HYBRIDGE

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

WP5: Control of uncertain hybrid systems

Monte Carlo conflict resolution algorithm with simulation examples

A. Lecchini[1], W. Glover[1], J. Lygeros[2] and J. Maciejowski[1]

April 18, 2005

Version:	*0.4*
Task number:	*5.4*
Deliverable number:	*D5.4*
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:	*Monte Carlo conflict resolution algorithm with simulation examples*
Authors of document:	*A. Lecchini, W. Glover, J. Lygeros and J. Maciejowsi*
Deliverable number:	*D5.4*
Contract:	*IST-2001-32460 of European Comission*
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	01/01/05	All	First draft
0.2	28/02/05	All	Second draft
0.3	13/03/05	All	Submitted for internal review
0.4	18/04/05	All	Submitted to the consortium

Version 0.4		Organisation	Signature/Date
Authors	A. Lecchini	UCAM	
	W. Glover	UCAM	
	J. Lygeros	UPAT	
	J.Maciejowski	UCAM	
Internal reviewers	R. Irvine	EEC	
	H. Blom	NLR	

HYBRIDGE, IST-2001-32460
Work Package WP5, Deliverable D5.4

Monte Carlo conflict resolution algorithm with simulation examples

Prepared by:
A. Lecchini[*], W. Glover[*], J. Lygeros[†]and J. Maciejowski[*]

Abstract

Conflict resolution is one of the main tasks of Air Traffic Control. It consists in issuing the proper instructions to aircraft in order to avoid loss of safe separation between them and, at the same time, direct them to destination. Conflict resolution implies making decisions in the face of the considerable levels of uncertainty which affect the motion of aircraft. In this deliverable, we present a framework for conflict resolution which allows to take into account such levels of uncertainty through the use of a stochastic simulator. The conflict resolution task is posed as the problem of optimising an expected value criterion. Optimisation of the expected value resolution criterion is then carried out through an iterative procedure which consists in proposing resolution instructions and in comparing them on the basis of trajectories simulation. This iterative optimisation procedure takes the form of a Monte Carlo Markov Chain (MCMC). Simulation examples inspired by current Air Traffic Control practice in Terminal Maneuvering Area and Approach Sectors illustrate the proposed conflict resolution strategy.

[*] Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK, Tel. +44 1223 33260, Fax. +44 1223 332662, { al394,wg214,jmm}@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 Acronyms

ATC	Air Traffic Control
ATM	Air Traffic Management
BADA	Base of Aircraft Data
FMS	Flight Management System
MC	Monte Carlo
MCMC	Monte Carlo Markov Chain
TMA	Terminal Maneuvering Area (European equivalent of TRACON)
STAR	Standard Approach Route
FL	Flight Level

1 Aim and Scope

The objective of WP5 of HYBRIDGE is to develop algorithms for assisting air traffic controllers and pilots with conflict resolution maneuvers.
The main aim of Deliverable D5.4 is summarised in Task 5.4 of WP5:

> Preferences of air traffic controllers and pilots need to be addressed in order to ensure that the resolution manoeuvres are acceptable to be implemented by the humans involved. Heuristics will be employed to guide the optimisation procedure in order to improve the computational performance. Monte Carlo simulation on the detailed multi-aircraft model of WP1 will be used to validate the theoretical predictions.

Accordingly, the main contribution of this deliverable is the illustration of the use of the resolution algorithm developed in WP5 on some realistic benchmarks inspired by current ATC practice. These benchmark are implemented on the multi-aircraft simulator developed in WP1.

This deliverable is organised as follows. The next section is an introduction to the conflict resolution problem and to the approach pursued in WP5. In Section 3 we describe our formulation of the conflict resolution problem as an optimisation problem. The stochastic optimisation procedure that we adopt to solve the problem is described in Section 4. The remaining sections are devoted to illustrate the approach. Section 5 introduces ATC in Terminal Maneuvering Area and Approach Sectors. The use of our conflict resolution procedure in some typical situations in these sectors is illustrated in Section 6. Section 7 contains conclusions and future objectives.

2 Introduction

In the current organisation of Air Traffic Management the centralised Air Traffic Control (ATC) is in complete control of the air traffic and ultimately responsible for safety. Before take off, aircraft receive flight plans which cover the entire flight. During the flight, ATC sends additional instructions to them, depending on the actual traffic, in order to improve traffic flow and avoid dangerous encounters. The main objective of ATC is to maintain safe separation. The level of accepted minimum safe separation can vary with the density of the traffic and the region of airspace. For example, a largely accepted value for horizontal minimum safe separation between two aircraft at the same altitude is 5 nmi in general en-route airspace which is reduced to 3 nmi in approach sectors for aircraft landing and departing. A conflict is defined as the situation of loss of minimum safe separation between two aircraft. If it is possible, ATC tries also to fulfil the (possibly conflicting) requests of aircraft and airlines; for example, desired paths to avoid turbulence or desired time of arrivals to meet schedule. In order to improve performance of ATC, mainly in anticipation of increasing levels of traffic, research effort has been spent in the last decade on creating tools for conflict detection and resolution. A review of research work in this area of ATC is presented in [15].

Uncertainty is introduced in air traffic by the action of the wind field, incomplete knowledge of the physical coefficients of the aircraft and unavoidable imprecision in the execution of ATC instructions. In conflict detection one has to evaluate the possibility of future conflict starting from the current state of the airspace and taking into account uncertainty in the future position of aircraft while they follow their flight plans. For this task, one needs a model to predict the future. In a probabilistic setting, the model could be either an empirical distribution of future position or a stochastic differential equation that describes the aircraft motion and defines implicitly a distribution for future aircraft positions. On the basis of the prediction model one can evaluate metrics related to safety. One example of a possible metric is conflict probability over a certain time horizon. Several methods have been developed to estimate different metrics related to safety for a number of prediction models, e.g [1, 12, 13, 18, 19]. Among other methods, Monte Carlo (MC) methods have the main advantage of allowing flexibility in the complexity of the prediction model since the model is used only as a simulator and, in principle, it is not involved in explicit calculations. In all methods a trade off exists between computational effort (simulation time in the case of MC methods) and accuracy of the model. Techniques to accelerate MC methods especially for rare event computations are under development, see e.g. [14].

In conflict resolution the objective is to calculate suitable maneuvers to avoid a predicted conflict. A number of conflict resolution algorithms have been proposed for a deterministic setting, for example [7, 11, 21]. In a stochastic setting, the research effort has been concentrated mainly on conflict detection, with only a few resolution strategies proposed. Simple conflict resolution maneuvers in a stochastic setting have been considered in [18, 19]. The main reason for this is the complexity of stochastic prediction models which, even if it does not make it impossible to estimate conflict probability through Monte Carlo methods, it makes the quantification of the effects of possible control actions intractable.

In this contribution we present a Monte Carlo Markov Chain (MCMC) framework [20] for conflict resolution in a stochastic setting. The interesting point in this approach is that it extends the advantages of Monte Carlo techniques, in terms of flexibility and complexity of the problems that can be tackled, to conflict resolution. The approach is borrowed from Bayesian statistics [16, 17]. We will consider a resolution criterion that takes into account separation and other factors (e.g. aircraft requests). Then, the MCMC optimisation procedure of [16] is employed to estimate the resolution maneuver that optimises the expected value criterion. The proposed approach is illustrated in simulation on some realistic benchmarks inspired by current ATC practice. These benchmarks have been implemented in an air traffic simulator developed in previous work [8, 9, 10].

