

## Simulation Based Bayesian Optimal Design of Aircraft Trajectories for Air Traffic Management

N. Kantas<sup>1,\*</sup>, A. Lecchini-Visintini<sup>2</sup>, J.M. Maciejowski<sup>1</sup>

<sup>1</sup> *Control Group, Cambridge University Engineering Dept.,  
Cambridge CB2 1PZ, UK, {nk234, jmm}@eng.cam.ac.uk*

<sup>2</sup> *Dept. of Engineering, University of Leicester,  
Leicester, LE1 7RH, UK, alv1@leicester.ac.uk*

### SUMMARY

In this paper we consider a specific Air Traffic Management problem, where multiple aircraft in a specific region are required to reach a different target zone in minimum expected time, while maintaining safe separation. The problem is complicated by the presence of random wind disturbances. We propose a realistic policy to automatically generate optimal and safe manoeuvres for each aircraft. The parameters of the optimal policy are computed using a Sequential Monte Carlo approach. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: Bayesian optimal design; sequential Monte Carlo; air traffic management; conflict resolution

### 1. Introduction

In this paper we address specific Air Traffic Management (ATM) problems, where multiple aircraft are interested in cruising through a monitored area in order to reach some specific point or area in the minimum expected time. The problem is complicated by the necessity of all aircraft maintaining safe separation. Although we will consider here only level flight, our proposed methodology is generic for a wide class of problems such as approaching terminal areas or changing flight levels. Our approach can be extended without much effort, for various aircraft dynamics, and for most established statistical wind models.

Planning aircraft trajectories to cross an area in expected minimum time in the presence of constraints can be posed as a non-convex optimisation problem. This is common in many problems in Air Traffic Management and the resulting costs or rewards in general exhibit often many local minima or maxima. This motivates the use of global optimisation algorithms,

---

\*Correspondence to: Control Group, Cambridge University Engineering Dept., Cambridge CB2 1PZ, UK.

Contract/grant sponsor: European Commission under project iFly and EPSRC; contract/grant number: FP6-TREN-037180 and EP/H021558/1 respectively

which in some cases guarantee asymptotic convergence to a global optimum. In most cases such algorithms employ a randomised search strategy to ensure that the search process is not trapped in some local optimum. A popular example is Simulated Annealing (SA).

In addition to the issue of multiple local optima, solving such problems becomes more complicated when stochastic processes are used to represent model uncertainties, such as the effect of the wind velocity on each aircraft. The underlying distribution of the wind field is usually modelled by time-varying spatial stochastic processes. The resulting costs or rewards that need to be optimised are usually expectations over high-dimensional probability distributions, for which it seems impossible to devise analytical solutions or approximations except for a few very simple cases. Even computing such high-dimensional integrals can be a formidable task.

Markov Chain Monte Carlo (MCMC) methods [22], and Sequential Monte Carlo (SMC) methods [4, 7, 18] are currently the most successful methods for evaluating such expectations over relatively high-dimensional spaces under very weak assumptions, and have been widely applied in many areas such as finance, robotics and communications. An interesting point, often overlooked by the applied optimisation community, is that Monte Carlo has also been applied for performing global optimisation, mainly in inference problems such as Bayesian optimal design of experimental settings, standard Maximum Likelihood or Maximum a Posteriori estimation [1, 14, 15, 21]. Considerable theoretical support exists for both MCMC and SMC under very weak assumptions [22, 4].

Recently, in [17] a MCMC algorithm similar to Simulated Annealing (SA) developed in [21] has been applied for scheduling approaching aircraft to a terminal area and computing optimal way points for their manoeuvres. In order to avoid the long execution times that are typically needed for stochastic optimisation routines, it is important to consider carefully the computational complexity of the algorithms and take advantage of any structure that might be available in the problem to speed up computation. In the problems we will be considering this seems to be easier to manage when the SMC framework is used, yielding computationally more efficient algorithms, that are usually easier to implement. In addition, the implementation of SMC algorithms can be easily parallelised. In practice, this can lead to much shorter execution times, when these algorithms are implemented on conventional Graphics Processor Units (GPU) [16].

The outline of this paper is as follows: in Section 2 we formulate the problem as an open-loop control problem by constructing a dual mode policy, which can be easily parameterised. In Section 3 we present a Bayesian interpretation of Simulated Annealing and in Section 4 we show how SMC can be used to compute the parameters of the optimal policy. In Section 5 we illustrate the performance of our approach using simulations. Finally, in Section 6 we provide some concluding remarks.

## 2. Problem Formulation

In this section we provide a probabilistic framework for a specific conflict resolution problem. We will consider level flight between  $M$  aircraft cruising with constant speed in a specified region. First, we will present the dynamics of each aircraft separately and then define the constraint region so that each aircraft maintains a safe separation from the rest, given some pre-specified tolerance probability. Then, we will consider the problem of finding optimal control

inputs to meet these constraints, so that each aircraft arrives at a target exit region in minimum expected time. Finally, we will use the following notation: for a  $j$  dimensional vector we denote  $z_{1:j} = (z_1, \dots, z_j)$ ,  $\|z\| = (z_1^2 + \dots + z_j^2)^{\frac{1}{2}}$ ,  $\delta_y(dz)$  the Dirac unit mass delta measure centered at  $z = y$  and  $\mathbb{I}_A(z)$  the indicator function of set  $A$ .

## 2.1. Dynamics

*2.1.1. Dynamic model for each aircraft* We consider a standard two-dimensional Cartesian fixed-speed model for each aircraft  $i$  [8], whose state  $x_t^i (\in \mathbb{R}^2)$  is composed of the  $x$  and  $y$  Cartesian coordinates of the aircraft. Let the true aircraft speed of aircraft  $i$  be  $v_{tas}^i (\in \mathbb{R})$ , and let also the control input at each time be the heading,  $h_t^i (\in [-\frac{\pi}{2}, \frac{\pi}{2}])$ , giving the following recursive dynamics for the state:

$$x_t^i = x_{t-1}^i + \begin{bmatrix} \cos h_t^i \\ \sin h_t^i \end{bmatrix} v_{tas}^i \delta + w_{t-1}(x_{t-1}^i) \delta \quad (1)$$

where  $\delta$  is the common time discretisation interval, and  $w_t(x)$  is a random disturbance representing the wind velocity at point  $x$  of the state space at time  $t$ . We assume that the sequence of headings  $h_{1:t}^i$  can be parameterised by some fixed (possibly low) dimensional parameter  $\theta^i$  and  $w_{t-1}$  is a spatio-temporal varying random variable defined on some open subset of  $\mathbb{R}^2$ . Also, for each aircraft  $i$ , we will assume there exists some exit point  $e^i (\in \mathbb{R}^2)$ . The aim of the control inputs is to drive each aircraft  $i$  to some neighbourhood of  $e^i$ , denoted by  $B^i (\subset \mathbb{R}^2)$ , which we will refer to frequently in the paper as the ‘target set’ or ‘exit region’.

*2.1.2. Parameterisation of control inputs* We will show how  $h_{1:t}^i$  can be represented as a deterministic function of a parameter  $\theta^i$ . Our approach uses ‘blocking’ of control inputs to reduce the dimension of the control space. The motivation for doing this arises from both the practical issues in the specific application and the computational aspects of the underlying methodology. As far as the application is concerned the resulting manoeuvres need to be simple enough to be implemented in practice either by a pilot or an autopilot. In terms of the methodology it is also important to reduce the control space to a reasonable sized parameter space, so that a fast and efficient search for the optimal control law can be made possible. We will refer to the control law also as a ‘policy’ and consider the parameter of the policy to be a deterministic variable to be computed.

More specifically for our parameterisation, we will divide the sequence of  $h_{1:t}^i$  to a sequence of  $H$  blocks of constant inputs as follows:

$$h_{1:t}^i = \underbrace{[h_1^i, \dots, h_1^i]}_{\substack{h_1 \\ \text{Block 1}}}, \underbrace{[h_2^i, \dots, h_2^i]}_{\substack{h_2 \\ \text{Block 2}}}, \dots, \underbrace{[h_H^i, \dots, h_H^i]}_{\substack{h_H \\ \text{Block } H}},$$

In this case, for every  $k = 1, \dots, H$ , each  $h_k$  is composed by the stacked vector of every  $h_k^i$ , which in turn can be computed by using the change in heading, namely  $\theta_k^i$ , compared to the heading in the first block  $h_1^i$ . This gives the following control update for  $k > 1$ :

$$h_k^i = h_1^i + \theta_k^i, \quad (2)$$

where we will for the time being set  $\theta_1^i = 0$ . Then the parameter is simply given by  $\theta^i = [h_1^i, \theta_2^i, \dots, \theta_H^i]$ . Note that we have not made so far any assumptions on each block size, which may vary with  $k$  and could be used as a design or tuning variable.

