zero component orthogonal to the reference path then there will be residual steady state errors in heading and cross track deviation. If deemed necessary the inclusion of integral action in the linear part of the controller would be a straightforward way to eliminate this steady state error and force the aircraft to track the reference path exactly even in the presence of wind with non-zero orthogonal component. We have decided to not implement integral action at this stage, to simplify the process of tuning the model to match empirical cross track deviation statistics. Absence of integral control implies there is one less parameter to tune and makes it easier to generate the linear growth in the variance of the cross track deviations observed in practice; see section 7.2.

5 Wind Model

The wind is modelled as two components, nominal and stochastic. The nominal wind will be modelled as a look-up table, similar in temporal and spatial resolution to meteorological data available to air traffic controllers. At the moment, however, the nominal wind is assumed to be zero and all wind is considered to be stochastic.

The Stochastic wind is modelled as jointly Gaussian random variables with a correlation structure in both time and space. The algorithm we use to compute this random field and the assumptions needed are described in Section 6.1. We also assume that the stochastic field has zero mean.

To gain some insight into the correlation structure for the stochastic component we have gained access to Rapid Update Cycle (RUC) wind data. This is used in the USA, and is freely available over the Internet. This is covered in more detail in Section 7.1.

6 Simulation

6.1 Wind Correlation

The stochastic wind component is modelled as a random field:

$$w : \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}^3$$

where $w(t, P)$ represents the wind at point $P \in \mathbb{R}^3$ at time $t \in \mathbb{R}$.

To ensure we can find an efficient algorithm to create wind fields we make the following assumptions.

Assumption 6.1 *Wind, $w(\cdot, \cdot)$, has the following properties:*

1. *$w(t, P)$ is a Gaussian random variable with mean $\mu(t, P)$ and covariance matrix $\Sigma(t, P)$.*
2. *The random field is Isotropic in x and y, i.e. the correlation structure is preserved under rotations in the horizontal plane.*
3. *$w_1(t, P)$, $w_2(t, P)$, $w_3(t, P)$ are independent of one another for all $t \in \mathbb{R}, P \in \mathbb{R}^3$*

Under these assumptions one can show the following.

Proposition 6.2 *Let $\rho(t, P, t', P') = E[(w(t, P) - \mu(t, P))(w(t', P') - \mu(t', P'))^T]$ for $t, t' \in \mathbb{R}$ and $P, P' \in \mathbb{R}^3$. Then*

$$\rho(t, P, t', P') = \begin{pmatrix} \rho_{xy}(t, P, t', P') & 0 & 0 \\ 0 & \rho_{xy}(t, P, t', P') & 0 \\ 0 & 0 & \rho_z(t, P, t', P') \end{pmatrix}$$

It appears that not much is known about the properties of the function ρ that governs the correlation of wind over space and time. A study of its properties based on RUC data will be given in Section 7.1.

We now consider how this correlation structure can be implemented in simulation. Consider N aircraft in some region of airspace. Let $P_i(t) \in \mathbb{R}^3$ denote the position of the i-th aircraft at time t, i.e. $P_i(t)$ consists of the x_1, x_2 and x_3 components of the state of aircraft i at time t. The procedure discretises the wind computation in intervals of Δ seconds; currently $\Delta = 15$ seconds is used. Let

$$t_0 = 0 \text{ and } t_k = k\Delta.$$

At each step k we start with the aircraft positions

$$P_1(t_k), \ldots, P_N(t_k)$$

and compute wind speeds for each aircraft

$$w(t_k, P_i(t_k))$$

using the procedure outlined below. We then integrate the aircraft and FMS models over the interval $[t_k, t_{k+1}]$ keeping the wind constant, i.e.

$$w(t, P_i(t)) = w(t_k, P_i(t_k)) \text{ for all } t \in [t_k, t_{k+1}].$$

The process is then repeated.

Proposition 6.2 enables us to simplify the calculation as the wind vectors are decoupled, so we can consider the three components of $w(t, P)$ separately. We need only calculate wind velocities at the current aircraft positions so to simplify notation let:

$$P_k = \begin{pmatrix} P_1(t_k) \\ \vdots \\ P_N(t_k) \end{pmatrix} \in \mathbb{R}^{3N}$$

$$w_k = \begin{pmatrix} w(t_k, P_1(t_k)) \\ \vdots \\ w(t_k, P_N(t_k)) \end{pmatrix} \in \mathbb{R}^{3N}$$

$$\mu_k = \begin{pmatrix} \mu(t_k, P_1(t_k)) \\ \vdots \\ \mu(t_k, P_N(t_k)) \end{pmatrix} \in \mathbb{R}^{3N}$$

$$\rho_{i,j} = \begin{pmatrix} \rho(t_i, P_1, t_j, P_1) & \ldots & \rho(t_i, P_1, t_j, P_N) \\ \vdots & & \vdots \\ \rho(t_i, P_N, t_j, P_1) & \ldots & \rho(t_i, P_N, t_j, P_N) \end{pmatrix} \in \mathbb{R}^{3N \times 3N}.$$

The following fact is easy to establish.

