HYBRIDGE

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

WP3: Reachability analysis for probabilistic hybrid systems

Probabilistic Aircraft Conflict Detection

Maria Prandini[*] and Oliver J. Watkins[**]

30th May 2005

Version:	*1.2*
Task number:	*3.2*
Deliverable number:	*D3.2*
Contract:	*IST-2001-32460 of European Commission*

[*] University of Brescia, Italy
[**] University of Cambridge, U.K.

DOCUMENT CONTROL SHEET

Title of document:	*Probabilistic Aircraft Conflict Detection*
Authors of document:	*M. Prandini and O.J. Watkins*
Deliverable number:	*D3.2*
Contract:	*IST-2001-32460 of European Commission*
Project:	*Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design (HYBRIDGE)*

DOCUMENT CHANGE LOG

Version #	Issue Date	Sections affected	Relevant information
0.1	20/02/2005	All	1st draft
0.2	13/03/2005	All	2nd draft
0.3	20/03/2005	All	Revised based on Reviewers' comments
1.0	27/04/05	All	Revised based on Reviewers' comments
1.1	17/05/05	All	Revised based on Reviewers' comments
1.2	30/05/05	All	

Version 1.1		Organisation	Signature/Date
Authors	Maria Prandini	UniBs	
	Oliver Watkins	UCAM	
Internal reviewers	M.C. Campi	UniBs	
	Henk Blom	NLR	
External reviewers			

HYBRIDGE, IST-2001-32460
Work Package WP3, Deliverable D3.2

Probabilistic Aircraft Conflict Detection

Prepared by:
Maria Prandini[*] and Oliver J. Watkins[†]

Abstract

In this deliverable we study algorithms for mid-term aircraft conflict detection. A two-aircraft conflict occurs when some prescribed separation distance between two aircraft is violated. In a mid-term time scale, the contribution of the different sources of uncertainty affecting the aircraft motion cannot be neglected when predicting the aircraft future positions. Based on this consideration, we address mid-term aircraft conflict detection according to a probabilistic viewpoint. We consider, in particular, the probability of conflict as criticality measure. Computing the probability of conflict is generally a difficult task. A possible solution to this issue is the simulation-based approach, characterized by repeated simulations of the aircraft trajectories according to the model describing their motion: the fraction of trajectories that generate a conflict is an unbiased estimate of the probability of conflict. This approach has the advantage of being applicable to general models and general contexts. Obviously, the more complex the model, the more time-consuming is generating simulations according to it. Given the time constraints of our specific application context, we look into solutions to speed-up the simulation-based method. These solutions are inspired by the multi-level approach to rare-event probability estimation and the sequential Monte Carlo approach.

[*] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, prandini@elet.polimi.it

[†] Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, U.K., ojw26@eng.cam.ac.uk.

Contents

List of Figures

1 Introduction

This is the second deliverable under workpackage 3 of the HYBRIDGE project, dealing with aircraft conflict detection.

The primary concern of all advanced air traffic management systems (ATMs) is to guarantee safety of air travel. In particular, one of the main objectives in ATM is ensuring appropriate separation between flight trajectories. Examples of separation criteria are the minimum *vertical separation* between flight levels and the minimum *horizontal separation* between two aircraft that have been assigned the same flight level. Currently, for en-route airspace the minimum horizontal separation is 5 nautical miles (nmi), while inside the Terminal Radar Approach Control (TRACON) area it is reduced to 3 nmi. The minimum vertical separation is 1000 feet (ft). A conflict occurs when two aircraft violate these separation criteria.

To prevent conflicts, ATMS resorts to a two stage process. In the first stage *conflict detection* is performed: the positions of the aircraft in the future are predicted based on their current positions and flight plans, and they are compared so as to detect potential situations of conflict. Once a potential conflict has been detected, the trajectories of the aircraft involved in the conflict are re-planned in the *conflict resolution* stage.

Conflict detection and resolution is actually given consideration at three different levels of the ATM process, which can be classified according to the considered look-ahead time horizon as:

i) long term conflict detection and resolution, where flight plans and airline schedules are composed (on a daily basis, for example) to ensure that airport and sector capacities are not exceeded ([1, 2, 3]);

ii) mid-term conflict detection and resolution, carried out by Air Traffic Controllers (ATCs), over horizons of the order of tens of minutes. It involves modifying the pre-planned flight plan on-line to ensure adequate aircraft separation.

iii) short term conflict prediction and resolution over horizons of seconds to minutes, carried out both on the ground at the ATC level (Short Term Conflict Alert System - STCA), and on board the aircraft at the Flight Management System (FMS) level (Traffic Alert and Collision Avoidance System - TCAS, [4, 5]).

Here we consider mid-term aircraft conflict detection.

Mid-range conflict detection involves challenging issues, such as predicting the aircraft positions along the look-ahead time horizon, predicting a conflict, issuing an alert to ATCs sufficiently in advance so that effective remedial actions may be taken, as opposed to emergency actions.

At the ATC level, the information available for predicting the aircraft future positions are the radar measurements on the aircraft current positions and their flight plans. Some information are also available regarding weather and wind perturbations.

One of the difficulties in predicting the aircraft future position consists in modeling the perturbations influencing its motion. The actual motion of the aircraft is in fact affected by uncertainty, due mainly to wind, but also to errors in tracking, navigation, and control. In mid-term time-horizons, the different sources of uncertainty affecting the aircraft motion generally cause deviations from the nominal flight path that cannot be neglected when performing conflict detection. In [6, 7, 8], a description of the global effect of the perturbations affecting the aircraft motion over a 20 minutes time horizon is given in probabilistic terms, by characterizing the resulting tracking error at each time instant along the look-ahead time horizon as a Gaussian random variable with time varying variance. This model was indeed verified by air traffic data in [9, 10]

In the probabilistic approach to conflict detection, the uncertainty affecting the aircraft motion is taken into account by considering the ensemble of sample paths and computing the probability of projected conflicts. Conflict assessment based on this criticality measure, which weights the admissible aircraft trajectories according to their likelihood, avoids the conservativeness of the worst-case approach where the admissible aircraft trajectories are all considered equally likely.

The use of realistic models for predicting the aircraft positions makes the problem of computing the probability of conflict impossible to solve analytically. Various methods have been proposed to address the issue of conflict probability estimation. Here, we investigate the simulation-based approach. Simulation-based methods are characterized by repeated simulations of the two-aircraft trajectories according to the model describing their motion: the fraction of trajectories that generate a conflict is an unbiased estimate of the probability of conflict. This approach has the advantage of being applicable to general models and general contexts. However, the more complex the model, the more time-consuming is generating simulations according to it. Given the time constraints of our specific application, we look into solutions to speed-up the simulation-based method. These solutions are inspired by the multi-level approach for rare-event probability estimation and the particle filtering approach. In both cases, the computational burden is reduced by decreasing the number of simulations needed for achieving certain accuracy and confidence levels with respect to the standard simulation-based approach.

Starting from the observation that the larger is a probability, the lower is the number of simulations needed to estimate it with a certain accuracy, the idea of the multi-level approach for probability estimation is to decompose the probability of interest into the product of a certain number of (larger) probabilities. The estimate of the probability of interest can be obtained as the product of the estimates of the component probabilities.

The probability of conflict has to be updated at each time instant when new radar measurements on the aircraft positions become available. According to the standard simulation-based approach, this would require simulating new trajectories starting from the newly updated initial positions. The idea of the particle filtering-based approach is to update the probability of conflict based on the re-use of the trajectories previously simulated after appropriately weighting their contribution according to their likelihood, suitably refreshed based the new radar measurements. This leads to the so-called Sequential Monte Carlo (SMC) method.

Based on the value of the probability of conflict, a decision should be taken on whether alerting air traffic controllers and pilots of forthcoming conflict situations or not. A threshold-based alerting logic is typically used where as soon as a certain value is overcome then an alert is issued. This value is typically chosen so as to compromise between the number of successful alerts and missed detections, without taking the geometry of the encounter into account. In this work we point out the dependence of the ‘optimal’ conflict alert threshold from the airspace configuration.

The deliverable is structured as follows. We start by presenting a comparative study of probabilistic methods for conflict detection (Section 2). We then review basic notions on the simulation-based approach to probability estimation (Section 3.1), and propose improvements of the computational efficiency of the standard simulation-based method through the multi-level decomposition approach (Section 3.2) and the sequential Monte Carlo method (Section 3.3).

The application of Sequential Monte Carlo methods to aircraft conflict detection is presented in the final section 4. In this section, the sequential Monte Carlo method is compared with other methods in the literature for probabilistic conflict detection.

2 Probabilistic methods for aircraft conflict detection

A comprehensive overview of the different conflict detection and resolution schemes presented in the literature is given in the survey paper [11], the main classification criterion being the modeling method used for projecting the aircraft position in the future. Here we restrict our attention to probabilistic conflict detection, hence the approaches are classified with a greater detail level than in [11].

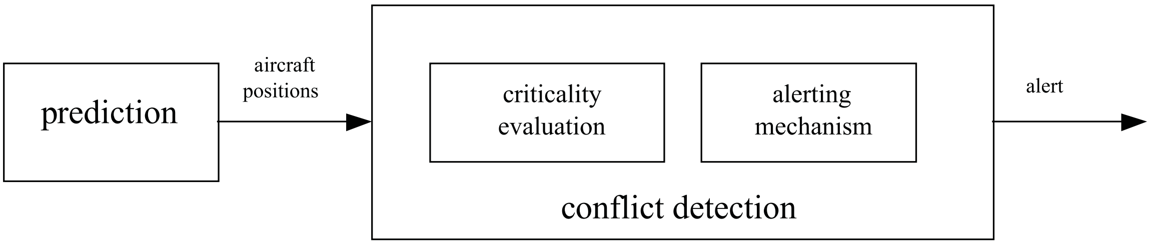

Figure 1: Conflict detection process

A conflict detection scheme is characterized by the following blocks (see figure 1): i) a prediction component that projects the aircraft positions in the look-ahead time horizon, and ii) a conflict detection component that computes a criticality measure to assess how unsafe is a certain configuration of the airspace and that decides based on it whether issuing an alert to ATCs or not.

We next describe different approaches for the different tasks involved in the implementation of these blocks. This will lead to the classification of probabilistic conflict detection methods in Table 1.

2.1 Classification criteria

2.1.1 Prediction of the aircraft position

Prediction of the aircraft future position is performed using radar measurements of the aircraft current position and a prediction model, which incorporates information on the aircraft flight plan and the characteristics of the disturbances affecting the aircraft motion. The prediction model is one of the characteristics by which the most high-level distinctions are made; along with the measure of criticality this tends to be one of the elements by which authors distinguish their work.

The prediction model is normally composed of three elements:

1. **Flight Plan.** The flight plan is typically given in terms of a sequence of waypoints and speeds between waypoints. We distinguish between conflict detection methods using

    - Full flight plan [**FP**].
    - Straight line flight plan [**SL**].

Although a complete flight plan is undoubtedly more accurate, the detection horizon tends to be of the order of 20 minutes, during which only one or even no waypoints may be passed. This motivates the use of a straight line flight plans, which represent subsections of complete flight plans.

