Monte Carlo Optimization for Conflict Resolution in Air Traffic Control

Andrea Lecchini Visintini, William Glover, John Lygeros, *Senior Member, IEEE*, and Jan Maciejowski, *Senior Member, IEEE*

***Abstract*—The safety of flights, and, in particular, separation assurance, is one of the main tasks of air traffic control (ATC). Conflict resolution refers to the process used by ATCs 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. In this paper, a framework for conflict resolution that allows one to take into account such levels of uncertainty using a stochastic simulator is presented. The conflict resolution task is posed as the problem of optimizing an expected value criterion. It is then shown how the cost criterion can be selected to ensure an upper bound on the probability of conflict for the optimal maneuver. Optimization of the expected value resolution criterion is carried out through an iterative procedure based on Markov chain Monte Carlo. Simulation examples inspired by current ATC practice in terminal maneuvering areas and approach sectors illustrate the proposed conflict resolution strategy.**

***Index Terms*—Air traffic control, conflict resolution, Monte Carlo methods, optimization, stochastic models.**

NOMENCLATURE

ATC	Air traffic control/controller.
ATM	Air traffic management.
BADA	Base of Aircraft Data.
FMS	Flight management system.
ETA	Expected time of arrival.
MC	Monte Carlo.
MCMC	Markov chain Monte Carlo.
STAR	Standard approach route.
TMA	Terminal maneuvering area.

I. 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 is ultimately responsible for safety. Before takeoff, aircraft file flight plans that 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 also tries to fulfill the (possibly conflicting) requests of aircraft and airlines (e.g., 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 has been devoted over the last decade on creating tools to assist ATC with conflict detection and resolution tasks. A review of research in this area of ATC is presented in [1].

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 [2] or a dynamical model, such as a stochastic differential equation (see, e.g., [3]–[5]) 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., [2]–[6]. In particular, 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 tradeoff exists between computational effort (simulation time in the case of MC methods) and accuracy of the model. Techniques to

Manuscript received June 23, 2005; revised February 12, 2006, March 28, 2006, and July 23, 2006. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/C014006/1, the European Commission under Project HYBRIDGE IST-2001-32460, the HYCON Network of Excellence under Contract FP6-IST-511368, and Eurocontrol under Project C20051E/BM/03. The Associate Editor for this paper was P. Kostek.

A. Lecchini Visintini is with the Department of Engineering, University of Leicester, Leicester LE1 7RH, U.K., and also with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: al394@eng.cam.ac.uk).

W. Glover and J. Maciejowski are with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: wg214@eng.cam.ac.uk; jmm@eng.cam.ac.uk).

J. Lygeros is with the Automatic Control Laboratory, ETH Zurich, CH-8092, Switzerland (e-mail: lygeros@control.ee.ethz.ch).

Digital Object Identifier 10.1109/TITS.2006.883108

accelerate MC methods, especially for rare event computations, are under development; see, e.g., [7].

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, e.g., [8]–[11]. In the stochastic setting, the research effort has concentrated mainly on conflict detection, and only a few simple resolution strategies have been proposed [2], [5]. 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 Markov chain MC (MCMC) framework [12] for conflict resolution in a stochastic setting. The aim of the proposed approach is to extend to conflict resolution the advantages of MC techniques in terms of flexibility and the complexity of the problems that can be tackled. The approach is motivated from Bayesian statistics [13]–[15]. We consider an expected value resolution criterion that takes into account separation and other factors (e.g., aircraft requests). Ideally, one would like to formulate the conflict resolution problem as a constrained optimization problem: Among the maneuvers that are safe with high enough probability, select the most efficient one. We show how such a constrained optimization problem can be approximated by an optimization problem with an expected value criterion so that the optimal maneuver for the latter ensures a high probability of constraint satisfaction. The MCMC optimization procedure of Mueller [13] is then 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 works [16]–[18].

This paper is organized in five sections. Section II 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 III. Section IV is devoted to the benchmark problems used to illustrate our approach. Section IV-A introduces the problems associated with ATC in terminal and approach sectors, and Section IV-B provides a brief overview of the simulator used to carry out the experiments. Sections IV-C and D present results on benchmark problems in terminal and approach sectors, respectively. Conclusions and future objectives are discussed in Section V.

II. Conflict Resolution With an Expected Value Criterion

A. Approximation of Constrained Optimization Problems

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 $\boldsymbol{\Omega}$. 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 ω is denoted by

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

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

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 ω; therefore, performance is measured by a function $\mathrm{perf}(\cdot,\cdot) : \boldsymbol{\Omega} \times \mathbf{X_f} \to [0,1]$. The expected performance for a given $\omega \in \boldsymbol{\Omega}$ is denoted by $\mathrm{PERF}(\omega)$, where

$$\mathrm{PERF}(\omega) = \int\limits_{x\in\mathbf{X_f}} \mathrm{perf}(\omega,x)p_\omega(x)dx.$$

Ideally, one would like to select ω 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

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

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

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

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

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 suboptimality in the resulting expected performance.

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

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

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

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

Instead of the constrained optimization problem (1)–(2), we solve the unconstrained optimization problem

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

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

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

$$\bar{\mathrm{P}}(\bar{\omega}) \leq \frac{1}{\Lambda} + \left(1 - \frac{1}{\Lambda}\right)\bar{\mathrm{P}}_{\min} \tag{4}$$

$$\mathrm{PERF}(\bar{\omega}) \geq \mathrm{PERF}_{\max|\bar{\mathbf{P}}} - (\Lambda - 1)(\bar{\mathbf{P}} - \bar{\mathrm{P}}_{\min}). \tag{5}$$

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

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

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

$$\begin{aligned}\mathrm{PERF}(\bar{\omega}) &+ \Lambda - (\Lambda - 1)\bar{\mathrm{P}}(\bar{\omega}) \\ &\geq \mathrm{PERF}(\omega) + \Lambda - (\Lambda - 1)\bar{\mathrm{P}}(\omega), \qquad \forall\omega \quad (6)\end{aligned}$$

which can be rewritten as

$$\bar{\mathrm{P}}(\bar{\omega}) \leq \frac{\mathrm{PERF}(\bar{\omega}) - \mathrm{PERF}(\omega)}{\Lambda - 1} + \bar{\mathrm{P}}(\omega), \qquad \forall\omega. \quad (7)$$

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

$$0 < \mathrm{PERF}(\omega) \leq \mathrm{P}(\omega). \quad (8)$$

Therefore, we can use (8) to obtain an upper bound on the right-hand side of (7) from which we obtain

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

Finally, 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 $\omega \in \mathbf{\Omega}$. In particular, we know from (7) that

$$\begin{aligned}\mathrm{PERF}(\bar{\omega}) \geq \mathrm{PERF}(\omega) - (\Lambda - 1)\left[\bar{\mathrm{P}}(\omega) - \bar{\mathrm{P}}(\bar{\omega})\right] \\ \forall\omega : \bar{\mathrm{P}}(\omega) \leq \bar{\mathbf{P}}.\end{aligned}$$

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

$$\begin{aligned}\mathrm{PERF}(\bar{\omega}) \geq \mathrm{PERF}(\omega) - (\Lambda - 1)\left[\bar{\mathbf{P}} - \bar{\mathrm{P}}_{\min}\right] \\ \forall\omega : \bar{\mathrm{P}}(\omega) \leq \bar{\mathbf{P}}.\end{aligned}$$

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

Proposition 2.1 suggests a method for choosing Λ to ensure that the solution $\bar{\omega}$ of the optimization 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

$$\Lambda \geq \frac{1}{\bar{\mathbf{P}}} \quad (9)$$

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

$$\Lambda = \frac{1 - \hat{\mathrm{P}}}{\bar{\mathbf{P}} - \hat{\mathrm{P}}} \quad \text{and} \quad \Lambda = \frac{1}{\bar{\mathbf{P}}}. \quad (10)$$

B. Relation to Conflict Resolution

We formulate conflict resolution as a constrained optimization problem. Roughly speaking, 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. 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 (e.g., the sequence of positions observed by ATC every 6 s, which is a typical time interval between two successive radar sweeps) during the resolution maneuver is 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 ω to maximize the expected value of some measure of performance associated with the execution of the resolution maneuver, while ensuring a small probability of conflict.

