# A Stochastic Hybrid Model for Air Traffic Control Simulation

William Glover[1], John Lygeros[2]

[1] Department of Engineering
University of Cambridge, Cambridge, CB2 1PZ, U.K.
wg214@eng.cam.ac.uk

[2] Department of Electrical and Computer Engineering
University of Patras, Patras, GR26500, Greece,
lygeros@ee.upatras.gr

**Abstract.** A method for modelling the evolution of multiple flights from the point of view of an air traffic controller is developed. The model is multi-agent, hybrid and stochastic. It 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 validating conflict detection and resolution algorithms.

## 1 Introduction

### 1.1 Air Traffic Management Context

The Air Traffic Management (ATM) system has operated reliably in its present form for many years. The increasing demand for air travel is stressing it to its limits, however. Projections of air traffic levels range from an increase of 50% to 200% over the next 10 years [5]. This increase could lead to both safety and performance degradation in the near future, and place an additional burden on the already overloaded human operators. It is believed, for example, that it is one of the major causes that contributed to a 33% increase in air traffic controller error over the period 1996-2000 [1].

One of the most promising solutions for this problem is increasing the level of automation. It is believed that by doing this, the efficiency of ATM can be improved and the tasks of human operators can be simplified. This may allow them to handle the increased demand in air traffic in a more reliable way, enhancing the level of safety over the current system. A number of different approaches to increasing the level of automation in the ATM process have been proposed in the literature (see, for example, [7, 14, 2, 8, 17, 9, 10]).

Separation assurance forms a major part of the current Air Traffic Control (ATC) workload. If the level of automation in the ATM process increases, some

Preprint submitted to HSCC 2004

of the separation assurance tasks can be transferred to the automated system. One approach for doing this is to rely on Conflict Detection and Resolution (CDR) strategies to assist ATC. CDR strategies try to predict the trajectory of aircraft within the managed airspace, analyse these trajectories to decide if there is a substantial possibility of loss of separation (conflict detection) and, if there is, issue advisories to the ATC and/or pilots on how to resolve the problem (conflict resolution).

### 1.2 Model Based CDR

Models of the ATM process are needed in different phases of the development of CDR strategies.

**Conflict Prediction.** Models are needed to predict the "future" of an encounter to determine whether it is likely to lead to a conflict or not.

**Conflict Resolution.** Models are also useful for conflict resolution. In the simplest case, possible resolution manoeuvres can be analysed in an "if-then" manner, with the model being used to predict the implications of each one. Moreover, most of the conflict resolution strategies based on control theoretic and path planning methods rely on an underlying model to generate the manoeuvres and analyse their performance.

**Validation.** Validation is a crucial step in the development of CDR algorithms. Eventually the algorithms will have to be validated in *field trials* involving air traffic controllers. This type of validation is likely to be very costly and time consuming. An alternative may be to try to validate the algorithms on *real data*. This approach is appealing, but suffers from a number of drawbacks. For example, flight track data is difficult to obtain (usually proprietary); conflict data is of course particularly sensitive! Moreover, conflict situations are (thankfully) extremely rare. Track data includes implicitly the actions of the air traffic controllers, whose job is precisely to prevent conflicts from occurring. To validate our algorithms, on the other hand, we would like to identify situations that would result in a conflict *if ATC were to take no action*. We would then notify ATC if one of these situations arises to ensure that *they do take action*.

A promising alternative is to validate the algorithms on *synthetic data*. The idea here is to simulate a realistic model of the aircraft and weather conditions and use the data generated by the model as the "real world". This approach does not suffer from any of the disadvantages listed above. However, it requires one to develop and tune a realistic model of how aircraft fly (from the point of view of ATC) as well as the uncertainty that enters into the process (due to a large extend to wind, but possibly also to under-modelling and other factors).

In this paper we take a first step in this direction. We develop a multi-aircraft model that allows us to capture

- 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).