Proposition 6.3 *Under Assumption 6.1 the matrix $\rho_{i,i}$ is symmetric.*

We show by induction that one can find matrices $B_{i,j} \in \mathbb{R}^{3N \times 3N}$ such that for all k the vectors

$$w_k = B_{k,k} n_k + B_{k,(k-1)} n_{k-1} + \ldots + B_{k,0} n_0 + \mu_k$$

have the desired correlation structure

$$E((w_i - \mu_i)(w_j - \mu_j)^T) = \rho_{i,j}$$

where $n_k \in \mathbb{R}^{3N}$ are independent, identically distributed with $n_k \sim N(0, I)$.

At time t_0 we set

$$w_0 = B_{0,0}n_0 + \mu_k$$

with $n_0 \sim N(0, I)$. Clearly

$$\begin{aligned} E((w_0 - \mu_0)(w_0 - \mu_0)^T) &= E(B_{0,0}n_0(B_{0,0}n_0)^T) \\ &= E(B_{0,0}n_0n_0^T B_{0,0}^T) \\ &= B_{0,0}B_{0,0}^T. \end{aligned}$$

We want $E((w_0 - \mu_0)(w_0 - \mu_0)^T) = \rho_{0,0}$. Assuming $\rho_{0,0}$ is positive definite, the Cholesky decomposition of $\rho_{0,0}$ to find $B_{0,0}$.

At time t_k we have already computes the 3N dimensional vectors

$$P_0, P_1, \ldots, P_k, n_0, n_1, \ldots n_{k-1}$$

We also have the $3N \times 3N$ matrices

$$\begin{matrix} B_{0,0} & & & \\ B_{1,0}, & B_{1,1} & & \\ \vdots & \vdots & \ddots & \\ B_{k-1,0}, & B_{k-1,1} & \ldots, & B_{(k-1)(k-1)} \end{matrix}$$

We set

$$w_k = B_{k,k}n_k + B_{k,(k-1)}n_{k-1} + \ldots + B_{k,0}n_0 + \mu_k$$

with $n_k \sim N(0, I)$. We wish to calculate $B_{k,1}, \ldots, B_{k,k}$ such that w_k will respect wind statistics. First we calculate $B_{k,0}$.

$$\begin{aligned} E\left[(w_k - \mu_k)(w_0 - \mu_0)^T\right] &= E\left[\left(B_{k,k}n_k + B_{k,k-1}n_{k-1} + \ldots + B_{k,0}n_0\right)\left(B_{0,0}n_0\right)^T\right] \\ &= E\left[\left(B_{k,k}n_k + \ldots + B_{k,0}n_0\right)\left(n_0^T B_{0,0}^T\right)\right] \\ &= B_{k,0}B_{0,0}^T, \end{aligned}$$

since $E(n_i n_j^T) = 0$ if $i \neq j$ and $E(n_i n_j^T) = 1$ if $i = j$ (recall that $n_i \in \mathbb{R}^3$ are independent, identically distributed with $n_i \sim N(0, I)$). Therefore,

$$B_{k,0} = \rho_{k,0}\left(B_{0,0}^T\right)^{-1}$$

(recall that $B_{0,0}$ is invertible by construction.)

Next, we calculate $B_{k,i}$ recursively on i.

$$E\left((w_k - \mu_k)(w_i - \mu_i)^T\right) = E\left[\left(B_{k,k}n_k + \ldots + B_{k,0}n_0\right)\left(B_{i,i}n_i + \ldots + B_{i,0}n_0\right)^T\right]$$

$$= \sum_{l=0}^{i} B_{k,l} B_{i,l}^T,$$

since $n_i \in \mathbb{R}^3$ are independent, identically distributed with $n_i \sim N(0, I)$. Therefore

$$B_{k,i} = \left[\rho_{k,i} - \sum_{l=0}^{i-1} B_{k,l} B_{i,l}^T \right] (B_{i,i}^T)^{-1}.$$

Finally we calculate $B_{k,k}$.

$$E\left((w_k - \mu_k)(w_k - \mu_k)^T \right) = \sum_{l=0}^{k} B_{k,l} B_{k,l}^T \implies B_{k,k} B_{k,k}^T = \rho_{k,k} - \sum_{l=0}^{k-1} B_{k,l} B_{k,l}^T.$$

Assuming that the matrix

$$\rho_{k,k} - \sum_{l=0}^{k-1} B_{k,l} B_{k,l}^T$$

is positive definite (it is clearly symmetric) we can compute $B_{k,k}$ by its Cholesky decomposition.