3 Conflict resolution with an expected value criterion

In our approach we formulate conflict resolution as a constrained optimisation problem. Given a set of aircraft involved in a conflict, the conflict resolution maneuver is determined by a parameter ω which defines the nominal paths of the aircraft The actual execution of the maneuver is affected by uncertainty. Therefore, the sequence of actual positions of the aircraft (for example: the sequence of positions every 6 seconds which is a typical time interval between two successive radar sweeps) during the resolution maneuver is, a-priori of its execution, a random variable, denoted by X. A conflict is defined as the event that the positions of two aircraft during the execution of the maneuver get too close. The objective is to select ω in order to maximise the expected value of some measure of performance associated to the execution of the resolution maneuver while ensuring a small probability of conflict. In this section we introduce the formulation of the problem in a general fashion.

Let X be a random variable whose distribution depends on some parameter ω. The distribution of X is denoted by $p_\omega(x)$ with $x \in \mathbf{X}$. The set of all possible values of ω is denoted by $\mathbf{\Omega}$. We assume that a constraint on the random variable X is given in terms of a feasible set $\mathbf{X_f} \subseteq \mathbf{X}$. We say that a realisation x, of random variable X, violates the constraint if $x \notin \mathbf{X_f}$. Moreover, we assume that for a realisation $x \in \mathbf{X_f}$ some definition of performance of x is given. In general performance can depend also on the value of ω, therefore performance is measured by a function $\mathrm{perf}(\cdot,\cdot) : \mathbf{\Omega} \times \mathbf{X_f} \to (0,1]$. The probability of satisfying the constraint is denoted by $\mathrm{P}(\omega)$

$$\mathrm{P}(\omega) = \int_{x \in \mathbf{X_f}} p_\omega(x) dx \,.$$

The probability of violating the constraint is denoted by $\bar{\mathrm{P}}(\omega) = 1 - \mathrm{P}(\omega)$. The expected performance for a given $\omega \in \mathbf{\Omega}$ is denoted by $\textsc{Perf}(\omega)$, where

$$\textsc{Perf}(\omega) = \int_{x \in \mathbf{X_f}} \mathrm{perf}(\omega, x) p_\omega(x) dx \,.$$

Ideally one would like to select ω to maximise the performance, subject to a bound on the probability of constraint satisfaction. Given a bound $\bar{\mathbf{P}} \in [0,1]$, this corresponds to solving the constrained optimisation problem

$$\textsc{Perf}_{\max|\bar{\mathbf{P}}} = \sup_{\omega \in \mathbf{\Omega}} \textsc{Perf}(\omega) \tag{1}$$

$$\text{subject to } \bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}. \tag{2}$$

Clearly, for feasibility we must assume that there exists $\omega \in \mathbf{\Omega}$ such that $\bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}$, or, equivalently,

$$\bar{\mathrm{P}}_{\min} = \inf_{\omega \in \mathbf{\Omega}} \bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}.$$

This optimisation problem is generally difficult to solve, or even to approximate by randomised methods. Here we approximate this problem by an optimisation problem with penalty terms. We show that with a proper choice of the penalty term we can enforce the desired maximum bound on the probability of violating the constraint, provided that such a bound is feasible, at the price of sub-optimality in the resulting expected performance. Let us introduce the function $u(\omega, x)$ defined as

$$u(\omega, x) = \begin{cases} \mathrm{perf}(\omega, x) + \Lambda & x \in \mathbf{X_f} \\ \\ 1 & x \notin \mathbf{X_f}, \end{cases}$$

where $\Lambda > 1$. The parameter Λ represents a reward for constraint satisfaction. The expected value of $u(\omega, x)$ is given by

$$U(\omega) = \int_{x \in \mathbf{X}} u(\omega, x) p_\omega(x) dx \qquad \omega \in \mathbf{\Omega}\,.$$

Instead of the constrained optimisation problem (1)–(2) we solve the unconstrained optimisation problem:

$$U_{\max} = \sup_{\omega \in \mathbf{\Omega}} U(\omega). \tag{3}$$

Assume the supremum is attained and let $\bar{\omega}$ denote the optimum solution, i.e. $U_{\max} = U(\bar{\omega})$. The following proposition introduces bounds on the probability of violating the constraints and the level of suboptimality of $\textsc{Perf}(\bar{\omega})$ over $\textsc{Perf}_{\max|\bar{\mathrm{P}}}$.

Proposition 3.1 *The maximiser, $\bar{\omega}$, of $U(\omega)$ satisfies*

$$\begin{aligned} \bar{\mathrm{P}}(\bar{\omega}) &\le \frac{1}{\Lambda} + \left(1 - \frac{1}{\Lambda}\right) \bar{\mathrm{P}}_{\min}\,, & (4) \\ \textsc{Perf}(\bar{\omega}) &\ge \textsc{Perf}_{\max|\bar{\mathrm{P}}} - (\Lambda - 1)(\bar{\mathbf{P}} - \bar{\mathrm{P}}_{\min})\,. & (5) \end{aligned}$$

Proof: The optimisation criterion $U(\omega)$ can be written in the form

$$U(\omega) = \textsc{Perf}(\omega) + \Lambda - (\Lambda - 1)\bar{\mathrm{P}}(\omega)\,.$$

By the definition of $\bar{\omega}$ we have that $U(\bar{\omega}) \ge U(\omega)$ for all $\omega \in \mathbf{\Omega}$. We therefore can write

$$\textsc{Perf}(\bar{\omega}) + \Lambda - (\Lambda - 1)\bar{\mathrm{P}}(\bar{\omega}) \ge \textsc{Perf}(\omega) + \Lambda - (\Lambda - 1)\bar{\mathrm{P}}(\omega) \qquad \forall \omega$$

which can be rewritten as

$$\bar{\mathrm{P}}(\bar{\omega}) \le \frac{\textsc{Perf}(\bar{\omega}) - \textsc{Perf}(\omega)}{\Lambda - 1} + \bar{\mathrm{P}}(\omega) \qquad \forall \omega\,. \tag{6}$$

Since $0 < \mathrm{perf}(\omega, x) \le 1$, $\textsc{Perf}(\omega)$ satisfies

$$0 < \textsc{Perf}(\omega) \le P(\omega)\,. \tag{7}$$

Therefore we can use (7) to obtain an upper bound on the right-hand side of (6) from which we obtain

$$\bar{\mathrm{P}}(\bar{\omega}) \leq \frac{1}{\Lambda} + \left(1 - \frac{1}{\Lambda}\right) \bar{\mathrm{P}}(\omega) \qquad \forall \omega \in \mathbf{\Omega}.$$

We eventually obtain (4) by taking a minimum to eliminate the quantifier on the right-hand side of the above inequality.