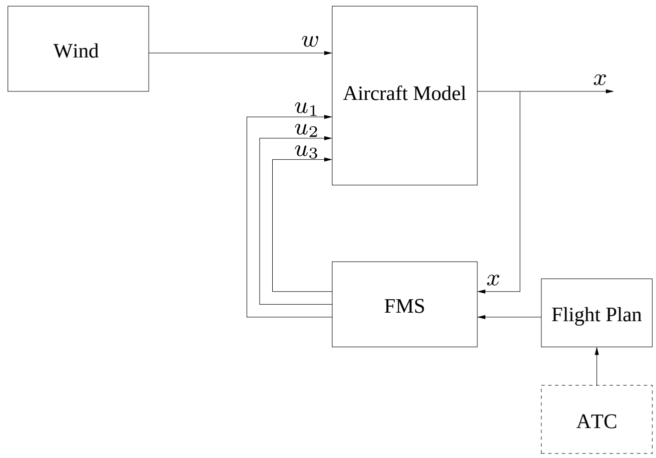


**Fig. 1.** Block diagram of multi-aircraft model components

– Stochastic dynamics, arising primarily because of the effect of wind on the aircraft tracks[3].

The model described here was implemented in a Java based object oriented simulator. To demonstrate possible applications of this work, we use Monte-Carlo simulation to perform a numerical study of the effect of the spatio-temporal correlation of wind on the probability of conflict. Other possible applications of the model include randomized conflict detection algorithms based on particle filter methods (as outlined, for example, in [19]).

### 1.3 Model Overview

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. Our model is, however, intended to be more accurate than the simplified models used by most conflict detection and resolution approaches to predict the future motion of aircraft.

The 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.

[3] 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. The air traffic controller is not modelled in detail however; for work in this direction see [6] and the references therein.

– 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

– 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.

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 Eurocontrol's Base of Aircraft Data (BADA) database [4]. We also use flight plans based on real flight data provided by Eurocontrol's Central Flow Management Unit (CFMU). For the remaining parameters (FMS gains, wind statistics), one would ideally like to determine values through formal system identification experiments. However, for reasons highlighted in [3] (unavailability of data, multiple time scales, etc.), this approach may be unrealistic. Instead we adopt a more heuristic approach. 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 observations in these studies.

### 1.4 Paper Overview

The rest of this paper provides some 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 presents the tuning of the model parameters, based on publicly available weather data and Monte-Carlo simulation of the model itself. Section 7 describes a numerical case study on the effect of wind correlation on the probability of conflict carried out using the simulator presented in this paper.

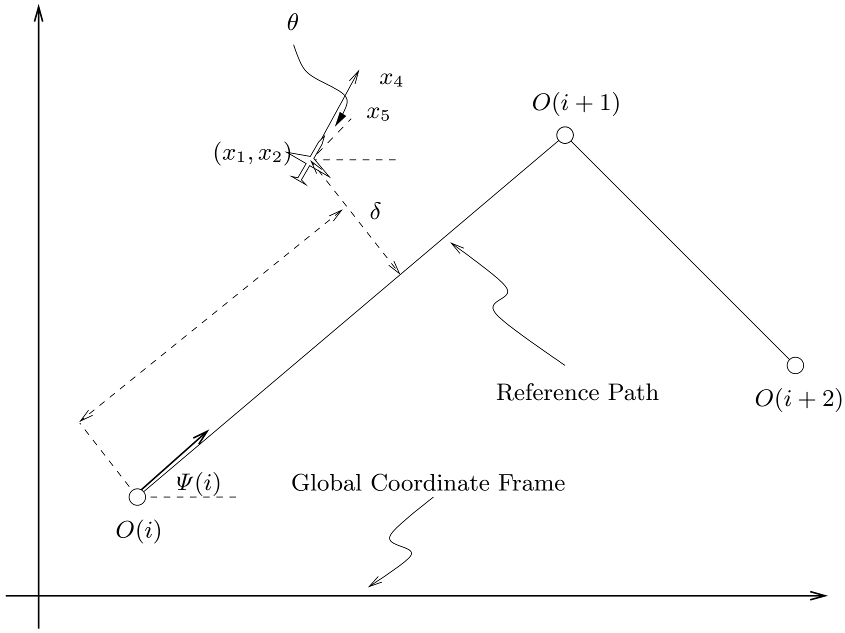


**Fig. 2.** Top view of a typical flight plan showing way-points and angles

## 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 corresponds to real flights and comes from Eurocontrol's CFMU.

The sequence of 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 heading*, $\Psi(i)$, 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).