In this paper we shall use the same length  $p$ , for each block, except the last one, which will usually be a longer block. In the last block we propose the use of either *tracking a straight-line course* or a *fixed heading*:

**Tracking:** we propose that each element in the final block is time varying in this case. In tracking the heading counteracts the cross-track action of the wind [9]. The inputs are computed at time  $t$  so that  $x_{t+1}^i$  tracks the straight line connecting  $x_{t-(H-1)p}^i$  to the exit point  $e^i$  for aircraft  $i$ . Therefore, each element in  $\underline{h}_H = [\mathbf{h}_{t-(H-1)p+1}^i, \dots, \mathbf{h}_t^i]^T$  can be computed recursively for  $m = t - (H - 1)p + 1, \dots, t$  as follows:

$$\mathbf{h}_{m+1}^i = \varphi_m^i + \sin^{-1} \left( \frac{\|w_m(x_m^i)\| \sin(\tan^{-1}(\frac{w_m(x_m^i)(2)}{w_m(x_m^i)(1)}) - \varphi_m^i)}{\delta v_{tas}} \right) \quad (3)$$

where  $\|\cdot\|$  denotes the 2-norm and for any time  $t$  and aircraft  $i$ , the current angle,  $\varphi_t$ , towards the exit-point  $e^i$ , is given by

$$\varphi_t^i = \tan^{-1} \left( \frac{e^i(2) - x_t^i(2)}{e^i(1) - x_t^i(1)} \right) \quad (4)$$

**Fixed heading:** in this case we propose that  $\underline{h}_H$  is the stacked vector composed of  $t - (H - 1)p$  repetitions of  $\mathbf{h}_H^i$ , which is given by the heading angle after the last input of  $\underline{h}_{H-1}$  is applied, i.e.

$$\mathbf{h}_H^i = \varphi_{t-(H-1)p}^i. \quad (5)$$

Note that in both cases, in contrast to the previous blocks, the size of the final block is not fixed but varies depending on  $t$ . In our application  $t$  will keep growing until  $x_t^i$  has reached its target set  $B^i$ . As a result the size of the last block can be thought of as random, but in each case the input heading is given by a deterministic update scheme, as shown in (3)-(5). The benefits of this strategy in each case is that  $\theta^i$  is fixed and possibly low-dimensional, regardless of the time required for the state trajectories to reach the target sets. The final block is somewhat analogous to the ‘terminal control law’ that is commonly used in ‘dual-mode’ model predictive control schemes, as introduced in [20] – we think of the first  $H - 1$  blocks as ‘mode 1’, and of the final block as ‘mode 2’.

*2.1.3. Multiple aircraft state augmentation and constraints* So far we have considered the dynamics of each aircraft separately. Now, we will assume that  $M$  aircraft are always present in the area of surveillance, so  $i = 1, \dots, M$ . Also we denote  $x_t$  to be the stacked vector of every  $x_t^i$ , i.e.  $x_t = [x_t^{1T}, \dots, x_t^{MT}]^T$  and similarly for the control parameters we have  $\theta = [\theta^{1T}, \dots, \theta^{MT}]^T$ .

We will also assume that for a fixed known value of  $\theta$ ,  $X_t$  is a Markov chain generated by the law  $\mathbb{P}_{x_0, \theta, t}(dx_{0:t}) = \prod_{k=1}^t f(x_k | x_{k-1}) dx_{0:t}$  with initial distribution  $\delta_{x_0}(dx)$ . The dynamics can be also written as a joint Markov transition density  $f_\theta$  with respect to the Lebesgue measure  $dx_t$ :

$$\mathbb{P}_{x_0, \theta}(X_t \in dx_t | X_{1:t-1} = x_{1:t-1}) = \mathbb{P}_{x_0, \theta}(X_t \in dx_t | X_{t-1} = x_{t-1}) = f_\theta(x_t | x_{t-1}) dx_t. \quad (6)$$

Each individual transition density  $f_\theta^i(x_t^i | x_{t-1}^i)$  with respect to  $dx_t^i$  defined as

$$\mathbb{P}_{x_0, \theta}(X_t^i \in dx_t^i | X_{1:t-1} = x_{1:t-1}) = \mathbb{P}_{x_0, \theta}(X_t^i \in dx_t^i | X_{t-1} = x_{t-1}) = f_\theta^i(x_t^i | x_{t-1}^i) dx_t^i$$

can be obtained through marginalisation

$$f_{\theta}^i(x_t^i|x_{t-1}) = \int f_{\theta}(x_t|x_{t-1})dx_t^{-i}, \quad (7)$$

where  $x^{-i}=[x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^M]$ .

For the purpose of air traffic management, we are primarily interested in maintaining at least some minimum safe separation distance between each aircraft. Therefore at each time  $t$  for the state vector  $x_t$  we define the feasible state space as

$$\mathcal{X}_t = \left\{ x_t^i \in \mathbb{R}^2 : \forall j \neq i \left\| x_t^i - x_t^j \right\| > d, i, j = 1, \dots, M \right\}, \quad (8)$$

and the path space of the feasible trajectories as  $\mathcal{X}_{0:t} = \mathcal{X}_0 \times \dots \times \mathcal{X}_t$ . Moreover, denote  $\Theta$  as the set of admissible actions  $\theta$  such that for a given  $t$  the probability of safe separation ( $PSS_t$ ) satisfies

$$PSS_t := \mathbb{P}_{x_0, \theta, t}(X_{0:t} \in \mathcal{X}_{0:t}) \geq 1 - \varepsilon, \quad (9)$$

where  $\varepsilon \in [0, 1)$  is some tolerance level. Note that in problems with stochastic dynamics it might not be possible to use deterministic constraints and we therefore opt for the use of some pre-specified tolerance level  $\varepsilon$  which can be interpreted as an allowed or safe probability of conflict. Note that, although it was natural to pose the joint dynamics in a distributed fashion as done for the case of each aircraft earlier, this is not straightforward for the case of the constraints, which involve pairwise comparisons of the aircrafts' states.

*2.1.4. Feynman-Kac model for the augmented process* In order to avoid dealing with the random sets  $\mathcal{X}_t$  defined in (8) we will use the discrete time Feynman-Kac model formalism described in detail in [4]. For convenience, let  $\mathbf{X}_t = X_{1:t}$  denote the sequence of  $t$  consecutive augmented states which is generated by the law of the Markov chain  $\mathbb{P}_{x_0, \theta, t}(dx_{0:t})$ . We will use the potential function  $\mathcal{G}_t(x_{0:t}) = \prod_{k=1}^{t-1} G_k(x_k)$  where

$$G_k(x_k) = \begin{cases} 1 & \inf_{i \neq j} \left\| x_k^i - x_k^j \right\| > d, i, j = 1, \dots, M \\ 0 & \text{otherwise} \end{cases}$$

to define the probability measure of the path of feasible state sequences up to time  $t$  as

$$\mathbb{Q}_{x_0, \theta, t}(\mathbf{X}_t \in \mathcal{A}) = \frac{1}{Z_t} \mathcal{G}_t(\mathbf{X}_t) \mathbb{P}_{x_0, \theta, t}(\mathbf{X}_t \in \mathcal{A}),$$

where  $\mathcal{A} \in \mathbb{R}^{2M}$  is an arbitrary set and  $Z_t$  is the normalising constant given by

$$Z_t = \mathbb{E}_{\mathbb{P}_{x_0, \theta, t}} \left[ \prod_{k=1}^{t-1} G_k(x_k) \right],$$

with  $\mathbb{E}_{\mathbb{P}_{x_0, \theta, t}}$  denoting the expectation under the probability measure  $\mathbb{P}_{x_0, \theta, t}$ .