More precisely, consider a region of the airspace (e.g., an ATC sector) with N aircraft moving in it. The examples presented below are for $N = 2$, but there is no conceptual limitation to considering arbitrary N. Each aircraft has a flight plan, which is typically encoded as a sequence of points in $\mathbb{R}^3$ together with an expected time of arrival (ETA) at each point. If the flight plan of aircraft i consists of N_i such points, we have

$$\left\{O_i^j, T_i^j\right\}_{j=1}^{N_i}, \qquad O_i^j \in \mathbb{R}^3,\ T_i^j \in \mathbb{R}_+,\ i = 1, \ldots, N.$$

We assume that waypoint coordinates are given with respect to a fixed reference frame on the ground, with horizontal distances measured in meters and with altitudes measured in feet.

In many cases, some freedom is allowed in certain elements of the flight plan. This is often the case with the ETA. With current technology, most commercial aircraft use what is known as a three-dimensional (3-D) FMS; in other words, they try to reach certain points in the airspace without worrying about whether they get there at a specific time. With a 3-D FMS, aircraft generally try to track a given speed, which depends on altitude but not on how much they are ahead or behind their ETA. Similar freedom sometimes exists in the vertical direction. When cruising, aircraft are required to track their flight level very accurately. This is not always the case when climbing or descending, however. For example, when descending to land at an airport, aircraft may follow a descent profile that takes them over specific points on the ground but not necessarily at a specific altitude. As long as certain constraints are met (e.g., the aircraft gets low enough to intercept the localizer before final approach), the descent rate is to a large extent determined by the settings of the FMS, wind conditions, etc. Similar freedom is allowed when taking off: The aircraft are required to fly over specific points on the ground, but their climb rate is largely determined by the thrust settings of their FMS, local wind conditions, their weight, etc.

In the examples, we will use an $*$ ("wild card") to denote elements of the flight plan that allow freedom. For example, a way point of the form

$$O_i^j = \begin{bmatrix} 0 \\ 50\,000 \\ 10\,000 \end{bmatrix}, \quad T_i^j = *$$

will mean that aircraft i should go over the point (0, 50 000) on the ground at an altitude of 10 000 ft but has some freedom over exactly when it will do so. Likewise

$$O_i^j = \begin{bmatrix} 50\,000 \\ 50\,000 \\ * \end{bmatrix}, \quad T_i^j = *$$

will mean that the aircraft should go over the point (50 000, 50 000) on the ground but has some freedom over exactly how high it will be when it gets there and exactly when this will happen.

ATCs can change the flight plans to prevent conflicts and get aircraft to their destinations more efficiently. Some elements of the flight plan are fixed, e.g., the beginning of the runway at an airport, the entry and exit gateways to the terminal maneuvering area (TMA), etc. The remaining elements that the controller can use to affect the motion of the aircraft will be the decision variables ω in our optimization problem. The examples given below will each have two decision variables denoted by ω_1 and ω_2; again, there is no conceptual limitation to considering more decision variables. For example

$$O_i^j = \begin{bmatrix} \omega_1 \\ -50\,000 \\ * \end{bmatrix}$$

means that ATC can determine the first coordinate of the jth waypoint of aircraft i; the second coordinate of the way point is fixed to $-50\,000$ m, and the altitude is free. The decision variables will be constrained to take values in a given set $\boldsymbol{\Omega}$, so that the waypoints are restricted to lie in a given ATC sector.

Even if the flight plan is fixed, there is considerable uncertainty in the movement of aircraft. We therefore treat the trajectory of each aircraft as a random sequence of points

$$\{x_i(k)\}_{k=0}^{M_i}, \qquad i = 1, \ldots, N.$$

In the case studies presented below, these sequences will be generated by sampling every 6 s the trajectories of the continuous-time stochastic hybrid model presented in [16]. The set of these sequences for all aircraft will play the role of the random variable X of the previous section, i.e.,

$$X = \left\{ x_i(k) \in \mathbb{R}^3 \;\middle|\; \begin{matrix} i=1,\ldots,N \\ k=1,\ldots,M_i \end{matrix} \right\} \in \mathbb{R}^{\sum_{i=1}^{N} 3M_i} = \mathbf{X}.$$

Notice that the distribution of X can be very complicated. For example, because of the spatiotemporal correlation of the wind, $x_i(k)$ may be correlated not only for different times k but also for different aircraft i. Even the length M_i of the trajectories may be random. In the case studies, M_i will be a "stopping time" for the stochastic process, more specifically the exit time from the TMA, or the landing time. Because we will use MC methods to carry out the optimization, these complications have very little effect on our method; as long as a simulator is available to generate aircraft trajectories for a given flight plan, the method can be applied.

In ATC, safety is typically encoded in terms of conflicts, i.e., situations where two aircraft come too close to one another. The definition of "too close" depends on the region of the airspace, weather conditions, etc. Following [5], we define a conflict zone

$$\mathcal{C} = \left\{ (x,y,z) \in \mathbb{R}^3 \;\middle|\; \begin{matrix} \|(x,y)\| \le 5 \text{ nmi} \\ |z| \le 1000 \text{ ft} \end{matrix} \right\} \subseteq \mathbb{R}^3.$$

We then define the constraint set $\mathbf{X}_F$ for the random variable X as

$$\mathbf{X}_F = \left\{ X \in \mathbf{X} \;\middle|\; \begin{matrix} \forall i,j = 1,\ldots,N \text{ with } i \neq j \\ \forall k = 1,\ldots,\min\{M_i, M_j\} \\ x_i(k) - x_j(k) \notin \mathcal{C}. \end{matrix} \right\}.$$

Again, there is no conceptual limitation to considering more complex constraint sets. For example, an additional constraint will be imposed in the second case study to ensure that the aircraft successfully intercept the localizer at the end of the maneuver so that they can proceed to landing.

The final element of the optimization problem discussed above, i.e., the cost function "perf," will be used to introduce other nonsafety-related considerations. For example, in the first case study, the aim will be for the two aircraft to exit the TMA with adequate time separation to facilitate the work of the downstream ATC. This will be encoded as a cost

$$\text{perf}(\omega, x) = e^{-6a||M_1 - M_2| - 300|}$$

on the difference between the exit times M_1 and M_2 of the two aircraft; 300 is the desired time separation in seconds, a is a design parameter, and the factor 6 is the sampling period needed to convert the number of samples into seconds. In the second case study, the objective will be for aircraft 2 to land as quickly as possible to conserve fuel or meet its schedule. This will be encoded by a cost function

$$\text{perf}(\omega, x) = e^{-6aM_2}.$$

III. MC 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 $u(\omega, x)$ pointwise. This optimization procedure is in fact a general procedure for the optimization of expected value criteria, which was originally proposed in the Bayesian statistics literature [13], [15].

The optimization strategy relies on extractions of a random variable Ω whose distribution has modes that coincide with the optimal points of $U(\omega)$. These extractions are obtained by MCMC simulation [12]. 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 tradeoff between estimation accuracy of the optimizer and computational effort. In particular, the distribution of Ω is proportional to $U(\omega)^J$, where J is a positive integer that 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 [19]. A formal parallel between these two strategies has been derived in [14].

The MCMC optimization 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

$$h(\omega, x_1, x_2, \ldots, x_J) \propto \prod_j u(\omega, x_j) p_\omega(x_j) \tag{11}$$

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

$$h(\omega) \propto \left[\int u(\omega, x) p_\omega(x) dx\right]^J = U(\omega)^J. \tag{12}$$

This means that if we can extract realizations of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$, then the extracted Ω 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 $(\Omega, X_1, X_2, X_3, \ldots, X_J)$, with the desired joint probability density given by (11), can be obtained through MCMC simulation. The algorithm is presented in Table I. 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 $h(\omega)$.