Notice the above computation relies on certain matrices being positive definite. Whether or not the matrices have this property has to do with the structure of the correlation function ρ and the interval Δ used. In our experience, this condition is met for the function ρ discussed in the next section and $\Delta = 15$.

The above computation produces all three components of the wind speed at the same time. The computation can also be decoupled into three separate computations for the w_1, w_2 and w_3 components. This offers computational advantages. It may also be necessary in cases like the one discussed in the next section, where the w_3 component is set to zero (in this case some matrices would become singular in the joint w_1, w_2, w_3 computation). An additional saving comes from the fact that for the w_1 and w_2 components of the wind the same $B_{i,j}$'s can be used but with different random samples.

$$\begin{aligned} w_{1k} &= B_{k,k} n_{xk} + B_{k,(k-1)} n_{xk-1} + \ldots + B_{k,0} n_{x0} + \mu_{xk} \\ w_{2k} &= B_{k,k} n_{y_k} + B_{k,(k-1)} n_{y_{k-1}} + \ldots + B_{k,0} n_{y_0} + \mu_{yk} \end{aligned}$$

This is because the wind is assumed to be isotropic in the horizontal plane. However the vertical component of the wind may have a completely different correlation function in which case a new set of $B_{i,j}$'s will need to be calculated.

6.2 Discrete Transition Uncertainty

The code also allows for the implementation of a discrete transition uncertainty about when an aircraft begins its descent to the airport. The uncertainty is assumed to be governed by a transition rate function, $\lambda : \mathbb{R}^6 \to \mathbb{R}^+$, like the one used in Piecewise Deterministic Markov Processes. This gives the instantaneous probability of transition. The probability of survival in a period $[t_0, t_1]$ will be given by $\exp\left(-\int_{t_0}^{t_1} \lambda(x(t))dt\right)$. This is approximated at each time step by $\exp\left(-\lambda(x(t_0))\delta t\right)$. The transition rate function can be chosen such that it continually increases after some initial location. There will also be a boundary specifying the latest time for which the aircraft can descend which will be a forced transition, as opposed to a spontaneous transition (see [1], [10].

6.3 Numerical Aspects

The computer simulation program simulates the flight of an arbitrary number of aircraft. Each aircraft has a set of three dimensional way-points and a departure time. The aircraft is modelled in the system from its departure time until it reaches its last way-point. The aircraft can have any type contained within the BADA database. The system is solved using the Runge-Kutta 4,5 with variable time step and with discrete transitions.

The wind field is calculated every n seconds. “Older” wind fields are discarded after a m samples. There is a trade off between the speed and memory usage of the simulation and accuracy of the wind field governed by m and n. For large numbers of aircraft the speed of the algorithm could be further improved by using a different correlation function that decreases to zero after a certain distance. This would lead to a sparse correlation matrix. Sparse matrix techniques could then be used to decrease the cost of multiplying matrices together and of the Cholesky decomposition. This has not been implemented.

The program can read way point data specified either in a native format or from data obtained from CFMUfiles. We have access to two days of aircraft flight plans from CFMU. The CFMU data can be used to produce realistic conflict geometries. The aircraft trajectories resulting from simulations can then be used to verify conflict prediction algorithms.

6.4 Usage

To run the program it is essential to have a Java Runtime Environment (JRE)installed. These can be downloaded from the sun Web site (java.sun.com). The more recent JRE are much faster than the older versions. To run the Graphical User Interface (GUI)mode of the program type “java ModelInterface” in the command line. To run the non-interactive mode type “java nonLinearModel *filename*”. Where *filename* is the name of the file containing the way-point data.

The program must have information on aircraft parameters. This can either be direct from the BADA database, or can be in a custom file format. The custom file format allows one to use the program without access to the BADA database by supplying the aircraft parameters.

File Formats The output of the program is saved in files named "trajectory*n*.txt" where n is a number from 0 upwards. Each file represents the trajectory of one aircraft. Each line in each file represents the position of the aircraft at a certain time. Each line contains 4 fields separated by spaces. The first field is the time in seconds. The next three fields are the x, y, z coordinates in metres.

The flightplan file consists of a series of tokens and values separated by newlines. Each line contains a pair of strings separated by a ":" character. The first is the variable name and the second is the value of the variable. *aircraft* is a four letter code designating the aircraft type. This is either the code used to identify an aircraft in the BADA database or the code used in the custom parameter file. DepartTime is the time in minutes from the beginning of the simulation that the aircraft appears at the first way-point. Px, Py, Pz are the x, y and z coordinates of each way-point in metres.

The custom aircraft parameter file is similar in format to the flightplan file. Instead of the positions of way-points it specifies values of parameters. The names of the parameters are similar to the names used in BADA. See the example file included with the program for more information.