In our implementation the expected times of arrival are ignored. Instead we assume that the speed when flying between two way-points is dictated by the speed profiles provided by BADA. 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. [18] 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), which can be easily derived from basic aerodynamics (Figure 3). After some simplifying assumptions (appropriate for airliners which normally execute only benign manoeuvres) the model can be described as a control system with six states: the horizontal position ($x_1$ and $x_2$) and altitude ($x_3$) of the aircraft, the true airspeed ($x_4$), the heading angle ($x_5$) and the mass of the aircraft ($x_6$). The control inputs are the engine thrust ($u_1$), the bank angle ($u_2$) and the flight path angle ($u_3$). The movement of the aircraft is also affected by the wind, which acts as a disturbance. We will model the wind through its speed $W = (w_1, w_2, w_3) \in \mathbb{R}^3$. After some algebra (omitted in the interest of space), the equations simplify to

$$\dot{x} = \begin{bmatrix} x_4 \cos(x_5)\cos(u_3) + w_1 \\ x_4 \sin(x_5)\cos(u_3) + w_2 \\ x_4 \sin(u_3) + w_3 \\ -\frac{C_D S \rho(x_3)}{2}\frac{x_4^2}{x_6} - g\sin(u_3) + \frac{1}{x_6}u_1 \\ \frac{C_L S \rho(x_3)}{2}\frac{x_4}{x_6}\sin(u_2) \\ -\eta u_1 \end{bmatrix}. \tag{1}$$

Here $C_L$ and $C_D$ are aerodynamic lift and drag coefficients, $S$ is the total wing surface area, $\rho(x_3)$ is air density and $\eta$ is a coefficient relating thrust to fuel consumption.

In addition to its continuous state, 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.

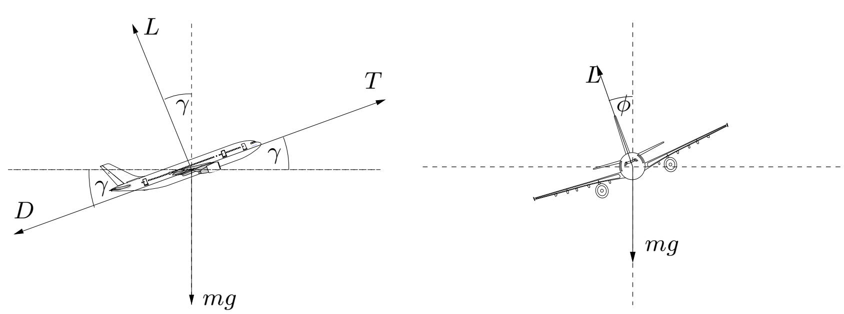


**Fig. 3.** Forces acting on an aircraft.

## 4 Flight Management System

### 4.1 Continuous Control

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, $u$.

In our model 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

$$\dot{x}_5 = \frac{C_L S \rho(x_3)}{2} \frac{x_4}{x_6} \sin(u_2).$$

$x_5$ can then be used to control the horizontal position of the aircraft $(x_1, x_2)$ through the equations[4]

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

Our model assumes that the FMS sets the bank angle based on the heading error and the cross track deviation from the reference path (Figure 2). The controller operates in continuous time and consists of a linear feedback part

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

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. Methods for setting the values of the gains $k_1$ and $k_2$ are discussed in subsequent sections; for more details see [11].

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, $V_{nom}$. $V_{nom}$ depends on altitude and aircraft type and is set according to a lookup table, based on data contained in BADA. 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

$$\dot{x}_4 = -\frac{C_D S \rho(x_3)}{2} \frac{x_4^2}{x_6} + \frac{1}{x_6} u_1. \tag{2}$$

[4] 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.

When climbing or descending, on the other hand, the thrust is set to a fixed value. The flight path angle is then adjusted to control the speed through equation (2). 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.

### 4.2 Discrete State

In our model, the values of the inputs $u$ are determined to some extent by conventional, continuous controllers. 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.

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*). In the interest of space, we only provide some details on the four most important discrete variables: flight level (*FL*), way-point index (*WP*), acceleration mode (*AM*), climb mode (*CM*).

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

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

The value of *FL* is updated based exclusively on the continuous state $x_3$. Among other things, *FL* is used to set the nominal speed, $V_{nom}$; $V_{nom}$ in turn determines the settings for the trust through the discrete state *AM*. the discretization levels of *FL* reflect the quantization of the lookup table used for setting $V_{nom}$.