In order to obtain (5) we proceed as follows. By definition of $\bar{\omega}$ we have that $U(\bar{\omega}) \geq U(\omega)$ for all $\omega \in \mathbf{\Omega}$. In particular, we know that

$$\textsc{Perf}(\bar{\omega}) \geq \textsc{Perf}(\omega) - (\Lambda - 1)\left[\bar{\mathrm{P}}(\omega) - \bar{\mathrm{P}}(\bar{\omega})\right] \qquad \forall \omega \,:\, \bar{\mathrm{P}}(\omega) \leq \bar{\mathbf{P}}\,.$$

Taking a lower bound of the right-hand side, we obtain

$$\textsc{Perf}(\bar{\omega}) \geq \textsc{Perf}(\omega) - (\Lambda - 1)\left[\bar{\mathbf{P}} - \bar{\mathrm{P}}_{\mathrm{min}}\right] \qquad \forall \omega \,:\, \bar{\mathrm{P}}(\omega) \leq \bar{\mathbf{P}}\,.$$

Taking the maximum and eliminating the quantifier on the right-hand side we obtain the desired inequality. ■

Proposition 3.1 suggests a method for choosing Λ to ensure that the solution $\bar{\omega}$ of the optimisation problem will satisfy $\bar{\mathrm{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$. In particular it suffices to know $\bar{\mathrm{P}}(\omega)$ for some $\omega \in \mathbf{\Omega}$ with $\bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}$ to obtain a bound. If there exists $\omega \in \mathbf{\Omega}$ for which $\hat{\mathrm{P}} = \bar{\mathrm{P}}(\omega)$ is known, then any

$$\Lambda \geq \frac{1 - \hat{\mathrm{P}}}{\bar{\mathbf{P}} - \hat{\mathrm{P}}}$$

ensures that $\bar{\mathrm{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$. If we know that there exists a parameter $\omega \in \mathbf{\Omega}$ for which the constraints are satisfied almost surely, a tighter (and potentially more useful) bound can be obtained. If there exists $\omega \in \mathbf{\Omega}$ such that $\bar{\mathrm{P}}(\omega) = 0$, then any

$$\Lambda \geq \frac{1}{\bar{\mathbf{P}}} \tag{8}$$

ensures that $\bar{\mathrm{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$. Clearly to minimise the gap between the optimal performance and the performance of $\bar{\omega}$ we need to select Λ as small as possible. Therefore the optimal choices of Λ that ensure the bounds on constraint satisfaction and minimise the suboptimality of the solution are $\Lambda = \frac{1-\hat{\mathrm{P}}}{\bar{\mathbf{P}}-\hat{\mathrm{P}}}$ and $\Lambda = \frac{1}{\bar{\mathbf{P}}}$ respectively.

4 Monte Carlo Optimisation

In this section we describe a simulation-based procedure, to find approximate optimisers of $U(\omega)$. The only requirement for applicability of the procedure is to be able to obtain realisations of the random variable X with distribution $p_\omega(x)$ and to evaluate $u(\omega, x)$ pointwise. This optimisation procedure is in fact a general procedure for the optimisation of expected value criteria. It has been originally proposed in the Bayesian statistics literature [16].

The optimisation strategy relies on extractions of a random variable Ω whose distribution has modes which coincide with the optimal points of $U(\omega)$. These extractions are obtained through Monte Carlo Markov Chain (MCMC) simulation [20]. The problem of optimising the expected criterion is then reformulated as the problem of estimating the optimal points from extractions concentrated around them. In the optimisation procedure, there exists a tunable trade-off between estimation accuracy of the optimiser and computational effort. In particular, the distribution of Ω is proportional to $U(\omega)^J$ where J is a positive integer which allows the user to increase the "peakedness" of the distribution and concentrate the extractions around the modes at the price of an increased computational load. If the tunable parameter J is increased during the optimisation procedure, this approach can be seen as the counterpart of Simulated Annealing for a stochastic setting. Simulated Annealing is a randomised optimisation strategy developed to find tractable approximate solutions to complex deterministic combinatorial optimisation problems, [22]. A formal parallel between these two strategies has been derived in [17].

The MCMC optimisation procedure can be described as follows. Consider a stochastic model formed by a random variable Ω, whose distribution has not been defined yet, and J conditionally independent replicas of random variable X with distribution $p_\Omega(x)$. Let us denote by $h(\omega, x_1, x_2, \ldots, x_J)$ the joint distribution of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$. It is straightforward to see that if

$$h(\omega, x_1, x_2, \ldots, x_J) \propto \prod_j u(\omega, x_j) p_\omega(x_j) \tag{9}$$

then the marginal distribution of Ω, also denoted by $h(\omega)$ for simplicity, satisfies

$$h(\omega) \propto \left[\int u(\omega, x) p_\omega(x) dx\right]^J = U(\omega)^J . \tag{10}$$

This means that if we can extract realisations of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$ then the extracted Ω's will be concentrated around the optimal points of $U(\Omega)$ for a sufficiently high J. These extractions can be used to find an approximate solution to the optimisation of $U(\omega)$.

Realisations of the random variables $(\Omega, X_1, X_2, X_3, \ldots, X_J)$, with the desired joint probability density given by (9), can be obtained through Monte Carlo Markov Chain simulation. The algorithm is presented below. In the algorithm, $g(\omega)$ is known as the instrumental (or *proposal*) distribution and is freely chosen by the user; the only requirement is that $g(\omega)$ covers the support of $h(\omega)$.

MCMC Algorithm

Given $\omega(k)$, $x_j(k)$, $j = 1, \dots, J$ realisations of random variable $X(k)$ with distribution $p_{\omega(k)}(x)$, and $u_J(k) = \prod_{j=1}^{J} u(\omega(k), x_j(k))$:

1 Extract

$$\tilde{\Omega} \sim g(\omega)$$

2 Extract

$$\tilde{X}_j \sim p_{\tilde{\Omega}}(x) \quad j = 1, \dots, J$$

and calculate

$$\tilde{U}_J = \prod_j u(\tilde{\Omega}, \tilde{X}_j)$$

3 Extract the new state of the chain as

$$[\Omega(k+1),\, U_J(k+1)] = \begin{cases} [\tilde{\Omega},\, \tilde{U}_J] & \text{with prob. } \rho(\omega(k), u_J(k), \tilde{\Omega}, \tilde{U}_J) \\ \\ [\omega(k),\, u_J(k)] & \text{with prob. } 1 - \rho(\omega(k), u_J(k), \tilde{\Omega}, \tilde{U}_J) \end{cases}$$

where

$$\rho(\omega, u_J, \tilde{\omega}, \tilde{u}_J) = \min \left\{ 1, \frac{\tilde{u}_J}{u_J} \frac{g(\omega)}{g(\tilde{\omega})} \right\}$$

This algorithm is a formulation of the Metropolis-Hastings algorithm for a desired distribution given by $h(\omega, x_1, x_2, \dots, x_J)$ and proposal distribution given by

$$g(\omega) \prod_j p_\omega(x_j)\,.$$

In this case, the acceptance probability for the standard Metropolis-Hastings algorithm is

$$\frac{h(\tilde{\omega}, \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_J)}{h(\omega, x_1, x_2, \dots, x_J)} \frac{g(\omega) \prod_j p_\omega(x_j)}{g(\tilde{\omega}) \prod_j p_\omega(\tilde{x}_j)}\,.$$

By inserting (9) in the above expression one obtains $\rho(\omega, u_J, \tilde{\omega}, \tilde{u}_J)$. Under minimal assumptions, the Markov Chain $\Omega(k)$ is uniformly ergodic with stationary distribution $h(\omega)$ given by (10). Results that characterise the convergence rate to the stationary distribution can be found for example in [20].

A general guideline to obtain faster convergence is to concentrate the search distribution $g(\omega)$ where $U(\omega)$ assumes nearly optimal values. The algorithm represents a trade-off between computational effort and the "peakedness" of the target distribution. This trade-off is tuned by the parameter J which is the power of the target distribution and also the number of extractions of X at each step of the chain. Increasing J concentrates the distribution more around the optimisers of $U(\omega)$, but also increases the number of simulations one needs to perform at each step. Obviously if the peaks of $U(\omega)$ are already quite sharp, this implies some advantages in terms of computation, since there is no need to increase further the peakedness of the criterion by running more simulations. For the specific $U(\omega)$ proposed in the previous section, a trade-off exists between its peakedness and the parameter Λ, which is related to probability of constraint violation. In particular, the greater Λ is the less peaked the criterion $U(\omega)$ becomes, because the relative variation of $u(\omega, x)$ is reduced, and therefore more computational effort is required for the optimisation of $U(\omega)$.