This algorithm is a formulation of the Metropolis–Hastings algorithm [12] for a desired distribution given by $h(\omega, x_1, x_2, \ldots, x_J)$ and a proposal distribution given by

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

Under minimal assumptions, the Markov Chain generated by $\Omega(k)$ is uniformly ergodic with stationary distribution $h(\omega)$ given by (12). Therefore, after a "burn-in" period, the extractions $\Omega(k)$ accepted by the algorithm will concentrate around the modes of $h(\omega)$, which, by (12), coincide with the optimal points of $U(\omega)$. Results that characterize the convergence rate to the stationary distribution can be found, e.g., in [12].

TABLE I
MCMC ALGORITHM

```
initialization:
  Set $k = 0$
  Generate $\Omega_0 \sim g(\omega)$
  Generate $X_0^j \sim p_{\Omega_0}(x)\,, \quad j = 1, \ldots, J$
  Set $U_0 = \prod_{j=1}^J u(\Omega_0, X_0^j)$
repeat:
  Set $k = k + 1$
  Generate $\tilde{\Omega} \sim g(\omega)$
  Generate $\tilde{X}^j \sim p_{\tilde{\Omega}}(x)\,, \quad j = 1, \ldots, J$
  Set $\tilde{U} = \prod_{j=1}^J u(\tilde{\Omega}, \tilde{X}^j)$
  Set $\rho_k = \min\left\{\frac{\tilde{U}}{g(\tilde{\Omega})} \frac{g(\Omega_{k-1})}{U_{k-1}}, 1\right\}$
  Set $[\Omega_k\,, U_k] = \begin{cases} [\tilde{\Omega}\,, \tilde{U}] & \text{with probability } \rho_k \\ [\Omega_{k-1}\,, U_{k-1}] & \text{with probability } 1 - \rho_k \end{cases}$
until True
```

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 ρ; otherwise, it is rejected, and the previous state of the Markov chain $[\Omega(k), U_J(k)]$ is maintained. In practice, 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).

A general guideline for obtaining a faster convergence is to aim at concentrating the search distribution $g(\omega)$ in the regions where $U(\omega)$ assumes nearly optimal values. Of course, these regions are not known, at least initially, but it is demonstrated in Sections IV-C and D how one can refine the selection of $g(\cdot)$ during the optimization. The algorithm represents a tradeoff between computational effort and peakedness of the target distribution. This tradeoff is tuned by the parameter J, which is the exponent in 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 increases the number of simulations one needs to perform at each step. Obviously, if the peaks of $U(\omega)$ are already quite sharp, then J can be kept small, which 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 tradeoff exists between its peakedness and the parameter Λ, which is related to the 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 optimization of $U(\omega)$.

IV. ATC IN TERMINAL AND APPROACH SECTORS

A. Current Practice

The TMA and approach sectors are perhaps the most difficult areas for ATC. The management of traffic in this case

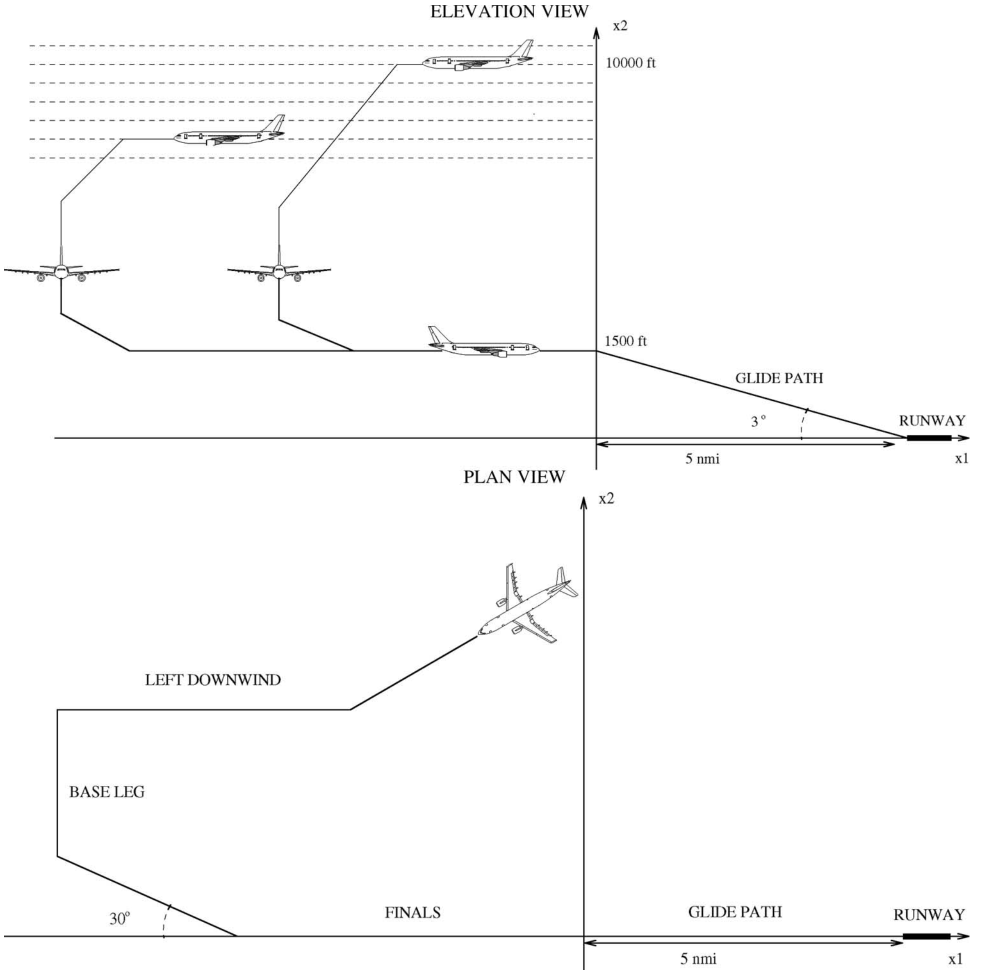


Fig. 1. Schematic representation of approach maneuver.

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 [20] and [21].

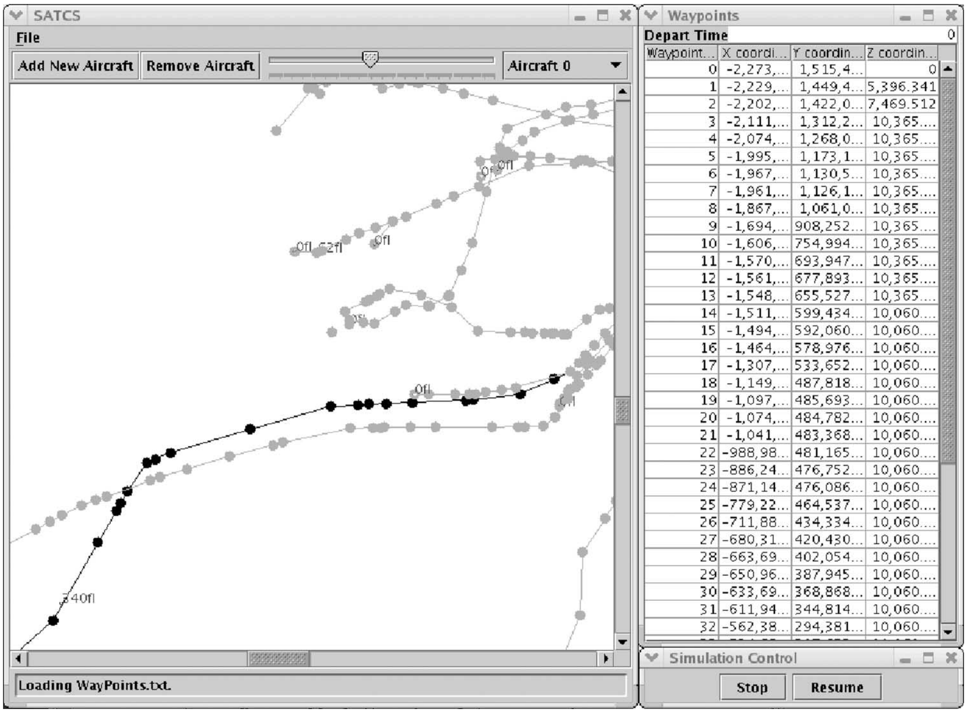
During most of the flight, aircraft stay at cruising altitudes (above 30 000 ft). In the current organization, the traffic at these altitudes has an en-route structure, which facilitates the action of ATC. Aircraft follow prespecified corridors at different "flight levels." Flight levels are given in three-digit numbers, representing hundreds of feet; for example, the altitude of 30 000 ft is denoted by FL300.