*Way Point Index.* The discrete variable representing the way point index takes integer values reflecting the number of waypoints 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 $O(i+1) \in \mathbb{R}^3$.

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 4.2. To determine the domain and guard of the automaton we assume that the aircraft performs "fly-by" turns, i.e. turns from one segment of the flight plan to the next without passing directly over the waypoint. We assume that the aircraft turns at constant speed and at a fixed bank angle. A simple geometric calculation leads to a linear condition for when the aircraft should begin its turn.

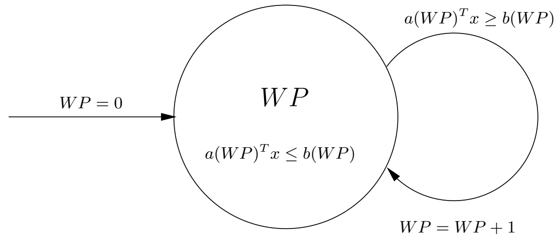


**Fig. 4.** Finite State Machine for WP. $a(WP)$ is a matrix and $b(WP)$ is a vector and together they determine the linear condition for where an aircraft should begin its turn.

*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. The state diagram for $CM$ is very similar to that of $AM$.

$CM$ and $AM$ together are used to set the thrust input, as discussed in the previous section.

## 5 Wind Model

The wind is modelled as the sum of two components, nominal and stochastic. The nominal wind will eventually 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 component is modelled as a random field:

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

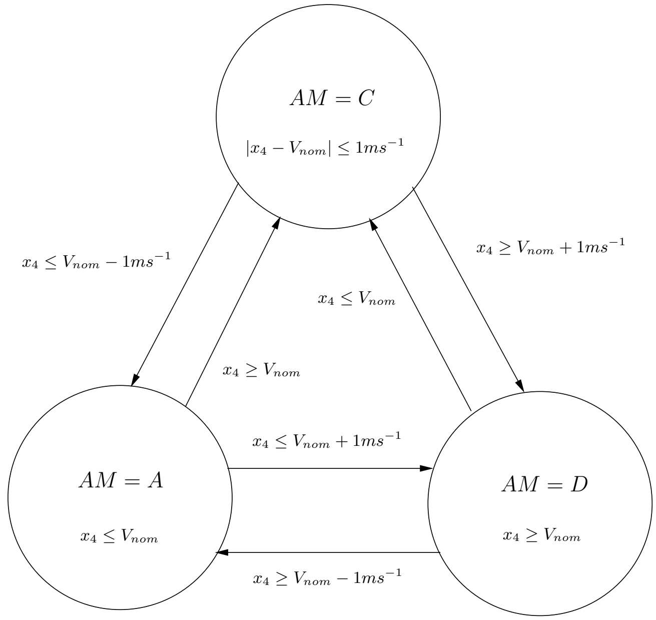


**Fig. 5.** Finite State Machine for AM.

where $w(t, P)$ represents the wind at point $P \in \mathbb{R}^3$ at time $t \in \mathbb{R}$. We assume that the wind field is jointly Gaussian, and that we know the correlation structure, i.e. we know $R(t, P, t', P') = E[w(t, P)]E[w(t', P')]^T$, for all $t, t' \in \mathbb{R}$, $P, P' \in \mathbb{R}^3$.

The main problem from the point of view of simulation is to find a computationally efficient method for generating a random field with the required statistical structure. We developed a method for computing the wind in discrete time and only at the positions of the aircraft[5], a *model-directed synthesis*. The algorithm is similar to a Cholesky decomposition, but is implemented progressively because the positions at which the wind is calculated are dependent on the aircraft positions, which in turn depend on the wind the aircraft experienced at earlier times. The details are omitted in the interest of space, but can be found in [11].

## 6 Tuning

The model presented above needs to be tuned to ensure that the trajectories it generates are realistic. In particular, we need to select values for

– The control parameters $k_1$ and $k_2$

[5] The alternative of gridding both in space and in time would be much more expensive computationally.

– The wind parameters (mean, variance, covariance function).

To tune these parameters we made use of two sources of information:

– Experimental statistics of aircraft deviations from their flight plans [15, 13, 16, 12]
– Wind field data from Rapid Update Cycle (RUC).