Using the Interactive Simulator The interactive simulator starts with three windows: The main window, used for visualising flight plans, the table window, used to give a textual representation of the flight plans and the Simulation Control Window.

The main window. The main window consists of a menu bar, a tool bar and the flight plan view which takes up the majority of the window. The flight plan view shows the flight plans of the aircraft as a series of circles representing the positions of each way-point connected by lines and arcs representing the predicted path of each aircraft. The radius of the arc segments is determined by assuming the aircraft has reached the height of the way-point when it passes it, so it will not always be accurate. The menu options File−new, File−open, File−save in the main window are self explanatory. The slider is used to adjust the level of zoom, i.e. the scale of the flight plan in the main window. To edit the way-points of an aircraft first of all select the aircraft from the combo-box. The selected flight plan will be coloured black, whereas the others will be coloured grey. To add a new way point to the end of a flight plan just click in the main window. To move a way point click and drag the way point. Only way points from the selected flight plan can be changed. To add a new aircraft click on the button labelled "add new aircraft".

The table window. The table window shows a table of the way points for the selected aircraft. The position in metres for x, y and z coordinates of each way point is shown in the table and above the table is a field for the departure time. All of the fields can be edited by clicking on them.

The Simulation Control Window. This window contains just two buttons: The start/stop button and the pause/resume button. Both buttons are fairly self explanatory.

7 Tuning and Evaluation

7.1 Wind field data from RUC

There is up to date wind data provided by the RUC, a weather system operated by the National Center for environmental Prediction (NCEP) for the National Oceanic and Atmosphere Administration (NOAA) in the U.S.A. This data is available at 37 flight levels with a spatial resolution of 40km×40km laterally and covers the entire United States. It contains substantial data for each grid point including the horizontal wind direction. We were not able to find data regarding vertical wind speed so for this section we will assume that $w_3 = 0$.

Using this data we can try to estimate the correlation function. In this section we discuss how this can be done based on Assumption 6.1 and the following.

Assumption 7.1 *Let $P = (x, y, z)$ and, as in Proposition 6.2, let $\mu : \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}^3$ denote the mean and $\rho : \mathbb{R} \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ the correlation function of the wind random field. We assume that the wind random field is stationary in x, y and t and has a time-space separable structure, i.e.*

$$\begin{aligned} \mu(t, P) &= \mu_z(z) \\ \rho_{xy}(t, P, t', P') &= R_\tau(|t - t'|) R_{xy}\left(\left\| \begin{pmatrix} x - x' \\ y - y' \end{pmatrix} \right\|\right) R_z(z, z') \\ \rho_z(t, P, t', P') &= 0. \end{aligned}$$

This allows us to do independent analysis to find R_τ, $R_x y$ and R_z; without this assumption we would not have enough data to produce reasonable results. In all figures we distinguish the north-south (u) from the east-west (v) component of the wind.

We find the mean and variance of the wind speed as a function of altitude by using all of the available grid points. The results of this are shown in Figure 14. Notice that the data suggests that the wind violates the isotropic assumption made in the previous section.

To find the covariance as a function of distance we use discrete bins of size 50km. We can find the covariance by taking random pairs of samples at the same altitude and time since iterating over all possible samples would take too long. The results for the covariance calculations are shown in Figures 15, 16 and 17.

The figures show that, as expected, the wind statistics depend very strongly on altitude and that the wind field is not really isotropic in the horizontal plane: there is a marked difference between the north-south and east-west directions. The graphs also show some strange characteristics such as negative correlation in Figure 15. This is possibly genuine. For example, typical weather patterns may involve vortices and weather fronts that cause the wind to be negatively correlated over certain distances. It is more likely that the wind is inhomogeneous and that this is causing the apparent negative correlation. So, the wind is typically stronger in some areas than others. It should be noted that this is unlikely to be much of a problem in our case, since the distance at which the negative correlation occurs is approximately 30,000km. The maximum scale we would be interested for conflict detection and resolution purposes is around 1000km. Over this range the correlation can be closely approximated by the exponential function

$$\rho_{xy}(t, P, t', P') = \sigma(z)\sigma(z') \exp\left(-\lambda|t - t'|\right) \exp\left(-\beta \left| \begin{array}{c} x - x' \\ y - y' \end{array} \right|\right) \exp\left(-\gamma|z - z'|\right) \qquad (13)$$

(recall that we assume that $\rho_z(t, P, t', P') = 0$). The deviation from the exponential is possibly due to variation in average wind velocity during night and day, i.e. the wind velocity not being homogeneous in time. The correlation also seems to increase with altitude. The correlation appears to be quite substantial.

The mean of the v-component of the average wind velocity is small so we can model it as being zero. The mean of the u-component can be modelled fairly accurately with a cubic function of altitude. The same applies for the standard deviation of both the u and v components of the wind.