5 Air Traffic Control in Terminal Maneuvering Area and Approach Sectors

Terminal Maneuvering Area (TMA) and Approach Sectors are perhaps the most difficult areas for ATC. The management of traffic, in this case, includes tasks such as determining landing sequences, issuing of "vector" maneuvers to avoid collisions, holding the aircraft in "stacks" in case of congested traffic, etc. Here, we give a schematic representation of the problem as described in [2, 6].

During most of the flight, aircraft stay at cruising altitudes, above 30000 ft. In the current organisation, the traffic at these altitudes has an en-route structure, which facilitates the action of ATC. Aircraft follow prespecified corridors at different *flight levels*. Flight levels are given in 3 digit numbers, representing hundreds of feet; for example, the altitude of 30000 ft is denoted by FL300.

Towards the end of the flight, aircraft enter TMA where ATC guides them from cruising altitudes to the entry points of the Approach Sector, which are typically between FL50 and FL150. Ideally, aircraft should enter the Approach Sector in a sequence properly spaced in time. The controllers of the Approach Sector are then responsible for guiding the aircraft towards the proper runway. The tasks of ATC in the Approach Sectors include:

1) Maintain safe separation between aircraft. This is the most important requirement for safety, in any sector and during all parts of the flight. Aircraft must always maintain a minimum level of separation. A conflict between two aircraft is defined as the situation of loss of minimum safe separation between them. Safe separation is defined by a protected zone centred around each aircraft. The level of accepted minimum separation can vary with the density of the traffic and the region of the airspace. A largely accepted shape of the protected zone is defined by a vertical cylinder, centred on the aircraft with having radius 5 nmi and height 2000 ft, so that aircraft which do not have 5 nmi of horizontal separation must have 1000 ft of vertical separation.

2) Descend aircraft from entry altitude to intercept localiser. Once aircraft have entered the Approach Sector, ATC must guide them from the entry altitude (FL50 to FL150) to FL15. This is the altitude at which they can intercept the *localiser*, i.e. the radio beacons which will guide them onto the runway. The point at which the aircraft will actually start the descent towards the runway is an important variable which has to be carefully chosen since it can affect the rest of the maneuver and the coordination with other aircraft. The reason is that aircraft fly following prespecified speed profiles which depend on the altitude; they fly faster at high altitudes and slower at low altitudes. This implies that aircraft, flying at lower altitudes, are slower in joining the landing queue.

3) Sequence aircraft towards the runway. The air traffic controllers must direct the aircraft towards the runway in a properly spaced queue. This is done by adjusting the waypoints (corners) of a standard approach route (STAR) — see Figure 1. Typically the route is composed of four legs. During their descent, aircraft are first aligned, on one of the two sides of the runway, in the direction of the runway but with opposite heading. This leg is called the *left/right downwind leg*, since aircraft are expected to land against the wind. Aircraft then perform a turn of approximately 90°, to approach the localiser. This second segment is called the *base leg.* Aircraft perform an additional turn in order to intercept the plane of the localiser with an angle of incidence of approximately 30°. The reason is that 30° is a suitable angle for pilots to perform the final turn in the direction of the runway as soon as possible when the localiser has been intercepted. It is required that aircraft intercept the localiser plane at least 5 nmi from the beginning of the runway and at an altitude of $1000 - 1500$ ft, so that they can follow a $3^\circ - 5^\circ$ glide path to the runway.

This approach geometry (which is referred to as the "trombone" manoeuvre) is advantageous to air traffic controllers as it allows them great flexibility in spacing aircraft by adjusting the length of the downwind leg.

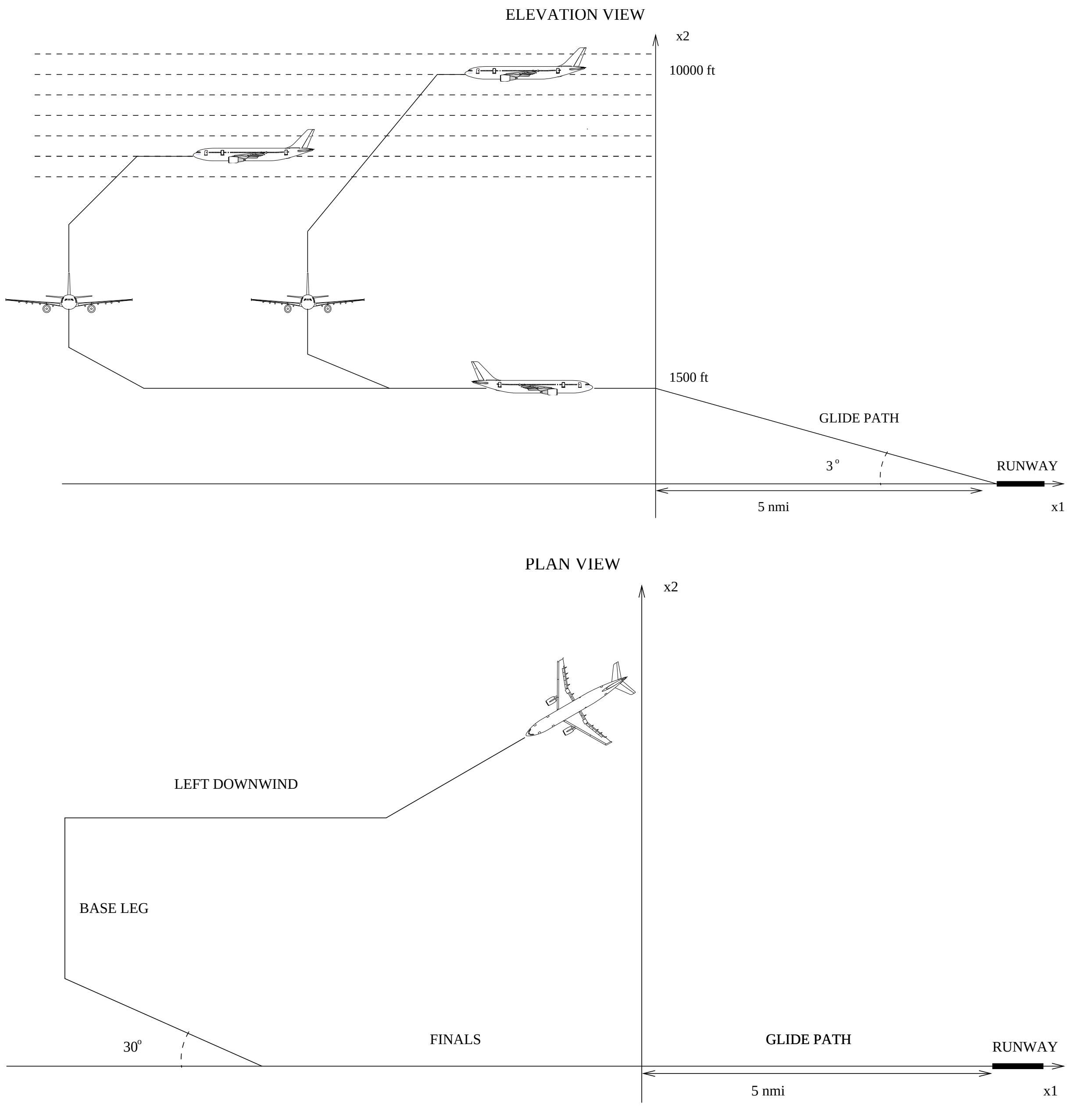


Figure 1: Schematic representation of approach maneuver

6 Simulation examples

In earlier work we developed an air traffic simulator that simulates adequately the behaviour of a set of aircraft from the point of view of ATC [8, 9, 10]. The simulator implements realistic models of current commercial aircraft described in the Base of Aircraft Data (BADA) [5]. The simulator contains also realistic stochastic models of the wind disturbance [3]. The aircraft models contain continuous dynamics, arising from the physical motion of the aircraft, discrete dynamics, arising from the logic embedded in the Flight Management System, and stochastic dynamics, arising from the effect of the wind and incomplete knowledge of physical parameters (for example, the aircraft mass, which depends on fuel, cargo and number of passengers). The simulator has been coded in Java and can be used in different operation modes either to generate accurate data, for validation of the performance of conflict detection and resolution algorithm, or to run faster simulations of simplified models. The nominal path for each aircraft is entered in the simulator as a sequence of way-points. The actual trajectories of the aircraft are then a perturbed version of the nominal path, depending on the particular realizations of wind disturbances and uncertain parameters. The reader is referred to [9, 10] for a more detailed description of the simulator.