An alternative definition for the law of the feasible state sequence path up to time  $t$  can be given using an arbitrarily chosen bounded test function  $\Phi : \mathbb{R}^{2t} \rightarrow \mathbb{R}$  as follows:

$$\mathbb{Q}_{x_0, \theta, t}(\Phi) = \frac{1}{Z_t} \mathbb{E}_{\mathbb{P}_{x_0, \theta, t}} \left[ \Phi(X_{0:t}) \prod_{k=1}^{t-1} G_k(x_k) \right].$$

Similarly, using a bounded test function  $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$  we can derive a sequence of distributions  $\eta_t$  for each  $X_t$  as

$$\eta_t(\phi) := \frac{\gamma_t(\phi)}{\gamma_t(1)} = \frac{1}{Z_t} \mathbb{E}_{\mathbb{P}_{x_0, \theta, t}} \left[ \phi(X_t) \prod_{k=1}^{t-1} G_k(x_k) \right]. \quad (10)$$

Given  $\eta_k(G_k) = \frac{Z_{k+1}}{Z_k}$  it is straightforward to show that for the unnormalised measure  $\gamma_t$  that

$$\gamma_t(\phi) = \eta_t(\phi) \prod_{k=1}^{t-1} \eta_k(G_k). \quad (11)$$

For the problem of interest one can interpret the normalising constant  $\gamma_t(G_t) = \gamma_{t+1}(1)$  as being  $PSS_t$ . Consequently from (10)-(11) we obtain the recursion

$$PSS_t = \gamma_{t+1}(1) = Z_{t+1} = \prod_{k=1}^t \eta_k(G_k), \quad (12)$$

which will be used later to compute  $PSS_t$  numerically.

*2.1.5. Monte Carlo approximations and relaxing the constraints* Although the Feynman-Kac model described earlier provides a convenient interpretation of  $PSS_t$  as the normalising constant  $Z_{t+1}$ , we note that in most cases this normalising constant cannot be computed analytically. In this section we will briefly sketch how to sample from  $\mathbb{Q}_{x_0, \theta, t}$  and numerically estimate  $PSS_t$  by a Monte Carlo approximation based on Importance Sampling (IS) [18]. The aim is to relax the constraints so that they remain valid for the approximation of  $PSS_t$  instead.

First we assume that it is possible to sample from the true dynamics using samples from the transition density  $f_\theta$  and we can therefore obtain  $\kappa$  independent identically distributed (iid) samples  $\{\mathbf{x}_t^{(l)}\}_{l=1}^\kappa$  from  $\mathbb{P}_{x_0, \theta, t}$ . We can then check whether any conflicts are present at any time and select samples as samples of  $\mathbb{Q}_{x_0, \theta, t}$  only when  $\mathcal{G}_t(\mathbf{x}_t^{(l)}) = 1$ . The Monte Carlo estimate of  $PSS_t$ , namely  $\widehat{PSS}_t$ , is given by:

$$\widehat{PSS}_t = \frac{1}{\kappa} \sum_{l=1}^{\kappa} \mathcal{G}_t(\mathbf{x}_t^{(l)})$$

One could then soften the constraints to take the following form,  $\widehat{PSS}_t \geq 1 - \varepsilon$ .

In the subsequent methodology we will assume that it is indeed possible to simulate from the true transition density  $f_\theta$ . Even if this is not the case our methodology can be still used provided that one can still obtain unbiased samples  $x_{1:t}$  from the dynamics. The minimal requirement is that point-wise computation of  $f_\theta$  is available. Therefore, the methodology and algorithms presented in the rest of the paper are generic and can be used not only for level flight, but for example in problems with changing altitude.

## 2.2. Cost function

We define for each aircraft the time of arrival  $T^i$  at the target set  $B^i$  as

$$T^i = \inf_{t \geq 1} \{t : X_t^i \in B^i\}.$$

Notice that in the context of stochastic processes the time of arrival corresponds to the random hitting time of set  $B^i$ . We will assume that once some aircraft first enters in  $B^i$  with a the realisation of the first hitting time  $T^i$  being  $\tau^i$ , then the state corresponding sequence  $\mathbf{x}_{\tau^i+m}^i$ ,  $m > 1$ , remains in an arbitrary absorbing feasible set, where the subsequent trajectory does not violate the constraints.

We will write the expected time of arrival or hitting time at  $B^i$  as:

$$\mathbb{E}_{x_0, \theta}[T^i] = \sum_{\tau=1}^{\infty} \tau^i \pi_{T^i, \theta}(\tau^i) \text{ with } \pi_{T^i, \theta}(\tau^i) = \frac{\mathbb{Q}_{x_0, \theta, \tau^i+1}\{T^i = \tau^i\}}{\sum_{m=1}^{\infty} \mathbb{Q}_{x_0, \theta, m+1}\{T^i = m\}}. \quad (13)$$

We define the probability mass function (pmf) of the hitting times (or times of arrival),  $\pi_{T^i, \theta}(\tau^i) \propto \mathbb{Q}_{x_0, \theta, \tau^i+1}\{T^i = \tau^i\}$ , using  $\mathbb{Q}_{x_0, \theta, t}$  instead of the natural law of the process  $\mathbb{P}_{x_0, \theta, t}$  to avoid  $G_k(X_k) = 0$  occuring, which would lead to an infinite  $T^i$ . Thus we ensure that all times  $t \leq \tau^i$ , the trajectories  $\mathbf{X}_t^i$  remain in the feasible space  $\mathcal{X}_{0:t}$ . Furthermore, to avoid other pathological cases, we will also assume that for every  $i$ ,  $\mathbb{Q}_{x_0, \theta}\{T^i < \infty\} = 1$  holds for some value of  $\theta \in \Theta$  and in addition  $\mathbb{E}_{x_0, \theta}(T^i) < \infty$  for every  $\theta, i$ .

In order to compute  $\pi_{T^i, \theta}$  we will consider all possible feasible state sequences  $\mathbf{X}_{\tau^i}$ , for which the state of aircraft  $i$  just reaches the target set  $B^i$  at time  $\tau^i$  given the previous part of the trajectory has not done so. For this reason we define the function  $g_{\tau^i}^i : \mathbb{R}^{2M\tau^i} \rightarrow \{0, 1\}$

$$g_{\tau^i}^i(\mathbf{x}_{\tau^i}) = \mathbb{I}_{x_{\tau^i}^i \in B^i} \prod_{k=1}^{\tau^i-1} \mathbb{I}_{x_k^i \notin B^i} \quad (14)$$

and use it to compute  $\pi_{T^i, \theta}(\tau^i)$  as

$$\pi_{T^i, \theta}(\tau^i) \propto \mathbb{Q}_{x_0, \theta, \tau^i+1}\{T^i = \tau^i\} = \frac{\mathbb{E}_{\mathbb{P}_{x_0, \theta, \tau^i}} \left[ g_{\tau^i}^i(\mathbf{x}_{\tau^i}) \prod_{k=1}^{\tau^i} G_k(\mathbf{x}_k) \right]}{\mathbb{E}_{\mathbb{P}_{x_0, \theta, \tau^i}} \left[ \prod_{k=1}^{\tau^i} G_k(\mathbf{x}_k) \right]} = \eta_{\tau^i+1}(g^i). \quad (15)$$

It is implied that by sampling any trajectory from  $\mathbb{Q}_{x_0, \theta, \tau^i+1}$ , one can determine if  $\tau^i$  is a sample from  $\pi_{T^i, \theta}(\tau^i)$  by checking whether  $g_{\tau^i}^i(\mathbf{x}_{\tau^i}) = 1$ . In this case we will require that for every time  $t > \tau^i$ ,  $g_t^i = 1$  also holds, because  $B^i$  was formulated as an absorbing set.

In this paper we shall consider minimising the expected maximum time of arrival with respect to all aircraft. More precisely we define the stopping time to be minimised as  $T = \max_i T^i$  and cast the problem as to compute

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}_{x_0, \theta}[T], \text{ s.t. } \widehat{PSS}_T \geq 1 - \epsilon$$

where the relaxed constraints impose the need of all aircraft staying within the feasible region with some confidence probability  $1 - \epsilon$ . This is a difficult cost to minimise by conventional methods, such as analytical computations or the use of gradients. This motivates the use of simulated based methods to estimate  $\theta^*$ .

In addition, we can use that  $\mathbb{E}_{x_0, \theta}(T) < \infty$  and therefore can be bounded from above by a constant  $\bar{\tau}_{x_0} = \sup_{\theta} \mathbb{E}_{x_0, \theta}(T)$ , in order to transform the minimisation problem to a maximisation problem and estimate  $\theta^*$  as the maximisers of

$$J_{x_0}(\theta) = \int u(\tau) \pi_{T, \theta}(\tau) d\tau, \quad (16)$$

where

$$u(\tau) = \bar{\tau} - \tau \text{ and } \pi_{T,\theta}(\tau) = \sum_{\tau=1}^{\infty} \frac{\mathbb{Q}_{x_0,\theta,\tau+1}\{T = \tau\}}{\sum_{m=1}^{\infty} \mathbb{Q}_{x_0,\theta,m+1}\{T = m\}} \delta_{\tau}(d\tau) \quad (17)$$

and  $\bar{\tau} > \bar{\tau}_{x_0}$  is an arbitrary constant so that  $u > 0$ . Having to solve a maximisation problem makes it easier to use the annealing methods found in [1, 14, 15, 21].