The correlation as a function of time difference for u and v wind components can be closely approximated as a negative exponential function to within 0.03. The correlation as a function of altitude difference can also be closely modelled by the exponential function. The correlation as a function of horizontal separation appears to be more complicated but over short distances the exponential function is adequate.

Even though the correlation functions all appear to be fairly complicated, by approximating them with the exponential function we can at least get some idea of amount of correlation. The correlation in all three cases is a lot greater than expected.

We have obtained useful results from the RUC data but to extend it further we would need a finer grid of data, possibly obtained from measurements from aircraft. A greater range of times and data covering the entire world would also help create more representative results. Formal system identification methods for random fields could be used to gain more insight into the correlation structure, if it is decided this is worth pursuing further.

The figures for λ, β and γ that are used in the simulator are $\lambda = 6e^{-6}m^{-1}$, $\beta = 1.6e^{-6}m^{-1}$ and $\gamma = 1.5e^{-5}m^{-1}$.

7.2 FMS

In this section we find suitable values for the FMS parameters k_1 and k_2. We use the results of the previous section and compare the predictions of our model to statistics from [9, 4, 11, 6].

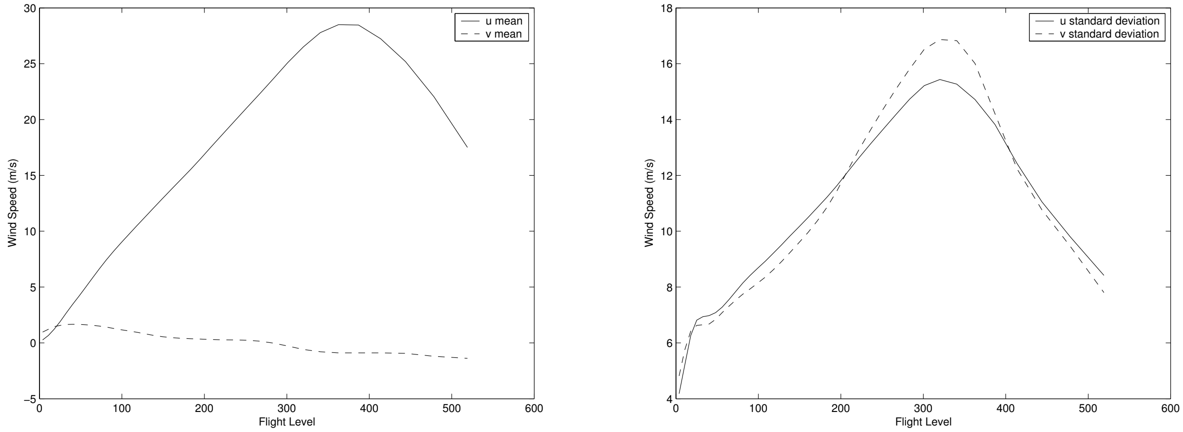


Figure 14: Mean and variance for wind in both u and v components as a function of altitude.

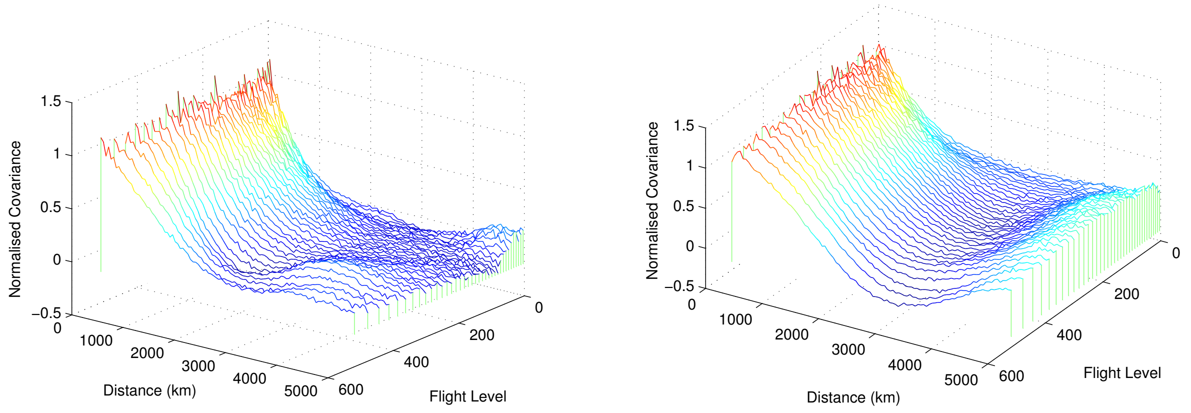


Figure 15: Covariance of u and v components respectively as a function of horizontal separation and altitude