Important remark

The air traffic simulator has been used to produce the examples presented in this section. The full accurate aircraft, FMS and wind models have been used both during the Monte Carlo optimisation procedure and to obtain Monte Carlo estimates of post-resolution conflict probabilities. The simulator was invoked from Matlab on a Linux workstation with a Pentium 4 3GHz processor. Under these conditions the simulation of the flight of two aircraft for 30 minutes, which is approximately the horizon considered in the examples, took 0.2 sec on the average. Notice that this simulation speed (5 simulations/sec) is quite low for a Monte Carlo framework. This is mainly due to the fact that no attempt has been made to optimise the code at this stage. For example, the process of executing the Java simulator from a Matlab environment introduced unnecessary and substantial computational overhead. As we will report the computational times required for the following examples, the reader is required to evaluate these times keeping in mind that the simulation environment was far from being optimal.

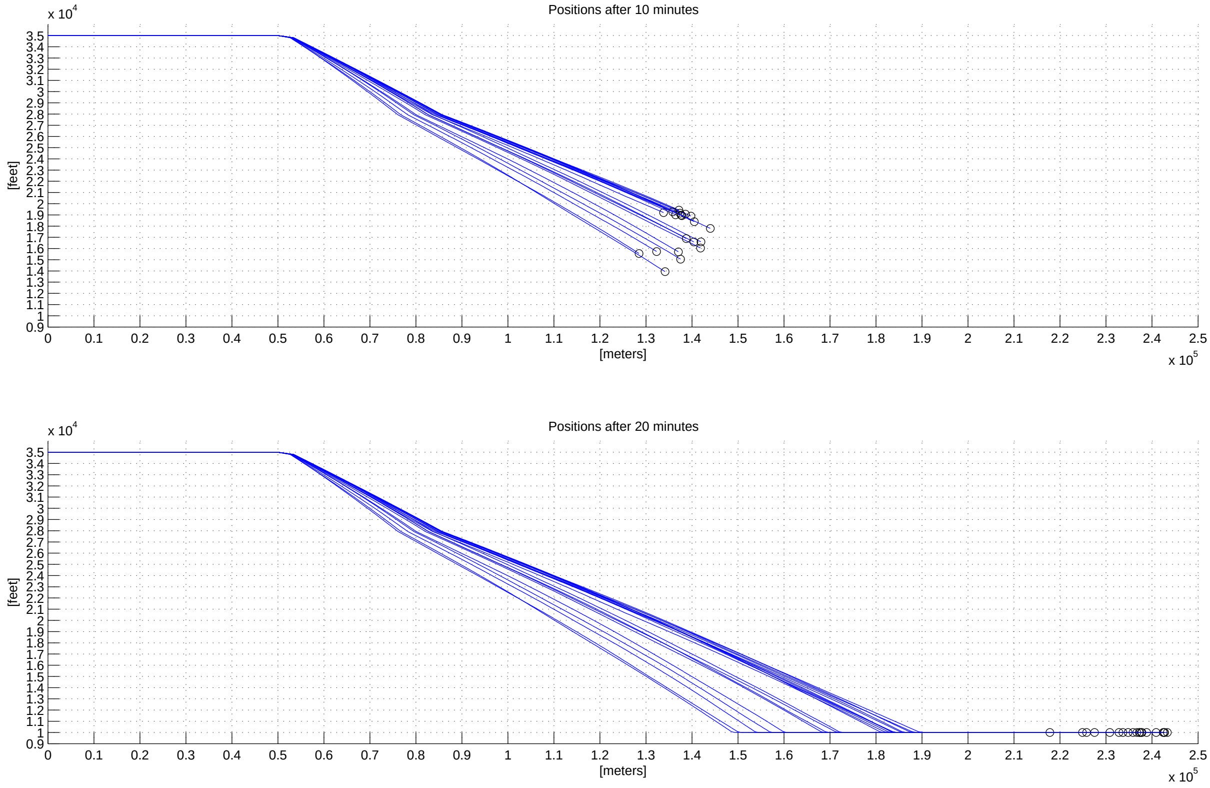
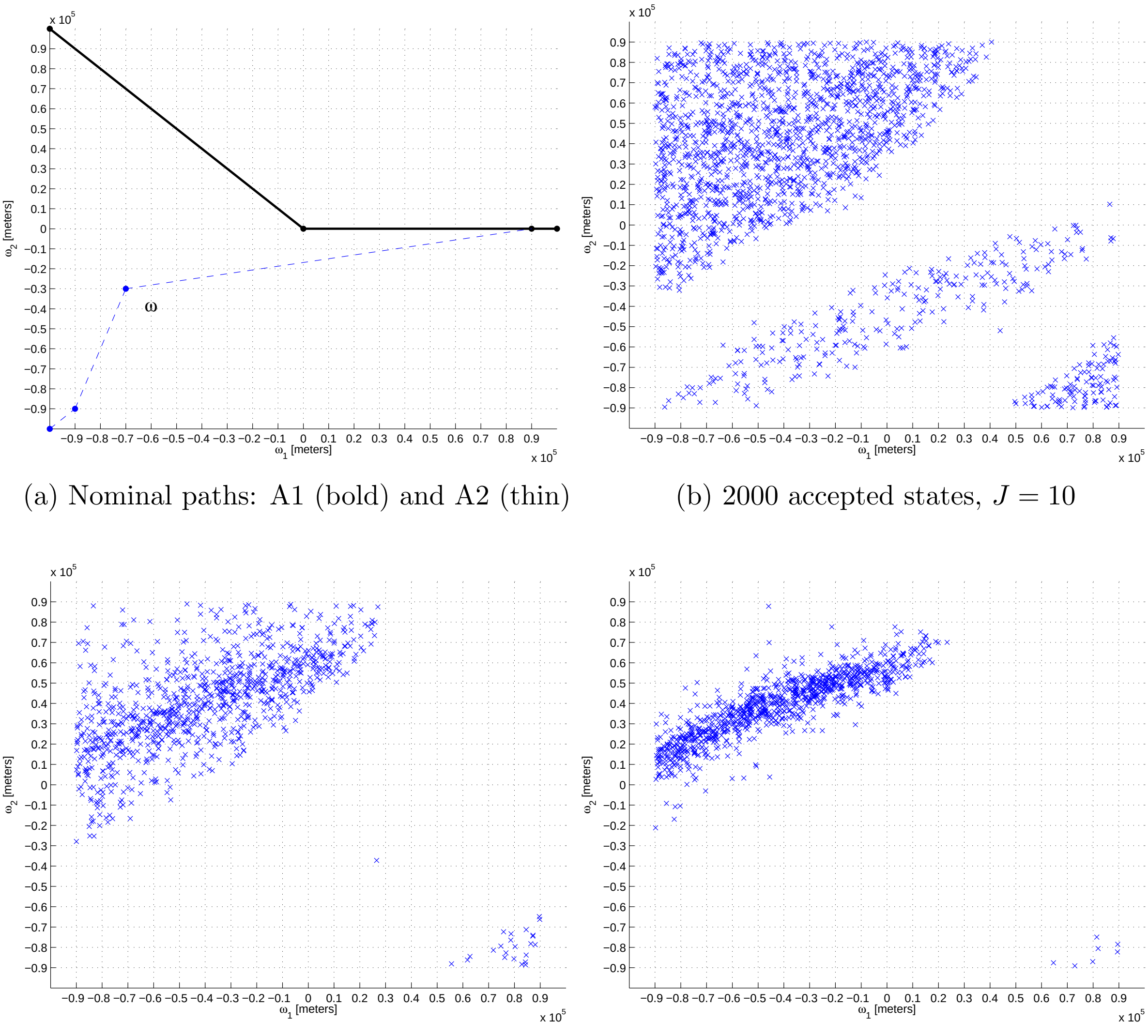


