

# Monte Carlo Optimization for Conflict Resolution in Air Traffic Control \*

A. Lecchini<sup>†</sup>, W. Glover<sup>\*</sup>, J. Lygeros<sup>‡</sup> and J. Maciejowski<sup>\*</sup>

## Abstract

The safety of the flights, and in particular separation assurance, is one of the main tasks of Air Traffic Control. Conflict resolution refers to the process used by air traffic controllers to prevent loss of separation. Conflict resolution involves issuing instructions to aircraft to avoid loss of safe separation between them and, at the same time, direct them to their destinations. Conflict resolution requires decision making in the face of the considerable levels of uncertainty inherent in the motion of aircraft. We present a framework for conflict resolution which allows one 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 optimizing an expected value criterion. Optimization of the expected value resolution criterion is carried out through an iterative procedure based on Monte Carlo Markov Chain. Simulation examples inspired by current air traffic control practice in terminal maneuvering areas and approach sectors illustrate the proposed conflict resolution strategy.

## 1 Introduction

In the current organization of the Air Traffic Management (ATM) system the centralized Air Traffic Control (ATC) is in complete control of air traffic and ultimately responsible for safety. Before take off, aircraft file flight plans which cover the entire flight. During the flight, ATC sends additional instructions to them, depending on the actual traffic, to improve traffic flow and avoid dangerous encounters. The primary concern of ATC is to maintain safe separation between the aircraft. The level of accepted minimum safe separation may depend on the density of air traffic and the region of the 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; this is reduced to 3 nmi in approach sectors for aircraft landing and departing. A conflict is defined as a situation of loss of minimum safe separation between two aircraft. If safety is not at stake, ATC

---

\*Work supported by the European Commission under project HYBRIDGE, IST-2001-32460, and the HYCON Network of Excellence, FP6-IST-511368, and by Eurocontrol under project C20051E/BM/03.

<sup>†</sup>Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK, Tel. +44 1223 33260, Fax. +44 1223 332662, {a1394, wg214, jmm}@eng.cam.ac.uk

<sup>‡</sup>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

also tries to fulfill the (possibly conflicting) requests of aircraft and airlines; for example, desired paths to avoid turbulence, or desired time of arrivals to meet schedule. To improve the performance of ATC, mainly in anticipation of increasing levels of air traffic, research effort has been devoted over the last decade on creating tools to assist ATC with conflict detection and resolution tasks. A review of research work in this area of ATC is presented in [15].

Uncertainty is introduced in air traffic by the action of wind, incomplete knowledge of the physical coefficients of the aircraft and unavoidable imprecision in the execution of ATC instructions. To perform conflict detection one has to evaluate the possibility of future conflicts given the current state of the airspace and taking into account uncertainty in the future position of aircraft. 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 aircraft positions [18], or a dynamical model, such as a stochastic differential equation (see, for example, [1, 12, 19]), 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. An example of such a 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 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 Monte Carlo methods) and the accuracy of the model. Techniques to accelerate Monte Carlo methods especially for rare event computations are under development, see for example [14].

For conflict resolution, the objective is to provide suitable maneuvers to avoid a predicted conflict. A number of conflict resolution algorithms have been proposed in the deterministic setting, for example [7, 11, 21]. In the stochastic setting, the research effort has concentrated mainly on conflict detection, and only a few simple resolution strategies have been proposed [18, 19]. The main reason for this is the complexity of stochastic prediction models which makes the quantification of the effects of possible control actions intractable.

In this paper we present a Monte Carlo Markov Chain (MCMC) framework [20] for conflict resolution in a stochastic setting. The aim of the proposed approach is to extend 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 motivated from Bayesian statistics [16, 17]. We consider an expected value resolution criterion that takes into account separation and other factors (e.g. aircraft requests). Then, the MCMC optimization procedure of [16] is employed to estimate the resolution maneuver that optimizes the expected value criterion. The proposed approach is illustrated in simulation, on some realistic benchmark problems, inspired by current ATC practice. The benchmarks were implemented in an air traffic simulator developed in previous work [8, 9, 10].

The material is organized in 5 sections. Section 2 presents the formulation of conflict resolution as an optimization problem. The randomized optimization procedure that we adopt to solve the problem is presented in Section 3. Section 4 is devoted to the benchmark problems used to illustrate our approach. Section 4.1 introduces the problems associated

with ATC in terminal and approach sectors and Section 4.2 provides a brief overview of the simulator used to carry out the experiments. Sections 4.3 and 4.4 present results on benchmark problems in terminal and approach sectors respectively. Conclusions and future objectives are discussed in Section 5. For the readers convenience the acronyms used in the paper are summarized in an appendix.

## 2 Conflict resolution with an expected value criterion

We formulate conflict resolution as a constrained optimization problem. Given a set of aircraft involved in a conflict, the conflict resolution maneuver is determined by a parameter  $\omega$  which defines the nominal paths of the aircraft. From the point of view of the ATC, the execution of the maneuver is affected by uncertainty, due to wind, imprecise knowledge of aircraft parameters (e.g. mass) and Flight Management System (FMS) settings, etc. Therefore, the sequence of actual positions of the aircraft (for example, the sequence of positions observed by ATC 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 two aircraft get too close during the execution of the maneuver. The goal is to select  $\omega$  to maximize 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 this problem in a general framework.