Toward 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 toward the runway for landing. The tasks of ATC in the approach sectors are discussed below.

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 5 nmi and height 2000 ft (i.e., aircraft that do not have 5 nmi of horizontal separation must have 1000 ft 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 that will guide them onto the runway. The point at which the aircraft will actually start the descent from cruising altitude (known as the "top of descent") is an important variable that 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 Toward the Runway: The ATCs must direct the aircraft toward the runway in a properly spaced queue. This is done by adjusting the waypoints (corners) of a standard approach route (STAR)—see Fig. 1. Typically, the



Depart Time 0

Waypoint...	X coordi...	Y coordin...	Z coordin...
0	-2,273,...	1,515,4...	0
1	-2,229,...	1,449,4...	5,396.341
2	-2,202,...	1,422,0...	7,469.512
3	-2,111,...	1,312,2...	10,365....
4	-2,074,...	1,268,0...	10,365....
5	-1,995,...	1,173,1...	10,365....
6	-1,967,...	1,130,5...	10,365....
7	-1,961,...	1,126,1...	10,365....
8	-1,867,...	1,061,0...	10,365....
9	-1,694,...	908,252...	10,365....
10	-1,606,...	754,994...	10,365....
11	-1,570,...	693,947...	10,365....
12	-1,561,...	677,893...	10,365....
13	-1,548,...	655,527...	10,365....
14	-1,511,...	599,434...	10,060....
15	-1,494,...	592,060...	10,060....
16	-1,464,...	578,976...	10,060....
17	-1,307,...	533,652...	10,060....
18	-1,149,...	487,818...	10,060....
19	-1,097,...	485,693...	10,060....
20	-1,074,...	484,782...	10,060....
21	-1,041,...	483,368...	10,060....
22	-988,98...	481,165...	10,060....
23	-886,24...	476,752...	10,060....
24	-871,14...	476,086...	10,060....
25	-779,22...	464,537...	10,060....
26	-711,88...	434,334...	10,060....
27	-680,31...	420,430...	10,060....
28	-663,69...	402,054...	10,060....
29	-650,96...	387,945...	10,060....
30	-633,69...	368,868...	10,060....
31	-611,94...	344,814...	10,060....
32	-562,38...	294,381...	10,060....

Loading WayPoints.txt.

Fig. 2. Screen shot of a simulation of all flights using Barcelona airport on a given day.

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 known as 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 localizer. This second segment is known as the "base leg." Aircraft perform an additional turn to intercept the plane of the localizer 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 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°–5° glide path to the runway.

This approach geometry (which is referred to as the "trombone maneuver") is advantageous to ATCs as it allows them great flexibility in spacing aircraft by adjusting the length of the downwind leg (hence the name trombone).

B. Simulation Procedure

A necessary step in the implementation of the MCMC algorithm is the extraction of $\tilde{X}_j$ for a given value of ω. 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) is necessary in the algorithm implementation. In general, the trajectories will be 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 [16]–[18]. The simulator implements realistic models of current commercial aircraft described in the Base of Aircraft Data (BADA) [22]. The simulator also contains realistic stochastic models of the wind disturbance [23]. The aircraft models contain continuous dynamics, arising from the physical motion of the aircraft, discrete dynamics, arising from the logic embedded in the FMS, and stochastic dynamics, arising from the effect of the wind and incomplete knowledge of physical parameters (e.g., 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 waypoints, 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 online (move waypoints, 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 (Fig. 2) and/or stored in files for postprocessing. The reader is referred to [17] and [18] 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 MC optimization procedure and to obtain MC estimates of postresolution conflict probabilities. The simulator was invoked from Matlab on a Linux workstation with a 3-GHz Pentium 4 processor. Under these conditions, the simulation of the flight of two aircraft

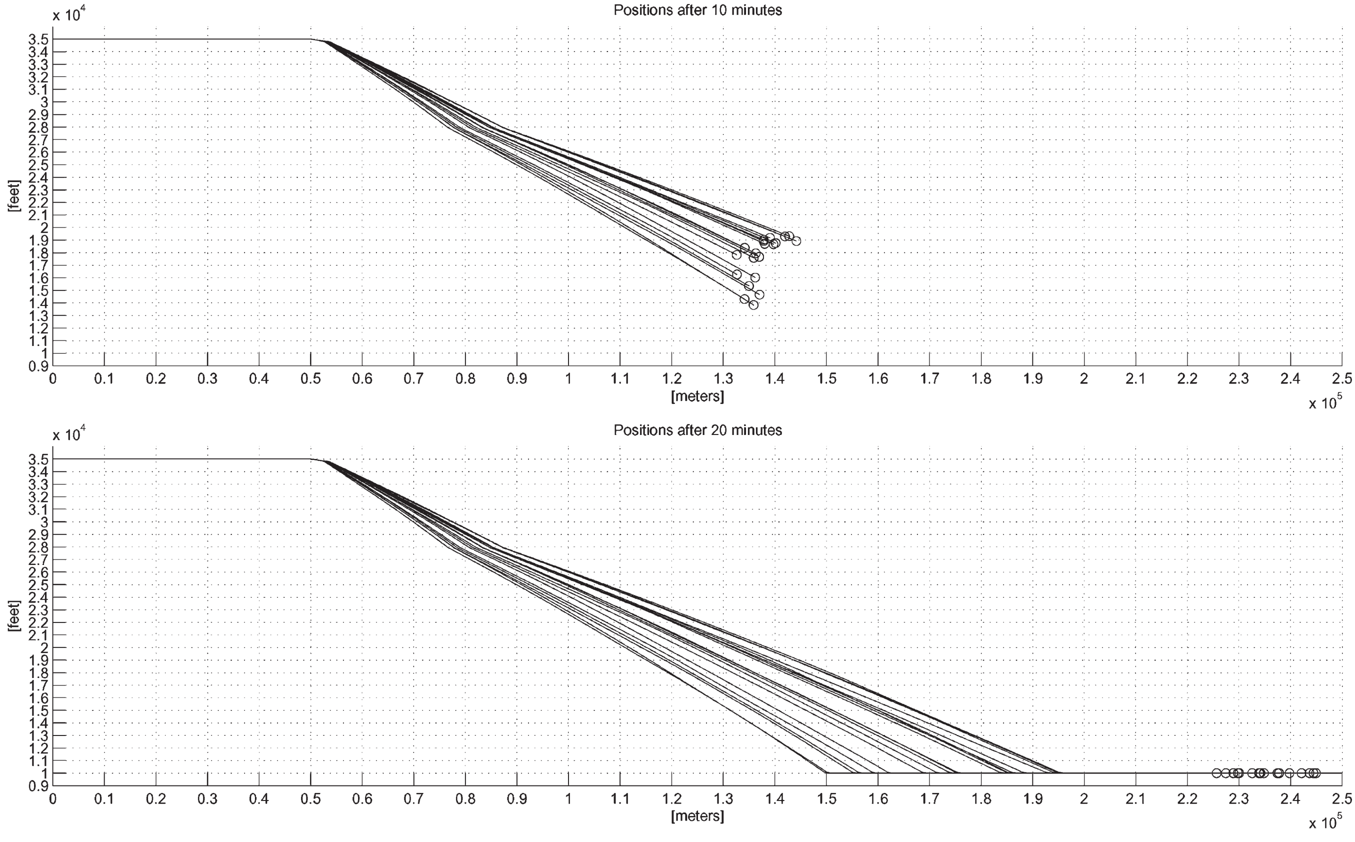


Fig. 3. Several trajectory realizations of aircraft descent.

for 30 min, which is approximately the horizon considered in the examples, took 0.2 s on average. This simulation speed (5 simulations/s) is quite low for an MC 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. Keep this fact in mind when evaluating the computation times reported in the following examples.

