Monte Carlo Optimization Strategies for Air-Traffic Control

A. Lecchini[a, *], W. Glover[a, †], J. Lygeros[b, ‡], and J. Maciejowski[a, §]

[a]*Department of Engineering, University of Cambridge, UK*

[b]*Department of Electrical and Computer Engineering, University of Patras, GR*

The safety of the flights, and in particular conflict resolution for separation assurance, is one of the main tasks of Air Traffic Control. Conflict resolution requires decision making in the face of the considerable levels of uncertainty inherent in the motion of aircraft. We present a Monte Carlo framework for conflict resolution which allows one to take into account such levels of uncertainty through the use of a stochastic simulator. A simulation example inspired by current air traffic control practice illustrates the proposed conflict resolution strategy.

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 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 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[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[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[2–6] . 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[7] .

*Research Associate, E-mail: `al394@cam.ac.uk`

†Research Associate, E-mail: `wg214@cam.ac.uk`

‡Assistant Professor, E-mail: `lygeros@ee.upatras.gr`

§Reader, E-mail: `jmm@eng.cam.ac.uk`

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[8–10] . 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 Monte Carlo Markov Chain (MCMC) framework[11] 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[12,13] . 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[12] 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[14–16] . The reader is referred also to the extended version of the present paper[17] .

This paper is organized as follows. Section II presents the formulation of conflict resolution as an optimization problem. The Monte Carlo optimization procedure that we adopt is presented in Section III. Section IV illustrate a simulation example. Conclusions and future objectives are discussed in Section V.

II. 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 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 (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 ω 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 ω. The distribution of X is denoted by $p_\omega(x)$ with $x \in \mathbf{X}$. The set of all possible values of ω is denoted by $\mathbf{\Omega}$. We assume that a constraint on the random variable X is given in terms of a feasible set $\mathbf{X_f} \subseteq \mathbf{X}$. We say that a 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 $\mathrm{P}(\omega)$

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

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

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 $\mathrm{perf}(\cdot,\cdot) : \mathbf{\Omega} \times \mathbf{X_f} \to [0,1]$. The expected performance for a given $\omega \in \mathbf{\Omega}$ is denoted by $\textsc{Perf}(\omega)$, where

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

Ideally one would like to select ω to 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

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

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

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

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

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

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

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

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

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

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

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

Proposition II.1[17] *The maximizer, $\bar{\omega}$, of $U(\omega)$ satisfies*

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

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

Proposition II.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 $\hat{\mathrm{P}} = \bar{\mathrm{P}}(\omega) < \bar{\mathbf{P}}$ is known, then any

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

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

$$\Lambda \geq \frac{1}{\bar{\mathbf{P}}} \tag{6}$$

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 sub-optimality of the solution are $\Lambda = \frac{1-\hat{\mathrm{P}}}{\bar{\mathbf{P}}-\hat{\mathrm{P}}}$ and $\Lambda = \frac{1}{\bar{\mathbf{P}}}$ respectively.

III. 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[12] .

The optimization strategy relies on extractions of a random variable Ω whose distribution has modes which coincide with the optimizers of $U(\omega)$. These extractions are obtained through Monte Carlo Markov

Chain (MCMC) simulation[11] . 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 Ω is proportional to $U(\omega)^J$ where J is a positive integer which allows the user to increase the "peakedness" of the distribution and concentrate the extractions around the modes at the price of an increased computational load. If the tunable parameter J is increased during the 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[18] . A formal parallel between these two strategies has been derived in Ref. 13 .

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

$$h(\omega, x_1, x_2, \ldots, x_J) \propto \prod_{j=1}^{J} u(\omega, x_j) p_\omega(x_j) \tag{7}$$

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

$$h(\omega) \propto \left[\int u(\omega, x) p_\omega(x) dx\right]^J = U(\omega)^J . \tag{8}$$

This means that if we can extract realizations of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$ then the extracted Ω's will be concentrated around the optimal points of $U(\Omega)$ for a sufficiently high J. These extractions can be used to find an approximate solution to the 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 (7), 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 $X_j(0) \sim p_{\Omega(0)}(x) \quad j = 1, \ldots, J$

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

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

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

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

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 ρ 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 $h(\omega, x_1, x_2, \ldots, x_J)$ 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

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

By inserting (7) 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 (8). Therefore, after a burn in period, the extractions $\Omega(k)$ accepted by the algorithm will concentrate around the modes of $h(\omega)$, which, by (8) coincide with the optimal points of $U(\omega)$. Results that characterize the convergence rate to the stationary distribution can be found, for example, in Ref. 11.