Note that at first glance choosing such a  $\bar{\tau}$  might seem restrictive, but this is mitigated by the fact that in practice it might be easy to find some  $\bar{\tau}$  large enough so that the problem remains unaffected. To simplify exposition in the subsequent sections we formulated  $\pi_{T,\theta}(\tau)$  as a probability density function composed from a train of Dirac delta functions. As regards to computing  $\pi_{T,\theta}(\tau)$  one can use  $\mathbb{Q}_{x_0,\theta,\tau+1}\{T = \tau\} \propto \eta_{\tau+1}(\prod_{i=1}^M g_{\tau^i}^i)$  similarly to equation (15).

### 2.3. Wind modelling

In this section we shall discuss how the wind velocity  $w_t(x)$  can be modelled at each point  $x \in \mathbb{R}^2$ . Our approach is close to that of [2, 10, 19] and similar ideas have appeared in [11]. More formally, we assume that  $W_t$  is a measurable process taking values in  $\mathbb{R}^2$ . We will assume that the mean of the process  $\mu_t$  is a spatio-temporal varying two dimensional surface, that depends on the past history of the process through a Markov relation

$$\mathbb{E}[W_t(x)|W_{1:t-1} = w_{1:t-1}] = \mathbb{E}[W_t(x)|W_{t-1} = w_{t-1}] = \mu_t(x; s_{\mu}(w_{t-1})),$$

where  $s_{\mu}(w_{t-1})$  is a vector of fixed dimensional sufficient statistics, for example some linear combination of  $s_{\mu}(w_{t-2})$ , the previous mean and covariance. Similarly for the variance we will assume

$$\mathbb{E}\left[(W_t(x) - \mu_t(x; s_{\mu}(w_{t-1})))^2 | W_{t-1} = w_{t-1}\right] = \Sigma_t(x; s_{\Sigma}(w_{t-1})),$$

with  $s_{\Sigma}(w_{t-1})$  being a the sufficient statistics for the variance. As we are assuming a stationary model, note that the sufficient statistics can be computed recursively.

This level of abstraction formulates the problem in its most general setting, but for the purposes of this paper, we will use the linear Gaussian model

$$w_t(x) = \mu_t(x; s_{\mu}(w_{t-1})) + v_t,$$

where  $v_t \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \Sigma_t(x; s_{\Sigma}(w_{t-1})))$ . In [2, 10, 19] the authors assumed that  $s_{\mu}$  and  $s_{\Sigma}$  depend on the complete path of  $W_{1:t-1}$ . This choice was influenced by the exhaustive data analysis on weather forecasts carried out in [3]. There the datasets analysed spanned a very large period of time, order of magnitudes larger than the few tens of minutes we are interested in. In this paper we choose to use Markov modelling rather than using a time varying autoregressive process as in [12]. This was further validated in the datasets used for simulations in Section 5. Again, we stress that the methodology and the simulation based algorithm presented later in the paper does not rely on this specific model choice.

In practice, the surface  $\mu_t(x)$  is unknown and what is available is gridded data, which corresponds to Rapid Update Cycle (RUC) weather forecasts [3] at specific points of the state space. The forecasts typically appear in intervals of three hours for the next three days. They are updated daily and can be downloaded from the National Oceanic and Atmospheric Administration (NOAA) server. We shall use this gridded data to provide smooth approximations of  $\mu_t$  and  $\Sigma_t$  both in space and time.



As far as  $\mu_t$  is concerned, obtaining approximations for the surfaces drops to a regression problem. For both the spatial and temporal regression we opted for the use of plain linear interpolation as in [19]. These choices were largely influenced by the inherent simplicity and computational efficiency. In addition early, but not exhaustive, numerical validation indicated that using more advanced regression techniques, such as Bayesian analysis [24], or splines [11] for the spatial regression, could yield better approximations but at a significant computational cost. In any case, the benefit of these methods is marginal in terms of illustrating the performance of our approach. Therefore these and other advanced methods should not be ruled out, when it comes to implementing a more realistic or complex example, but their further investigation is beyond the scope of this paper.

As regards obtaining approximations for  $\Sigma_t$ , we used a simplification of the work found in [2, 19], suitable for implementation of the spatial component of a Markovian wind field. Based on the additional assumption that the variance retains a diagonal isotropic structure for the two dimensional model, we use the following covariance:

$$\mathbb{E}[(W_t - \mu_t(x))(W_t - \mu_t(x')) | W_{t-1} = w_{t-1}] = \text{diag}[\sigma^2 \sigma_x(|x(i) - x'(i)|)]_{i=1,2},$$

where  $\text{diag}[\varphi(i)]_{i=1,2}$  denotes the  $2 \times 2$  diagonal matrix with  $\varphi(i)$  being the  $i$ -th element on the diagonal,  $\sigma$  is a constant, and  $\sigma_x(\epsilon) = -0.006 + 1.006 \exp(\frac{-\epsilon}{337000})$ . The authors in [10, 19] also provide a recursive method to compute  $s_{\Sigma}(w_{t-1})$  and  $\Sigma_t(x | s_{\Sigma}(w_{t-1}))$  for this model.

### 3. Bayesian optimal design

Bayesian statistical methodology relies on the fact that all variables of the problem, such as state, parameters, or observations if any, are modelled as random variables. Moreover prior distributions are assigned to each one of them, which encapsulates any prior information or expert knowledge not captured by the model. Then inference is performed using the joint or marginal posterior appropriately. As such information is not likely to be easily available, vague priors distributions can be used. In this section we show how Simulated Annealing can be used to maximise the Bayesian interpretation  $J_{x_0}(\theta)$ . Applying these ideas in the area of Air Traffic Management has been initiated by [17] for a different application, but the results were promising so that the methodology can be further exploited and used.

#### 3.1. Bayesian inference on times of arrival

More specifically, in the case of the stopping time  $T$ , which is already defined as a random variable, assigning a prior  $p(\tau)$  can emphasise the search or sampling in more practically useful regions, for example by considering only cases where  $T < \bar{\tau}$ . In this case, the simplest choice of the prior  $p(\tau)$  can be the uniform distribution  $\mathcal{U}_{[1, \bar{\tau}]}$ . In the remainder of this paper we shall use this prior, as typically one might be able to deduce extreme or worst case time of arrivals  $\bar{\tau}$ . Of course this is not the only possible choice. Other possibly diffuse distributions with most of its density mass concentrated in  $[1, \bar{\tau}]$  can be considered, e.g. a negative binomial, without actually requiring  $T < \bar{\tau}$ ,  $\mathbb{Q}_{x_0, \theta}$ -almost surely (*a.s.*). From a Bayesian perspective we would be actually interested in actually sampling  $\tau$  from the following posterior

$$\tilde{\pi}_{T, \theta}(\tau) \propto \pi_{T, \theta}(\tau) p(\tau).$$

As expected, for the specific choice of prior being a uniform distribution, sampling from  $\tilde{\pi}_{T^i, \theta^i}(\tau)$  will not return any samples  $\tau > \bar{\tau}$ .

In the context of Bayesian optimal design for the parameter  $\theta$  we would be interested instead in minimising the a posteriori mean of the time of arrival with respect to  $\tilde{\pi}_{T, \theta}(\tau)$  rather than (18), thus yielding an alternative a posteriori reward criterion to be maximised given by

$$\tilde{J}_{x_0}(\theta) = \int u(\tau) \tilde{\pi}_{T, \theta}(\tau) d\tau, \quad (18)$$

It is clear that we are modifying the problem formulation of Section 2.2 here. This does not involve the particular choice of prior, which implies  $T < \bar{\tau}$ ,  $\mathbb{Q}_{x_0, \theta}$ -a.s.; if a negative binomial prior was used instead this would not be the issue, but the shape of the posterior would be a result of twisting the shape of  $\pi_{T, \theta}(\tau)$  by multiplying it with the prior distribution. This modification of the problem to be solved is well justified in the Bayesian statistics literature (see [22] for a few examples) and in some sense we use the prior as a flexible means of inserting valuable information not captured by modelling the times of arrival strictly as the hitting time of some set  $B^i$  by the marginal of a Markov process  $x_t^i$ .

---

**Algorithm 1** Obtaining samples  $\tau^i \sim \tilde{\pi}_{T^i, \theta^i}$  and  $\tau \sim \tilde{\pi}_{T, \theta}$  given  $\theta, x_0, \bar{\tau}; i = 1, \dots, M$ .