To gain some insight into the correlation structure for the stochastic component we started with the RUC wind data. This data is used in the USA and is freely available over the Internet. First we found the mean and variance of the wind from the RUCdata and verified that the along track aircraft deviation statistics is consistent with these findings. We then estimated the values of $k_1$ and $k_2$ which give the correct cross-track deviation statistics. This can be done using Monte-Carlo methods. Alternatively, if we assume that the wind is roughly constant over periods of the order of a few minutes, we can derive an analytic expression. If we assume the standard deviation of the cross track deviation is $2nmi$, $\sigma = 8ms^{-1}$ and $v = 250ms^{-1}$ then

$$\frac{k_2}{k_1} \simeq E(y_\infty^2) \left( \frac{v}{w_y} \right)^2 = 115000m \tag{3}$$

Here $y_\infty$ denotes the steady state cross track error.

More systematic methods for tuning the model (e.g. using techniques developed for random field identification) are a subject of ongoing work.

## 7 Simulation

The model presented here was coded in a Java object oriented simulator. A graphical user interface was developed to allow visualizing the results and running the simulation in interactive mode (where the used can change flight plans on-line, more or less as an air traffic controller would). Figure 6 shows the simulation of a simple 20 min, 4 waypoint flightplan.

To demonstrate possible uses of the model developed here we have conducted a numerical study of the effect of correlation in the wind on the probability that aircraft will come dangerously close to one another (conflict probability). This is a topic of contention among researchers and practitioners in the field of air traffic management. Some authors believe that correlation is important and must be accounted for, while others believe that correlation is unimportant and can be ignored; both classes provide heuristic arguments that support their view!

Our simulator can be used to quantify this effect. We conducted a number of simulations of a pair of aircraft both following a simple flight plan. We find the probability distribution of the aircraft separation after 20 min. This will give an idea of the chance of the aircraft coming into conflict. In the first set of simulations we do not have correlation between the aircraft. In the second we do.

Figures 7, 8 show the results of the two experiments. When there is correlation between the two aircraft the variance of the separation is very small (the standard deviation is 305m vs 5000m without separation). This suggests that conflict detection probes that ignore correlation would be too conservative in this case. Interestingly, the variance of the average position is larger (the standard deviation is 7120m vs 5250m). This suggests that although a conflict could be predicted more accurately if correlation is included, the prediction of the actual position of the conflict would be more variable.

These results allow one to draw some interesting conclusions. Clearly correlation between aircraft makes a difference to the statistics of the positions of each aircraft and hence the probability of conflict. Generally speaking the variance of the separation between aircraft will be less with correlation, although this is dependent on the geometry of the flightplans. This will lead to the probability of conflict being lower for situations where the aircraft do not nominally come into conflict, and higher for situations in which they do nominally come into conflict. The fact that the position of conflict is less well known will not make a large difference to the accuracy of conflict probes. In summary the effect of ignoring correlation in conflict probes will be to create an unnecessary number of false alerts, so it is important to develop methods that can account for correlation.

**Acknowledgement**: Research supported by the European Commission under HYBRIDGE, IST-2001-32460 and Eurocontrol under contract C20051E/BM/03. The authors would like to thank Eurocontrol's Experimental Centre and Central Flow Management Unit for providing access to the BADA database and flight plan data. They are grateful to H. Blom, R. Irvine, A. Lecchini and O. Watkins for discussions that provided invaluable insight into the ATM problem.

## References

1. Federal Aviation Authority. Safer skies [online]. 2001. Available from World Wide Web: `http://www.faa.gov/apa/safer_skies/saftoc.htm`.
2. K. Blin. Stochastic conflict detection for air traffic management. Technical report, Eurocontrol Experimental Centre, BP 15, 91222 Brétigny-sur-Orge, France, 2000.
3. 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.
4. 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/`.
5. U.K. Department of Transport. Air traffic forecasts for the United Kingdom 1997. Technical report, U.K. Governement, Department of Transport, 1997. Available from World Wide Web: `http://www.aviation.dft.gov.uk/aed/air/aircont.htm`.
6. M.D. di Benedetto and G. Pola. Inventory of error evolution control problems in air traffic management. Technical Report WP7, Deliverable D7.1, HYBRIDGE, November 4, 2002.