2. **Model of the aircraft motion.** There is great scope for variation in the complexity of the model of aircraft flight. Most models will fit at points along a continuum between the following ones:

   - Basic [**B**]: The aircraft proceeds along a known straight line flight path at known velocity.
   - Kinematic [**K**]: Inclusion of basic kinematic equations of aircraft flight enables control of the aircraft and tracking of reference trajectories given disturbances.
   - Dynamic [**D**]: Dynamic models typically include all factors with a significant effect on aircraft trajectory. The BADA model [12], for instance takes account of the dynamics of individual aircraft, their changing mass, thrust, lift, atmospheric conditions and flight phase (procedural information).

   It is worth noting that while a “Basic” model cannot capture complex behavior, in the majority of cases it adequately replicates the straight and level nominal flight of most models.

3. **Uncertainty.** The uncertainty may directly enter the equations governing the aircraft motion or being superimposed to the nominal aircraft motion. We then distinguish between:

   - Stochastic difference/differential equation [**SDE**]: The uncertainty is modeled via some disturbance entering the equations describing the aircraft motion. The disturbance may be either independent [**SDE,I**] or correlated in either space or time (or both) [**SDE,C**].
   - Superimposed stochastic component [**SSC**]: Uncertainty is modeled by superimposing onto the deterministic trajectory a stochastic component.

2.1.2 Conflict detection

1. **Criticality measure.** The criticality measure is one of key elements in characterizing a conflict detection method. Most methods focus on conflicts between aircraft pairs. The most well known measures in the literature for assessing the criticality of a two-aircraft encounter are:

   - Overall probability of conflict [**PC**]: This is the probability at a conflict will occur during the look-ahead time horizon.
   - Maximum probability of conflict [**max-PC**]: This is the maximum value over the look-ahead time horizon of the instantaneous probability of conflict.

- Collision probability [**CP**]: This is the probability at a collision will occur during the look-ahead time horizon.
- Incrossing probability [**IC**]: An incrossing is defined as the event of one aircraft entering the exclusion zone of another. The incrossing probability depends on the incrossing risk, which is the integral over the prediction horizon of the incrossing rate. Incrossing probability is used to evaluate collisions rather than conflicts.

2. **Computation Method.** It is generally difficult to compute a criticality measure exactly, especially in the case of complex prediction models. Different approximation methods have been considered in the literature. The choice of the approximation method is strongly dependent on the adopted criticality measure and prediction model. For instance, calculating the overall conflict probability for complex models may only be practical through Monte Carlo simulation.

   - Analytical approximation [**AA**]: Analytic –though approximate– solutions to the problem of computing the criticality measure are adopted. The drawback is that, when the assumptions made to enable an analytical solution will break down under certain conditions, then conflict detection will become unreliable.
   - Randomized algorithms [**RA**]: The max-PC measure of conflict probability is estimated using stochastic methods. A randomized estimation is performed for computing the instantaneous probability of conflict and for maximizing it. This is less efficient than an analytical approximation but typically straightforward.
   - Gridding methods [**GM**]: Grid-based methods are characterized by a discretization both in time and in space of the stochastic processes describing the aircraft motions. Grid-based methods are computationally intensive. On the other hand, the outcome of the grid-based algorithms is a map that associates to each admissible aircraft position pairs the corresponding estimate of the criticality measure, which could be used not only for conflict detection, but also for designing feedback control for conflict avoidance.
   - Monte Carlo methods [**MC**]: Monte Carlo methods are characterized by repeated simulations of the two-aircraft trajectories according to the model describing their motion to estimate the criticality measure of interest by an averaging procedure. They do have the advantage that sophisticated models for the aircraft motion may be used. They are generally less computationally intensive than gridding methods.

3. **Alerting Mechanism.** The alerting mechanism describes the method by which a criticality measure is converted into an alert for ATC. It is typically a threshold-based mechanism where as soon as the criticality measure overcomes a certain threshold value, an alert is issued. This can be combined with an hysteresis mechanism, such as in the sequential probability ratio test. Multiple thresholds can be adopted to issue alerts of different severity level, eventually combined with a hysteresis mechanism.

   An alerting mechanism is an extra layer of functionality, and some authors do not include details of such a mechanism, but conclude with the calculation of the criticality measure.

The air traffic controller is faced with a dynamical decision problem under uncertainty. This problem can be directly formulated as a stochastic control problem where the criticality measure is embedded in the cost function to optimize and the appropriate timing of the resolution action is automatically selected. This is, for instance, the point of view in [13]. In [14], the criticality measure is used for continuously adjusting the aircraft trajectories according to a decentralized control scheme, inspired by the potential field method.

We distinguish among the following alerting mechanisms:

- Single threshold [**T**].
- Single threshold with hysteresis [**H**].
- Multiple thresholds [**MT**], eventually with hysteresis [**MH**].

2.1.3 Performance assessment

1. **Validation.** Validation data are critical for evaluating the performance of a conflict detection scheme. Genuine aircraft track data is the most appealing in performing validation, however, being proprietary to the air traffic service providers, it is not easy to obtain. Other evaluation methods are therefore used in most areas of the literature. We distinguish two cases: validation data are generated by the same model as used to perform conflict prediction (the prediction model), and validation data are generated by the a more complex model than the prediction model. It is worth noting at this point that 'validation' is a contentious term in the ATC community as it implies testing in a real ATC setting. We therefore use the term 'Complementary Model Evaluation'. The following methods have been used to generate data suitable for complementary model evaluation.

    - Track data [**TD**]: A significant drawback of using real track data is that it will contain either very few or, more likely, no conflicts; as conflict data is both rare and sensitive. Validation or tuning of a conflict probe then becomes impossible. The validation performed in [15] is achieved by translating different tracks into potentially conflicting positions, then testing the probe. While appealing, this method destroys potential information on spatial and temporal correlation of wind disturbances.
    - Erzberger-Paielli model [**EP**]: The Erzberger-Paielli model [6] is a simple kinematic model which incorporates the most pertinent features of aircraft dynamics, and as such is a good first step in generating synthetic trajectory information. The drawback of this model is that disturbances are not explicitly included.
    - SDEs replicating EP statistics [**SDE**]: A variety of these have been proposed ([14, 16, 17]) to generate stochastic synthetic track data. Although Erzberger and Paielli's seminal flight deviation statistics [15] are of great use they do not give sufficient information for the development of a dynamic model. A question mark therefore hangs over all SDE models, in that there is an unlimited number of models which can be formulated to respect EP statistics, so some other insight is required to generate trajectories.

- Detailed models [**DM**]: Few detailed models for conflict detection validation have been proposed. The BADA model [12] is an excellent model of dynamics in the vertical plane although takes little account of turning maneuvers and no account of disturbances. Models in [18, 19] make the relatively simple extension to three dimensions. The concept of wind correlation in space and time is introduced in [20] and [19, 21], and appears to apply time and space correlated disturbances successfully, while respecting EP statistics. Although complex and therefore somewhat computationally intensive, these models appear to generate valuable synthetic track data.

Alternatively, one can adopt the following methodology for validation:

- Bias and uncertainty assessment [**B&U**]: The model used for predicting the aircraft position tries to cover the significant aspects of the reality that are believed to be relevant for conflict detection. In bias and uncertainty assessment, the effect of model assumptions on the estimated risk measure is evaluated, providing an expected ("realistic") value with a corresponding credibility assessment ([22]).

2. **Performance measure.** All methods rely on some off-line tuning of the alerting threshold, to achieve an optimal trade off between false and missed alerts. Other cost measures can be introduced to quantify the alerting system performance. In those methods using a simulation-based approximation, the number of simulations used is tuned to trade off the accuracy against the computation time. Another parameter that may be tuned is the prediction horizon.

   The following performance measures have been either explored or had their value highlighted in the literature:

   - SOC curves [**SOC**]: Introduced in [23], SOC curves have gained a substantial following as a performance measure (see e.g. [14, 17, 24]). The SOC curve plots the probability of false alert against the probability of successful alert, parameterized by alert threshold. The closer the approach of the curve to the point $(0, 1)$ (the ideal situation of zero probability of false alert, certainty of successful alert), the better the performance of the conflict probe.
   - Sensitivity to encounter geometry [**SEG**]: This topic is considered only in a few contributions. In a real ATC setting conflict geometries of all types will be encountered, and the ability of the alerting mechanism to successfully detect conflicts across this range is crucial to its performance.
   - Cost function [**CF**]: The problem of conflict avoidance can be directly formulated as an optimal control problem with the cost function to be optimized incorporating the safety requirements. In this case, conflict detection and resolution are accomplished at the same time, and performance can be measured in terms of the value taken by the cost function.
   - Computational intensity [**CI**]: Many conflict detection schemes are designed with efficiency in mind. Even though here must be a trade off between accuracy and prohibitively high computation times, this is a difficult measure to quantify due to the continuing leaps in available computing power.

- Controller acceptance [**CA**]: In the near and mid-term ATC situation, air traffic controllers will be responsible for the implementation of conflict probe output to air traffic situations. Any conflict detection method therefore must be acceptable to air traffic controllers. This topic has been considered in a number of human factors studies ([25]).

It is worth noticing these performance measures can be jointly used to quantify the alerting system performance.

2.2 Classification of probabilistic methods

Table 1 shows the classification of the methods for probabilistic conflict detection most prominent in the literature. Several of these methods do not attempt to address all aspects of conflict detection, but concentrate on one or two particular elements. These models will therefore have several fields with a '–' entry. This is no reflection on the results of the particular paper cited, simply the scope of the work within. Each entry contains either a single bibliography reference or multiple references when they belong to the same research stream. The acronyms appearing in Table 1 have been defined in the previous subsection.

	Prediction			**Conflict detection**			**Performance**	
	Flight plan	**Aircraft motion**	**Uncertainty**	**Criticality measure**	**Comput. method**	**Alert. mech.**	**Valid. data**	**Perform. measure**
[6]	SL	K	SSC	PC	AA	–	SDE,C	–
[24]	FP	B	SDE,I	PC	MC	T	SDE,I	SOC
[26]	SL	K	SSC	–	AA	–	–	–
[16]	SL	B	SDE,I	PC	AA	–	PE	–
[14]	FP	B	SSC	maxPC	RA	T	SDE,I	SOC
[27]	SL	B	SSC	PC	AA	–	–	–
[28]	FP	K	SDE,C	maxPC	RA	–	–	–
[29]	FP	D	SDE,C	PC	MC	–	PE	–
[17]	SL	B	SDE,I	PC	MC	T	SDE,I	SOC
[20, 30, 31]	FP	K	SDE,C	PC	GM	–	–	–
[32, 33, 34, 35]	FP	K	SDE,C	IC,CP	AA,MC	–	B&U	SEG,CI,CA

Table 1: Classification of probabilistic conflict detection methods in the literature.

3 Simulation-based approach

The problem of estimating the probability of some event can be reformulated as that of estimating the mean of the indicator function of the event of interest, which is a binary random variable equal to 1 when the event is verified, 0 otherwise. In turn, the mean of this random variable can be estimated according to the Monte Carlo method, that is by extracting multiple samples and taking the average of the outcomes of the extractions. By appropriately selecting the number of extractions, one can ensure a desired accuracy of the estimate with a probability arbitrarily close to 1.

In order to estimate small probabilities with adequate accuracy, a large number of extractions is needed. This is particularly evident in the extreme case of rare events: only by extracting many samples, one can get a nonzero estimate of the probability of a rare event.