Let  $X$  be a random variable whose distribution depends on some parameter  $\omega$ . The distribution of  $X$  is denoted by  $p_\omega(x)$  with  $x \in \mathbf{X}$ . The set of all possible values of  $\omega$  is denoted by  $\Omega$ . 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 realization  $x$ , of random variable  $X$ , violates the constraint if  $x \notin \mathbf{X}_f$ . The probability of satisfying the constraint for a given  $\omega$  is denoted by  $P(\omega)$

$$P(\omega) = \int_{x \in \mathbf{X}_f} p_\omega(x) dx.$$

The probability of violating the constraint is denoted by  $\bar{P}(\omega) = 1 - P(\omega)$ .

For a realization  $x \in \mathbf{X}_f$  we assume that we are given some definition of performance of  $x$ . In general performance can depend also on the value of  $\omega$ , therefore performance is measured by a function  $\text{perf}(\cdot, \cdot) : \Omega \times \mathbf{X}_f \rightarrow [0, 1]$ . The expected performance for a given  $\omega \in \Omega$  is denoted by  $\text{PERF}(\omega)$ , where

$$\text{PERF}(\omega) = \int_{x \in \mathbf{X}_f} \text{perf}(\omega, x) p_\omega(x) dx.$$

Ideally one would like to select  $\omega$  to maximize 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 optimization problem

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

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

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

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

The optimization problem (1)-(2) is generally difficult to solve, or even to approximate by randomized methods. Here we approximate this problem by an optimization 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.

We introduce a function  $u(\omega, x)$  defined on the entire  $\mathbf{X}$  by

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

with  $\Lambda > 1$ . The parameter  $\Lambda$  represents a reward for constraint satisfaction. For a given  $\omega \in \Omega$ , the expected value of  $u(\omega, x)$  is given by

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

Instead of the constrained optimization problem (1)-(2) we solve the unconstrained optimization problem:

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

Assume the supremum is attained and let  $\omega^*$  denote the optimum solution, i.e.  $U_{\max} = U(\omega^*)$ . The following proposition establishes bounds on the probability of violating the constraints and the level of sub-optimality of  $\text{PERF}(\bar{\omega})$  over  $\text{PERF}_{\max|\bar{P}}$ .

**Proposition 2.1** *The maximizer,  $\bar{\omega}$ , of  $U(\omega)$  satisfies*

$$\bar{P}(\omega^*) \leq \bar{P}_{\min} + \frac{1}{\Lambda}(1 - \bar{P}_{\min}) \quad (4)$$

$$\text{PERF}(\omega^*) \geq \text{PERF}_{\max|\bar{P}} - (\Lambda - 1)(\bar{P} - \bar{P}_{\min}). \quad (5)$$

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

$$\begin{aligned} U(\omega) &= \int_{x \in \mathbf{X}_f} (\text{perf}(\omega, x) + \Lambda) p_\omega(x) dx + \int_{x \notin \mathbf{X}_f} p_\omega(x) dx \\ &= \text{PERF}(\omega) + \Lambda - (\Lambda - 1)\bar{P}(\omega). \end{aligned}$$

By the definition of  $\omega^*$  we have that  $U(\omega^*) \geq U(\omega)$  for all  $\omega \in \Omega$ . We therefore can write

$$\text{PERF}(\omega^*) + \Lambda - (\Lambda - 1)\bar{P}(\omega^*) \geq \text{PERF}(\omega) + \Lambda - (\Lambda - 1)\bar{P}(\omega) \quad \forall \omega \in \Omega.$$

Since  $0 \leq \text{perf}(\omega, x) \leq 1$ ,  $\text{PERF}(\omega)$  satisfies

$$0 \leq \text{PERF}(\omega) \leq P(\omega) = 1 - \bar{P}(\omega). \quad (6)$$

Therefore we obtain

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

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

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

$$\text{PERF}(\bar{\omega}) \geq \text{PERF}(\omega) - (\Lambda - 1) [\bar{\mathbf{P}}(\omega) - \bar{\mathbf{P}}(\bar{\omega})] \quad \forall \omega : \bar{\mathbf{P}}(\omega) \leq \bar{\mathbf{P}}.$$

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

$$\text{PERF}(\bar{\omega}) \geq \text{PERF}(\omega) - (\Lambda - 1) [\bar{\mathbf{P}} - \bar{\mathbf{P}}_{\min}] \quad \forall \omega : \bar{\mathbf{P}}(\omega) \leq \bar{\mathbf{P}}.$$

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

Proposition 2.1 suggests a method for choosing  $\Lambda$  to ensure that the solution  $\bar{\omega}$  of the optimization problem will satisfy  $\bar{\mathbf{P}}(\bar{\omega}) \leq \bar{\mathbf{P}}$ . In particular it suffices to know  $\bar{\mathbf{P}}(\omega)$  for some  $\omega \in \Omega$  with  $\bar{\mathbf{P}}(\omega) < \bar{\mathbf{P}}$  to obtain a bound. If there exists  $\omega \in \Omega$  for which  $\hat{\mathbf{P}} = \bar{\mathbf{P}}(\omega) < \bar{\mathbf{P}}$  is known, then any

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

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

$$\Lambda \geq \frac{1}{\bar{\mathbf{P}}} \tag{7}$$

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

### 3 Monte Carlo Optimization

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

The optimization strategy relies on extractions of a random variable  $\Omega$  whose distribution has modes which coincide with the optimizers of  $U(\omega)$ . These extractions are obtained through Monte Carlo Markov Chain (MCMC) simulation [20]. The problem of optimizing the expected criterion is then reformulated as the problem of estimating the optimal points from extractions concentrated around them. In the optimization procedure, there exists a tunable parameter that governs the trade-off between estimation accuracy of the optimizer and computational effort. In particular, the distribution of  $\Omega$  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 optimization procedure, this approach can be seen as the counterpart of Simulated Annealing for a stochastic setting. Simulated Annealing is a randomized optimization strategy developed to find tractable approximate solutions to complex deterministic combinatorial optimization problems, [22]. A formal parallel between these two strategies has been derived in [17].