---

```

1: Initialise permitted number of restarts at each  $k = 1, \dots, \bar{\tau}$ : set all  $\varkappa_k = 1, \varkappa_{\max} = \bar{\varkappa}$ .
2: for  $k = 1, \dots, \bar{\tau}$  do
3:   Sample  $x_k \sim f_{\theta}(\cdot | x_{k-1})$ 
4:   if  $\mathbb{I}_{x_k \in \mathcal{X}} = 1$ , then
5:     Initialise  $\mathcal{I} = 1, \dots, M$ .
6:     for each  $i \in \mathcal{I}$  do
7:       if  $\mathbb{I}_{x_k^i \in (B^i)} = 1$ , then
8:         set  $\tau^i = k$ , augment  $\mathbf{x}_{\tau}^i = [\mathbf{x}_{k-1}^i, x_k^i]$ , set  $\mathcal{I} \leftarrow \mathcal{I} \setminus \{i\}$  and return  $\tau^i, \mathbf{x}_{\tau}^i$ 
9:       else
10:        augment  $\mathbf{x}_k^i = [\mathbf{x}_{k-1}^i, x_k^i]$  and proceed to  $k + 1$ 
11:      end if
12:    end for
13:  else if  $k \neq \bar{\tau}$  and  $\varkappa_k < \bar{\varkappa}$  then
14:    set  $\varkappa_k = \varkappa_k + 1$ , go to step 3
15:  else
16:    terminate without sample.
17:  end if
18: end for
19: Set  $\tau = \max_i \tau^i$  and return  $\tau, \widehat{PSS}_{\tau} = \prod_{k=1}^{\tau} \frac{1}{\varkappa_k}$ .
```

---

In Algorithm 1 we present a sequential Importance Sampling algorithm that samples from  $\tilde{\pi}_{T, \theta}(\tau)$  for a fixed  $\theta$ . First we generate samples  $\tau^i \sim \tilde{\pi}_{T^i, \theta^i}$  for every  $i$  and then we set the sample of the stopping time to be  $\tau = \max_i \tau_i$ . A convenient by-product of Algorithm 1 is that we can reuse the samples from  $\tilde{\pi}_{T, \theta}(\tau)$  to obtain estimates of the probability of safe separation  $PSS_{\tau}$ . Noting that  $\eta_k(G_k) \approx \frac{1}{\varkappa_k}$  and using equation (12) we obtain the estimate  $\widehat{PSS}_{\tau} = \prod_{k=1}^{\tau} \frac{1}{\varkappa_k}$ . To improve on the variance of this estimator one may use  $L \gg 1$  iid

samples of Algorithm 1 and use the Monte Carlo estimator  $\widehat{PSS}_\tau = \frac{1}{L} \sum_{l=1}^L \prod_{k=1}^{\tau^{(l)}} \frac{1}{x_k^{(l)}}$ . When  $\widehat{PSS}_\tau \geq 1 - \varepsilon$ , then  $\theta$  can be regarded as a feasible parameter.

Clearly, the specific choice of parameterisation for the policy is crucial to obtain samples efficiently, without wasting too much sampling effort in regions of the state space where the probability of aircraft  $i$  reaching  $B^i$  is extremely low.

### 3.2. Simulated annealing

As far as  $\theta$  is concerned, so far it has been treated as a deterministic parameter of the policy that needs to be computed. Using a Bayesian approach, we will treat it as a random variable and assign it a prior with density  $p(\theta)$ . In our application a possible choice of  $p(\theta)$  can be a density peaked where each  $h^i_1, \theta^i_k$  are near  $\varphi^i_0$  and zero respectively.

In the standard Bayesian interpretation of Simulated Annealing, we are interested in sampling  $\theta^i$  from a density  $\tilde{\pi}_\gamma$

$$\tilde{\pi}_\gamma(\theta) \propto p(\theta) \tilde{J}_{x_0}(\theta)^\gamma,$$

where  $p(\theta)$  is the pdf of an arbitrary prior distribution, which must contain the maximisers  $\theta^*$ . Under weak assumptions, as  $\gamma \rightarrow \infty$ ,  $\tilde{\pi}_\gamma(\theta)$  becomes concentrated on the set of maximisers of  $J$  [13].

Alternatively, we can introduce  $\gamma_n$  artificial replicates of  $\tau$ , all stacked into a joint variable  $\tau_{1:\gamma_n}$  and define the following joint posterior density:

$$\pi_n(\theta, \tau_{1:\gamma_n}) \propto p(\theta) \prod_{j=1}^{\gamma_n} u(\tau_j) \pi_{T,\theta}(\tau_j) p(\tau_j) \quad (19)$$

We will denote the marginal posterior as  $\pi_n(\theta)$ , for which it is easy to show that it is indeed proportional to  $\tilde{\pi}_\gamma$ , i.e.

$$\tilde{\pi}_\gamma(\theta) \propto \int \pi_n(\theta, \tau_{1:\gamma_n}) d\tau_{1:\gamma_n}.$$

$\{\gamma_n\}_{n \geq 0}$  is a strictly increasing integer infinite sequence, which will play the role of the inverse temperature. Under weak assumptions, for a logarithmic schedule and large  $n$ , one can obtain formal convergence results, and as  $\gamma_n \rightarrow \infty$ ,  $\pi_n$  becomes concentrated on the set of maximisers of  $J_{x_0}(\theta)$ , [21]. It has been shown empirically that logarithmic schedules can lead to slow convergence and therefore more quickly increasing rates, e.g. linear, are used. In addition, as infinite sequences are impossible to use, finite ones,  $\{\gamma_n\}_{n=1, \dots, n_{\max}}$ , are used instead. Also **note** that iteration index  $n$  should not be confused with time index  $t$ .

## 4. Simulation based optimisation

In general it is impossible to sample directly from  $\pi_n$ , hence various Monte Carlo simulation schemes have been proposed. In [1, 21] this is achieved by Markov Chain Monte Carlo (MCMC). Instead, in this paper we propose to adopt the Sequential Monte Carlo (SMC) samplers framework of [6]. Our approach is almost identical to the application of SMC samplers presented in [14] for Maximum Likelihood parameter estimation and later in [15] for Bayesian decision theory. Later in this section we shall propose an algorithm similar to [14] and [15], which differs only in the application and context used.

#### 4.1. SMC samplers

SMC methods are a sequential implementation of importance sampling to sample from a general distribution with density  $\pi(\mathbf{y}) = \frac{v(\mathbf{y})}{Z}$ , where  $v(\mathbf{y})$  can be evaluated point-wise and the normalisation constant  $Z$  is unknown. A swarm of samples  $\{y_{1:n}^{(l)}\}_{l=1}^L$ , called particles, evolves towards the distribution of interest by passing through a sequence of intermediate bridging distributions of increasing dimensions with densities  $\{\pi_n(y_{1:n}) = \frac{v_n(y_{1:n})}{Z_n}\}_{n \leq T}$ , such that  $\mathbf{y} = y_{0:T}$  and  $\pi_T = \pi$ . This is implemented through a procedure of sampling via *importance sampling and resampling* (SISR), which provide appropriate approximations  $\{\hat{\pi}_n(y_{0:n})\}_{n \leq T}$ . A detailed exposition of the background theory can be found in [4]. In the context state estimation or optimal Bayesian filtering for hidden Markov models SMC algorithms have been also known as Particle Filters [7]. In the most general setting SMC extends beyond this popular context [6, 18].

SMC samplers [6] are a generalisation of traditional SMC methods, such as particle filtering [7]. Given a sequence of distributions with  $\{\pi_n\}_{n \geq 0}$  defined on the same or different spaces, the aim is to generate samples  $\{y_n^{(l)}\}_{l=1}^L$  from  $\pi_n$ , constructed based on samples from  $\pi_{n-1}$ . The core idea is to use an artificial auxiliary distribution on a space of increasing dimension, whose density  $\{\bar{\pi}_n\}$  admits the density of interest  $\pi_n$  as its marginal. More specifically let,

$$\bar{\pi}_n(y_{1:n}) = \pi_n(y_n) \prod_{k=n-1}^1 \bar{q}_k(y_k | y_{k+1}),$$

where  $\bar{q}_k$  is an appropriate backward Markov density. Then traditional SMC methods, e.g. SISR, can be used to sample from  $\bar{\pi}_n(y_{1:n})$ . New samples are proposed according to a forward Markov transition density  $q_n$ . These particles are weighted recursively as follows:

$$\frac{w_n(y_{n-1:n})}{w_{n-1}(y_{n-2:n-1})} \propto \frac{\pi_n(y_n) \bar{q}_{n-1}(y_{n-1} | y_n)}{\pi_{n-1}(y_{n-1}) q_n(y_n | y_{n-1})}$$

to provide approximations

$$\hat{\pi}_n(dy) = \sum_{l=1}^L W_n^{(l)} \delta_{y_n^{(l)}}(dy), \text{ with } W_n^{(l)} = \frac{w_n(y_{n-1:n}^{(l)})}{\sum_{l=1}^L w_n(y_{n-1:n}^{(l)})}.$$

Then a resampling procedure is introduced to copy (or multiply) particles with high weights and therefore discarding particles with low weights. The resampling procedure serves to focus the computational effort on the “promising” regions of the state-space. The simplest resampling scheme is multinomial resampling: generate  $L$  independent samples from  $\hat{\pi}_n(y_n)$ . We shall be referring to  $\{y_n^{(l)}, W_n^{(l)}\}_{l=1}^L$  as the particle approximation  $\hat{\pi}_n$  to  $\pi_n$ .