Figure 2: Several trajectory realisations of aircraft descent

6.1 Sequencing aircraft in Terminal Maneuvering Area

We consider the problem of sequencing two aircraft. This is a typical task of ATC in TMA where aircraft descend from cruising altitude and need to be sequenced and separated by a certain time interval before entering in the final Approach Sector. In Figure 2 several possible trajectory realisations of a descending aircraft corresponding to the same nominal path are displayed. In this figure, the aircraft descends from 35000 ft to 10000 ft. In addition to stochastic wind terms, uncertainty about the mass of the aircraft is introduced as an uniform distribution between two extreme values. The figure suggests that the resulting uncertainty in the position of aircraft is of the order of magnitude of some kilometres.

We consider the problem of sequencing two descending aircraft as illustrated in Figure 3-(a). The initial position of the first aircraft (A1) is $[-100000\ 100000]$ (where coordinates are expressed in meters) and altitude 35000 ft. The path of this aircraft is fixed. This aircraft proceeds to waypoint $[-90000\ 90000]$ where it will start a descent to 15000 ft. The trajectory of A1, while descending, is determined by an intermediate waypoint in $[0\ 0]$ and a final waypoint in $[100000\ 0]$, where the aircraft is assumed to exit the sector and enter the approach sector. The second aircraft (A2) is initially at $[-100000\ -100000]$ and altitude 35000 ft. This aircraft proceeds to waypoint $[-90000\ -90000]$ where it will



(a) Nominal paths: A1 (bold) and A2 (thin)

(b) 2000 accepted states, $J = 10$

(c) 1000 accepted states, $J = 50$

(d) 1000 accepted states, $J = 100$

Figure 3: Accepted states during MCMC simulation

start the descent to 15000 ft. The intermediate waypoint $\omega = [\omega_1 \ \omega_2]$ must be selected in the range $\omega_1 \, , \omega_2 \in [-90000 \ 90000]$. The aircraft will then proceed to waypoint [90000 0] and then to the exit waypoint [100000 0].

We assume that the objective is to obtain a time separation of 300 sec between the arrivals of the two aircraft at the final waypoint. Performance in this sense is measured by perf $= e^{-a\cdot(|T_1-T_2|-300)}$ where T_1 and T_2 are the arrival times at the final waypoint and $a = 5 \cdot 10^{-3}$. The constraint is that the trajectory of the two aircraft should not be conflicting. In our simulations we define a conflict as the situation in which two aircraft have less than 5 nmi of horizontal separation and less than 1000 ft of vertical separation (in the TMA of large airports horizontal minimal separation is sometimes reduced to 3 nmi). In these simulation examples, safe separation is defined by a protected zone centred around each aircraft having radius 5 nmi and height 2000 ft. We optimise initially with an upper bound on probability of constraint violation given by $\bar{\mathbf{P}} = 0.1$. It is easy to see that there exists a maneuver in the set of optimisation parameters that gives negligible conflict probability. Therefore, based on inequality (8), we select $\Lambda = 10$ in the optimisation criterion.

The results of the optimisation procedure are illustrated in Figures 3-(b-d). Each figure shows the scatter plot of the accepted parameters during MCMC simulation for different choices of J and search distribution g. In all cases the first 10% of accepted parameters was discarded as a *burn in* period to allow convergence of the chain to its stationary distribution. For each case we give also the ratio between accepted and proposed states during MCMC simulation. Figure 3.(b) illustrates the case $J = 10$. Regions characterised by a low density of accepted parameters can be clearly seen in the figure. These are parameters which correspond to nominal paths with high probability of conflict. The figure also shows distinct “clouds” of accepted maneuvers. They correspond to different sequences of arrivals at the exit point: either A1 arrives before A2 (top left and bottom right clouds) or A1 arrives after A2 (middle cloud). In this case the proposal distribution g was uniform over the parameter space. The ratio accepted/proposed states was 0.27. This means that approximately $1000 \cdot 10/0.27$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/seconds, the required computational time to obtain 1000 states was then approximately 2 hours. Figure 3.(c) illustrates the case $J = 50$. In this case the proposal distribution g was a sum of 2000 Gaussian distributions $N(\mu, \sigma^2 I)$ with variance $\sigma^2 = 10^7\, m^2$. The means of Gaussian distributions were 2000 parameters randomly chosen from those accepted in the MCMC simulation for $J = 10$. The choice of this proposal distribution gives clear computational advantages since less computational time is spent searching over regions of non optimal parameters. In this case the ratio accepted/proposed states was 0.34. This means that approximately $1000 \cdot 50/0.34$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/seconds, the required computational time to obtain 1000 states was then approximately 8 hours. Figure 3.(d) illustrates the case $J = 100$ and proposal distribution constructed as before from states accepted for $J = 50$. Here the ratio accepted/proposed states was 0.3. This means that approximately $1000 \cdot 100/0.3$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/seconds, the required computational time to obtain 1000 states was then approximately 18 hours. Figure 3.(d) indicates that a nearly optimal maneuver is $\omega_1 = -40000$ and $\omega_2 = 40000$. The probability of conflict for this maneuver, estimated by 1000 Monte-Carlo runs, was zero. The estimated expected time separation between arrivals was 283 sec.