C. Sequencing Aircraft in TMAs

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 the approach sector. Fig. 3 shows several possible trajectory realizations of a descending aircraft, corresponding to the same nominal path. 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 the aircraft on arrival at FL100 is of the order of several kilometers.

We consider the problem of sequencing two descending aircraft as illustrated in Fig. 4(a). The initial position of the first aircraft (A1) is (−100 000, 100 000) (where coordinates are expressed in meters) and FL350. The path of this aircraft is fixed: The aircraft proceeds to waypoint (−90 000, 90 000), where it will start a descent to FL150. The trajectory of A1 while descending is determined by an intermediate waypoint at (0, 0) and a final waypoint at (100 000, 0), where the aircraft is assumed to exit the TMA and enter the approach sector. In other words, the flight plan of A1 is

$$O_1^1 = \begin{bmatrix} -100\,000 \\ 100\,000 \\ 35\,000 \end{bmatrix}, \quad O_1^2 = \begin{bmatrix} -90\,000 \\ 90\,000 \\ 35\,000 \end{bmatrix}$$

$$O_1^3 = \begin{bmatrix} 0 \\ 0 \\ * \end{bmatrix}, \quad O_1^4 = \begin{bmatrix} 100\,000 \\ 0 \\ 15\,000 \end{bmatrix}.$$

The times are all free except $T_1^1 = 0$, which is used to set the beginning of the scenario.

The second aircraft (A2) is initially at (−100 000, −100 000) and FL350. This aircraft proceeds to waypoint (−90 000, −90 000), where it will start its descent to FL150. The intermediate waypoint $\omega = (\omega_1, \omega_2)$ must be selected in the range ω_1, $\omega_2 \in [-90\,000, 90\,000]$. The aircraft will then proceed to waypoint (90 000, 0) and then to the exit waypoint (100 000, 0). In other words, the flight plan of A2 is

$$O_2^1 = \begin{bmatrix} -100\,000 \\ -100\,000 \\ 35\,000 \end{bmatrix}, \quad O_2^2 = \begin{bmatrix} -90\,000 \\ -90\,000 \\ 35\,000 \end{bmatrix}, \quad O_2^3 = \begin{bmatrix} \omega_1 \\ \omega_2 \\ * \end{bmatrix}$$

$$O_2^4 = \begin{bmatrix} 90\,000 \\ 0 \\ 15\,000 \end{bmatrix}, \quad O_2^5 = \begin{bmatrix} 100\,000 \\ 0 \\ 15\,000 \end{bmatrix}.$$

The times are again free except $T_2^1 = 0$, i.e., A2 starts its trajectory simultaneously with A1. The decision variables take values in the set $\mathbf{\Omega} = [-90\,000, 90\,000]^2$.

We assume that the objective is to obtain a time separation of 300 s between the arrivals of the two aircraft at the exit waypoint (10 000, 0). Performance in this sense is measured by the function

$$\text{perf} = e^{-a\cdot||T_1-T_2|-300|}$$

where T_1 and T_2 are the arrival times of A1 and A2 at the exit waypoint 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 5 nmi of horizontal separation and less than 1000 ft of vertical separation.[1] We optimize initially with an upper bound on the probability of constraint violation of $\bar{\mathbf{P}} = 0.1$. 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 (9), we select $\Lambda = 10$ in the optimization criterion.

The results of the optimization procedure are illustrated in Fig. 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.

Fig. 4(b) illustrates the case $J = 10$. In this case, the proposal distribution g was uniform over the parameter space, and the ratio of accepted/proposed states was 0.27. The algorithm was executed until 5100 states had been accepted; Fig. 4(b) shows the last 2000 accepted states. The figure clearly indicates regions characterized by a low density of accepted parameters. These are parameters that 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). Notice that even for this relatively small value of J, the algorithm allows one to separate safe from unsafe maneuvers and distinguish discrete decisions that the controller will have to make (in this case, whether to allow A1 or A2 to exit the TMA first). Moreover, because the density of the clouds is proportional to the cost function raised to the power J, the figure also suggests that the optimal maneuver is likely to be found among the ones in the top left cloud.

The properties of the algorithm discussed in Section III suggest that the last fact should become clearer as we increase the number of simulations J. Fig. 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 the 2000 parameters accepted in the MCMC simulation for $J = 10$ that are shown in Fig. 4(b). The choice of this proposal distribution gives clear computational advantages since less computation time is spent searching over regions of nonoptimal parameters; it still, however, ensures coverage of the parameter space as required for convergence of the algorithm. The figure shows 1000 accepted states; an additional 100 states accepted initially were discarded as a burn-in period. In this case, the ratio of accepted to proposed states was 0.34. It is clear from the figure that the middle cloud (where A2 exits the TMA before A1) has effectively vanished; this indicates that the optimal maneuver most likely requires A1 to exit before A2. In addition, the fact that the top left cloud concentrates most of the accepted states suggests that the optimal maneuver should be sought in this region and not in the bottom right cloud.

This fact becomes even clearer in Fig. 4(d). The figure shows the case $J = 100$ with a proposal distribution constructed as before, by superimposing Gaussian distributions $N(\mu, \sigma^2 I)$ with $\sigma^2 = 10^7$ m^2 on the states accepted for $J = 50$. In this case, the ratio of accepted/proposed states was 0.3; the figure again shows the 1000 states accepted after a 100-state burn-in period. It is clear from the figure that the bottom right cloud has effectively vanished. Moreover, at this relatively high value of J, the maneuvers in the top left cloud are all likely to give very similar performance.

To test this performance, we selected one maneuver from Fig. 4(d) from the area where the top left cloud is the densest; in this case, we selected $\omega_1 = -40\,000$ and $\omega_2 = 40\,000$. We then used this maneuver as a flight plan in 1000 MC runs of the simulator. None of these runs exhibited a conflict; in other words, the probability of conflict for this maneuver estimated by 1000 MC runs was zero. Moreover, the expected time separation between the exit times of the two aircraft estimated by these 1000 simulations was 283 s, which is very close to the desired 300 s. We note that the flight plan was fixed at the beginning of the simulation to the selected maneuver and was not changed subsequently. The performance is likely to be even better if our algorithm is used in a receding horizon manner, repeating the optimization periodically and changing the flight plan as the trajectories of the aircraft evolve and measurements of their positions are taken.

To assess the computational cost of the algorithm, we also provide the time needed to obtain 1100 accepted states; this ensures a reasonable sample of 1000 states, assuming the first 100 accepted states discarded as a burn-in period. For the case $J = 10$, where the ratio of accepted to proposed states was 0.27, approximately $1100 \times 10/0.27 = 40\,740$ simulations were needed to obtain 1000 accepted states. At the average simulation speed of 5 simulations/s, the required computation time to obtain 1000 accepted states was approximately 2 h. In the case $J = 50$, approximately $1100 \times 50/0.34 = 161\,764$ simulations were needed to obtain 1000 accepted states; these simulations required approximately 9 h of computation time. Finally, in the case $J = 100$, approximately $1100 \times 100/0.3 = 366\,666$ simulations were needed to obtain 1000 accepted states, and the required computation time was approximately 20 h.

D. Coordination of Approach Maneuvers

In this section, we optimize an approach maneuver with coordination between two aircraft. In Fig. 5, several trajectory realizations of an aircraft performing the approach maneuver described in this section are displayed. Here, the aircraft is initially at FL100 and descends to FL15 during the approach. Uncertainty in the trajectory is due to the action of the wind and randomness in the aircraft mass, as before. In the final leg,

[1] In the TMA of large airports, horizontal minimal separation is sometimes reduced to 3 nmi, but this fact is ignored here.