Many sharp convergence results are available for SMC algorithms; under fairly weak assumptions one can obtain almost sure convergence results with respect to the population size  $L$

$$\sum_{l=1}^L W_n^{(l)} \delta_{y_n^{(l)}}(dy) \xrightarrow[L \rightarrow \infty]{a.s.} \pi_n(dy),$$

as well as central limit theorems, see [6] for SMC samplers and [4] for a book length review. For more details on traditional SMC and SMC samplers and we refer to [7, 18] and [6] respectively.

#### 4.2. SMC samplers for optimal policy design

In our case, we set  $y_{1:n} = (\theta, \tau_{1:\gamma_n})$  and  $y_n = (\theta, \boldsymbol{\tau}_n)$  with  $\boldsymbol{\tau}_n = \tau_{\gamma_{n-1}+1:\gamma_n}$ . A possible candidate for  $\pi_n$  can take the form of (19), but we shall further generalise that formulation to include more flexible non integer annealing schedules following the ideas in [14, 15]. Instead of (19) we shall use

$$\pi_n(\theta, \tau_{1:\gamma_n}) = p(\theta) \left[ \prod_{j=1}^{\gamma_n} u(\tau_j) \pi_{T,\theta}(\tau_j) p(\tau_j) \right] u(\tau_{\gamma_n})^{\nu_n}, \quad (20)$$

where  $\{\gamma_n\}_{n=1,\dots,n_{\max}}$  is a monotone increasing integer sequence with  $\gamma_n \in \mathbb{N}_+^*$  and  $\{\nu_n\}_{n=1,\dots,n_{\max}}$  is a real sequence such that  $\nu_n \in [0, 1]$ , with the only restriction being that  $\nu_{n_{\max}} = 0$ . Note that when  $\nu_n = 0$ , the marginal  $\pi_n(\theta)$  remains as before.

To obtain samples for this specific distribution sequence  $\{\pi_n(\theta_n, \boldsymbol{\tau}_n)\}$ , we propose to propagate  $\theta$  at each  $n$  using a random walk transition density for  $q_n(\theta_n|\theta_{n-1})$  and then once  $\theta_n$  is sampled, sample the hitting times from the true dynamics,  $\boldsymbol{\tau}_n \sim \prod_{j=\gamma_{n-1}+1}^{\gamma_n} \pi_{T,\theta_n}(\tau_j)$ . We will also use the backward transition density,

$$\bar{q}_{n-1}(\theta_{n-1}, \boldsymbol{\tau}_{n-1}|\theta_n, \boldsymbol{\tau}_n) \propto q_n(\theta_n|\theta_{n-1}) u(\tau_{\gamma_{n-1}})^{\nu_{n-1}},$$

which serves as an approximation that can resemble closely the optimal backward transition density with respect to the variance of the weights presented in [6]. The weights then can be propagated according to

$$\frac{w_n(\boldsymbol{\tau}_n)}{w_{n-1}(\boldsymbol{\tau}_{n-1})} \propto u(\tau_{\gamma_n})^{\nu_n} \prod_{j=\gamma_{n-1}+1}^{\gamma_n} u(\tau_j).$$

In Algorithm 2 we summarise the proposed SMC algorithm which can be used to maximise  $J_{x_0}(\theta)$  for each aircraft.

#### 4.3. Implementation notes and related approaches

The resampling step adds some variance to the weights in order to keep the particle-set alive, i.e. to prevent all particles having negligible weight apart very few or even a single one. In addition, a key element to the success of any proposed SMC algorithm is that the shape of  $\pi_n$  should be close to that of  $\pi_{n-1}$ . This is reflected in the variance of the importance weights. Therefore it is important to carefully choose  $\bar{q}_n$  and  $q_n$  in order to achieve good mixing properties and keep the variance of the importance weights small. Since the resampling step tends to contribute to an increase in the variance of the weights it has been proposed to use resampling only when it is necessary, i.e. when the variance of the unnormalised weights exceeds some threshold. This can be implemented by monitoring the *effective sample size* (ESS) [18]

$$ESS_n = \left( \sum_{l=1}^L (W_n^l)^2 \right)^{-1}$$

and resampling only if it drops under some pre-specified value, e.g.  $\frac{L}{2}$ . The ESS is always less than  $L$  and can be interpreted as the number of perfect samples from the target distribution that would yield the same estimator variance. Note that we expect that Algorithm 2 will yield lower  $ESS_n$  compared to the ones found in [14, 15] or in other more traditional applications of

**Algorithm 2** Simulation based trajectory planning

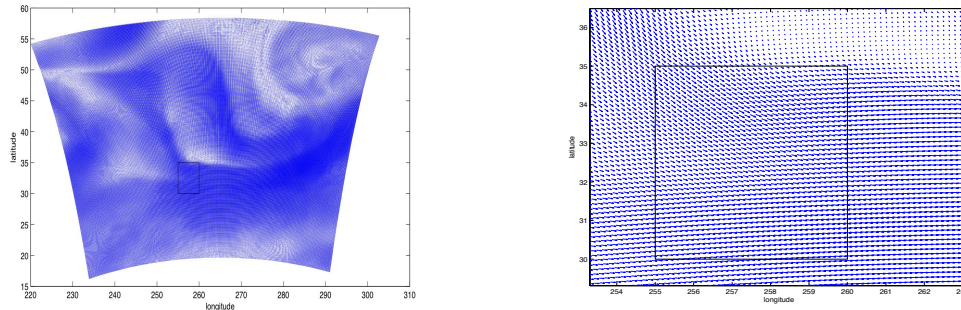
- 
- 1: Initialisation, at  $n = 0$ : for each particle  $l = 1, \dots, L$  : sample from the prior  $\theta_0^{(l)} \sim p(\cdot)$  and assume  $W_{-1}^l = \frac{1}{L}$ .
  - 2: **for**  $n = 0, \dots, n_{\max}$  : **do**
  - 3:     **for** each particle  $l = 1, \dots, L$  : **do**
  - 4:         **if**  $n \geq 1$ , sample control parameter particles  $\theta_n^{(l)} \sim q_n(\cdot | \theta_{n-1}^{(l)})$
  - 5:         **for**  $j = \gamma_{n-1} + 1, \dots, \gamma_n$  using  $\theta_n^{(l)}$  and Algorithm 1, sample  $\tau_j^{(l)}$
  - 6:         Weighting particles: set  $w_n^{(l)} = W_{n-1}^{(l)} u(\tau_{\gamma_n}^{(l)})^{\nu_n} \prod_{j=\gamma_{n-1}+1}^{\gamma_n} u(\tau_j^{(l)})$ ,
  - 7:         **if** Algorithm 1 returns no sample, set  $w_n^{(l)} = 0$ .
  - 8:     **end for**
  - 9:     Normalise weights:  $W_n^{(l)} = w_n^{(l)} / \sum_{l'=1}^L w_n^{(l')}$ .
  - 10:     Resample, if  $\text{ESS}_n < L_{\text{thresh}}$ , to get new particle set  $\{(\theta_n^{(l)}, \tau_n^{(l)})\}_{l=1}^L$  with equal weights  $W_n^{(l)} = \frac{1}{L}$ .
  - 11: **end for**
  - 12: Compute the estimate of optimal  $\theta$
- 

SMC such as nonlinear filtering. This is because the algorithm will be rejecting samples which violate constraints, although hopefully the frequency of the latter should reduce with  $n$ .

A common criticism can be that as in any annealing algorithm, some tuning might be required for the annealing schedule. The tradeoff is clearly between fast convergence and the chance to be trapped in local optima. The non integer annealing schedules  $\{\gamma_n, \nu_n\}_{n=1, \dots, n_{\max}}$  can make tuning slightly easier, and it is also possible to adapt  $\gamma_n, \nu_n$  to the drop in the ESS and use a slower annealing schedule to spend more time exploring areas where the algorithm is prone to get stuck. Similar ideas have been used recently in the context of approximate Bayesian computation in [5].