The MCMC optimization procedure can be described as follows. Consider a stochastic model formed by a random variable  $\Omega$ , 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  $h(\omega, x_1, x_2, \dots, x_J)$  the joint distribution of  $(\Omega, X_1, X_2, X_3, \dots, X_J)$ . It is straightforward to see that if

$$h(\omega, x_1, x_2, \dots, x_J) \propto \prod_{j=1}^J u(\omega, x_j) p_\omega(x_j) \quad (8)$$

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

$$h(\omega) \propto \left[ \int u(\omega, x) p_\omega(x) dx \right]^J = U(\omega)^J. \quad (9)$$

This means that if we can extract realizations of  $(\Omega, X_1, X_2, X_3, \dots, X_J)$  then the extracted  $\Omega$ '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 optimization of  $U(\omega)$ . Realizations of the random variables  $(\Omega, X_1, X_2, X_3, \dots, X_J)$ , with the desired joint probability density given by (8), 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  $h(\omega)$ .

### Algorithm 1 (MCMC Algorithm)

**initialization:**

- Extract  $\Omega(0) \sim g(\omega)$
- Extract  $X_j(0) \sim p_\Omega(x) \quad j = 1, \dots, J$
- Compute  $U_J(0) = \prod_{j=1}^J u(\Omega(0), X_j(0))$
- Set  $k = 0$

**repeat**

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

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

Compute  $\tilde{U}_J = \prod_{j=1}^J u(\tilde{\Omega}, \tilde{X}_j)$

Set  $\rho = \min \left\{ 1, \frac{\tilde{U}_J g(\omega(k))}{u_J(k) g(\tilde{\Omega})} \right\}$

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

Set  $k=k+1$

**until** True

In the description of the algorithm, lower- and upper-case symbols denote respectively quantities that are known at the iteration  $k$  and quantities that are extracted at the iteration  $k$ . Notice for example that  $[\omega(k), u_J(k)]$  denotes the current state and that  $[\Omega(k+1), U_J(k+1)]$  denotes the subsequent state of the chain. In the initialization step the state  $[\Omega(0), U_J(0)]$  is always accepted. In subsequent steps the new extraction  $[\tilde{\Omega}, \tilde{U}_J]$  is accepted with probability  $\rho$  otherwise it is rejected and the previous state of the Markov chain  $[\omega(k), u_J(k)]$  is maintained. Practically, the algorithm is executed until a certain number of extractions (say 1000) have been accepted. Because we are interested in the stationary distribution of the Markov chain, the first few (say 10%) of the accepted states are discarded to allow the chain to reach its stationary distribution (“burn in period”).

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

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

By inserting (8) in the above expression one obtains  $\rho(\omega, u_J, \tilde{\omega}, \tilde{u}_J)$ . Under minimal assumptions, the Markov Chain generated by the  $\Omega(k)$  is uniformly ergodic with stationary distribution  $h(\omega)$  given by (9). Therefore, after a burn in period, the extractions  $\Omega(k)$  accepted by the algorithm will concentrate around the modes of  $h(\omega)$ , which, by (9) coincide with the optimal points of  $U(\omega)$ . Results that characterize 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 optimizers 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