In the multi-level approach to rare event probability estimation, the number of extractions is reduced by considering a set of nested events containing the event of interest so that its probability can be expressed as the product of the conditional probabilities of the nested events. These conditional probabilities are larger than the probability of interest, thus their estimate requires a smaller number of extractions, which results in an improvement of the computational efficiency for the overall probability estimation.

A stochastic reachability problem consists in determining the probability that the trajectories of a given stochastic process enters a target set starting from some initial state. The event of interest is related to the evolution of the stochastic process, and computing its probability is generally a difficult problem to solve analytically. The Monte Carlo approach turns out to be a convenient solution to this issue. In this case, extracting a sample from the indicator function of the event translates into running a simulation of the process and verifying if the resulting trajectory enters the target set. For this reason, we refer to the Monte Carlo approach for reachability analysis as "simulation-based approach". In a dynamically evolving setting, as soon as updated information on the current process position is available, the estimation of the probability should be updated. According to the standard Monte Carlo approach, new simulations should be run. Inspired by the particle filtering method, in the sequential Monte Carlo approach simulations are re-used for updating the probability estimate thus improving computational efficiency.

In our ATM application context, the event of interest is that of two aircraft coming closer than a minimum allowed distance along a 20 minutes horizon, given their flight plans and current positions made available by radar measurements. The two aircraft motions are described through a pair of coupled stochastic difference/differential equations. An extraction from the indicator function of the event of interest can be obtained by generating an instance of the two aircraft trajectories: if the generated trajectories get closer than the prescribed separation minima, then a conflict occurs and the outcome of the random extraction is 1, otherwise no conflict occurs and the outcome is 0. As soon as a new radar measurement becomes available, the estimate of the probability of conflict should be updated. The use of both the multi-level and the particle-filtering approaches to improve the computational efficiency of the simulation-based approach is investigated.

We start by recalling some fundamental notions on the Monte Carlo method for estimating probabilities. We then describe the multi-level decomposition and the sequential Monte Carlo methods. We study in particular the finite sample properties of these methods.

3.1 Monte Carlo method

We recall some basic concepts on the Monte Carlo method for mean estimation. We then apply these concepts to the problem of probability estimation. The application of the Monte Carlo method to stochastic reachability analysis is also briefly discussed.

3.1.1 Mean estimation

Let X be a random variable defined on some probability space $(\Omega, \mathcal{F}, P)$. Denote by μ_X its mean.

The empirical average estimator of μ_X based on n independent extractions from X is given by

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X^{(i)},$$

where $X^{(i)}$, $i = 1, 2, \ldots, n$, are independent random variables identically distributed as X. $\overline{X}_n$ is a random variable defined over the product probability space $(\Omega_n, \mathcal{F}_n, P_n)$, where $\Omega_n := \Omega \times \Omega \times \ldots \times \Omega$, $\mathcal{F}_n := \mathcal{F} \otimes \mathcal{F} \otimes \ldots \otimes \mathcal{F}$, and $P_n := P \times P \times \ldots \times P$, ($n$ times). To assess the asymptotic properties of the estimator as $n \to \infty$, we shall refer to the infinite product probability space denoted as $(\bar{\Omega}, \bar{\mathcal{F}}, \bar{P})$.

Let $\bar{E}$ be the expectation operator with respect to the probability $\bar{P}$. For any integer n the empirical average estimator is unbiased, i.e., its expected value is equal to the quantity to be estimated:

$$\mu_{\overline{X}_n} = \bar{E}[\overline{X}_n] = \bar{E}\left[\frac{1}{n} \sum_{i=1}^{n} X^{(i)}\right] = \frac{1}{n} \sum_{i=1}^{n} \bar{E}[X^{(i)}] = \frac{1}{n} \sum_{i=1}^{n} \mu_X = \mu_X.$$

The estimate obtained for a given realization of the random variables $X^{(i)}$, $i = 1, 2, \ldots, n$, is generally different from μ_X, and depends on the realization. One would expect that, as n grows to infinity, the estimate will get closer to μ_X, for any realization. This is formalized in the following notion of consistency.

Definition 1 *The estimator $\overline{X}_n$ of μ_X is consistent (in probability) if*

$$\lim_{n \to \infty} \bar{P}(|\overline{X}_n - \mu_X| > \epsilon) = 0.$$

Since by Chebyshev inequality

$$\bar{P}(|\overline{X}_n - \mu_X| > \epsilon|) \leq \frac{\bar{E}[(\overline{X}_n - \mu_X)^2]}{\epsilon^2},$$

and

$$\bar{E}[(\overline{X}_n - \mu_X)^2] = \bar{E}\left[\Big(\frac{1}{n}\sum_{i=1}^{n} X^{(i)} - \mu_X\Big)^2\right] = \frac{1}{n^2}\sum_{i=1}^{n}\bar{E}[(X^{(i)} - \mu_X)^2] = \frac{1}{n}\bar{E}[(X - \mu_X)^2],$$

it follows that, if the variance of X is bounded, then the empirical average estimator is consistent.

To decide how many extractions are needed to adequately estimate μ_X, it is important to evaluate the quality of the estimate for finite n (finite sample properties).

The estimate of μ_X depends on the value taken by $X^{(1)}, X^{(2)}, \dots, X^{(n)}$. Even if n is large, there are realizations of $X^{(1)}, X^{(2)}, \dots, X^{(n)}$ such that the corresponding estimate is not accurate. This means that the quality of the estimate can only be assessed in probabilistic terms. If bad samples may occur, but with small probability, then, the quality of the estimate is good.
The quality of the estimator is quantified in probabilistic terms through two parameters: the accuracy parameter ϵ and the confidence parameter δ.

Definition 2 *The average mean estimator $\overline{X}_n$ is characterized by an accuracy $\epsilon > 0$ with confidence $1 - \delta \in (0,1)$ if $\bar{P}(|\overline{X}_n - \mu_X| \leq \epsilon) > 1 - \delta$.* □

Note that in this definition ϵ is an absolute accuracy parameter. One can also assess the accuracy in terms of a relative accuracy parameter α:

$$\bar{P}\left(\left|\frac{\overline{X}_n - \mu_X}{\mu_X}\right| \leq \alpha\right) > 1 - \delta.$$

We now consider the problem of determining the value of n that guarantees a certain accuracy and confidence level for the average mean estimator.

Hoeffding's inequality is fundamental in this respect. We recall it here according to our notations.

Theorem 1 (Hoeffding's inequality) *Suppose that X takes values in some bounded interval $[a,b]$. Then,*

$$\bar{P}(|\overline{X}_n - \mu_X| > \epsilon) < 2e^{-\frac{2n\epsilon^2}{(b-a)^2}}.$$

□

Fix $\epsilon > 0$ and $\delta \in (0,1)$. If n is such that $e^{-\frac{2n\epsilon^2}{(b-a)^2}} \leq \delta$, that is

$$n \geq \frac{\log(2/\delta)(b-a)^2}{2\epsilon^2},$$

then by Hoeffding's inequality, one guarantees an accuracy ϵ for the average mean estimator of μ_X with confidence $1 - \delta$.

It is important to note that this result is distribution-free, in that it does not depend on the distribution of X. This is a main strength of Hoeffding's inequality, which makes it applicable in general contexts. However, it is also its weak point, since the bounds derived based on it are generally conservative. Improved bound can be found when the probability distribution of X is known. The point is that in many cases of interest the probability distribution of X is not known explicitly and difficult to determine.

3.1.2 Probability estimation

Given the probability space $(\Omega, \mathcal{F}, P)$. Consider an event $\mathcal{A} \in \mathcal{F}$.

Let us introduce the random variable X: $\Omega \to \mathbb{R}$ on $(\Omega, \mathcal{F}, P)$:

$$X(\omega) = \begin{cases} 1 \text{ se } \omega \in \mathcal{A} \\ 0 \text{ se } \omega \notin \mathcal{A} \end{cases}$$

This is the indicator function of $\mathcal{A}$. It is easily seen that $\mu_X = P(\mathcal{A})$. The probability of the event $\mathcal{A}$ can then be estimated by estimating the mean of X.

If one can execute independent extractions from X, $P_A := P(\mathcal{A})$ can be estimated through the average mean estimator:

$$\hat{P}_A = \frac{1}{n} \sum_{i=1}^{n} X^{(i)}.$$

The estimator $\hat{P}_A$ has the same properties of the average mean estimator.

As for the asymptotic properties, it is unbiased and consistent.

As for the finite sample properties, the results obtained by Hoeffding's inequality still holds true. In this case X takes values in $[a, b] = [0, 1]$. The number of extraction n to guarantee a certain accuracy ϵ with confidence $1 - \delta$ is then given by

$$n = \left\lceil \frac{\log(2/\delta)}{2\epsilon^2} \right\rceil. \tag{1}$$

From this expression for n, one can see that confidence is “cheap”, because the dependence of n on δ is logarithmic, whereas accuracy is “expensive”, because the dependence of n on ϵ is quadratic.
Accuracy should be chosen based on the order of the probability to be estimated. Hence, estimating rare events probability by standard Monte Carlo method is not possible in practice. A possible solution to this issue is the multi-level decomposition method.

3.1.3 Stochastic reachability analysis

In general terms, a reachability analysis problem consists in verifying if the trajectory of a given system will enter some set within a certain time horizon. In a stochastic setting the system evolution is described in probabilistic terms, and the reachability analysis problem is formulated as that of determining the probability that the system trajectory will enter the set.

Let X_t be the stochastic process in $\mathbb{R}^n$ defined on $(\Omega, \mathcal{F}, P)$. Suppose that X_t is the solution to the stochastic differential equation (SDE)

$$dX_t = a(X_t)dt + b(X_t)dW_t \tag{2}$$

where $a : \mathbb{R}^n \to \mathbb{R}^n$, $b : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ and W_t is a Wiener process in $\mathbb{R}^n$. The SDE (2) is initialized with X_0, which is a random variable independent of W_t and with probability density μ_0 with support on $B_0 \subset \mathbb{R}^n$.

We consider the problem of computing the probability P_A that X_t enters some given set A within some time horizon $[0, T]$. In the ATM application of interest, the SDE (2) may describe the aircraft relative position and A the protection area surrounding an aircraft.

The probability P_A is generally difficult to determine analytically. One can estimate it by the Monte Carlo method.

By introducing the indicator function of the set A:

$$I_A(x) = \begin{cases} 1, \ x \in A \\ 0, \ x \notin A \end{cases} ,$$

the indicator function of the event of interest is given by $\varphi_{X_{[0,T]}} := \max_{\tau\in[0,T]} I_A(X_\tau)$, and the probability P_A can be expressed as

$$P_A = P(\varphi_{X_{[0,T]}} = 1) = E[\varphi_{X_{[0,T]}}].$$

By considering $n = \lceil \log(2/\delta)/(2\epsilon^2) \rceil$ independent extractions from $\varphi_{X_{[0,T]}}$ and taking their average, we obtain an estimate of P_A with accuracy ϵ and confidence $1 - \delta$. The estimator $\hat{P}_A$ can then be expressed as

$$\hat{P}_A = \frac{1}{n} \sum_{i=1}^{n} \varphi^{(i)}_{X_{[0,T]}},$$

with $\varphi^{(i)}_{X_{[0,T]}}$, $i = 1, \ldots, n$, random variables independent and identically distributed as $\varphi_{X_{[0,T]}}$.