Finally, as we are interested in estimating the maximisers  $\theta^*$ , using the mean of  $\hat{\pi}_{n_{\max}}(\theta)$  can be a bad estimate in the case of multi-modalities. Therefore, one should be careful to use the mode of  $\hat{\pi}_{n_{\max}}(\theta)$  or the mean of some cluster of  $\left\{ \theta_{n_{\max}}^{(l)} \right\}_{l=1}^L$  with significant weight as estimates  $\theta^*$ .

From the presentation so far it has not been clear what are the contributions in optimal Bayesian design methodology. Our approach is based on [14, 15], but there are a number of novel contributions. Firstly, we address problems with a different utility and more specifically a minimum expected stopping time problem in contrast to Maximum Likelihood or Minimum Entropy problems considered in these papers. A different feature of our problem is that although it is possible to sample from the distribution of interest it might be hard or impossible to evaluate it point-wise, which leads to proposing propagation of  $\theta$  via a random walk. Another important element included in our approach is the explicit handling of constraints, which has not appeared earlier in [14, 15]. Moreover, readers familiar with the stochastic programming literature [23] might find some resemblance to our approach. Nevertheless, we clarify that our approach is completely different as we are approximating the solution of the true problem, i.e. estimating the true maximiser, rather than solving an approximate problem based on sample



(a) Velocity diagram of wind over CONUS. The rectangle is the area where a conflict appears, which we aim to resolve. A magnified plot of this area is plotted in Figure 1(b).

(b) U- and V- components of wind velocity over CONUS at time 20:00, 21 of February 2008. with respect to latitude and longitude.

Figure 1. Velocity diagram of wind velocity of a snapshot of the dataset recorded at time 20:00, 21 of February 2008.

averages. In addition we avoid the restrictive setting of iid sampling and convex problems very common to the literature of stochastic programming.

## 5. Simulations

We begin by discussing the details of the wind field model discussed in Section 2.3. We use RUC2 prediction gridded data for flight level (FL) 300 over the Continental United States (CONUS). The data was a wind velocity forecast for time 20:00, 21 of February 2008, and was downloaded from the NOAA server three hours earlier. In order to perform interpolation with respect to time we used also the same forecast for time 23:00. Each grid consists of 451 longitudinal and 337 points, where each point is separated by 13.545 km from others either horizontally or vertically. In Figure 1(a) we plot a velocity diagram the of the wind velocity for for time 20:00 and in Figure 2 we plot the  $u$  and  $v$  components separately with respect to the longitude and latitude.

We will construct an artificial conflict involving 4 aircraft heading towards each other in the rectangle defined by 255 to 260 degrees in longitude and 30 to 35 degrees in latitude and refer to this area as the area of conflict. The specific part of CONUS was chosen due to the severity of the winds appearing in that area during the time interval considered, which results in a harder conflict resolution problem. Aircraft are initialised in the different corners of the rectangle and are heading diagonally towards the opposite corner. We will assume that all aircraft are cruising at the same constant speed,  $253\text{ms}^{-1}$ , which is a typical value for the specific flight level [8]. In the remainder of this section, we shall present the performance of Algorithm 2 for each of the presented dual mode policies in Section 2.1.2, i.e. when fixed heading or tracking is used for the last block. Algorithm 2 will be implemented using a linear annealing schedule, with  $n_{\max} = 100$ ,  $L = 1000$ ,  $q_n$  a Gaussian kernel with decaying bandwidth,  $\sigma^2 = 5$  and for the policy we set  $H = 4$  and  $p = 3$ .



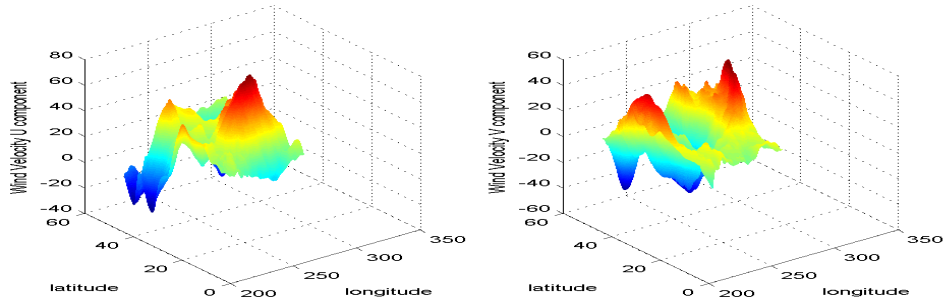


Figure 2. U- and V- components of wind velocity over CONUS at time 20:00, 21 of February 2008. with respect to latitude and longitude.

### 5.1. Fixed heading for mode 2

Here we consider the the case where in the final block the heading consists of the angle to exit after the headings in block  $H - 1$  have been applied. As explained in Section 2.1.2, the control inputs in the first  $H - 1$  blocks depend on  $\theta$ , which is computed using Algorithm 2. The second mode of the policy is block  $H$ , which is independent of  $\theta$ . Nevertheless, it affects the performance of the policy significantly in terms of the closeness to optimality, the statistical efficiency of Algorithms 1 and 2, as well as the sensitivity to constraints and size of target sets. Also the second mode is crucial for the resulting estimates of  $\theta^*$  and the underlying state trajectories, which in turn inherit some characteristic features due to the specific nature of the second mode.

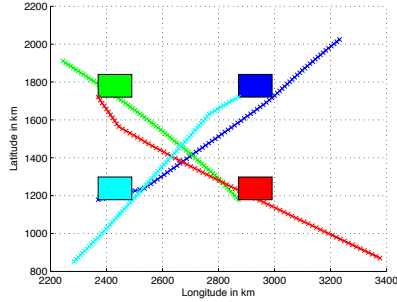
In Figure 3(a) we illustrate the resulting average aircraft trajectories after Algorithm 2 has been executed. One can see that there is a mild form of symmetry in the direction each aircraft turns to. We stress that turning in the same direction is not imposed by the policy or the algorithm, but is a result of the simulation based selection procedure in the algorithm. In Figures 3(b) and 3(c) we plot the approximate cost and the ESS with respect to iteration  $n$ . Finally in Figure 3(d) we plot the minimum, maximum and mean separation of each pair of aircraft, which are computed by using the estimated optimal policy in Algorithm 1 to obtain independent Monte Carlo samples of the aircraft trajectories.

It is clear that when fixed heading is used for the second mode of the policy, separation constraints are easier to handle, at the cost of requiring larger exit zones  $B^i$  to achieve reasonable performance and maintain reasonable sampling efficiency. This is because after the first mode of the policy completes, the aircraft turns towards the exit point, but then drifts away from it due to the unexpected wind.

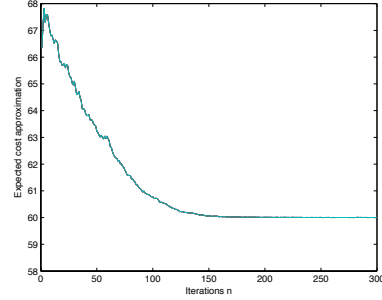
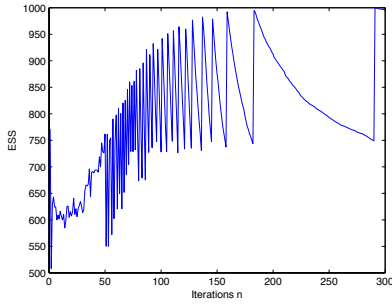
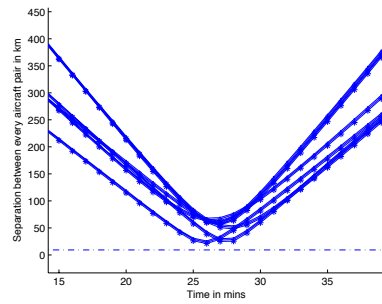
### 5.2. Tracking for mode 2

On the other hand when tracking is used, although the exit zones can be made arbitrarily small, the resulting trajectories will be closer to each other and the policy seems to be more sensitive to constraints. This was confirmed by our initial simulations. So that a fair comparison can be made when Algorithm 2 is used, we used a gradual smooth constraint-tightening approach. We used a varying minimum separation for each iteration  $n$ , which was





(a) Trajectories of resulting manoeuvres.