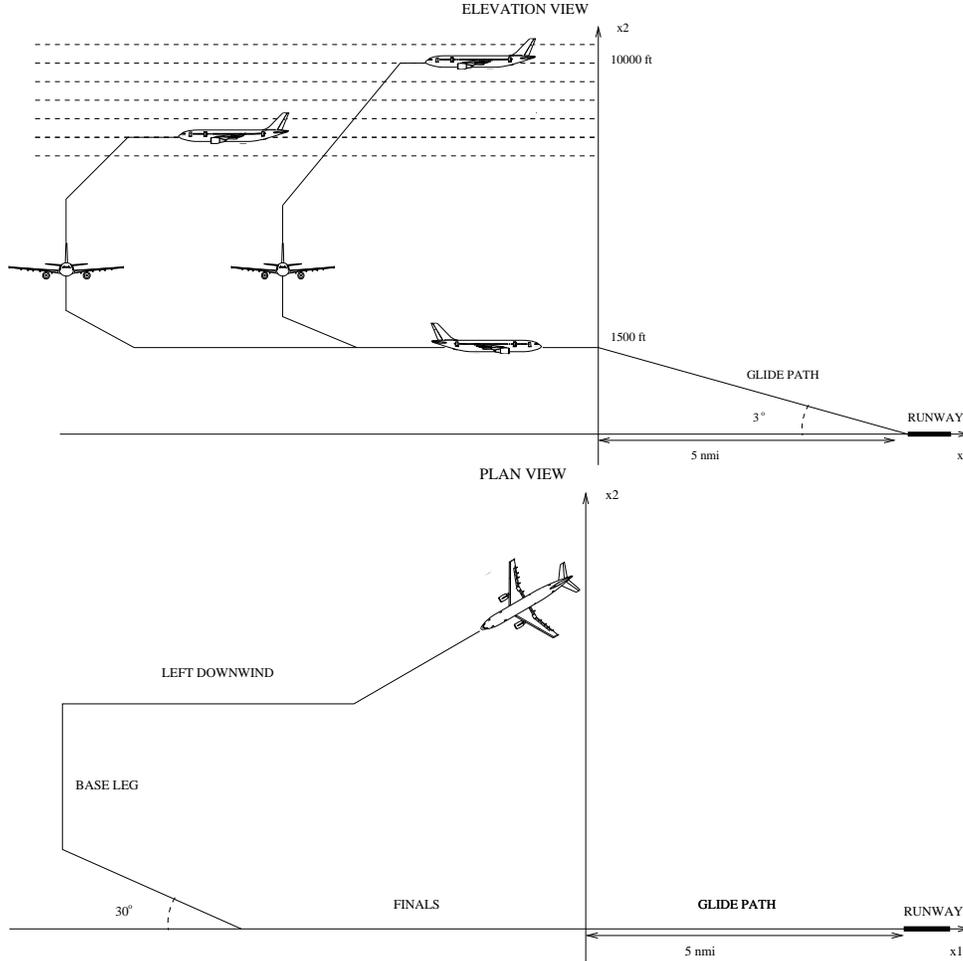


Figure 1: Schematic representation of approach maneuver

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  $\Lambda$ , which is related to probability of constraint violation. In particular, the greater  $\Lambda$  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 optimization of  $U(\omega)$ .

## 4 ATC in terminal and approach sectors

### 4.1 Current practice

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 organization, the traffic at these altitudes has an en-route structure, which facilitates the action of ATC. Aircraft follow pre-specified 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 the 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 runway for landing. 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 centered 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, centered on the aircraft with radius 5nmi and height 2000ft (i.e. aircraft which do not have 5nmi of horizontal separation must have 1000ft of vertical separation).

2) *Take aircraft from entry altitude down to intercept the localizer.* 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 *localizer*, 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 pre-specified 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 way-points (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  $90^\circ$ , to approach the localizer. This second segment is called the *base leg*. Aircraft perform an additional turn in order to intercept the plane of the localizer with an angle of incidence of approximately  $30^\circ$ . The reason is that  $30^\circ$  is a suitable angle for pilots to perform the final turn in the direction of the runway as soon as possible when the localizer has been intercepted. It is required that aircraft intercept the localizer 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 maneuver”) is advantageous to air traffic controllers as it allows them great flexibility in spacing aircraft by adjusting the length of the downwind leg (hence the name trombone).

## 4.2 Simulation procedure

A necessary step in the implementation of the MCMC algorithm is the extraction of the  $\tilde{X}_j$  for a given value of  $\omega$ . In the ATC case studies this corresponds to the extraction of aircraft trajectories for a given airspace configuration (current aircraft positions and given flight plans). The extraction of multiple such trajectories ( $J$  to be precise) will be necessary in the algorithm implementation. The trajectories will be in general different from one another because of the uncertainty that enters the process (due to wind, aircraft parameters unknown to ATC, etc.)

In earlier work we developed an air traffic simulator to simulate adequately the behavior 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, either by reading data files, or manually. The simulator can also be used in a so called interactive mode, where the user is allowed to manipulate the flight plans on-line (move way-points, introduce new way points, etc.)

The actual trajectories of the aircraft generated by the simulator are a perturbed version of the flight plan that depends on the particular realizations of wind disturbances and uncertain parameters. The trajectories of the aircraft can be displayed on the screen (Figure 2) and/or stored in files for post-processing. The reader is referred to [9, 10] for a more detailed description of the simulator.

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 optimization 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 seconds 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 optimize the code at this stage. For example, executing the Java simulator from a Matlab environment introduces unnecessary and substantial computational overhead. The reader is requested to evaluate the computation times reported in the following examples keeping this fact in mind.