An extraction from $\varphi_{X_{[0,T]}}$ can be obtained by simulating first a trajectory of X_t over the time interval $[0, T]$, according to the SDE (2) initialized at time 0 with the value for X_0 extracted from μ_0, and then computing $\varphi_{X_{[0,T]}} = \max_{\tau\in[0,T]} I_A(X_\tau)$ for this realization. The estimator $\hat{P}_A$ can then be reformulated as

$$\hat{P}_A = \frac{1}{n} \sum_{i=1}^{n} \varphi_{X^{(i)}_{[0,T]}}, \tag{3}$$

where $X^{(i)}_{[0,T]}$, $i = 1, \ldots, n$, are independent and distributed as $X_{[0,T]}$.

In practice, the estimate of P_A is the fraction of the n simulated trajectories that enters A.

3.2 Multi-level decomposition method

3.2.1 Description of the approach

Given an event $\mathcal{A} \in \mathcal{F}$ in some probability space $(\Omega, \mathcal{F}, P)$, let us consider a sequence of nested events $\mathcal{B}_i$, $i = 1, \ldots, m$: $\mathcal{A} = \mathcal{B}_m \subset \ldots \subset \mathcal{B}_1$.

Then, the probability of the event $\mathcal{A}$ can be expressed as follows: $P_A = \Pi_{i=1}^m P_i$, where we set $P_i := P(\mathcal{B}_i|\mathcal{B}_{i-1})$, $i = 2, \ldots, M$, and $P_1 = P(\mathcal{B}_1)$.

The idea of the multi-level decomposition method is to estimate P_A by

$$\hat{P}_A = \Pi_{i=1}^m \hat{P}_i,$$

where $\hat{P}_i$ denotes the estimate of P_i, $i = 1, \ldots, m$. If each one of the P_i's is large compared with P_A, then the number of samples needed to estimate each P_i by the Monte Carlo method is significantly smaller than that needed to estimate P_A.

The following proposition shows how is it possible to determine the accuracy of $\hat{P}_A = \Pi_{i=1}^m \hat{P}_i$ given the accuracy of each single $\hat{P}_i$.

Proposition 1 *Let $|q_i - p_i| \le \epsilon_i$, $i = 1, \ldots, m$, where q_i, p_i, and ϵ_i, $i = 1, \ldots, m$, are positive real numbers. Then*

$$\left|\Pi_{i=1}^m p_i - \Pi_{i=1}^m q_i\right| \le \left[\Pi_{i=1}^m (1+\alpha_i) - 1\right] \Pi_{i=1}^m p_i, \tag{4}$$

where $\alpha_i := \epsilon_i / p_i$, $i = 1, \ldots, m$.

Proof: The proof is by induction on m.

It is immediately verified that $q_1 q_2 - p_1 p_2$ can be expressed as follows:

$$\begin{aligned} q_1 q_2 - p_1 p_2 &= (p_1 - q_1)(p_2 - q_2) - 2 p_1 p_2 + p_1 q_2 + p_2 q_1 \\ &= (p_1 - q_1)(p_2 - q_2) - p_1(p_2 - q_2) - p_2(p_1 - q_1). \end{aligned}$$

From this equation, we have

$$\begin{aligned} |p_1 p_2 - q_1 q_2| &= |q_1 q_2 - p_1 p_2| \\ &\le |(p_1 - q_1)(p_2 - q_2)| + |p_1(p_2 - q_2)| + |p_2(p_1 - q_1)| \\ &= |p_1 - q_1||p_2 - q_2| + p_1|p_2 - q_2| + p_2|p_1 - q_1|. \end{aligned}$$

Since $|q_i - p_i| \le \epsilon_i$, $i = 1, 2$, we get

$$|p_1 p_2 - q_1 q_2| \le \epsilon_1 \epsilon_2 + p_1 \epsilon_2 + p_2 \epsilon_1.$$

If we set $\alpha_1 := \epsilon_1 / p_1$ and $\alpha_2 := \epsilon_2 / p_2$, then

$$|p_1 p_2 - q_1 q_2| \le (\alpha_1 \alpha_2 + \alpha_2 + \alpha_1) p_1 p_2 = [(1+\alpha_1)(1+\alpha_2) - 1] p_1 p_2,$$

which is equation (4) for $m = 2$.

Assume now by induction hypothesis that

$$\left|\Pi_{i=1}^k p_i - \Pi_{i=1}^k q_i\right| \le \left[\Pi_{i=1}^k (1+\alpha_i) - 1\right] \Pi_{i=1}^k p_i.$$

We next show that

$$\left|\Pi_{i=1}^{k+1} p_i - \Pi_{i=1}^{k+1} q_i\right| \le \left[\Pi_{i=1}^{k+1} (1+\alpha_i) - 1\right] \Pi_{i=1}^{k+1} p_i,$$

which concludes the proof of (4).

Define $\bar{p}_k := \Pi_{i=1}^k p_i$ e $\bar{q}_k := \Pi_{i=1}^k q_i$. By the induction hypothesis,

$$\left|\bar{p}_k - \bar{q}_k\right| \le \epsilon$$

with $\epsilon := \alpha\bar{p}_k$ and $\alpha := \Pi_{i=1}^{k}(1+\alpha_i) - 1$.

Based on the same derivations as for $m = 2$ with p_1 and p_2 replaced by $\bar{p}_k$ and p_{k+1}(hence α_1 and α_2 replaced by α and α_{k+1}), and q_1 and q_2 replace by $\bar{q}_k$ and q_{k+1} we obtain:

$$|\bar{p}_k p_{k+1} - \bar{q}_k q_{k+1}| \leq [(1+\alpha)(1+\alpha_{n+1}) - 1]\bar{p}_k p_{k+1}.$$

Substituting in this equation the expressions for $\bar{p}_k$, $\bar{q}_k$ and α, we finally have

$$\begin{aligned}\left|\Pi_{i=1}^{k+1} p_i - \Pi_{i=1}^{k+1} q_i\right| &= |\bar{p}_k p_{k+1} - \bar{q}_k q_{k+1}| \\ &\leq \left[(1 + \Pi_{i=1}^{k}(1+\alpha_i) - 1)(1+\alpha_{k+1}) - 1\right]\Pi_{i=1}^{k+1} p_i \\ &= \left[\Pi_{i=1}^{k+1}(1+\alpha_i) - 1\right]\Pi_{i=1}^{n+1} p_i.\end{aligned}$$

■

Suppose now that each P_i is estimated with accuracy ϵ_i and confidence $1 - \delta_i$ by the standard Monte Carlo method with $n_i = \lceil \log(2/\delta_i)/(2\epsilon_i^2) \rceil$ extractions (see (1)). Denote by $(\overline{\Omega}, \overline{\mathcal{F}}, \overline{P})$ the probability space where all the random variables involved in the estimates $\hat{P}_i$, $i = 1, \ldots, m$, are defined. Then,

$$\overline{P}\big(|P_i - \hat{P}_i| > \epsilon_i\big) < \delta_i,\ i = 1, \ldots, m.$$

This implies that

$$\overline{P}\big(|P_i - \hat{P}_i| \leq \epsilon_i,\ i = 1, \ldots, m\big) \geq 1 - \sum_{i=1}^{m} \delta_i.$$

Then, by Proposition 1 with $p_i := P_i$ and $q_i := \hat{P}_i$, we have

$$\overline{P}\left(|P_A - \hat{P}_A| \leq \alpha P_A\right) \geq 1 - \delta \tag{5}$$

where we set $\alpha := [\Pi_{i=1}^{m}(1+\alpha_i) - 1]$ with $\alpha_i = \epsilon_i/P_i$, and $\delta := \sum_{i=1}^{m} \delta_i$. The overall number of extractions to estimate P_A is

$$n = \sum_{i=1}^{m} n_i = \sum_{i=1}^{m} \left\lceil \frac{\log(2/\delta_i)}{2\epsilon_i^2} \right\rceil. \tag{6}$$

From (5) and the definition of α and δ, it follows that to ensure a relative accuracy α for the estimate $\hat{P}_A$, with confidence $1 - \delta$, we can choose

$$\begin{aligned}&\epsilon_i = \overline{\alpha} P_i,\ i = 1, \ldots, m, \text{ with } \overline{\alpha} \leq \sqrt[m]{1+\alpha} - 1, \\ &\delta_i = \tfrac{\delta}{m},\ i = 1, \ldots, m.\end{aligned} \tag{7}$$

Equation (7) combined with equation (6) provides the number of extractions needed to ensure a certain desired relative accuracy α and confidence $1-\delta$ for the estimate of P_A. Below we provide an example of evaluation of the computational efficiency of the multi-level decomposition method for different values of the probabilities P_i.

Example 1 Let $P_A = 10^{-10}$. Suppose that we introduce 5 nested events such that:
Case a: $P_1 = P_2 = P_3 = P_4 = P_5 = 10^{-2}$.

Case b: $P_1 = P_2 = P_3 = P_4 = 10^{-1}$ and $P_5 = 10^{-6}$.
To achieve a relative accuracy $\alpha = 0.1$ (absolute accuracy $\epsilon = \alpha P = 10^{-11}$) and a confidence $1 - 10^{-6}$, we can choose $\delta_i = 0.2 \cdot 10^{-6}$, $i = 1, \ldots, 5$, and $\overline{\alpha} = 1.92 \cdot 10^{-2}$.
The resulting number of extractions in case a and case b is, respectively, $n_a = \sum_{i=1}^{5} n_{a,i} \simeq 10^9$ and $n_b = \sum_{i=1}^{5} n_{b,i} \simeq \cdot 10^{16}$, whereas in the standard Monte Carlo approach the number of extractions would be $n \simeq 10^{22}$. □

The problem is that this procedure cannot be used in practice to decide the number of extractions for guaranteeing a certain relative accuracy α and confidence $1 - \delta$ for $\hat{P}_A$. This is because the value for ϵ_i is chosen based on the value of the probability P_i, which is not known.

In the standard Monte Carlo approach a similar problem arises, since the number of simulations to guarantee a relative accuracy α depends on P_A through $\epsilon = \alpha P_A$, and P_A is not known. However, it is often the case that the objective is actually verifying if P_A is greater than some value $\tilde{P}$. Hence, ϵ can be chosen based on $\tilde{P}$. In the multi-level approach the same kind of information on $\tilde{P}$ is not useful for deciding the single ϵ_i: is we set ϵ_i base on $\tilde{P}$ we would not get any computational advantage with respect to the standard Monte Carlo approach.

The novel idea presented in the following subsection is to make an a-posteriori evaluation of the accuracy, based on the obtained estimated value for P_i.

3.2.2 The problem of evaluating the quality of the multi-level probability estimate: a solution based on an a-posteriori evaluation

We propose an approach for guaranteeing a desired quality for the estimate of the overall probability P_A obtained by multi-level decomposition. The key characteristic of the approach is the a-posteriori assessment of the actual value for the accuracy ϵ_i, based on the value of $\hat{P}_i$, so as to achieve the desired accuracy ϵ for $\hat{P}_A$.

The following algorithm is formulated to determine an estimate $\hat{P}_A$ for P_A with guaranteed relative accuracy $\beta \in (0, 1)$ and confidence $1 - \delta$.