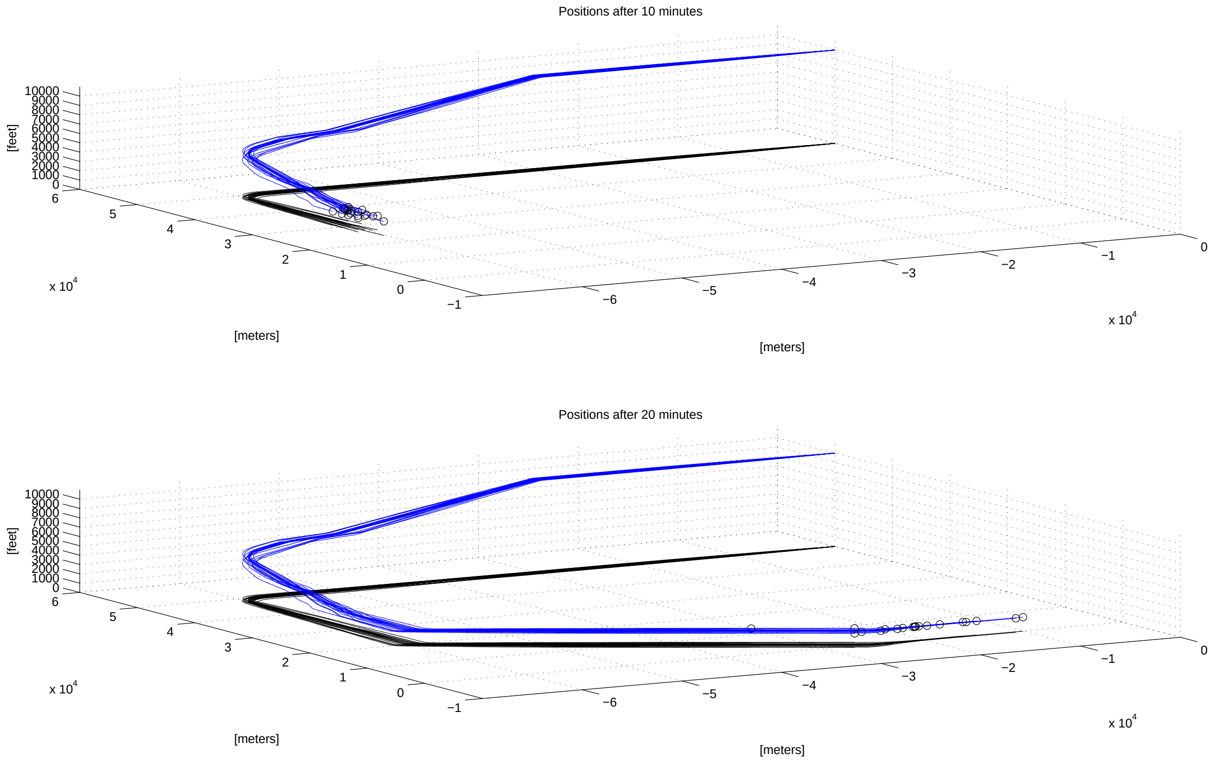


Figure 4: Several trajectory realisations of an approach maneuver

6.2 Coordination of approach maneuvers

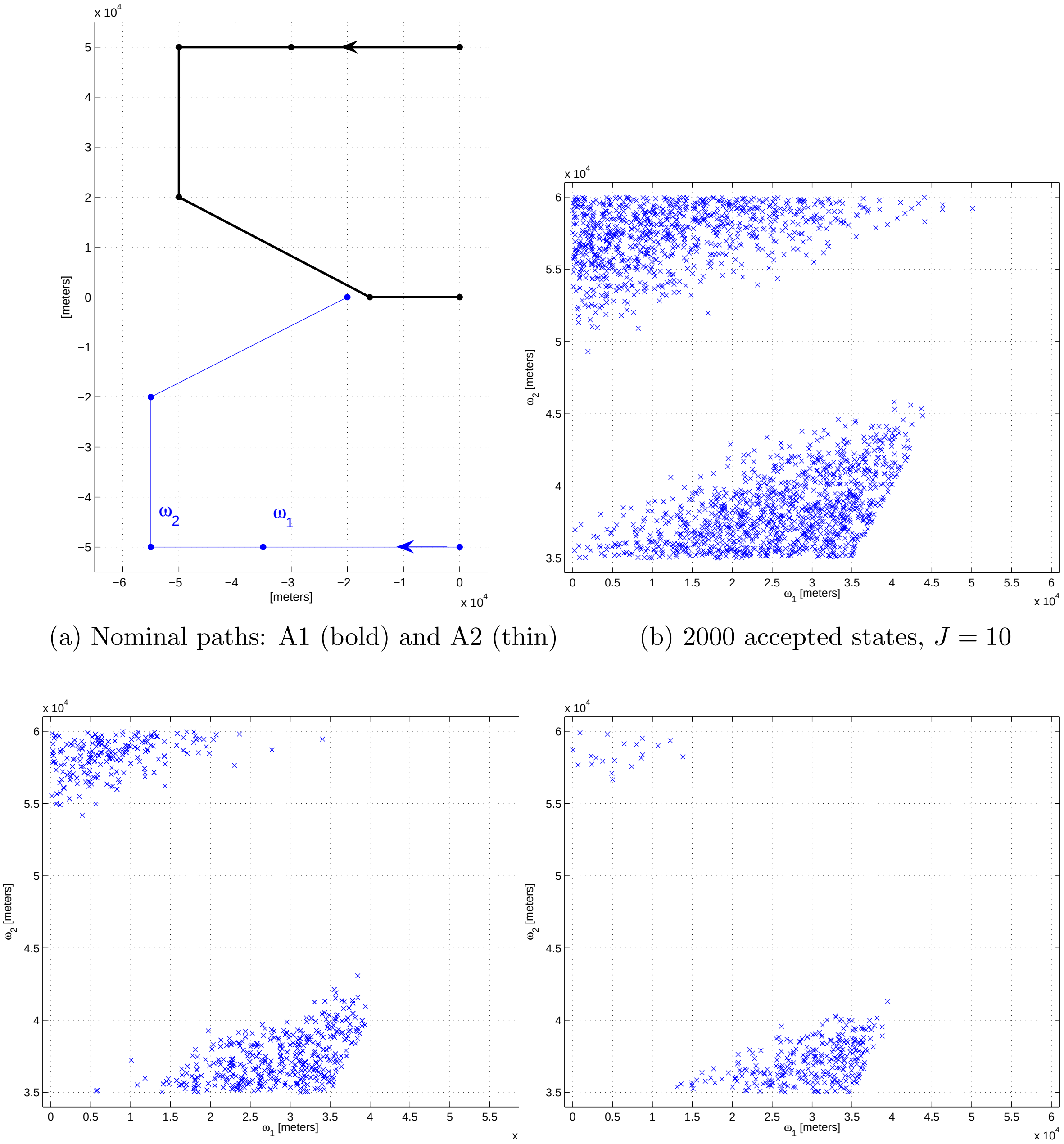
In this section, we optimise an approach maneuver with coordination between two aircraft. In Figure 4 several trajectory realisations, of an aircraft performing the approach maneuver described in Section 5, are displayed. Here, the aircraft is initially at 10000 ft and descends to 1500 ft during the approach. Uncertainty in the trajectory is due to the action of the wind and randomness in the aircraft' mass. In the final leg a function that emulates the localiser and so eliminates cross-track error is implemented.

The problem is illustrated in Figure 5-(a). We consider Aircraft One (A1) and Aircraft Two (A2) approaching the runway. The glide path towards the runway starts at the origin of the reference frame and coordinates are expressed in meters. The aircraft are initially in level flight. The parameters of the approach maneuver are the distance, from initial position, of the start of the final descent (ω_1) and the length of the downwind leg (ω_2).

The initial position of A1 is [0 50000] and altitude 10000 ft. The approach maneuver of this aircraft is fixed to $\bar{\omega}_1 = 30000$ and $\bar{\omega}_2 = 50000$. The initial position of A2 is [0 50000] and altitude 10000 ft. The parameters of its approach maneuver will be selected using the optimisation algorithm. The range of the optimisation parameters is $\omega_2 \in [35000,\ 60000]$ and $\omega_1 \in [0,\ \omega_2]$.

We assume that the performance of the approach maneuver is measured by the arrival time of A2 at the start of the glide path (T_2). The measure of performance is given by perf $= e^{-a \cdot T_2}$ with $a = 5 \cdot 10^{-4}$. The constraint is that the trajectory of A2 is not in conflict with the trajectory of A1. Aircraft 2 must also reach the altitude of 1500 ft before the start of the glide path. We optimise initially with an upper bound on probability of constraint violation given by $\bar{\mathbf{P}} = 0.1$. Since there exists a maneuver in the set of optimisation parameters that gives negligible conflict probability, we select $\Lambda = 10$ in the optimisation criterion according to inequality (8).