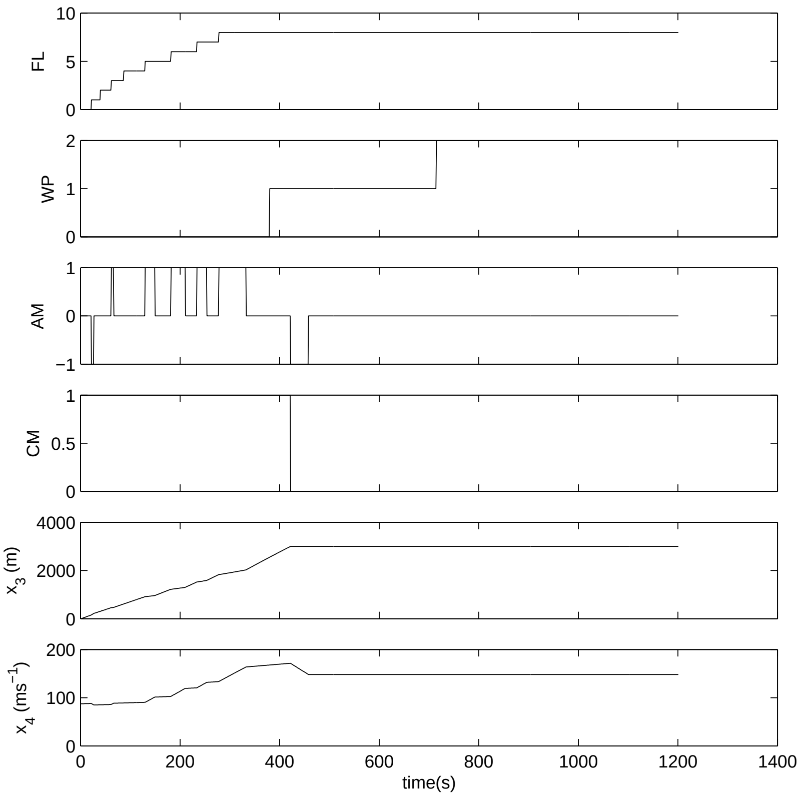


**Fig. 6.** Selected discrete and continuous variables from the simulation of a simple 20 minute flight plan.


7. Heinz Erzberger, Thomas J. Davis, and Steven Green. Design of Center-TRACON automation system. In *Proceedings of the AGARD Guidance and Control Syposium on Machine Intelligence in Air Traffic Management*, pages 11.1–11.12, 1993.
8. J. M. Hoekstra et al. Overview of NLR free flight project 1997-1999. Technical report, National Aerospace Laboratory, NLR, 2000.
9. Eurocontrol. Aircraft in the future air traffic management system. Technical report, Eurocontrol, 2001.
10. Defence Evaluation and Research Agency (UK). The more autonomous - aircraft in the future air traffic management system. Technical report, DERA, 2000.
11. W. Glover and J. Lygeros. A multi-aircraft model for conflict detection and resolution algorithm validation. Technical Report WP1, Deliverable D1.3, HYBRIDGE, July 23, 2003.

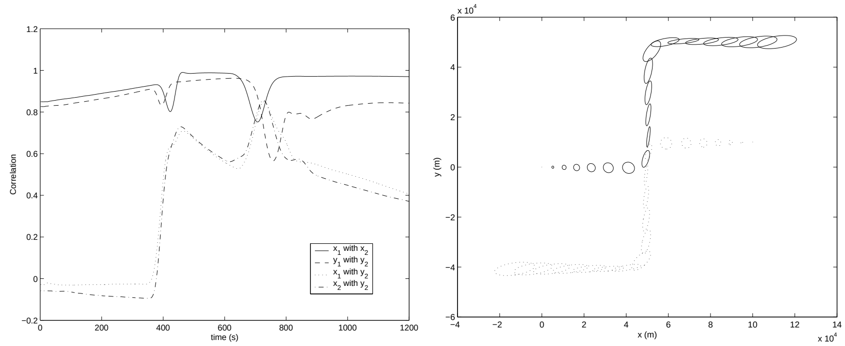


**Fig. 7.** Plot of correlation between the position components of both aircraft (left) and error ellipses for the simulated flight-plan (right).