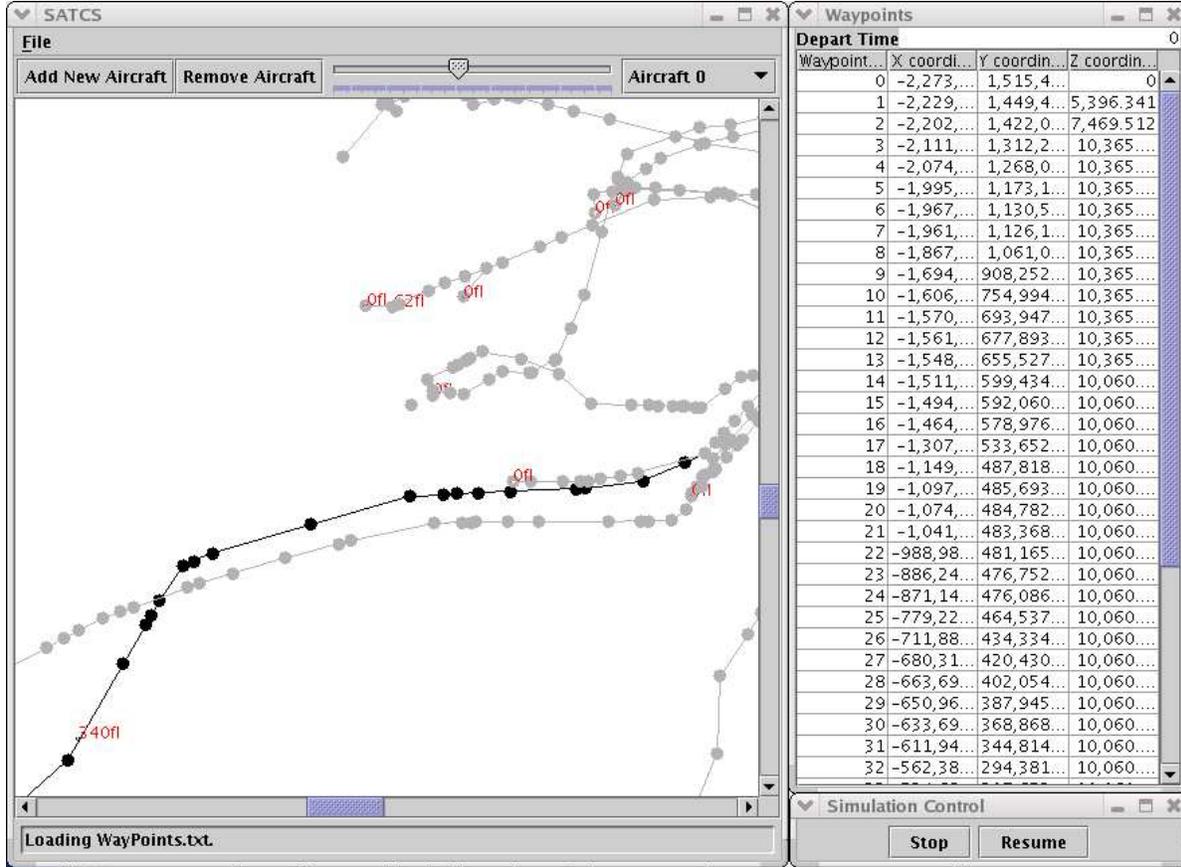


Figure 2: Screen shot of a simulation of all flights using the Barcelona airport on a given day.

### 4.3 Sequencing aircraft in Terminal Maneuvering Areas

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 3 several possible trajectory realizations of a descending aircraft corresponding to the same nominal path are displayed. In this figure, the aircraft descends from FL350 to FL100. In addition to stochastic wind terms, uncertainty about the mass of the aircraft is introduced as a 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 kilometers.

We consider the problem of sequencing two descending aircraft as illustrated in Figure 4(a). The initial position of the first aircraft (A1) is  $(-100000, 100000)$  (where coordinates are expressed in meters) and FL350. The path of this aircraft is fixed: The aircraft proceeds to way-point  $(-90000, 90000)$  where it will start a descent to FL150. The trajectory of A1, while descending, is determined by an intermediate way-point in  $(0, 0)$  and a final way-point in  $(100000, 0)$ , where the aircraft is assumed to exit the TMA and enter the approach sector. The second aircraft (A2) is initially at  $(-100000, -100000)$  and FL350. This aircraft proceeds to way-point  $(-90000, -90000)$  where it will start its

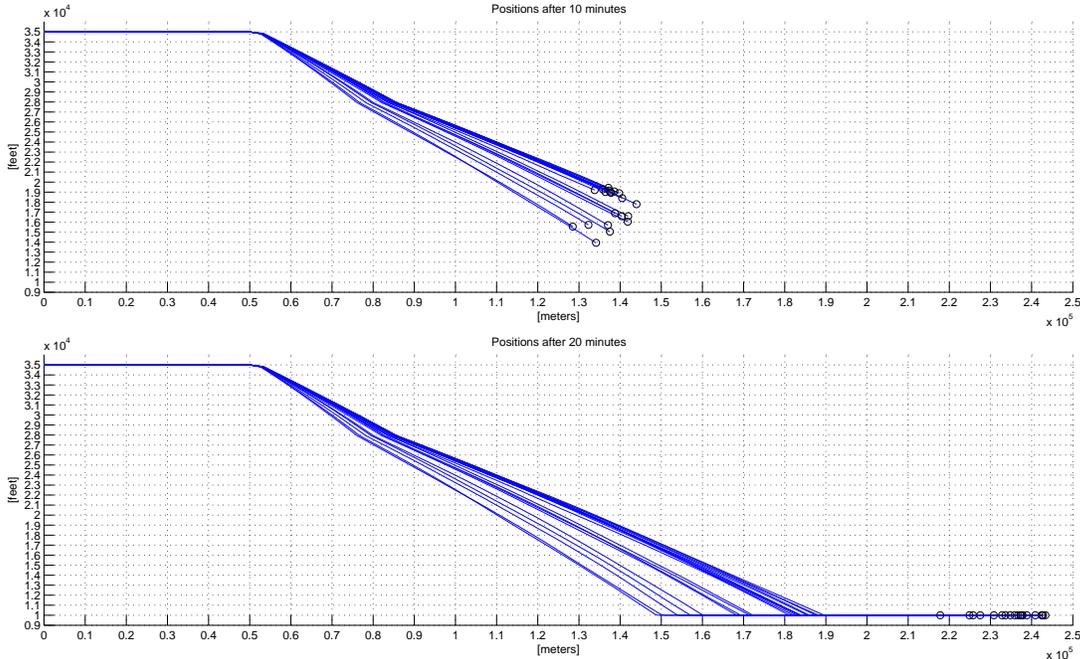


Figure 3: Several trajectory realizations of aircraft descent

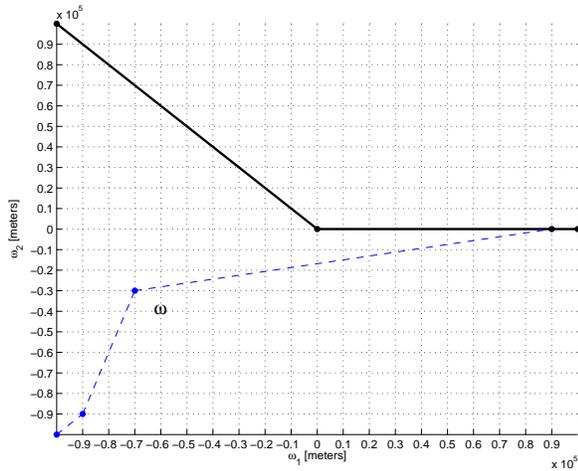
descent to FL150. The intermediate way-point  $\omega = (\omega_1, \omega_2)$  must be selected in the range  $\omega_1, \omega_2 \in [-90000, 90000]$ . The aircraft will then proceed to way-point  $(90000, 0)$  and then to the exit way-point  $(100000, 0)$ .