IV. Simulation Example

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[14–16] . The simulator implements realistic models of current commercial aircraft described in the Base of Aircraft Data (BADA)[19] . The simulator contains also realistic stochastic models of the wind disturbance[20] . The aircraft models contain continuous dynamics, arising from the physical motion of the aircraft, discrete dynamics, arising from the logic embedded in the Flight Management System, and stochastic dynamics, arising from the effect of the wind and incomplete knowledge of physical parameters (for example, the aircraft mass, which depends on fuel, cargo and number of passengers). The simulator has been coded in Java and can be used in different operation modes, either to generate accurate data for validation of the performance of conflict detection and resolution algorithm, or to run faster simulations of simplified models. The nominal path for each aircraft is entered in the simulator as a sequence of way-points. The actual trajectories of the aircraft generated by the simulator are a perturbed version of the nominal path that depends on the particular realizations of wind disturbances and uncertain parameters.

The air traffic simulator has been used to produce the example 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 example, 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 example keeping this fact in mind.

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 1 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.

The conflict resolution problem is illustrated in Figure 2(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

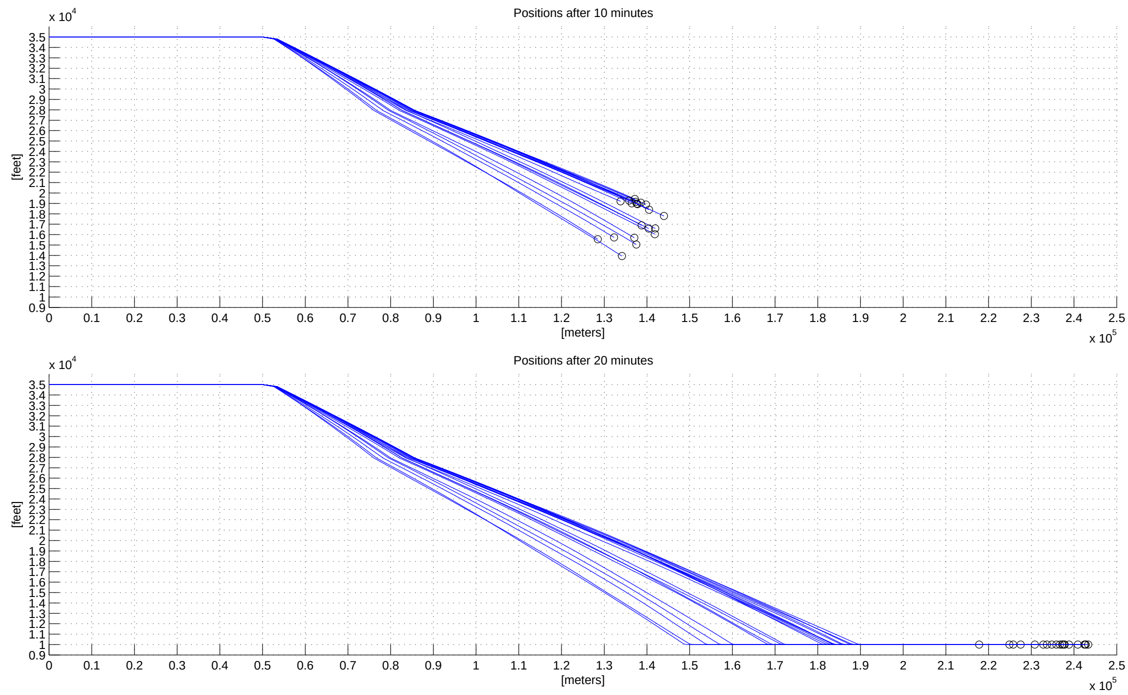


Figure 1. Several trajectory realizations of aircraft descent

second aircraft (A2) is initially at $(-100000, -100000)$ and FL350. This aircraft proceeds to way-point $(-90000, -90000)$ where it will start its 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 $(100000, 0)$.