12. D. Harrison and G. Moek. 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.
13. D.A. Hsu. Long-tailed distributions for position errors in navigation. *Applied Statistics*, 28:62–72, 1979.
14. C. Meckiff and P. Gibbs. PHARE highly interactive problem solver. Technical Report EEC 273/94, Eurocontrol Experimental Centre, BP 15, 91222 Brétigny-sur-Orge, France, November 1994.
15. R. A. Paielli. Empirical test of conflict probability estimation. Technical report, NASA Ames Research Centre, Moffett Field CA 94035-1000, U.S.A., 1998.
16. 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.
17. Conflict Detection RTCA Special Committee 186, Working Group-1 and Resolution Subgroup. Application of airborne conflict management: Detection, prevention and resolution. Technical report, RTCA, 2000.
18. R. Slattery and Y. Zhao. Trajectory synthesis for air traffic automation. *Journal of Guidance, Control and Dynamics*, 20:232–238, 1997.
19. O. Watkins and J. Lygeros. Stochastic reachability for discrete time systems: An application to aircraft collision avoidance. In *IEEE Conference on Decision and Control (CDC03)*, Hawaii, USA, December 9-12 2003. To appear.

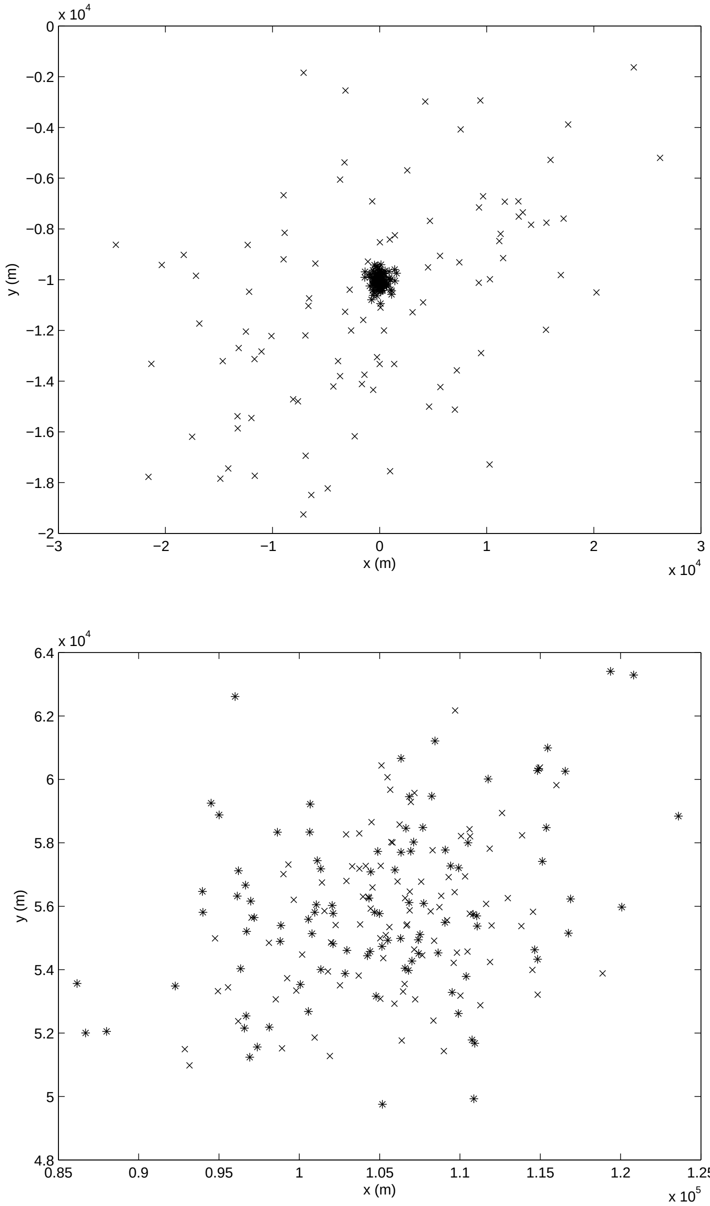


**Fig. 8.** Scatter plots of aircraft positions after twenty minutes. The first shows difference in aircraft position, the second shows average aircraft position. In both plots "×" denotes simulations with correlation, "✱" simulations without correlation.