Algorithm 1 (A-posteriori guaranteed quality estimate)

```
Input: Fix $\beta$, $\delta \in (0,1)$, $\bar{\epsilon} > 0$, and $r \in (0,1)$.

Initialization: Set $\overline{\alpha} = \sqrt[m]{1 + \beta/(1+\beta)} - 1$, $\delta_k = \delta/m$ and $\epsilon_k = \bar{\epsilon}$, $k = 1, \ldots, m$, $i = 1$.

For $i = 1$ to $m$ do:

    1. Determine an estimate $\hat{P}_i$ of $P_i$ with absolute accuracy $\epsilon_i$ and confidence $1 - \delta_i$.
    2. If $\epsilon_i/\hat{P}_i > \overline{\alpha}$, then set $\epsilon_i := r\epsilon_i$ and go to step 1, else $i := i + 1$.

Output: Set $\hat{P}_A := \Pi_{i=1}^{m} \hat{P}_i$.
```

$\bar{\alpha}$ is the relative accuracy for each $\hat{P}_i$ estimate, which would correspond to the desired relative accuracy β for the overall $\hat{P}_A$ estimate. The value for the absolute accuracy ϵ_i for each $\hat{P}_i$ is selected a-posteriori, based on $\hat{P}_i$, so as to guarantee the relative accuracy $\bar{\alpha}$. This is done iteratively, by eventually rescaling the current value of ϵ_i by an $r < 1$ factor.

We now show that $\hat{P}_A$ obtained through this algorithm is an estimate of P_A with relative accuracy β and confidence $1-\delta$.

The proof relies on the proposition below, which directly follows from Proposition 1.

Proposition 2 *Let* $|P_i - \hat{P}_i| \leq \epsilon_i$, $i = 1, \ldots, m$. *Then*

$$\left|\Pi_{i=1}^m P_i - \Pi_{i=1}^m \hat{P}_i\right| \leq \frac{\alpha}{1-\alpha}\Pi_{i=1}^m P_i, \tag{8}$$

where $\alpha := \left[\Pi_{i=1}^m (1+\alpha_i) - 1\right]$ *with* $\alpha_i := \epsilon_i/\hat{P}_i$, $i = 1, \ldots, m$.

Proof: By Proposition 1 with $q_i = P_i$ and $p_i = \hat{P}_i$, $i = 1, \ldots, m$, we have

$$\left|\Pi_{i=1}^m P_i - \Pi_{i=1}^n \hat{P}_i\right| \leq \left[\Pi_{i=1}^m (1+\alpha_i) - 1\right]\Pi_{i=1}^m \hat{P}_i \tag{9}$$

Based on (9), $\Pi_{i=1}^m \hat{P}_i \leq \frac{1}{1-\alpha}\Pi_{i=1}^m P_i$.

Plugging this bound into (9), we get (8). ■

Denote by $(\overline{\Omega}, \overline{\mathcal{F}}, \overline{P})$ the probability space where all the random variables involved in the estimates $\hat{P}_i$, $i = 1, \ldots, m$, are defined. Then,

$$\overline{P}\big(|P_i - \hat{P}_i| > \epsilon_i\big) < \delta_i, \; i = 1, \ldots, m.$$

This implies that

$$\overline{P}\big(|P_i - \hat{P}_i| \leq \epsilon_i, \; i = 1, \ldots, m\big) \geq 1 - \sum_{i=1}^m \delta_i = 1 - \delta.$$

Then, by Proposition 2, we have

$$\overline{P}\left(|P_A - \hat{P}_A| \leq \frac{\alpha}{1-\alpha}P_A\right) \geq 1 - \delta, \tag{10}$$

where $\alpha := \left[\Pi_{i=1}^m (1+\alpha_i) - 1\right]$ with $\alpha_i := \epsilon_i/\hat{P}_i$, $i = 1, \ldots, m$. Since ϵ_i is adaptively chosen so that $\alpha_i := \epsilon_i/\hat{P}_i \leq \bar{\alpha}$, by the definition of $\bar{\alpha}$ we have that $\alpha \leq \beta/(1+\beta)$. If we plug this bound in (10), we finally get that

$$\overline{P}\left(|P_A - \hat{P}_A| \leq \beta P_A\right) \geq 1 - \delta.$$

As for the computational efficiency of the algorithm, by Hoeffding's inequality, for estimating each P_i with accuracy ϵ_i and confidence $1 - \delta_i$, we need $n_i = \lceil \log(2/\delta_i)/(2\epsilon_i^2) \rceil$ independent extractions from the indicator function of the event $\mathcal{B}_i$. The value for ϵ_i is decided a-posteriori based on the resulting value for $\hat{P}_i$, starting from the initialization $\bar{\epsilon}$. This means that we cannot determine a-priori the computational effort needed to estimate P_A with given relative accuracy β and confidence $1-\delta$.

This is in fact not surprising, since we do not know the P_i values a-priori: the computational effort will be comparable with that of the standard Monte Carlo method if some P_i is close to P_A, and much lower if all the P_i's are equal.

3.2.3 Application to stochastic reachability analysis

Let us consider the stochastic reachability analysis problem described in Subsection 3.1.3.

Our objective is computing the probability P_A that the solution X_t to SDE equation (2) enters some given set A within some time horizon $[0, T]$.

We next illustrate how the multi-level decomposition method can be used to estimate P_A. For this purpose it is convenient to introduce the stopping time $\tau_A := \inf\{t \geq 0 : X_t \in A\}$ and express P_A as

$$P_A := P(\tau_A \leq T).$$

Under suitable assumptions on the a and b coefficients of the SDE equation (2), its solution X_t is a strong time-homogeneous Markov process with continuous trajectories.

Consider a sequence of m nested sets B_i, $i = 1, \dots, m$, in the state space $\mathbb{R}^n$ of the X_t process, such that:

$$A = B_m \subset \dots \subset B_1 \quad B_1 \cap B_0 = \emptyset.$$

An example is shown in figure 2 for the case of $n = 2$ and $m = 4$, with the target set A that is a circle of unitary radius centered at the origin. The sets B_i, $i = 1, 2, 3, 4$ are circles centered at the origin defined by $B_i = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq (5 - i)^2\}$, $i = 1, 2, 3, 4$.

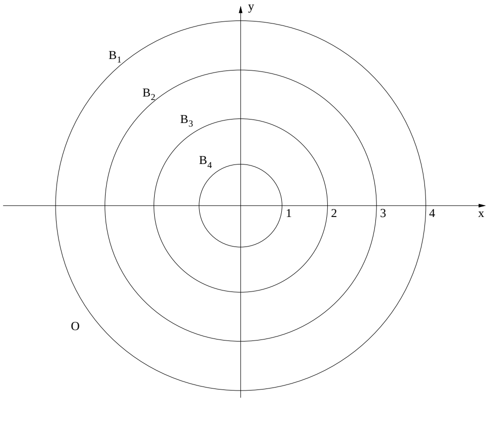

Figure 2: Example of multi-level decomposition in $\mathbb{R}^2$

Define the random variable τ_i representing the first time the process enters B_i:

$$\tau_i := \inf\{t \geq 0 : X_t \in B_i\},\ i = 1, 2, \dots, m.$$

Based on the continuity properties of the X_t process trajectories, X_t has to enter the sets B_1, B_2, ..., B_{m-1}, in this order, before entering the target set $A = B_m$. This implies that the probability of interest $P_A = P(\tau_m < T)$ can be expressed as follows ([36, 37]):

$$\begin{aligned} P_A &= P(\tau_k \leq T, k = 1, \dots, m) \\ &= P(\tau_m \leq T | \tau_k \leq T, k = 1, \dots, m-1) P(\tau_k \leq T, k = 1, \dots, m-1) \end{aligned}$$

$$
\begin{aligned}
&= P(\tau_m \leq T|\tau_{m-1} \leq T)P(\tau_k \leq T, k = 1, \ldots, m-1)\\
&= \ldots\\
&= P(\tau_m \leq T|\tau_{m-1} \leq T)P(\tau_{m-1} < T|\tau_{m-2} \leq T) \ldots P(\tau_1 \leq T).
\end{aligned}
$$

If we let $P_i = P(\tau_i \leq T|\tau_{i-1} < T)$ and $P_1 = P(\tau_1 \leq T)$, then, $P_A = \Pi_{i=1}^m P_i$.

Recalling the notations in subsection 3.2.1, the event $\mathcal{A}$ and the nested events $\mathcal{B}_i$ are defined as

$$
\begin{aligned}
&\mathcal{A} = \{\inf\{t \geq 0 : X_t \in A\} \leq T\},\\
&\mathcal{B}_i = \{\tau_i \leq T\} = \{\inf\{t \geq 0 : X_t \in B_i\} \leq T\},\ i = 1, \ldots, m,
\end{aligned}
$$

whereas the probabilities P_i are given by

$$P_i = P(\mathcal{B}_i|\mathcal{B}_{i-1}),\ i = 2, \ldots, m, \quad P_1 = P(\mathcal{B}_1).$$

In order to estimate P_A, one should be able to estimate each single probability P_i, $i = 1, \ldots, m$. For $i = 1$, one can simply run simulations of the process X_t according to the SDE (2) initialized with the given initial distribution π_0 of X_0. Instead, for $i > 1$, the event $\mathcal{B}_i$ depends on the evolution of the strong Markov process X_t after entering the set B_{i-1}. Given the strong Markov property of X_t, if the distribution of the process arrested at the hitting time τ_{i-1}, were known, then one could run simulations according to the SDE (2), re-initialized on the contour of B_{i-1} with such distribution. The problem is that the distribution of $X_{\tau_{i-1}}$ is not known and difficult to determine.

This problem is addressed in the different implementations of the multi-level decomposition method for stochastic reachability based on the following idea. The trajectories simulated for estimating P_{i-1} are stopped on the contour of B_{i-1}, as soon as they enter this set within time T. Multiple trajectories are then generated starting from these points, according to the SDE (2), so as to estimate P_i.

In the Interacting Particle System (IPS) algorithm ([37, 39, 36]), N_p trajectories are generated starting from N_p initial conditions extracted according to the empirical distribution obtained on the contour of B_{i-1}. Under strong Markov condition, the asymptotic analysis in [37] shows that the so-obtained estimator for P_A is unbiased and consistent. In the REpetitive Simulation Trials After Reaching Thresholds (RESTART) algorithm developed in [40], a fixed number of trajectories per hitting point is generated. It is important to note, however, that RESTART does not assume that the process is a strong Markov process, and does not provide any asymptotic analysis.

By letting N_p depend on the level i, one can easily integrate the IPS method in Algorithm 1 of subsection 3.2.2, by using it to determine the estimate $\hat{P}_i$ at step 1. However, it is important to note that, given the approximation of the distribution on the contour of B_{i-1}, the conclusions on the analysis of the algorithm are still valid only under some restrictive assumptions such as that the probability of entering B_i does not depend on the starting point on the contour of B_{i-1}.

An example of application of Algorithm 1 incorporating the IPS method is provided next. This example allows to verify the IPS algorithm performance, given that it is possible to compute the probability of interest exactly.

Example 2 (algebraic Brownian motion) Consider the algebraic Brownian motion in $\mathbb{R}$, solution to the SDE:

$$dX_t = \alpha dt + \sigma dW_t, \tag{11}$$

initialized with $X_0 = x$, where α and σ are positive constants.