The results of the optimisation procedure are illustrated in Figures 5-(b-d) for different values of J and proposal distribution g. Each figure shows the scatter plot of the accepted parameters during MCMC simulation. In all cases the first 10% of accepted parameters was discarded as a *burn in* period to allow convergence of the chain to its stationary distribution. Figure 5-(b) illustrates the case $J = 10$ and proposal distribution g uniform over the parameter space. In this case, the ratio between accepted and proposed parameters during MCMC simulation was 0.23. This means that approximately $1000 \cdot 10/0.23$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/seconds, the required computational time to obtain 1000 states was then approximately 2.4 hours. A region characterised by a low density of accepted parameters can be clearly seen in the figure. These are parameters which correspond to a conflicting maneuver where the aircraft are performing an almost symmetrical approach. The figure also shows two distinct "clouds" of accepted maneuvers. They correspond to a discrete choice that the air traffic controller has to make: either land A2 before A1 (bottom right cloud) or land A1 before A2 (top left cloud). Figure 5-(c) illustrates the case $J = 50$. In this case the proposal distribution g was a sum of 100 Gaussian distributions $N(\mu, \sigma^2 I)$ with variance $\sigma^2 = 10^5 \, m^2$. The means of Gaussian distributions were 100 parameters chosen from those accepted in the MCMC simulation for $J = 10$. In this case the ratio between accepted and proposed parameters was 0.25. This means that approximately $1000 \cdot 50/0.25$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/seconds, the required computational time to obtain 1000 states was then approximately 11 hours. Figure 5-(d) illustrates the case $J = 100$ and proposal distribution constructed as before from states accepted for $J = 50$. In this case the ratio between accepted and proposed parameters was 0.3. This means that approximately $1000 \cdot 100/0.3$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/seconds, the required computational time to obtain 1000 states was then approximately 18 hours. Figure 5-(d) indicates that a nearly optimal maneuver is $\omega_1 = 35000$ and $\omega_2 = 35000$. The probability of conflict for this maneuver, estimated by 1000 Monte-Carlo runs, was zero.



(a) Nominal paths: A1 (bold) and A2 (thin)

(b) 2000 accepted states, $J = 10$

(c) 1000 accepted states, $J = 50$

(d) 1000 accepted states, $J = 100$

Figure 5: Accepted states during MCMC simulation

7 Conclusions

In this paper we illustrated our current approach to air traffic conflict resolution in a stochastic setting based on the use of Monte Carlo methods. The main motivation for our approach is to enable the use of realistic stochastic hybrid models of aircraft flight; Monte Carlo methods appear to be the only ones that allow such models. We have formulated conflict resolution as the optimisation of an expected value criterion with probabilistic constraints. Here, a penalty formulation of the problem has been considered which guarantees constraint satisfaction but delivers a suboptimal solution. A side effect of the optimisation procedure is that structural differences between maneuvers are highlighted as "clouds" of maneuvers accepted by the algorithm. We presented the application of this method to two realistic scenarios inspired by manoeuvring in the terminal area and final approach respectively. The solutions proposed by our algorithm were tested by Monte-Carlo simulations. The solutions proposed by the algorithm gave very good performance in terms of post-resolution conflicts.

Our current research is concerned with overcoming the suboptimality imposed by the penalty formulation of the constrained optimisation problem considered in Section 3. A possible way is to use the Monte Carlo Markov Chain procedure presented in Section 4 to obtain optimization parameters that satisfy the constraint and then to optimise over this set in a successive step. We are also working on sequential Monte-Carlo implementations of the optimisation algorithm [4]. This will allow considerable computational savings, since it will enable the re-use of simulations from one step of the procedure to the next. Finally, we are continuing to work on modelling and implementation in the simulator of typical ATC situation with a realistic parameterisation of control actions and control objectives.

References

[1] H.A.P. Blom and G.J. Bakker. Conflict probability and incrossing probability in air traffic management. In *IEEE Conference on Decision and Control*, Las Vegas, Nevada, U.S.A., December 2002.

[2] EUROCONTROL Experimental Centre. Analysis of the current situation in the Paris, London and Frankfurt TMA. Technical Report Work Package 1: Final Project Report, FALBALA, April 27, 2004. Available from World Wide Web: `http://www.eurocontrol.int/care/asas/falbala-forum/falbala-diss.htm`.

[3] R.E. Cole, C. Richard, S. Kim, and D. Bailey. An assessment of the 60 km rapid update cycle (ruc) with near real-time aircraft reports. Technical Report NASA/A-1, MIT Lincoln Laboratory, July 15, 1998.

[4] A. Doucet, N de Freitas, and N. Gordon (eds). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[5] EUROCONTROL Experimental Centre. *User Manual for the Base of Aircraft Data (BADA) — Revision 3.3*. 2002. Available from World Wide Web: `http://www.eurocontrol.fr/projects/bada/`.

[6] EUROCONTROL Experimental Centre. *Air-Traffic Control Familiarisation Course*. 2004.

[7] E. Frazzoli, Z.H. Mao, J.H. Oh, and E. Feron. Aircraft conflict resolution via semi-definite programming. *AIAA Journal of Guidance, Control, and Dynamics*, 24(1):79–86, 2001.

[8] W. Glover and J. Lygeros. A stochastic hybrid model for air traffic control simulation. In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control*, number 2993 in LNCS, pages 372–386. Springer Verlag, 2004.

[9] W. Glover and J. Lygeros. A multi-aircraft model for conflict detection and resolution algorithm evaluation. Technical Report WP1, Deliverable D1.3, HYBRIDGE, February 18, 2004. Available from World Wide Web: `http://www.nlr.nl/public/hosted-sites/hybridge/`.

[10] W. Glover and J. Lygeros. Simplified models for conflict detection and resolution algorithms. Technical Report WP1, Deliverable D1.4, HYBRIDGE, June 17, 2004. Available from World Wide Web: `http://www.nlr.nl/public/hosted-sites/hybridge/`.

[11] J. Hu, M. Prandini, and S. Sastry. Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution. *AIAA Journal of Guidance, Control and Dynamics*, 25(5), 2002.

[12] J. Hu, M. Prandini, and S. Sastry. Aircraft conflict detection in presence of spatially correlated wind perturbations. In *AIAA Guidance, Navigation and Control Conf.*, Austin, Texas, USA, 2003.

[13] R. Irvine. A geometrical approach to conflict probability estimation. In *4th USA/Europe Air Traffic Management R&D Seminar*, Budapest, Hungary, 2001.

[14] J. Krystul and H.A.P. Blom. Monte Carlo simulation of rare events in hybrid systems. Technical Report WP8, Deliverable D8.3, HYBRIDGE, 2004. Available from World Wide Web: http://www.nlr.nl/public/hosted-sites/ hybridge/.

[15] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[16] P. Mueller. Simulation based optimal design. In *Bayesian Statistics 6*, pages 459–474. J.O. Berger, J.M. Bernardo, A.P. Dawid and A.F.M. Smith (eds.), Oxford University Press, 1999.

[17] P. Mueller, B. Sanso, and M. De Iorio. Optimal Bayesian design by inhomogeneous Markov chain simulation. Technical report, 2003. Available from World Wide Web: http://www.ams.ucsc.edu.

[18] R.A. Paielli and H. Erzberger. Conflict probability estimation for free flight. *Journal of Guidance, Control and Dynamics*, 20(3):588–596, 1997. Available from World Wide Web: `http://www.ctas.arc.nasa.gov/publications/papers/`.

[19] M. Prandini, J. Hu, J. Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(4), 2000.

[20] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.

[21] C. Tomlin, G. Pappas, and S. Sastry. Conflict resolution for Air Traffic Management: a case study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):5–521, 1998.

[22] P.J.M. van Laarhoven and E.H. Aarts. *Simulated Annealing: Theory and Applications*. D.Reidel Publishing Company, 1987.