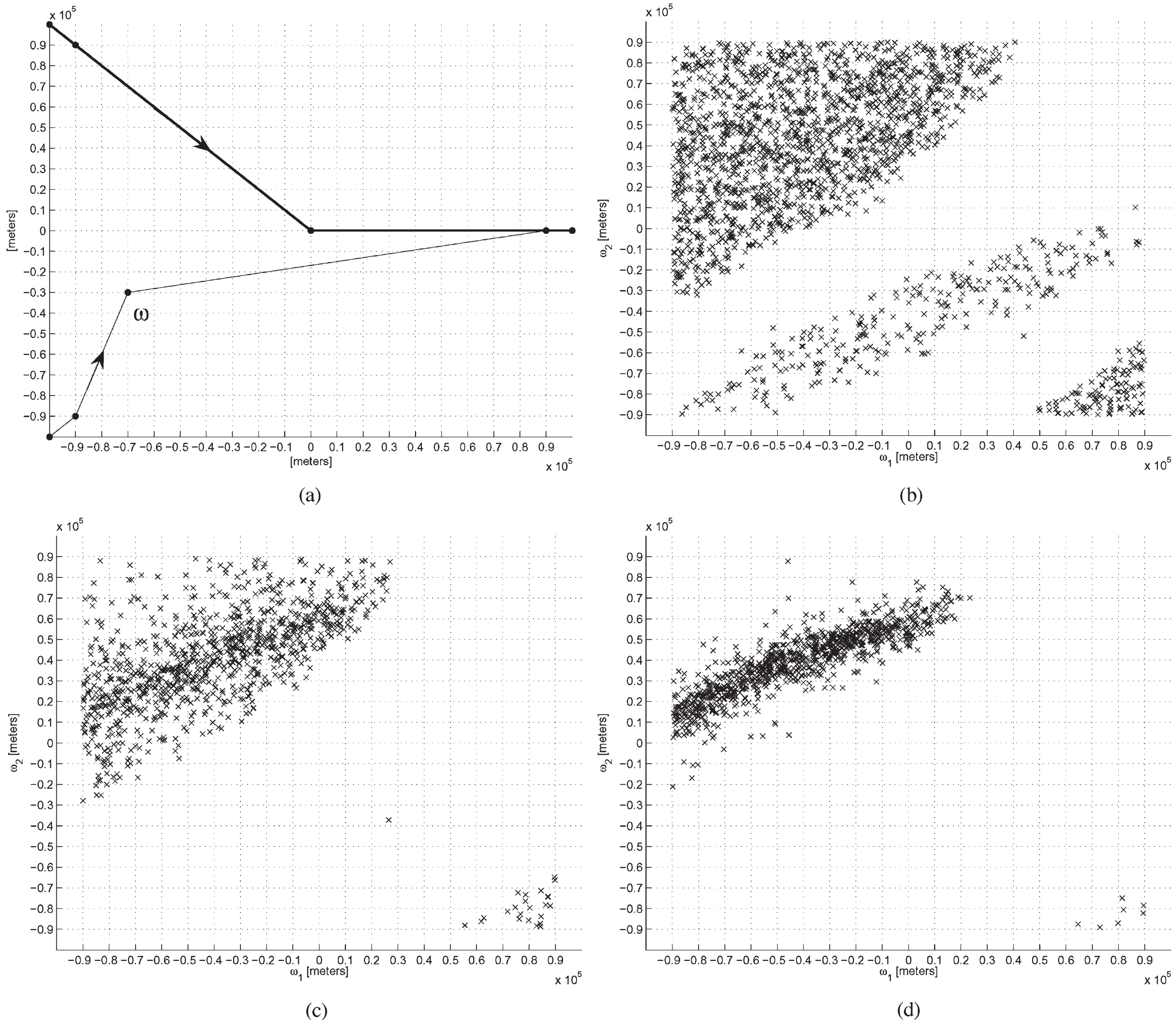


Fig. 4. Accepted states during MCMC simulation. (a) Nominal paths: A1 (bold) and A2 (thin). (b) 2000 accepted states, $J = 10$. (c) 1000 accepted states, $J = 50$. (d) 1000 accepted states, $J = 100$.

a function that emulates the localizer, and eliminates cross-track error, is implemented.

The problem formulation is illustrated in Fig. 6(a). In the interest of space, we will not state the values of O_i^j and T_i^j explicitly; determining these values from the discussion should be straightforward. We consider A1 and A2 approaching the runway. The glide path toward the runway starts at the origin of the reference frame, and the 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, 50 000) and FL100. The approach maneuver of this aircraft is fixed to $\bar{\omega}_1 = 30\,000$ and $\bar{\omega}_2 = 50\,000$. The initial position of A2 is (0, 50 000) and altitude FL100. The parameters of its approach maneuver will be selected using the optimization algorithm. The range of the optimization parameters is $\omega_2 \in [35\,000, 60\,000]$ 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 trajectories of the two aircraft are not in conflict. In addition, A2 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 the probability of constraint violation given by $\bar{\mathbf{P}} = 0.1$. 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 (9).

The results of the optimization procedure are illustrated in Fig. 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. Fig. 6(b) illustrates the case $J = 10$ and proposal distribution g

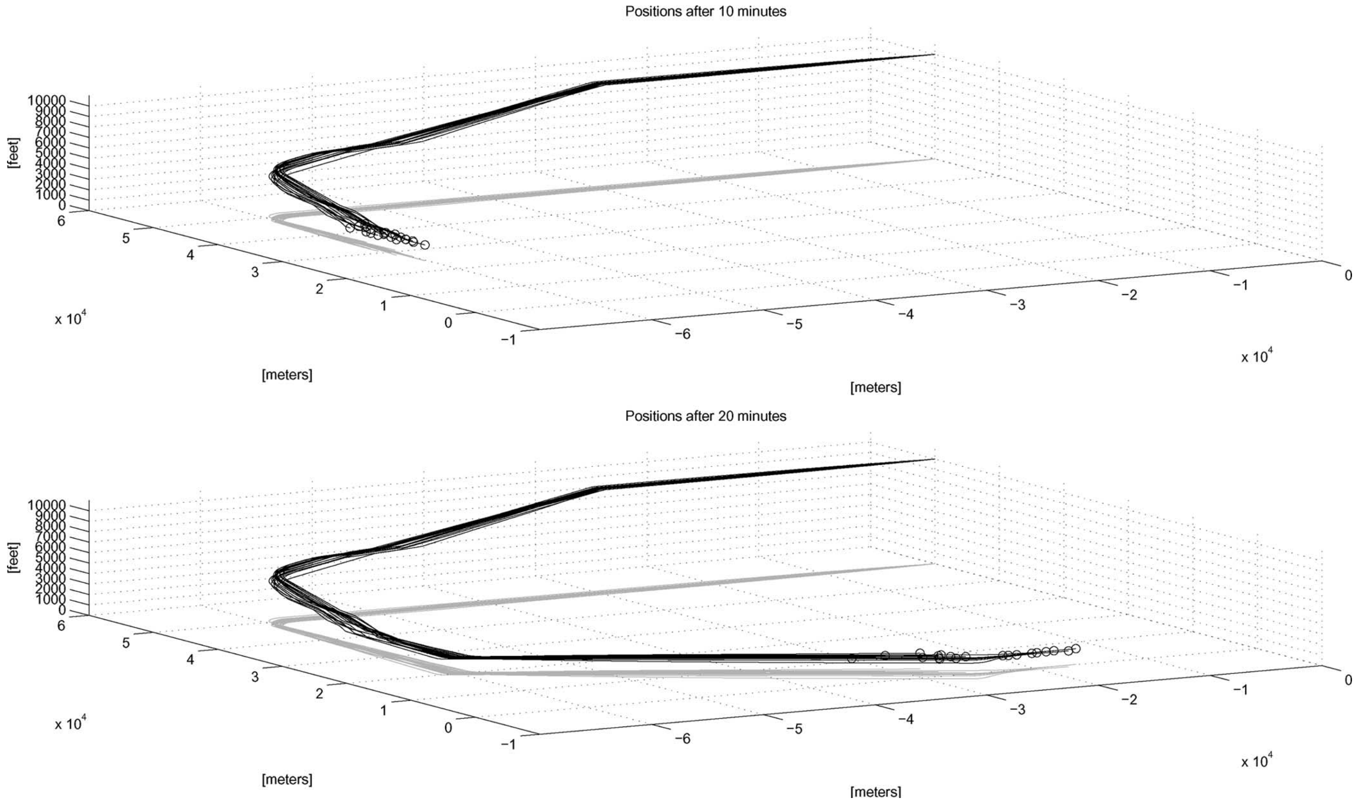


Fig. 5. Several trajectory realizations of an approach maneuver.