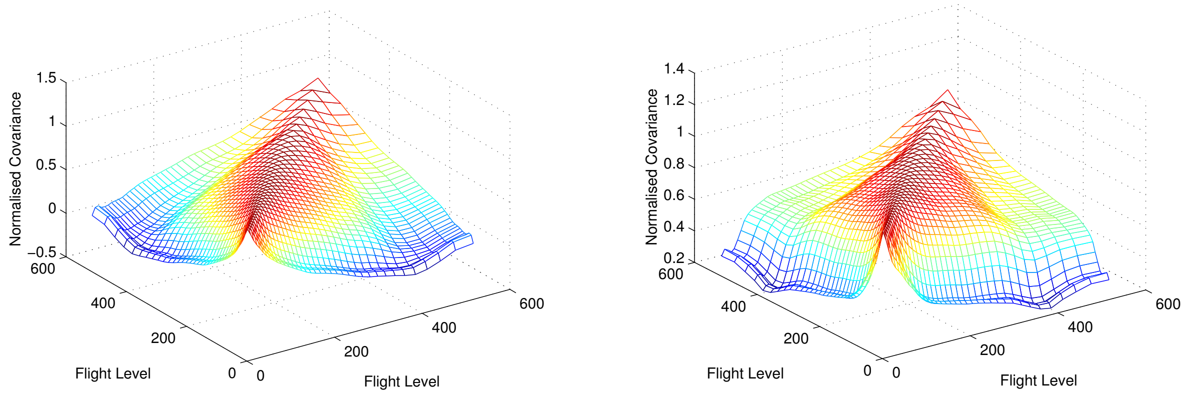


Figure 16: Covariance of u and v components respectively as a function of altitude

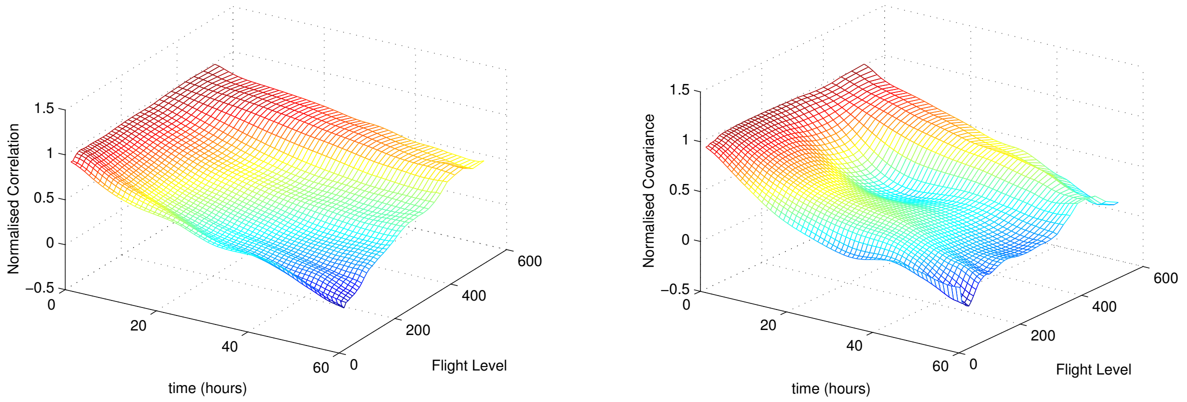


Figure 17: Covariance of u and v components respectively as a function of time difference and altitude

Since we do not have access to data concerning flight paths, we rely on earlier studies concerned with flight path statistics. In [9] there was a study of the performance of the Center-TRACON Automation System (CTAS) system which includes an analysis of cross-track and along-track deviations. We use the along track deviations from this study to estimate the variance for the wind. By running Monte-Carlo simulations we can then estimate the values for k_1 and k_2 such that the cross-track deviations are close to the real statistics. This includes information on the asymptotic behaviour of the deviations, i.e. the limit as $t \to \infty$ as well as the rate of convergence.

There have been two different probabilistic distributions for cross track-error arising from studies on aircraft track data. The first is from the Erzberger and Paielli work on the CTAS system and is a Gaussian distribution. The second arose from studies on organised track separations. Since radar coverage is sparse across the Atlantic and there is no air traffic control it is important that aircraft do not drift so far off track otherwise they may go into conflict with aircraft from a neighbouring track. During these studies it was discovered that the cross track distribution could be closely fit by a double exponential distribution

$$(1-\alpha)\left\{\frac{1}{2\kappa}\exp\left(-\frac{|d|}{\kappa}\right)\right\}+\alpha\left\{\frac{1}{2\lambda}\exp\left(-\frac{|d|}{\lambda}\right)\right\}. \tag{14}$$

which has much larger tails than the Gaussian distribution. In [4, 11] it was hypothesised that these heavy tails are caused by the combination of many Gaussian distributions with different variances and the fact that some aircraft perform particularly badly. The claim is that each aircraft has a Gaussian distribution, but because different aircraft have navigation systems of different standards, the variance of the cross-track error depends on the aircraft. The double exponential distribution arises from averaging over all aircraft. To include this effect we would have to somehow include the different variances in the simulation, either by altering the values of k_1 and k_2 or by introducing some new code to simulate pilot blunders, equipment degradation, etc.