We assume that the objective is to obtain a time separation of 300 seconds between the arrivals of the two aircraft at the exit way-point,  $(100000, 0)$ . Performance in this sense is measured by  $\text{perf} = e^{-a \cdot (|T_1 - T_2| - 300)}$  where  $T_1$  and  $T_2$  are the arrival times of A1 and A2 at the exit way-point and  $a = 5 \cdot 10^{-3}$ . The constraint is that the trajectories 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 5nmi of horizontal separation and less than 1000ft of vertical separation<sup>1</sup>. We optimize initially with an upper bound on the probability of constraint violation of  $\mathbf{P} = 0.1$ . It is easy to see that there exists a maneuver in the set of optimization parameters that gives negligible conflict probability. Therefore, based on inequality (7), we select  $\Lambda = 10$  in the optimization criterion.

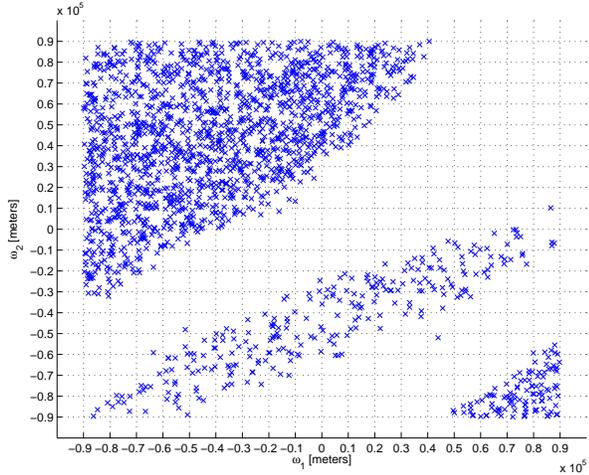
The results of the optimization procedure are illustrated in Figures 4(b-d). Each figure shows the scatter plot of the accepted parameters during MCMC simulation for different choices of  $J$  and search distribution  $g$ . In all cases the first 10% of accepted parameters was discarded as a burn in period, to allow convergence of the Markov chain to its stationary distribution.

Figure 4(b) illustrates the case  $J = 10$ . Regions characterized 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

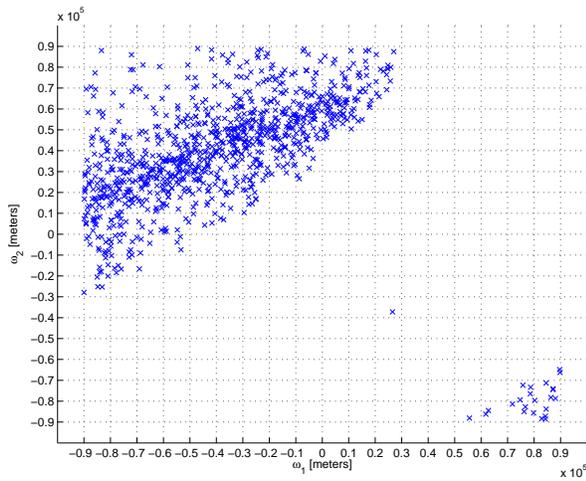
<sup>1</sup>In the TMA of large airports horizontal minimal separation is sometimes reduced to 3nmi, but this fact is ignored here.



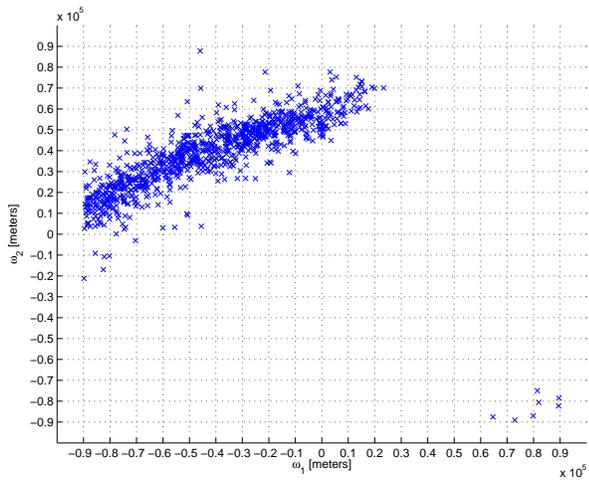
(a) Nominal paths: A1 (bold) and A2 (thin)



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



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



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

Figure 4: Accepted states during MCMC simulation

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 and the ratio of accepted/proposed states was 0.27. This means that approximately  $1100 \cdot 10 / 0.27 = 40740$  simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/second, the required computational time to obtain 1000 accepted states was then approximately 2 hours. In this simulation we actually extracted 5100 states. Figure 4(b) displays the last 2000 extracted states.