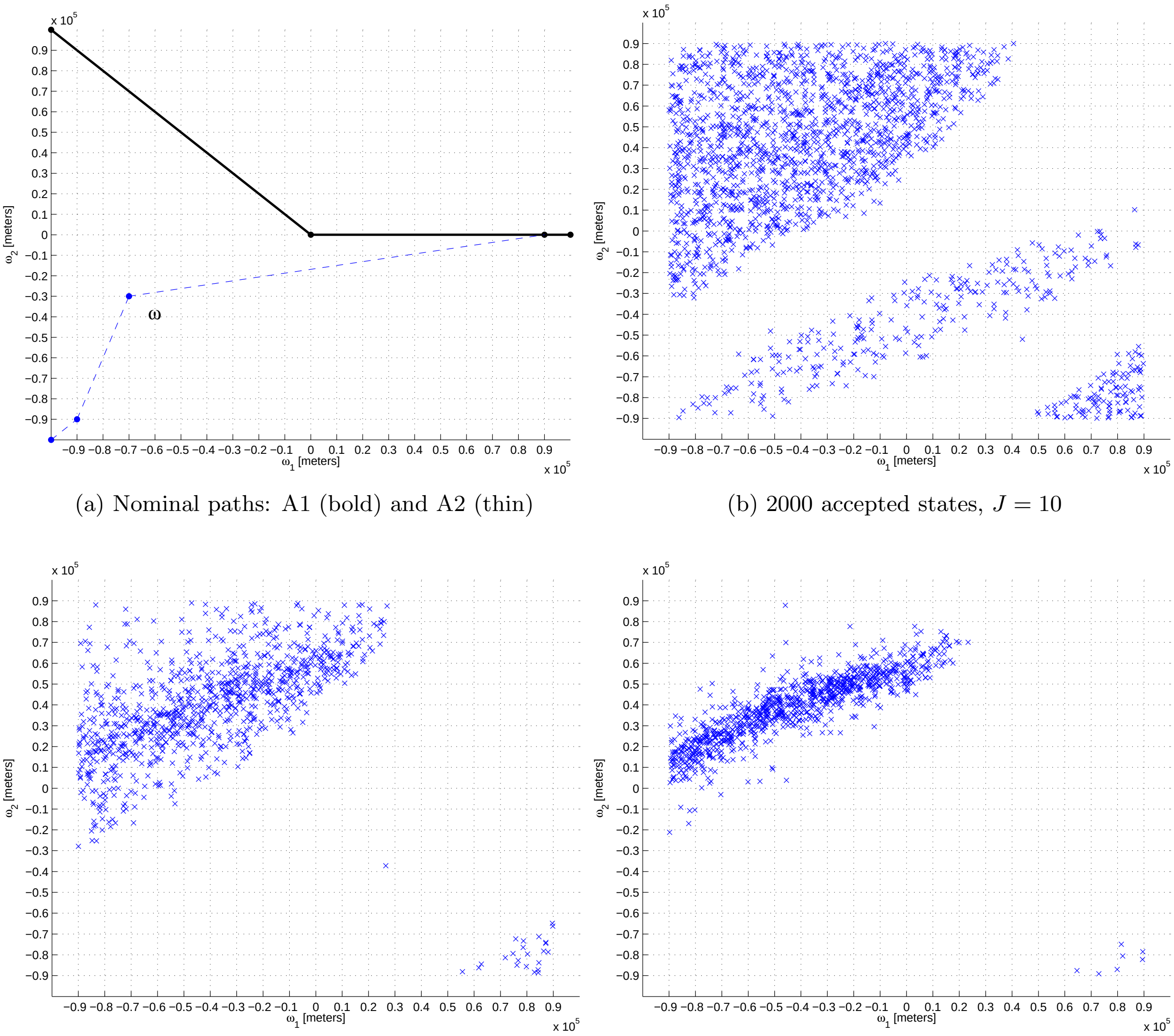
We assume that the objective is to obtain a time separation of 300 seconds between the arrivals of the two aircraft at the exit way-point, $(10000, 0)$. Performance in this sense is measured by $\text{perf} = e^{-a\cdot(|T_1-T_2|-300)}$ where T_1 and T_2 are the arrival times of A1 and A2 at the exit 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[a]. 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 (6), we select $\Lambda = 10$ in the optimization criterion.