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 $1100 \times 10/0.23 = 47\,826$ simulations were needed to obtain 1000 accepted states, which took approximately 2.6 h. A region with no accepted parameters can be clearly seen in the figure. These are parameters that 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 ATC has to make: Either land A2 before A1 (bottom right cloud) or land A1 before A2 (top left cloud). At this stage, however, it is difficult to tell which of the two clouds is denser and, therefore, which discrete decision is more likely to lead to a better maneuver.

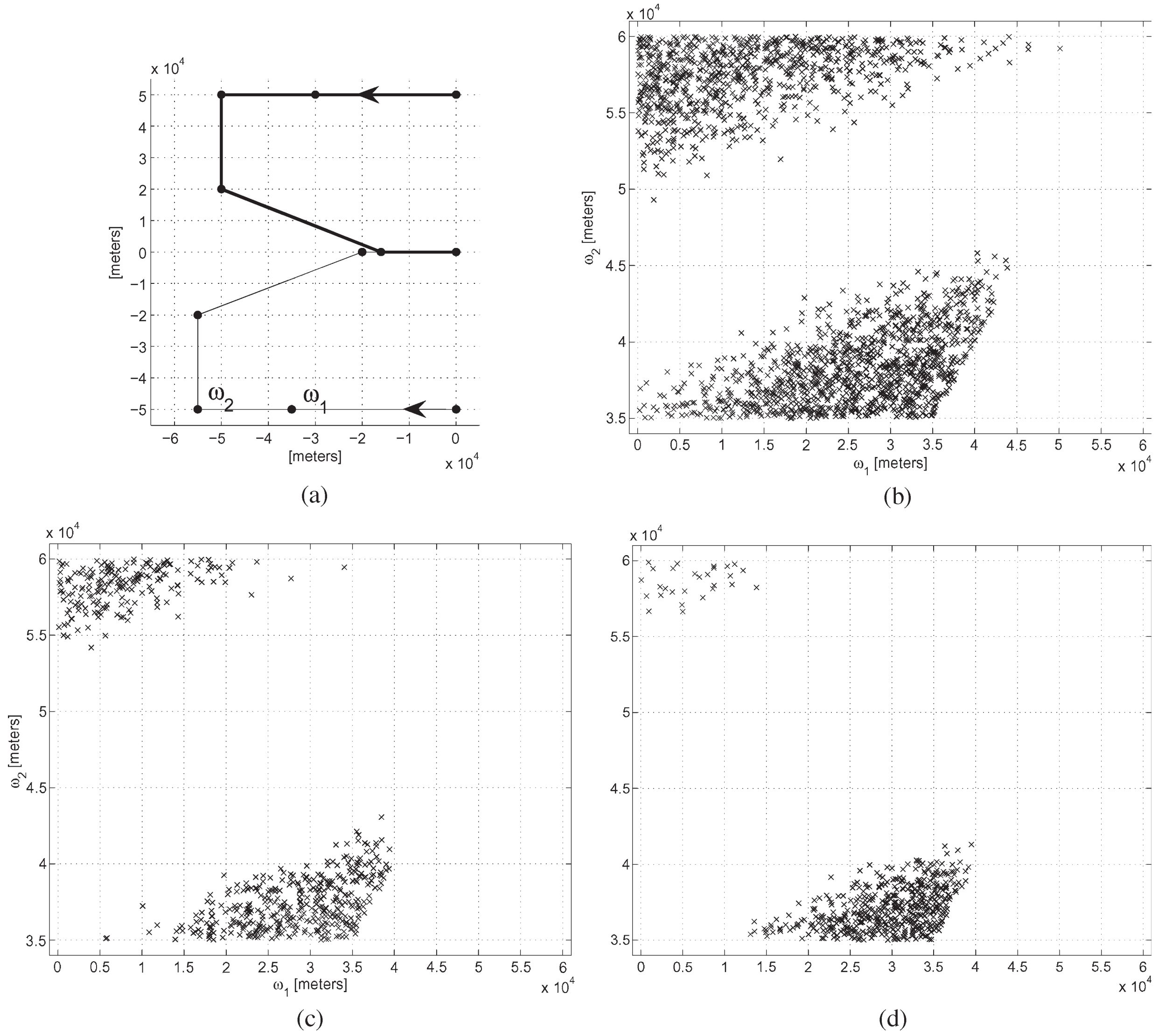
Fig. 6(c) illustrates the case $J = 50$. In this case, the proposal distribution g was a sum of 100 Gaussian distributions $N(\mu, \sigma^2 I)$ with means selected randomly among the maneuvers accepted for $J = 10$ and variance $\sigma^2 = 10^5$ m^2. In this case, the ratio between accepted and proposed parameters was 0.25. This means that approximately $1100 \times 50/0.25 = 220\,000$ simulations were needed to obtain 1000 accepted states, which required approximately 12 h of computation. The clouds are still quite dispersed, so selecting a near optimal maneuver is still difficult.

Fig. 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 $1100 \times 100/0.3 = 366\,666$ simulations were needed to obtain 1000 accepted states (approximately 20 h). Notice that the top left cloud has almost vanished in the figure. Moreover, with this relatively high level of J, all accepted maneuvers are likely to lead to similar costs. Fig. 6(d) indicates that a nearly optimal maneuver is $\omega_1 = 35\,000$ and $\omega_2 = 35\,000$. Once again, the probability of conflict for this maneuver, estimated by 1000 MC runs, was zero.

V. Conclusion

We have introduced an approach to air traffic conflict resolution in a stochastic setting based on MC methods. The main motivation for our approach is to enable the use of realistic stochastic hybrid models of aircraft flight to carry out trajectory prediction for the task of conflict resolution. Because of the complexity of the models, it is impossible to derive closed-form expressions for the probability distributions of aircraft positions. MC methods appear to be the only ones that allow the use of such complex 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 MC simulations and gave very good performance in terms of postresolution conflicts. Computation times were high, partly because no attempt was made to optimize the implementation of the algorithms. Note also that the algorithm can



(a) (b) (c) (d)

Fig. 6. Accepted states during MCMC simulation. (a) Nominal paths: A1 (bold) and A2 (thin). (b) 2000 accepted states, $J = 10$. (c) 1000 accepted states, $J = 50$. (d) 1000 accepted states, $J = 100$.

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 suboptimality imposed by the penalty formulation of the constrained optimization problem considered in Section II. A possible way is to use the MCMC procedure presented in Section III to obtain optimization parameters that satisfy the constraints and then to optimize over this set in a successive step. Although this seems conceptually appealing, its computational implications are unclear since a large number of extractions may be necessary to form a good impression of the set of maneuvers that meet the safety constraints. We are also working on sequential MC implementations of the optimization algorithm [24]. This will allow considerable computational savings since it will enable the reuse 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 online in a receding horizon manner. Finally, we are continuing to work on modeling and implementation in the simulator of typical ATC situations, with a realistic parameterization of control actions and control objectives, and on the possibility of integration with other existing tools (see, e.g., [8]).

References

[1] J. Kuchar and L. Yang, "A review of conflict detection and resolution methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 179–189, Dec. 2000.

[2] R. Paielli and H. Erzberger. (1997). Conflict probability estimation for free flight. *J. Guid. Control Dyn.* 20(3), pp. 588–596. [Online]. Available: http://www.ctas.arc.nasa.gov/publications/papers/

[3] H. Blom and G. Bakker, "Conflict probability and incrossing probability in air traffic management," in *Proc. IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2002, pp. 2421–2426.

[4] J. Hu, M. Prandini, and S. Sastry, "Aircraft conflict detection in presence of spatially correlated wind perturbations," in *Proc. AIAA Guid., Navigat., and Control Conf.*, Austin, TX, 2003. CD-ROM.