Figure 4(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  $1100 \cdot 50/0.34 = 161764$  simulations were needed to obtain 1000 accepted states. At an average of 5 simulations/second, the required computational time to obtain 1000 states was approximately 9 hours.

Figure 4(d) illustrates the case  $J = 100$  and a proposal distribution constructed as before from states accepted for  $J = 50$ . Here the ratio accepted/proposed states was 0.3. This means that approximately  $1100 \cdot 100/0.3 = 366666$  simulations were needed to obtain 1000 accepted states. At an average of 5 simulations/second, the required computational time to obtain 1000 states was approximately 20 hours. Figure 4(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 seconds.

#### 4.4 Coordination of approach maneuvers

In this section, we optimize an approach maneuver with coordination between two aircraft. In Figure 5 several trajectory realizations, of an aircraft performing the approach maneuver described in Section 4.4, are displayed. Here, the aircraft is initially at FL100 and descends to 1500ft 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 localizer and so eliminates cross-track error is implemented.

The problem formulation is illustrated in Figure 6(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 ( $\omega_1$ ) and the length of the downwind leg ( $\omega_2$ ).

The initial position of A1 is  $(0, 50000)$  and FL100. 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 FL100. The parameters of its approach maneuver will be selected using the optimization algorithm. The range of the optimization 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  $\text{perf} = e^{-a \cdot T_2}$  with  $a = 5 \cdot 10^{-4}$ . The constraint is that the trajectories of the two aircraft are not in conflict. In addition, aircraft 2 must also reach the altitude of 1500 ft before the start of the glide path, to ensure that it intercepts the localizer. We optimize initially with an upper bound on probability of constraint violation given by  $\mathbf{P} = 0.1$ . Since there exists a maneuver in the set of optimization parameters that gives negligible conflict probability, we select  $\Lambda = 10$  in the optimization criterion according to inequality (7).

The results of the optimization procedure are illustrated in Figures 6(b-d) for different values of  $J$  and proposal distribution  $g$ . Each figure shows the scatter plot of the accepted parameters during MCMC simulation. Again, the first 10% of accepted parameters were discarded in each case as a burn in period. Figure 6(b) illustrates the case  $J = 10$  and proposal distribution  $g$  uniform over the parameter space. In this case, the ratio

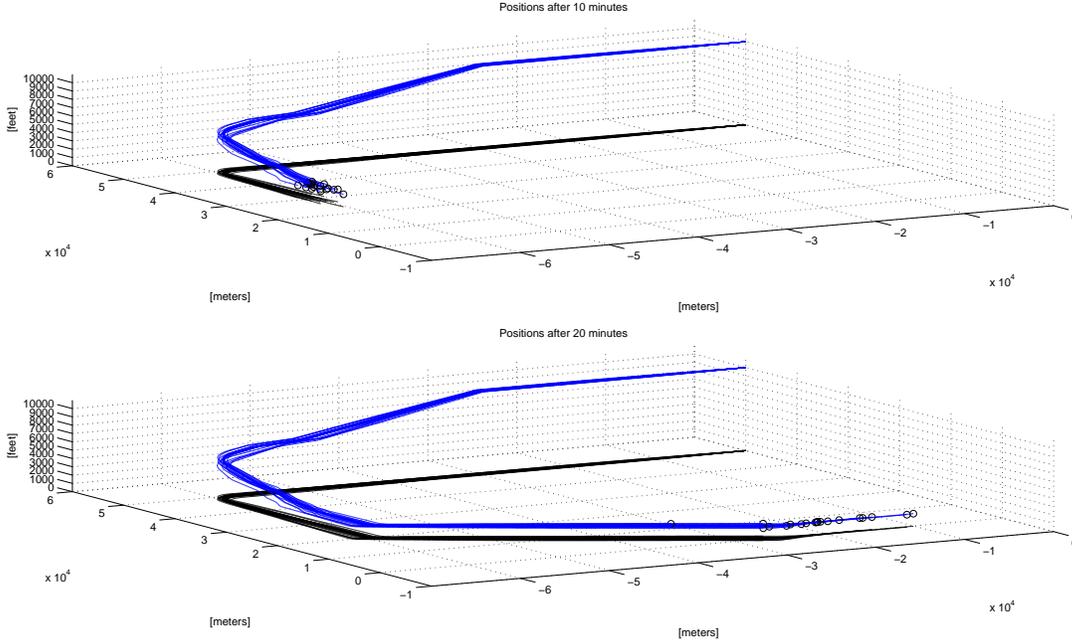


Figure 5: Several trajectory realizations of an approach maneuver

between accepted and proposed parameters during MCMC simulation was 0.23. This means that approximately  $1100 \cdot 10 / 0.23 = 47826$  simulations were needed to obtain 1000 accepted states, which took approximately 2.6 hours. A region characterized 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 6(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 means selected randomly among the maneuvers accepted for  $J = 10$  and variance  $\sigma^2 = 10^5 m^2$ . In this case the ratio between accepted and proposed parameters was 0.25. This means that approximately  $1100 \cdot 50 / 0.25 = 220000$  simulations were needed to obtain 1000 accepted states, which required approximately 12 hours of computation.

Figure 6(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  $1100 \cdot 100 / 0.3 = 366666$  simulations were needed to obtain 1000 accepted states (approximately 20 hours). Figure 6(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.