Suppose that we want to estimate the probability that X_t hits the barrier $d = 0$ within time T starting from $x > 0$. If we set $A = \{y \in \mathbb{R} : y \leq 0\}$ and $\tau(x) = \inf\{t > 0 : X_t \in A; X_0 = x\}$, then the probability of interest can be expressed as

$$P_A(x) = P(\tau(x) \leq T).$$

Now, $\tau(x)$ is a random variable with probability density function:

$$f_{\tau(x)}(t) = \frac{x}{\sigma\sqrt{2\pi t^3}} \cdot e^{-\frac{(x+\alpha t)^2}{2\sigma^2 t}}.$$

Therefore $P_A(x)$ can be computed as follows

$$P_A(x) = P(\tau(x) \leq T) = \int_0^T f_{\tau(x)}(t)dt.$$

We set $T = 1$, $\alpha = 1$, $\sigma = 1$. The corresponding values for $P_A(x)$, $x \in [1, 5]$, are reported in figure 3.

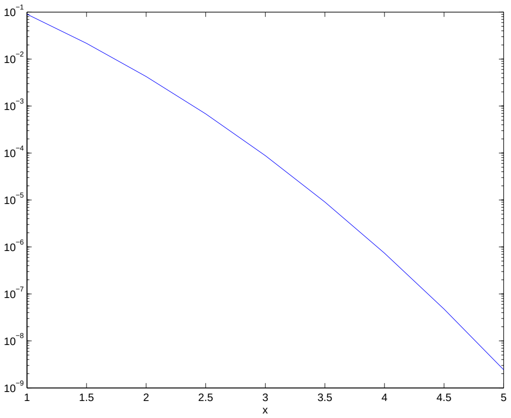

Figure 3: Probability $P_A(x)$ of hitting $d = 0$ as a function of the initial position x, $x \in [1, 5]$.

Let us consider the set of nested events $B_i = \{y \in \mathbb{R} : y \leq d_i\}$, $i = 1, \dots, m$, where the thresholds d_i are such that $0 = d_m < d_{m-1} < \cdots < d_1 < x$. In the reported examples, d_i are chosen so that the corresponding probabilities P_i are all greater than or equal to 0.1.

The results obtained through Algorithm 1 incorporating the IPS method are provided in Table 2. We set the desired relative accuracy and confidence for the estimate of $P_A(x)$ equal to $\beta = 0.5$ and $1 - \delta = 0.999$. The values for the relative accuracy and confidence of each single $\hat{P}_i$ are

$$\alpha_i = \overline{\alpha} = \sqrt[m]{1 + \beta/(1 + \beta)} - 1, \quad 1 - \delta_i = 1 - \delta/m,$$

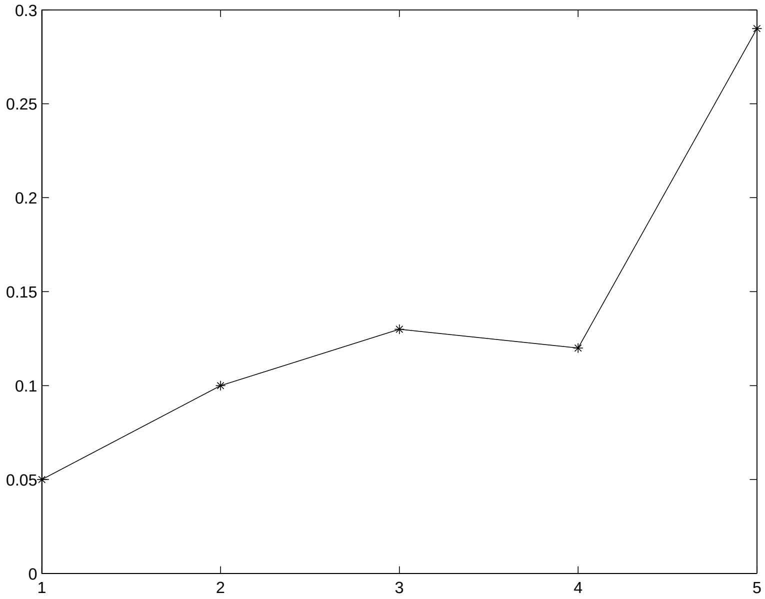

Figure 4: Actual relative accuracy as a function of the initial position x.

which depend on the number m of levels.

To implement the IPS model in practice we use a discrete-time approximation of (11) based on the Euler scheme with 1000 discretization time steps.

x	m	$\overline{\alpha}$	P_A	$\hat{P}_A$	time [s]
1	6	$4.86 \cdot 10^{-2}$	$9.04 \cdot 10^{-2}$	$8.56 \cdot 10^{-2}$	76
2	12	$2.4 \cdot 10^{-2}$	$4.26 \cdot 10^{-3}$	$3.80 \cdot 10^{-3}$	129
3	20	$1.4 \cdot 10^{-2}$	$8.8 \cdot 10^{-5}$	$7.65 \cdot 10^{-5}$	174
4	26	$1.1 \cdot 10^{-2}$	$7.39 \cdot 10^{-7}$	$6.47 \cdot 10^{-5}$	222
5	32	$9 \cdot 10^{-3}$	$2.42 \cdot 10^{-9}$	$1.72 \cdot 10^{-9}$	238

Table 2: Results of Algorithm 1 incorporating the IPS method ($\beta = 0.5$, $\delta = 0.01$)

In Figure 4 the relative accuracy for $\hat{P}_A(x)$ is plotted as a function of x. Note that the actual relative accuracy is smaller than $\beta = 0.5$. A reason for this is that the value for ϵ_i adaptively selected in Algorithm 1 may be much smaller than necessary, if the r factor is small. □

3.3 Sequential Monte Carlo approach to stochastic reachability analysis

The Sequential Monte Carlo (SMC) approach is introduced to improve the computational efficiency of the standard Monte Carlo approach to stochastic reachability analysis for systems evolving in a highly dynamic uncertain environment, where reachability has to be repeatedly verified on-line based on the updated information on the system behavior.

This is relevant to our application context, since in fact the probability of conflict has to be

updated at each time instant when new radar measurements on the aircraft positions become available. According to the standard Monte Carlo approach, this would require simulating new trajectories starting from the newly updated initial positions.
The idea of the SMC approach is to update the probability of conflict based on the re-use of the trajectories previously simulated after appropriately weighting their contribution according to their likelihood, suitably refreshed based the new radar measurements. This idea is inspired by the particle filtering methods for nonlinear filtering, whose main principle is recursively generating random measures that approximate distributions of interest. These random measures are represented by samples extracted from relevant distributions and propagated through the system dynamics, appropriately weighted based on the observations on the system.

The improvement obtained by re-using the trajectories previously simulated is particularly significant in the case when generating trajectories is computationally demanding.

We formulate the SMC method in a discrete time setting.

Let (Ω, F, P) be a probability space on which two vector-valued stochastic processes are defined; $X = \{X_t, t \in \mathbb{N}\}$ and $Y = \{Y_t, t \in \mathbb{N}\}$. X is the state process and Y the observation process. Let n_d be the dimension of the state space of each of X and Y, and let $\mathcal{B}(\mathbb{R}^{n_d})$ be the Borel σ-algebra on $\mathbb{R}^{n_d}$.

We assume that X is a Markov Process with initial probability density μ_{X_0} and probability transition kernel $K_t(x_t, x_{t-1})$, such that for all $S \in \mathcal{B}(\mathbb{R}^{n_d})$:

$$P\{X_k \in S | X_{k-1} = x_{k-1}\} = \int_S K_k(x, x_{k-1}) dx. \tag{12}$$

To simplify notation we shall denote the time varying kernel $K_k(\cdot,\cdot)$ by $K(\cdot,\cdot)$. Although the kernel is time-varying, the mapping being performed is implicit from the subscript k in x_k.

The joint probability density of $X_{0:t} := (X_0, X_1, \ldots, X_t)$ is given by

$$\mu_{X_{0:t}}(x_{0:t}) := \mu_{X_0}(x_0) \prod_{k=1}^{t} K(x_k, x_{k-1}).$$

The probability density of X_t can then be expressed by

$$\mu_{X_t}(x_t) = \int \mu_{X_0}(x_0) \prod_{k=1}^{t-1} K(x_k, x_{k-1}) K(x_t, x_{t-1}) dx_{0:t-1},$$

where $dx_{0:t-1}$ is a short-hand-notation for $dx_0 dx_1 \ldots dx_{t-1}$.

We consider the full information case where at each time step t the exact observation process state is received ($Y_t = X_t$). While this is unrealistic for highly complex systems, at the level of complexity at which we shall operate, information of the most significant state elements can be gathered.

At each time t, we wish to estimate the probability that the state of the system will reach a given set A over a time horizon of length T, given the observations Y_k collected up to time t:

$$P_A(t) := P(X_k \in A, \text{ for some } k \in [t, t+1, \ldots, t+T] | Y_k = y_k, k = 0, \ldots, t).$$

As X is a Markov process and $Y_k = X_k$, $k = 0, \dots t$, this is equivalent to:

$$P_A(t) = P(X_k \in A, \text{ for some } k \in [t, t+1, \dots, t+T] | Y_t = X_t = y_t). \tag{13}$$

Let $\varphi_{X_{t:t+T}} := \max_{\tau \in [t,t+1,\dots,t+T]} I_A(X_\tau)$, where I_A is the indicator function of set A and $X_{t:t+T}$ is the collection of random variables X_k, $k = t, \dots, t+T$. Then, the probability $P_A(t)$ can then be expressed as

$$P_A(t) = P(\varphi_{X_{t:t+T}} = 1 | Y_t = x_t) = E[\varphi_{X_{t:t+T}} | Y_t = y_t].$$

The conditional probability density of $X_{t:t+T}$ given $Y_t = X_t = y_t$ is given by

$$\mu_{X_{t:t+T}|Y_t=y_t}(x_{t:t+T}|y_t) = \delta_{y_t}(x_t) \prod_{k=t+1}^{t+T} K(x_k, x_{k-1}),$$

where $\delta_y(\cdot)$ is the δ function in y. As a result, $P_A(t)$ can be expressed as

$$P_A(t) = \int \varphi_{X_{t:t+T}} \delta_{y_t}(x_t) \prod_{k=t+1}^{t+T} K(x_k, x_{k-1}) dx_{t:t+T}. \tag{14}$$

Analytical solutions to equation (14) can be obtained only in a limited set of cases. We can use the standard Monte Carlo method to estimate $P_A(t)$. By extracting a certain number n of independent identically distributed (i.i.d.) paths according to the conditional joint distribution of positions across the time horizon

$$\tilde{X}^{(i)}_{t:t+T} \sim \delta_{y_t}(x_t) \prod_{k=t+1}^{t+T} K(x_k, x_{k-1}),\ i = 1, \dots, n, \tag{15}$$

we can then compute the estimate

$$\hat{P}_A(t) = \frac{1}{n} \sum_{i=1}^{n} \varphi_{\tilde{X}^{(i)}_{t:T+t}}. \tag{16}$$

By Hoeffding's inequality:

$$\bar{P}(|\hat{P}_A(t) - P_A(t)| > \epsilon) < e^{-2n\epsilon^2},$$

where $\bar{P}$ is the probability measure in the probability space where all random variables involved are defined. Then, to get an estimate of P_A with accuracy ϵ and confidence $1 - \delta$ we have to take $n = \lceil \log(2/\delta)/(2\epsilon^2) \rceil$.