The results of the optimization procedure are illustrated in Figures 2(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 2(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 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 2(b) displays the last 2000 extracted states.

Figure 2(c) illustrates the case $J = 50$. In this case the proposal distribution g was a sum of 2000

[a]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 2. Accepted states during MCMC simulation

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 2(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 2(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.

V. Conclusions

In this paper we illustrated our current approach to air traffic conflict resolution in a stochastic setting based on the use of Monte Carlo methods. The main motivation for our approach is to enable the use of realistic stochastic hybrid models of aircraft flight; Monte Carlo methods appear to be the only ones that allow such models. We have formulated conflict resolution as the optimisation of an expected value criterion with probabilistic constraints. Here, a penalty formulation of the problem has been considered which guarantees constraint satisfaction but delivers a suboptimal solution.

Our current research is concerned with overcoming the sub optimality imposed by the need to provide constraint satisfaction guarantees. A possible way is to use the Monte Carlo Markov Chain procedure presented in Section II to obtain optimisation parameters that satisfy the constraint and then to optimise over this set in a successive step. Formulation of the the conflict resolution procedure in the Sequential Monte Carlo[21] framework is also under investigation. Our current research is focused also on modelling and implementation in the simulator of typical Air-Traffic Control situation with a realistic parameterisation of control actions and control objectives.

Acknowledgments

This work was supported by the European Commission under project HYBRIDGE IST-2001-32460 and HYCON Network of Excellence FP6-IST-511368, and by EUROCONTROL under contract C20051E/BM/03.

References

[1] Kuchar, J. and Yang, L., "A review of Conflict Detection and Resolution Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 179–189.

[2] Paielli, R. and Erzberger, H., "Conflict Probability Estimation for Free Flight," *Journal of Guidance, Control and Dynamics*, Vol. 20, No. 3, 1997, pp. 588–596.

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

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

[5] Prandini, M., Hu, J., Lygeros, J., and Sastry, S., "A probabilistic Approach to Aircraft Conflict Detection," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000.

[6] Irvine, R., "A Geometrical Approach to Conflict Probability Estimation," *4th USA/Europe Air Traffic Management R&D Seminar*, Budapest, Hungary, 2001.

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

[8] Frazzoli, E., Mao, Z., Oh, J., and Feron, E., "Aircraft Conflict Resolution Via Semi-Definite Programming," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 79–86.

[9] Hu, J., Prandini, M., and Sastry, S., "Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 25, No. 5, 2002.

[10] Tomlin, C., Pappas, G., and Sastry, S., "Conflict Resolution for Air Traffic Management: a Case Study in Multi-Agent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 5–521.

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

[12] Mueller, P., "Simulation Based Optimal Design," *Bayesian Statistics 6*, J.O. Berger, J.M. Bernardo, A.P. Dawid and A.F.M. Smith (eds.), Oxford University Press, 1999, pp. 459–474.

[13] Mueller, P., Sanso, B., and De Iorio, M., "Optimal Bayesian Design by Inhomogeneous Markov Chain Simulation," Tech. rep., 2003, Available from World Wide Web: http://www.ams.ucsc.edu.

[14] Glover, W. and Lygeros, J., "A Stochastic Hybrid Model for Air Traffic Control Simulation," *Hybrid Systems: Computation and Control*, edited by R. Alur and G. Pappas, No. 2993 in LNCS, Springer Verlag, 2004, pp. 372–386.

[15] Glover, W. and Lygeros, J., "A Multi-aircraft model for conflict detection and resolution algorithm evaluation," Tech. Rep. WP1, Deliverable D1.3, HYBRIDGE, February 18, 2004.

[16] Glover, W. and Lygeros, J., "Simplified models for conflict detection and resolution algorithms," Tech. Rep. WP1, Deliverable D1.4, HYBRIDGE, June 17, 2004.

[17] Lecchini, A., Glover, W., Lygeros, J., and Maciejowski, J., "Monte Carlo Optimization for Conflict Resolution in Air-Traffic Control," *submitted for publication*, 2005.

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

[19] 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/`.

[20] Cole, R., Richard, C., Kim, S., and Bailey, D., "An assessment of the 60 km rapid update cycle (RUC) with Near Real-Time Aircraft Reports," Tech. Rep. NASA/A-1, MIT Lincoln Laboratory, July 15, 1998.

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