[5] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 199–220, Dec. 2000.

[6] R. Irvine, "A geometrical approach to conflict probability estimation," in *Proc. 4th USA/Eur. Air Traffic Manage. R&D Semin.*, Budapest, Hungary, 2001. CD-ROM.

[7] J. Krystul and H. Blom. (2004). Monte Carlo simulation of rare events in hybrid systems. HYBRIDGE, Amsterdam, The Netherlands, Tech. Rep. WP8. Deliverable D8.3. [Online]. Available: http://www.nlr.nl/public/hosted-sites/hybridge/

[8] T. Becher, D. Barker, and A. Smith. (2005, Feb.). Near-to-mid term solution for efficient merging of aircraft on uncoordinated terminal RNAV routes. MITRE Corp., Cambridge, MA, Tech. Rep. [Online]. Available: http://www.mitre.org/work/tech_papers_05/05_0123/

[9] E. Frazzoli, Z. Mao, J. Oh, and E. Feron, "Aircraft conflict resolution via semi-definite programming," *AIAA J. Guid. Control Dyn.*, vol. 24, no. 1, pp. 79–86, 2001.

[10] J. Hu, M. Prandini, and S. Sastry, "Optimal coordinated maneuvers for three-dimensional aircraft conflict resolution," *AIAA J. Guid. Control Dyn.*, vol. 25, no. 5, pp. 888–900, 2002.

[11] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 509–521, Apr. 1998.

[12] C. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[13] P. Mueller, "Simulation based optimal design," in *Bayesian Statistics 6*, J. O. Berger, J. M. Bernardo, A. P. David, and A. F. M. Smith, Eds., London, U.K.: Oxford Univ. Press, 1999, pp. 459-474.

[14] P. Müller, B. Sansó, and M. De Iorio, "Optical bayesian design by inhomogeneous Markov chain simulation," *J. Amer. Stat. Assoc.*, vol. 99, no. 467, pp. 788–798, 2004.

[15] A. Doucet, S. Godsill, and P. Robert, "Marginal maximum *a posteriori* estimation using Markov chain Monte Carlo," *Stat. Comput.*, vol. 12, no. 1, pp. 77–84, Jan. 2002.

[16] W. Glover and J. Lygeros, "A stochastic hybrid model for air traffic control simulation," in *Hybrid Systems: Computation and Control*, ser. LNCS, R. Alur and G. Pappas, Eds., New York: Springer-Verlag, 2004, pp. 372-386, no. 2993.

[17] ——. (2004, Feb. 18). A multi-aircraft model for conflict detection and resolution algorithm evaluation. HYBRIDGE, Amsterdam, The Netherlands, Tech. Rep. WP1. Deliverable D1.3. [Online]. Available: http://www.nlr.nl/public/hosted-sites/hybridge/

[18] ——. (2004, Jun. 17). Simplified models for conflict detection and resoution algorithms. HYBRIDGE, Amsterdam, The Netherlands, Tech. Rep. WP1. Deliverable D1.4. [Online]. Available: http://www.nlr.nl/public/hosted-sites/hybridge/

[19] P. van Laarhoven and E. Aarts, *Simulated Annealing: Theory and Applications*. Amsterdam, The Netherlands: Reidel, 1987.

[20] EUROCONTROL Experimental Centre. (2004, Apr. 27). Analysis of the current situation in the Paris London and Frankfurt TMA. FALBALA, EUROCONTROL Experimental Centre, Bretigny sur Orge, France, Tech. Rep. CARE/ASAS/SOF/04-057 Work Package 1: Final Project Report. [Online]. Available: http://www.eurocontrol.int/care-asas/public/standard_page/activities_4_falbala_diss.html

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

[22] ——. (2002). EUROCONTROL Experimental Centre, *User Manual for the Base of Aircraft Data (BADA)—Revision 3.3.* [Online]. Available: http://www.eurocontrol.fr/projects/bada/

[23] R. Cole, C. Richard, S. Kim, and D. Bailey, "An assessment of the 60 km rapid update cycle (RUC) with near real-time aircraft reports," Lincoln Laboratory, Mass. Inst. Technol., Lexington, MA, Tech. Rep. NASA/A-1, Jul. 15, 1998.

[24] *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. De Freitas, and N. Gordon, Eds., New York: Springer-Verlag, 2001.

Andrea Lecchini Visintini received the Laurea degree in computer engineering from the University of Pavia, Pavia, Italy, in 1997 and the Ph.D. degree in computer engineering from the University of Brescia, Brescia, Italy, in 2001.

From April 2001 to June 2003, he was a Postdoctoral Researcher with the Center for System Engineering and Applied Mechanics, Université-Catholique de Louvain, Louvain, Belgium. In November 2002, he was a Visiting Fellow with the Research School of Information Sciences and Engineering, Australian National University, Canberra, Australia. From July 2003 to June 2006, he was a Research Associate with the Control Group, University of Cambridge, Cambridge, U.K. Since July 2006, he has been a Lecturer of engineering with the University of Leicester, Leicester, U.K. His current research interests are in system identification and data-based control system design; optimization and predictive control for complex stochastic systems, with application to air traffic control; stochastic optimization; and Monte Carlo methods.

William Glover was born in Cambridge, U.K., in 1980. He received the M.Math degree in mathematics from the University of Warwick, Warwick, U.K., in 2002. He is currently pursuing the Ph.D. degree with the Institute for Manufacturing, Department of Engineering, University of Cambridge.

From 2002 to 2004, he was a Research Assistant with the Department of Engineering, University of Cambridge, where he worked on hybrid systems and air traffic control.

John Lygeros (SM'06) received the B.Eng. degree in electrical engineering and the M.Sc. degree in control from Imperial College, London, U.K., in 1990 and 1991, respectively, and the Ph.D. degree from the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, in 1996.

He has held a series of postdoctoral research appointments with the National Automated Highway Systems Consortium, Massachusetts Institute of Technology, and the University of California, Berkeley. He was also a part-time Research Engineer with SRI International, Menlo Park, CA, and a Visiting Professor with the Department of Mathematics, Universite de Bretagne Occidentale, Brest, France. Between July 2000 and March 2003, he was a University Lecturer with the Department of Engineering, University of Cambridge, Cambridge, U.K. Between March 2003 and July 2006, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Patras, Patras, Greece. Since July 2006, he has been an Associate Professor with the Automatic Control Laboratory, ETH Zurich, Switzerland. His research interests include modeling, analysis, and control of hierarchical hybrid systems with applications to biochemical networks and large-scale systems such as automated highways and air traffic management.

Jan Maciejowski (SM'06) received the B.Sc. degree (first class) in automatic control from Sussex University, Sussex, U.K., in 1971 and the Ph.D degree in control engineering from University of Cambridge, Cambridge, U.K., in 1978.

From 1971 to 1974, he was a Systems Engineer with Marconi Space and Defence Systems, Ltd., where he worked mostly on attitude control of spacecraft and high-altitude balloon platforms. He has held Visiting Professorships with the University of California, Santa Barbara; Delft Technological University, Delft, Netherlands; and Nanyang Technological University, Singapore. He has also consulted for several U.K. and U.S. companies on various control and signal processing problems. He is currently a Professor in control engineering with the Department of Engineering, University of Cambridge, and the Head of the Control Group. He is also a Fellow of Pembroke College, Cambridge. He has published about 100 journal and conference papers on control systems. He is the author of *Predictive Control with Constraints* (Prentice-Hall, 2001), which has recently been translated into Japanese. His current research interests are in predictive control, its application to fault-tolerant control, in hybrid systems, and in system identification.

Dr. Maciejowski is a Fellow of the Institution of Electrical Engineers and the Institute of Measurement and Control (InstMC). He is also a Distinguished Lecturer of the IEEE Control Systems Society. He was the President of the European Union Control Association from 2003 to 2005 and the InstMC in 2002. His book *Multivariable Feedback Design* (Addison-Wesley, 1989) received the IFAC Control Engineering Textbook Prize in 1996.