With this in mind and the fact that to match the tails of the double exponential distribution would take a very large number of simulations it was decided that the Erberger Paielli model is more appropriate. Another advantage of the Erberger Paielli model is that there are statistics for the growth of the variance, as opposed to the other studies that have just the average variance. This is important because we have 5 parameters to fit to the data and the average variance may not be enough to determine all of these parameters.

Once we have fit the data to the Gaussian distribution for one aircraft type we can then run the simulations over all aircraft types to see if this makes the double exponential distribution.

[8] provides a study of the accuracy of the CTAS prediction system over a time horizon of about 20 minutes. CTAS considers aircraft to be on their flight plans if they within 8 nmi (cross-track) of the reference path; otherwise they are considered off their flight plans. The study only considers aircraft on their flightplan. This explains the Gaussian probability density function, since the tails of the distribution are effectively ignored.

According to [8], the along-track variance grows quadratically with time. If we assume that this variation is caused mostly by the wind then we find that the wind must be almost constant over a duration of 20 minutes. This is consistent with Section 7.1. The figures of [8] can be used to deduce an estimate for σ, the variance of the wind field, which is consistent with the one in section 7.1.

Next, we can run Monte-Carlo simulations of straight level flight for 20 minutes with constant wind to estimate the controller gains k_1 and k_2. The only remaining problem is how to decouple λ (the time correlation) and β (the horizontal space correlation). This will be hard to do without access to real track data or some other source of weather information. Another possibility would be to use a study of cross-correlation between aircraft pairs. Unfortunately, to our knowledge there have been no such studies carried out for horizontal deviations although there is a study for vertical deviations in [6].

We now consider the system with a constant wind. Let y_∞ denote the limit of y as $t \to \infty$ and w_y represent the cross-track wind velocity. From the calculations used to determine the stability of the equilibria of the bank angle controller we see that:

$$\begin{aligned} E\left(y_\infty^2\right) &= \left(-\frac{k_2}{k_1}\right)^2 E\left(\sin^{-1}\left(-\frac{w_y}{v}\right)^2\right) \\ &= \left(-\frac{k_2}{k_1}\right)^2 E\left(\left(-\frac{w_y}{v}\right)^2 + 1/3\left(-\frac{w_y}{v}\right)^4 + \cdots\right) \text{By Taylor series expansion.} \end{aligned}$$

Let w_y be a zero mean Gaussian random variable, $w_y \sim N\left(0, \sigma^2\right)$, then

$$E\left(y_\infty^2\right) \simeq \left(\frac{k_2}{k_1}\right)^2 \left(\frac{w_y}{v}\right)^2 \left(1 + \left(\frac{w_y}{v}\right)^2\right) \tag{15}$$

$$\simeq \left(\frac{k_2}{k_1}\right)^2 \left(\frac{w_y}{v}\right)^2 \text{Assuming } w_y \ll v \tag{16}$$

If we assume the standard deviation of the cross track deviation is $2nmi$, $\sigma = 8ms^{-1}$ and $v = 250ms^{-1}$ then

$$\frac{k_2}{k_1} = 115000m \tag{17}$$

We now just need to run simulations to get the correct rate of convergence to the limit. The values we use in the simulator are $k_1 = 1e{-}5m^{-1}s^{-1}$ and $k_2 = 1.2s^{-1}$.

8 Concluding remarks

8.1 Assumptions and Limitations

For reference purposes we list all the simplifying assumptions made in the derivation of the model discussed so far. Improvements in the accuracy of the model can be made by gradually relaxing these assumptions. We feel however that it would be counter productive to pursue all of these improvements at this stage, before better insight becomes available into the accuracy of the predictions of the model and the needs of the conflict detection and resolution methods developed in WP3, WP5 and WP6 of HYBRIDGE.

1. Spoilers, flaps, etc. are not used as inputs. Their effect on the aerodynamic parameters is considered.

2. The angle of attack and side-slip angles are small. The flight path angle and the bank angle are treated as inputs rather than states.

3. Ground speed is given by a simple addition of wind speed to airspeed.

4. A 3D FMS is used. Aircraft do not correct for along track errors.

5. During final approach and landing, no attempt is made to track a glide path. As a consequence the aircraft generally tend to miss the beginning of the runway longitudinally.

6. The same controller is used to track straight paths and turns.

7. No nominal weather data is provided to the model. The nominal wind is zero and all wind is treated as a stochastic perturbation.