The concept of particles arises naturally in the Monte Carlo method; rather than referring to n i.i.d. sampled paths, we may simply refer to n 'particles'. A particle is a pair $(\tilde{X}^{(i)}_{t:t+T}, \xi^{(i)})$, consisting of a trajectory $\tilde{X}^{(i)}_{t:t+T}$ and an associated weight $\xi^{(i)}$ representing the 'significance' of this particle. The weights add up to 1:

$$\sum_{i=1}^{n} \xi^{(i)} = 1.$$

In the case of equation (16), for instance, the weight $\xi^{(i)} = 1/n$, as all particles are equally significant. The Monte Carlo estimate can then be expressed as

$$\hat{P}_A(t) = \sum_{i=1}^{n} \xi^{(i)} \varphi_{\tilde{X}^{(i)}_{t:T+t}}.$$

In this section we demonstrate how 'old' trajectories (e.g. from time 0) may be used to gives unbiased estimates of $P_A(t)$, up to certain values of t. This process is known as 'particle filtering'. Particle filtering methods have generated considerable research interest in recent years [41, 42, 43], principally as a method for performing non-linear filtering. Non-linear stochastic processes commonly do not admit analytical descriptions of their probability distributions, however, if the trajectories may be simulated then an empirical probability distribution may be constructed. Sequential Monte Carlo methods are usually used to weight the prior (simulated) distribution, given an uncertain state observation, to give a posterior distribution taking into account the effect of the observation on the state estimate. Resampling is then performed according to this weighted distribution and simulation continues. The ultimate goal is to achieve convergence to a stationary distribution, from which empirical properties may be estimated. In the case we present the circumstances are quite different: we have exact state observation (and hence perfect knowledge of the posterior distribution over the time horizon), but due to the potential complexity of our dynamical system are concerned with the possibly unwarranted computational effort of drawing new sample trajectories at each time step. We therefore introduce a novel method for re-using old sample paths in an optimal fashion.

Lemma 1 (SMC estimation) *The probability $P_A(t)$, given the observation $Y_t = y_t$, can be expressed as follows*

$$P_A(t) = \int \varphi_{y_t, x_{t+1:t+T}} q(x_{0:t+1}) \frac{K(x_{t+1}, y_t)}{\delta_{y_0}(x_0) \prod_{k=1}^{t+1} K(x_k, x_{k-1})} \delta_{y_0}(x_0) \prod_{k=1}^{t+T} K(x_k, x_{k-1}) dx_{0:t+T}, \quad (17)$$

where $q(x_{0:t+1})$ is any function satisfying $q(x_{0:t+1}) > 0$, for all $x_{0:t+1}$, and $\int q(x_{0:t}, x_{t+1}) dx_{0:t} = 1$, for all x_{t+1}.

Proof: At time t, the standard MC estimation of $P_A(t)$ is:

$$P_A(t) = \int \varphi_{x_{t:t+T}} \delta_{y_t}(x_t) \prod_{k=t+1}^{t+T} K(x_k, x_{k-1}) dx_{t:t+T},$$

which may be re-written as

$$P_A(t) = \int \varphi_{y_t, x_{t+1:t+T}} K(x_{t+1}, y_t) \prod_{k=t+2}^{t+T} K(x_k, x_{k-1}) dx_{t+1:t+T}. \quad (18)$$

Given that $q(x_{0:t+1})$ is such that $\int q(x_{0:t}, x_{t+1}) dx_{0:t} = 1$ and $q(x_{0:t+1}) > 0$, we may write:

$$P_A(t) = \int \varphi_{y_t, x_{t+1:t+T}} q(x_{0:t+1}) K(x_{t+1}, y_t) \prod_{k=t+2}^{t+T} K(x_k, x_{k-1}) dx_{0:t+T}.$$

Multiplying and dividing by $\delta_{y_0}(x_0)\prod_{k=1}^{T} K(x_k, x_{k-1})$, we get (17). ■

The choice of $q(\cdot,\cdot)$ is entirely free within the given constraints; the estimate of $P_A(t)$ will be unaffected as long as the distribution at time steps $t+1$ and beyond are not skewed; hence the condition that $\int q(x_{0:t}, x_{t+1}) dx_{0:t} = 1$. The choice of $q(\cdot,\cdot)$ depends on some considerations discussed later in this section.

Note now that

$$\delta_{y_0}(x_0)\prod_{k=1}^{t+T} K(x_k, x_{k-1})$$

appearing in (17) is the conditional probability density of $X_{0:t+T}$ given the observation $Y_0 = y_0$ $\mu_{X_{0:t+T}|Y_0=y_0}$. This is the probability density according to which sample paths are drawn for estimating $P_A(0)$ (see equation (15) with $t = 0$).

Define the weighting function

$$\xi(x_{0:t+1}; y_0, y_t) = q(x_{0:t+1})\frac{K(x_{t+1}, y_t)}{\delta_{y_0}(x_0)\prod_{k=1}^{t+1} K(x_k, x_{k-1})}.$$

By Lemma 1, we were able to express $P_A(t)$ as

$$P_A(t) = E_{\mu_{X_{0:t+T}|Y_0=y_0}}[\varphi_{y_t, X_{t+1:t+T}}\xi(X_{0:t+1}; y_0, y_t)] \tag{19}$$

where $E_{\mu_{X_{0:t+T}|Y_0=y_0}}$ denotes expectation with respect to the conditional density $\mu_{X_{0:t+T}|Y_0=y_0}$. This reformulation allows to obtain an estimate of $P_A(t)$ at time step t, from sample paths originating at time step 0, according to the standard Monte Carlo approach. Note however that for large values of t it will generally be the case that only a small number of the paths sampled according to $\mu_{X_{0:t+T}|Y_0=y_0}$ will match the y_t observation (and hence will give a significant contribution to the estimate). The Hoeffding's result still holds, since it is distribution-free. The problem is that the fraction of paths that give a significant contribution to the integral has a small probability, and it may end up being in the δ probability region where the accuracy ϵ is not guaranteed.
This is known as the degeneracy phenomenon in the particle filtering literature. We shall now show how it can be alleviated by appropriately selecting the q function.

We reiterate at this point that this is not SMC in a standard sense; due to the challenge of re-using sample paths given an exact state observation we introduce the new concept of weighting according to probability of transition to particle locations at the next time step.

3.3.1 SMC estimate

Based on the expression in Lemma 1, the SMC estimate is given by

$$\hat{P}_A(t) = \sum_{i=1}^{n} \varphi_{y_t, \tilde{X}^{(i)}_{t+1:t+T}} \xi^{(i)}, \tag{20}$$

$$\xi^{(i)} \propto q(\tilde{X}^{(i)}_{0:t+1}) \frac{K(\tilde{X}^{(i)}_{t+1}, y_t)}{\delta_{y_0}(\tilde{X}^{(i)}_0) \prod_{k=1}^{t+1} K(\tilde{X}^{(i)}_k, \tilde{X}^{(i)}_{k-1})}, \sum_{i=1}^{n} \xi^{(i)} = 1, \tag{21}$$

where $\tilde{X}^{(i)}_{0:t+T}$, $i = 1, \ldots, n$, are independent identically distributed (i.i.d.) paths according to the conditional joint distribution of positions across the time horizon $[0, t+T]$:

$$\tilde{X}^{(i)}_{0:t+T} \sim \delta_{y_0}(x_0) \prod_{k=1}^{t+T} K(x_k, x_{k-1}),\ i = 1, \ldots, n. \tag{22}$$

The weight normalization (21) is a standard step in SMC estimation, ensuring that $\hat{P}_A(t)$ lies in the range $[0, 1]$. Weighting in this manner assigns high weights to those particles closely matching the observation y_t, and low weights to those far from y_t. As $n \to \infty$ the convergence becomes 'almost sure': realizations leading to a failure of the SMC estimation exist, but have zero measure. In practice, however, i.e., when n is finite, the degeneration phenomenon deteriorate the estimate accuracy.

Degeneracy of the particle approximation is typically measured by the Effective Sample Size (ESS) criterion ([44, 45]):

$$n_{ess} = \frac{1}{\sum_{i=1}^{n} \left(\xi^{(i)}\right)^2}, \tag{23}$$

which takes values in $[1, n]$, given that $\xi^{(i)} \geq 0$ and $\sum_{i=1}^{n} \xi^{(i)} = 1$. $n_{ess} = 1$ when one weight is 1, and the others are 0, whereas $n_{ess} = n$ is all the weights are equal to $1/n$. High degeneracy corresponds to $n_{ess} = 1$. When the n_{ess} is below some threshold, then a new set of sample paths are drawn starting from the current y_t observation to estimate $P_A(t)$. This set will eventually be re-used for estimating $P_A(\tau)$ for $\tau > t$.

The larger the value of n_{ess}, the better the estimate. Given that the $\xi^{(i)}$, $i = 1, \ldots, n$, are identically distributed and they add up to 1, their mean is $1/n$. This jointly with the definition of n_{ess} in equation (23) leads to the fact that to maximize the n_{ess} the variance of $\xi^{(i)}$ must be minimized. The freedom available to achieve this is the choice of q. The optimal choice of $q(\cdot, \cdot)$ is:

$$q_{opt}(x_{0:t+1}) = \frac{\delta_{y_0}(x_0) \prod_{k=1}^{t+1} K(x_k, x_{k-1})}{\int \delta_{y_0}(x_0) \prod_{k=1}^{t+1} K(x_k, x_{k-1}) dx_{0:t}},$$

and is a particularization of the more general result in [45].

The corresponding weighting function is

$$\xi(x_{t+1}; y_0, y_t) = \frac{K(x_{t+1}, y_t)}{\mu_{X_{t+1}|Y_0=y_0}(x_{t+1}|y_0)},$$

where $\mu_{X_{t+1}|Y_0=y_0}(x_{t+1}|y_0) = \int \delta_{y_0}(x_0) \prod_{k=1}^{t+1} K(x_k, x_{k-1}) dx_{0:t}$ is the probability density of X_t given $Y_0 = y_0$. Note that in this case the weighting function depends only on x_{t+1}.

By plugging this expression in (20) and (21), we get

$$\hat{P}_A(t) = \sum_{i=1}^{n} \varphi_{y_t, \tilde{X}^{(i)}_{t+1:t+T}} \xi^{(i)}, \tag{24}$$

$$\xi^{(i)} \propto \frac{K(\tilde{X}_{t+1}^{(i)}, y_t)}{\mu_{X_{t+1}|Y_0=y_0}(\tilde{X}_{t+1}^{(i)}|y_0)}, \sum_{i=1}^{n} \xi^{(i)} = 1. \tag{25}$$

The weighting strategy is quite natural in its formulation. We are only interested in the future, given exact knowledge of y_t and the Markov property of the process. Our weighting must therefore skew the distribution $\mu_{X_{t+1}|Y_0=y_0}(x_{t+1}|y_0)$ to give the posterior distribution that would have been obtained had a new simulation been run starting at y_t. Figure 5 give a graphical representation of the weighting process. The weighting function is chosen such that it maps the prior distribution $\mu_{X_{t+1}|Y_0=y_0}(x_{t+1}|y_0)$ onto the posterior distribution $K(x_{t+1}, y_t)$.