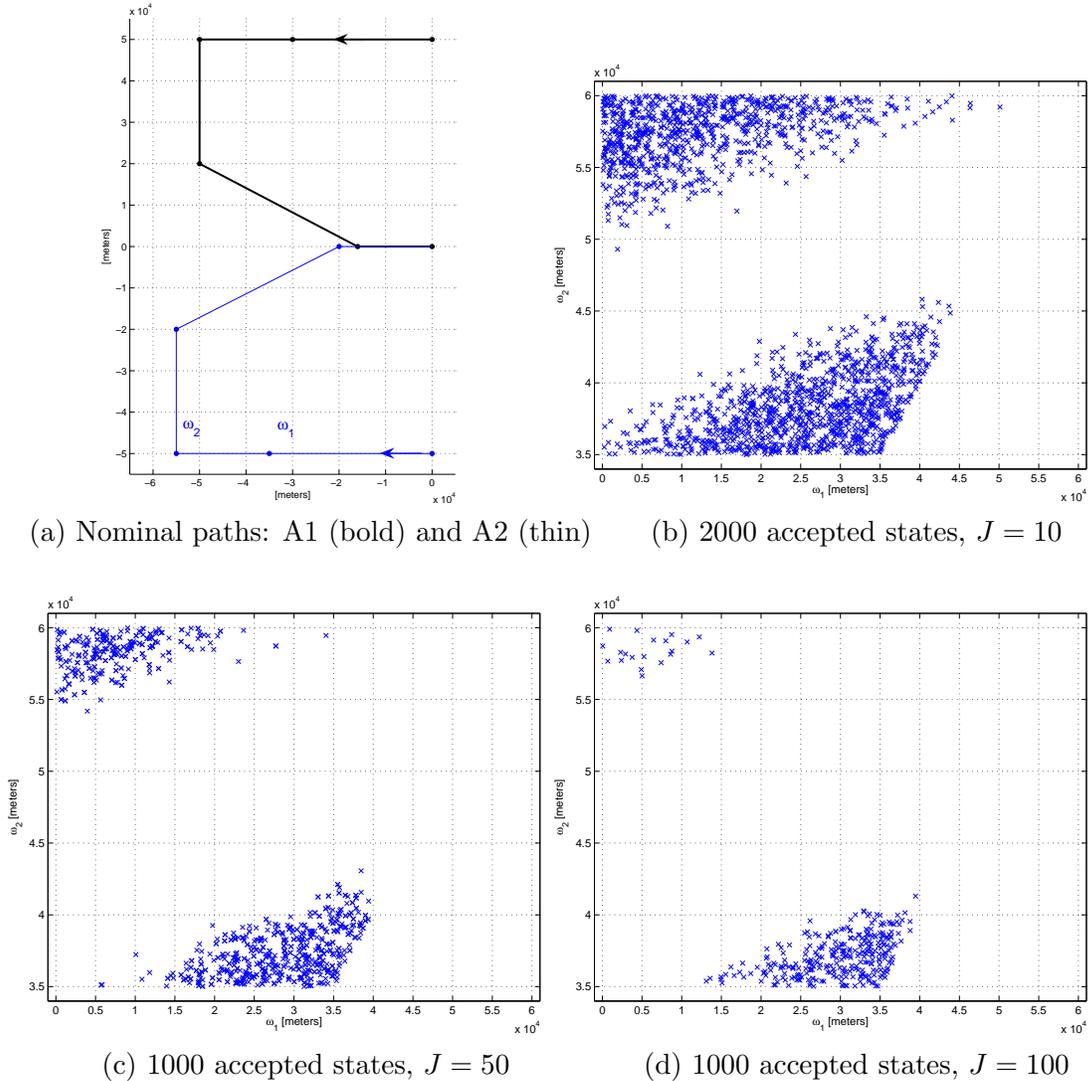


Figure 6: Accepted states during MCMC simulation

## 5 Conclusions

In this paper we illustrated an approach to air traffic conflict resolution in a stochastic setting based on 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 optimization of an expected value criterion with probabilistic constraints. Here, a penalty formulation of the problem was developed, which guarantees constraint satisfaction but delivers suboptimal solutions. A side effect of the optimization procedure is that structural differences between maneuvers are highlighted as “clouds” of maneuvers accepted by the algorithm, with the density of the clouds related to the cost of the particular maneuver.

We presented the application of this method to two realistic scenarios inspired by terminal area and final approach maneuvering respectively. The solutions proposed by

our algorithm were tested in Monte Carlo simulations and gave very good performance in terms of post-resolution conflicts. Computations times were high, partly because no attempt was made to optimize the implementation of the algorithms. Note also that the algorithm can provide a fairly accurate separation between “good” and “bad” maneuvers (i.e. maneuvers that meet, or violate the constraints) relatively cheaply, using a low  $J$ . It is only when we try to find an optimal maneuver among these that the computational load really kicks in.

Current research concentrates on overcoming the sub-optimality imposed by the penalty formulation of the constrained optimization problem considered in Section 2. A possible way is to use the Monte Carlo Markov Chain procedure presented in Section 3 to obtain optimization parameters that satisfy the constraints and then to optimize over this set in a successive step. We are also working on sequential Monte Carlo implementations of the optimization 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. It will also introduce feedback to the process, since it will make it possible to repeat the optimization on line in a receding horizon manner. Finally, we are continuing to work on modeling and implementation in the simulator of typical ATC situation with a realistic parameterization 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.

## A List of acronyms

ATC	Air Traffic Control/Controller
ATM	Air traffic Management
BADA	Base of Aircraft Data
FMS	Flight Management System
ft	Feet
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
nmi	Nautical miles
STAR	Standard Approach Route
TMA	Terminal Maneuvering Area