8. Aircraft turning at a way point ignore the measured wind and compute the point where they should start their turn as though the wind was zero.

9. The model ignores the effect of weather phenomena other than wind speed (e.g. fluctuations in humidity).

8.2 Extensions and Current Developments

Perhaps the biggest limitation of the current simulator is its inability to accurately capture the landing process. At the moment we are trying to include glide path information to ensure landing aircraft actually touch the ground at the beginning of the runway. This requires a change in the way the flight path angle and thrust are set in the approach and landing phases of the flight.

A second improvement that can be pursued is in the way the wind statistics are modelled. At the moment we ignore any nominal wind information available to the ATC and lump all wind together into a stochastic term. A consequence of this is that the wind terms are not zero mean and isotropic. An obvious improvement is to separate the wind into a known term for the nominal wind and a stochastic perturbation from this nominal. The RUC data could play the role of the nominal wind, but no data is available on deviations. We are hoping that we will be able to extract some such data if radar tracks are made available from Eurocontrol.

A more minor improvement that can be easily pursued is in the way that aircraft turn from one way point to the next. One can introduce a special turn controller, instead of simply changing the way point index WP and relying on the straight line controller to take care of the resulting cross track deviation. This should be straightforward and will require the introduction of additional discrete FMS states for turning. In this case, conditions also have to be given for the transition from turning to cruising. For example the aircraft could turn until it is heading for the next way-point, or could turn until it is parallel to the next segment of the reference path. Both of these alternatives have the advantage that the aircraft is guaranteed to complete the turn.

The simulator has already been used in a study of the effect of correlation in the wind on the probability of conflict. The results will be reported in D1.4 of HYBRIDGE.

8.3 Acknowledgements

The authors would like to thank Oliver Watkins, Laurent Cuillerier and Laurent Burlion for their early work on the aircraft model and the simulator. They are grateful to Henk Blom, Richard Irvine and Ted Lewis for the invaluable insight and references that they provided throughout the model and simulator development.

The authors would like to thank the BADA team at Eurocontrol Experimental Centre and Eurocontrol's CFMU for providing access to the BADA database and flight plan data respectively.

References

[1] M. L. Bujorianu, W. Glover, J. Lygeros, and G. Pola. A stochastic hybrid process modelling framework. Technical Report WP1, Deliverable D1.2, HYBRIDGE, May 29, 2003.

[2] Eurocontrol Experimental Centre. User manual for the base of aircraft data (BADA) revision 3.3 [online]. 2002. Available from World Wide Web: http://www.eurocontrol.fr/projects/bada/.

[3] Heinz Erzberger, Russell A. Paielli, Douglas R. Isaacson, and Michelle M. Eshow. Conflict detection and resolution in the presence of prediction error. In 1^{st} *USA/Europe Air Traffic Management R & D Seminar*, Saclay, France, 1997.

[4] D.A. Hsu. Long-tailed distributions for position errors in navigation. *Applied Statistics*, 28:62–72, 1979.

[5] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre. Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control. *Journal of Guidance, Control and Dynamics*, 21(1):29–38, January-February 1998.

[6] G. Moek and D. Harrison. European studies to investigate the feasability of using 1000 ft vertical seperation minima above fl290, part ii. precision radar data analysis and collision risk assessment. *Journal of Navigation*, 45:91, 1992.

[7] Stephane Mondoloni and Dianna Liang. Improving trajectory forecasting through adaptive filtering techniques. In *Proc. 5th USA/Europe ATM R&D Seminar*, June 2003.

[8] R. A. Paielli. Empircal test of conflict probability estimation. Technical report, NASA Ames Research Center, Moffett Field, CA 94035-1000, U.S.A., 1998. Available from World Wide Web: http://www.ctas.arc.nasa.gov/publications/papers/.

[9] R. A. Paielli. Empirical test of conflict probability estimation. Technical report, NASA Ames Research Centre, Moffett Field CA 94035-1000, U.S.A., 1998.

[10] G. Pola, M.L. Bujorianu, J. Lygeros, and M. Di Benedetto. Stochastic hybrid models: An overview with applications to air traffic management. In *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS03)*, pages 52–57, Saint Malo, France, June 16-18 2003.

[11] H.J. Rome and V. Krishnan. Causal probabilistic model for evalutating future transoceanic airlane seperations. *IEEE Tr. Aerospace and Electronic Systems*, 26:804–817, 1990.

[12] R. S. Schild. *Rule Optimization for Airborne Aircraft Separation.* PhD thesis, Vienna Technical University, 1998. Available from World Wide Web: http://www.eos.tuwien.ac.at/Oeko/RSchild/Rules/.

[13] Rhonda Slattery and Yiyuan Zhao. Trajectory synthesis for air traffic automation. *Journal of Guidance, Control and Dynamics*, 20:232–238, 1997.