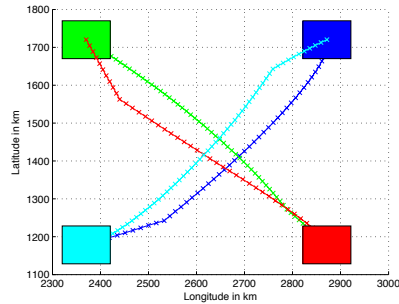
(b) Approximate expected cost in minutes with iteration  $n$ .(c) Effective sample size with iteration  $n$ .

(d) Separation for every pair of aircraft with time.

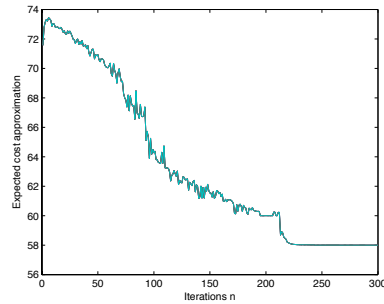
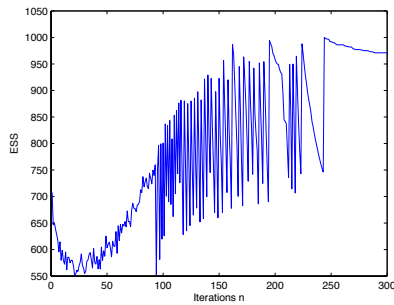
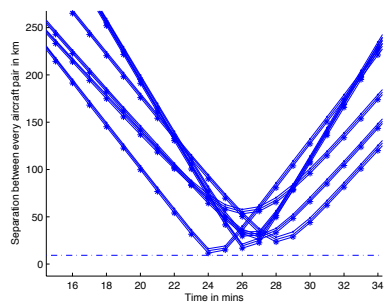
Figure 3. Performance of Algorithm 2 when fixed heading is used in mode 2: a common colour indexing is used to distinguish between each aircraft  $i$ . Simulations lasted 109 minutes when implemented in Matlab. (a) We plot the mean trajectories of the aircraft with dot connected lines of different colour. Each rectangle depicts  $B^i$  and is filled with the colour of  $i$ . (b) We plot the approximate expected cost,  $\hat{J}_{x_0} = \sum_{l=1}^L W_n^{(l)} \tau_n^{(l)}$ , for each aircraft  $i$  with respect to iteration  $n$ . (c) We plot the ESS for each aircraft  $i$  with respect to  $n$ . (d) To assess the computed estimate of  $\theta^*$  we use it to obtain 5000 independent samples of the trajectories using Algorithm 1. The plot shows only the time interval when separation is critical and close to the minimum safe value of 9.260 km, which is shown as a dotted horizontal line. We plot the maximum, minimum and mean separation observed for every pair of aircraft with respect to time. The line connecting dots and stars shows the maximum and minimum separations respectively and the solid line is the mean observed separation.

given by  $d_n = d/(0.8 + n^{-0.5})$ . In fact,  $d_{n_{\max}} < d$ , but the values are quite close. This allows comparisons to be made on a fair level, as the target zones for each aircraft used in our simulations were better suited for the fixed heading case.

In Figure 4 we show the same plots as in Figure 3 for the same problem, when tracking is used in the second mode of the policy together with gradual tightening of the constraints. We observe that the approximate expected costs this time are marginally lower but similar compared to the fixed heading case. As regards to the sampling efficiency monitored by the ESS, both policies seem to perform similarly.



(a) Trajectories of resulting manoeuvres.

(b) Approximate expected cost in minutes with iteration  $n$ .(c) Effective sample size with iteration  $n$ .

(d) Separation for every pair of aircraft with time.

Figure 4. Performance of Algorithm 2 when tracking is used in mode 2: for more details we refer to the caption of Figure 3. Simulations lasted 89 minutes when implemented in Matlab.

## 6. Conclusions

In this paper we considered the ATM problem of conflict resolution for level flight, when constant speed is used and only changing the heading of the aircraft is allowed at each time. We proposed a conveniently parameterised dual mode policy to automatically design manoeuvres using simulation, so that each aircraft reaches a target in the expected minimum time of arrival in the presence of stochastic wind disturbances modelled by a realistic model derived from actual forecast data. We showed how effective our approach is by appropriate simulations. As regards the resolution properties, an implicit feature of the presented policy is that each aircraft is forced earlier rather than later to act preemptively by executing manoeuvres in case a conflict arises. If one takes into account that the actual execution times of our implementations can be drastically reduced by at least one order of magnitude using parallel implementation GPUs [16], this makes our proposed approach suitable for mid/short term conflict resolution with a horizon of the order of tens of minutes.

## ACKNOWLEDGEMENTS

The authors are grateful to the anonymous reviewers for their suggestions to improve the quality and presentation of the paper.

## REFERENCES

1. Amzal B., Bois F.Y., Parent E., Robert C.P. (2006). Bayesian-Optimal Design via Interacting Particle systems. *Journal of the Am. Stat. Association*, Vol. 101, No. 474.
2. Chaloulos G. and Lygeros J. (2007) Effect of Wind Correlation on Aircraft Conflict Probability, *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1742-1752.
3. Cole, R. E., Richard, C. M., Kim, S. K., Bailey, D. B. (1998) An Assessment of the 60 km Rapid Update Cycle (RUC) with Near Real-Time Aircraft Reports, Project Report NASA/A-1, MIT Lincoln Laboratory, Lexington, MA.
4. Del Moral P. (2004) *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. New York: Springer Verlag.
5. Del Moral P., Doucet A., Jasra A. (2009) An Adaptive Sequential Monte Carlo Method for Approximate Bayesian Computation, preprint.
6. Del Moral P., Doucet A., Jasra A. (2006) Sequential Monte Carlo Samplers, *J. Royal Statist. Soc. B*, vol. 68, no. 3, pp. 411-436.
7. Doucet A., de Freitas N. and Gordon N.J. (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.
8. Eurocontrol Experimental Centre (2009) User manual for the Base of Aircraft Data (BADA) Revision 3.7, EEC Technical/Scientific Report No. 2009-003.
9. Federal Aviation Administration (2008) *Radar Vectoring, Pilot/Controller Glossary*.
10. Glover W. and Lygeros J. (2004) A multi-aircraft model for conflict detection and resolution algorithm evaluation, HYBRIDGE Deliverable D1.3 (EC IST-2001-32460).
11. Gyrfas B. and Dunbar W. B. (2007) Time Optimal Flight Paths in a General Wind Field with Convective Weather Hazards, submitted preprint, UCSC, 2007.
12. Huang Z. and Chalabi Z.S. (1995) Use of time series analysis to model and forecast wind speed data. *Journal of Wind Engineering and Industrial Aerodynamics*, vol 56, pp 311-322.
13. Hwang, C.R.(1980) Laplace's method revisited: weak convergence of probability measures. *Ann. Prob.* 8(6), 1177-1182.
14. Johansen A. M., Doucet A., Davy M. (2008). Particle methods for Maximum Likelihood Parameter Estimation in Latent Variable Models, *Statistics and Computing*, 18(1):47-57, 2008.
15. Kueck H., de Freitas N. and Doucet A. (2006) SMC Samplers for Bayesian Optimal Nonlinear Design. *In Proc. Nonlinear Statistical Signal Processing Workshop*, Cambridge, 2006.
16. Lee A., Yau C., Giles M., Doucet A. and Holmes C. (2009) On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. submitted.
17. Lecchini Visintini A., Glover W., Lygeros J., Maciejowski J.M. (2006) . Monte Carlo optimization for conflict resolution in air traffic control. *IEEE Trans. Int. Transp. Systems*, vol. 7, no. 4, pp 470-482.
18. Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. New York: Springer-Verlag.
19. Lymperopoulos I., Lygeros, (2010) Sequential monte carlo methods for multi-aircraft trajectory prediction in air traffic management, *this issue*.
20. Michalska H. and Mayne D.Q. (1993) Robust receding horizon control of constrained nonlinear systems, *IEEE Trans. Aut. Control*, vol. 38:11, pp 1623-1633
21. Muller P., Sanso B. G., De Iorio M. (2004). Optimal Bayesian design by Inhomogeneous Markov Chain Simulation. *Journal of the Am. Stat. Association*, pp. 788-798.
22. Robert, C. P. and Casella, G. (2004) *Monte Carlo Statistical Methods*. 2nd Ed., Springer.
23. Shapiro A., Dentcheva D. and Ruszczyński A. (2009) *Lectures on Stochastic Programming: Modelling and Theory*, SIAM, Philadelphia.
24. Stroud, J., Mueller, P., and Sanso, B. (2001). Dynamic Models For Spatio-Temporal Data. *Journal of the Royal Statistical Society, Series B*, 63, 673-689.