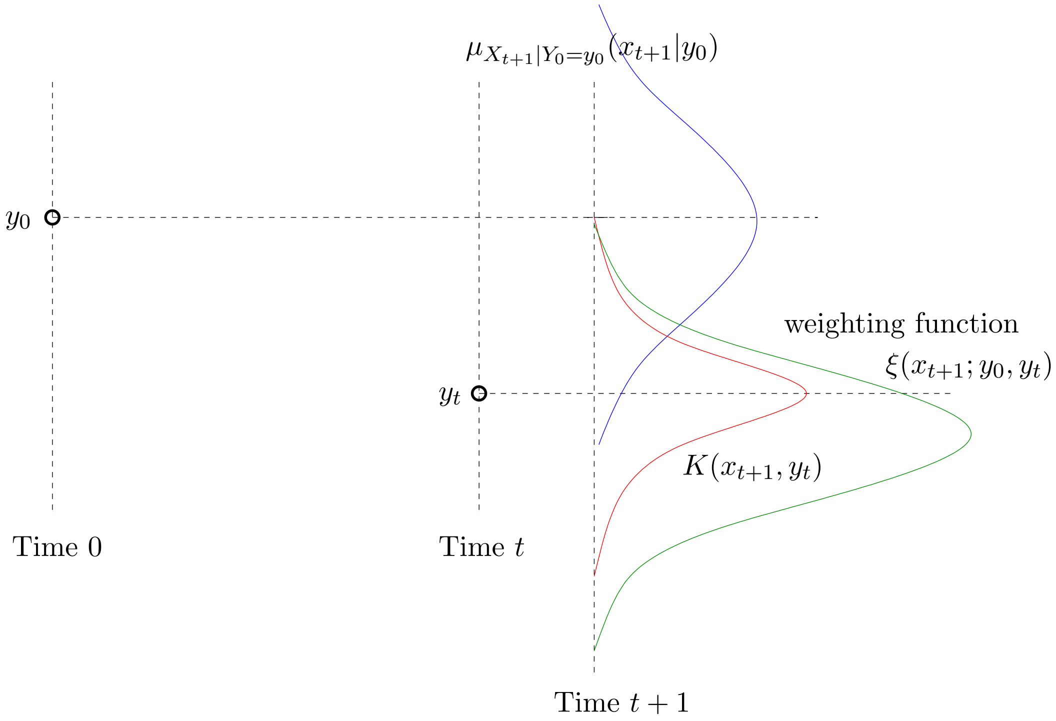

Figure 5: Prior distribution $\mu_{X_{t+1}|Y_0=y_0}(x_{t+1}|y_0)$, posterior $K(x_{t+1}, y_t)$, and weighting function.

Inspection of Figure 5 leads to a natural conclusion on the nature of particle algorithms. If either a particularly unlikely observation is received, or the particle cloud grows in a particularly unlikely way, then the resulting weighting function will concentrate all the weight to one or few particles.

Throughout the above discussion we have assumed the particles were originated at time 0. If, however, we wish to start particles at time t_0 from position y_{t_0} we need simply replace time 0 with time t_0 throughout and the results hold as before. Notational simplicity precluded this in the original discussion.

3.3.2 Implementation of SMC

A potential implementation of the SMC estimates above seems clear: perform Monte Carlo to obtain the necessary sample paths, then repeat SMC estimation at each time step until some time t_{crit} at which n_{ess} drops below some critical threshold $\bar{n}_{ess}$ where accuracy is no longer deemed acceptable. At this point SMC is restarted, by resampling paths initiating from $y_{t_{crit}}$. This procedure is quite appealing in its simplicity, but has the drawback that when $\bar{n}_{ess}$ is set high many resampling steps are required and efficiency is lost. To alleviate this problem we propose a novel method using two particle clouds. When the combined n_{ess} of the two sets drops below a threshold, we discard and resample the set of particles with the smallest n_{ess}. Two sets of particles $\mathcal{S}_1$ and $\mathcal{S}_2$ are sampled, and at each time step their trajectories are extended and checked for conflicts. If necessary the set with the lowest individual n_{ess} is resampled. This process will smooth the joint n_{ess} and reduce the overall computational effort, resulting in a larger number of re-uses for each set of particles. With two particle clouds, however, we will have two estimates $\hat{P}_{A,i}$, $i = 1, 2$, of the probability P_A, which must be combined in some way. We consider the linear combination

$$\hat{P}_A(t) = \alpha \hat{P}_{A,1}(t) + (1-\alpha)\hat{P}_{A,2}(t).$$

The coefficient α should be chosen so as to minimize the variance of the estimator $\hat{P}_A(t)$. Since

$$var(\hat{P}_A(t)) = \alpha^2 var(\hat{PC}_{A,1}(t)) + (1-\alpha)^2 var(\hat{P}_{A,2}(t)),$$

one should choose

$$\alpha^\star = \frac{var(\hat{P}_{A,2}(t))}{var(\hat{P}_{A,2}(t)) + var(\hat{P}_{A,1}(t))}.$$

By interpreting $n_{ess,i}$ as the number of independent samples in a standard MC estimate of P_A to obtain the same quality as the SMC estimate $\hat{P}_{A,i}(t)$ ([44]), we can approximate $var(\hat{P}_{A,i}(t))$ with $\frac{1}{n_{ess,i}} P_A(t)(1-P_A(t))$, which in turn can be approximated by $\frac{1}{n_{ess,i}} \hat{P}_{A,i}(t)(1-\hat{P}_{A,i}(t))$. This finally leads to the following choice for α:

$$\alpha = \hat{\alpha}^\star = \frac{n_{ess,1}\hat{P}_{A,2}(t)(1-\hat{P}_{A,2}(t))}{n_{ess,1}\hat{P}_{A,2}(t)(1-\hat{P}_{A,2}(t)) + n_{ess,2}\hat{P}_{A,1}(t)(1-\hat{P}_{A,1}(t))}. \tag{26}$$

In combining the $\hat{P}_{A,1}$ and $\hat{P}_{A,2}$ estimates we are essentially performing a two-step weighting procedure, which results in weighting cloud 1 by $\alpha\xi_1$ and cloud 2 by $(1-\alpha)\xi_2$. The effective sample size of the combined estimate is therefore given by:

$$n_{ess} = \frac{1}{\alpha^2 \sum_{i=1}^n \left(\xi_1^{(i)}\right)^2 + (1-\alpha)^2 \sum_{i=1}^n \left(\xi_2^{(i)}\right)^2},$$

which reduces to:

$$n_{ess} = \frac{[n_{ess,2}\hat{P}_{A,1}(t)(1-\hat{P}_{A,1}(t)) + n_{ess,1}\hat{P}_{A,2}(t)(1-\hat{P}_{A,2}(t))]^2}{n_{ess,1}[\hat{P}_{A,2}(t)(1-\hat{P}_{A,2}(t))]^2 + n_{ess,2}[\hat{P}_{A,1}(t)(1-\hat{P}_{A,1}(t))]^2}. \tag{27}$$

Having established these results SMC probability estimation with two sets of particles may be performed with a horizon of T, for some time τ, using the following algorithm.

```
Algorithm 2 (Efficient MC estimation of $P_A(t)$, $t = 0, 1, \dots, \tau$)
Initialization:
Generate $\mathcal{S}_1$ and $\mathcal{S}_2$ by extracting the i.i.d. sample paths $\tilde{X}^{(i)}_{j,0:T}$ $i = 1, \dots, n$, $j = 1, 2$ from $\delta_{y_0}(x_0) \prod_{k=1}^{T} K(x_k, x_{k+1})$.
Set $t_0 = 0$ and $t = 0$.
While $t < \tau$
  Repeat
    For $j = 1, 2$ compute
        $\hat{P}_{A,j}(t) = \sum_{i=1}^{n} \varphi_{y_t, \tilde{X}^{(i)}_{j,t+1:t+T}} \xi^{(i)}_j,$
        $n_{ess,j} = \frac{1}{\sum_{i=1}^{n} \left(\xi^{(i)}_j\right)^2}$
        $\xi^{(i)}_j \propto \frac{K(\tilde{X}^{(i)}_{j,t+1}, y_t)}{\mu_{X_{t+1}|Y_{t_0}=y_{t_0}}(\tilde{X}^{(i)}_{j,t+1}|y_{t_0})}, \sum_{i=1}^{n} \xi^{(i)}_j = 1.$
    Compute
        $\hat{P}_A(t) = \alpha \hat{P}_{A,1}(t) + (1-\alpha) \hat{P}_{A,2}(t),$
        $\alpha = \frac{n_{ess,1} \hat{P}_{A,2}(t)(1 - \hat{P}_{A,2}(t))}{n_{ess,1} \hat{P}_{A,2}(t)(1 - \hat{P}_{A,2}(t)) + n_{ess,2} \hat{P}_{A,1}(t)(1 - \hat{P}_{A,1}(t))}.$
    Set
        $n_{ess} = \frac{[n_{ess,2} \hat{P}_{A,1}(t)(1 - \hat{P}_{A,1}(t)) + n_{ess,1} \hat{P}_{A,2}(t)(1 - \hat{P}_{A,2}(t))]^2}{n_{ess,1} [\hat{P}_{A,2}(t)(1 - \hat{P}_{A,2}(t))]^2 + n_{ess,2} [\hat{P}_{A,1}(t)(1 - \hat{P}_{A,1}(t))]^2}.$
    $t \leftarrow t + 1$
     Receive $y_t$
    Extend sample paths to $T + t$: $X^{(i)}_{j,t+T} \sim K(x_{T+t}, X^{(i)}_{j,t+T-1})$, $j = 1, 2$.
  Until $n_{ess} < \bar{n}_{ess}$
Resample $\mathcal{S}_j$ with lowest $n_{ess,j}$ by extracting the i.i.d. sample paths $\tilde{X}^{(i)}_{j,t:t+T}$ $i = 1, \dots, n$, $j = 1, 2$ from $\delta_{y_t}(x_t) \prod_{k=t+1}^{t+T} K(x_k, x_{k+1})$.
Set $t_0 = t$.
end
```

If the process X_k is a discrete time version of an underlying continuous time process $X(t)$, then one should check if it is the case that the discrete time state X_k 'jumps over' the target set, while the continuous state $X(t)$ passes through it, not to underestimate the true probability of interest.

Suppose that $X(t)$ is governed by a SDE and X_k is obtained by some discretization scheme. Firstly we must ensure that Δt, the time step of the discrete time SDE, is small enough that the trajectory may be reasonably approximated by a straight line. We can then check whether if the straight line between consecutive points enters the target set:

$$\{\gamma x_k + (1-\gamma)x_{k-1} : \gamma \in [0,1]\} \cap A \neq \emptyset.$$

To account for this function φ can be modified as follows:

$$\varphi_{x_{t:T+t}} = \max_{k\in[t:T+t]} I_k = \begin{cases} 1, \ \{\gamma x_k + (1-\gamma)x_{k-1} : \gamma \in [0,1]\} \cap A \neq \emptyset \\ 0, \ \text{otherwise.} \end{cases}$$

For large values of Δt relative to A, this refinement gives valuable improvements in accuracy. Although this may be implemented analytically, with $2n$ line segments to check per time step (number of particle clouds × number of particles), it can be time consuming. A method to reduce the complexity of this checking is to first verify if the discrete time sampled path already enters A and apply the more complex expression for φ only to the remaining sample paths.

4 Application to probabilistic aircraft conflict detection

4.1 Models of the aircraft motion

One of the difficulties in predicting the aircraft future position consists in modeling the perturbations influencing its motion. The actual motion of the aircraft is in fact affected by uncertainty, due mainly to wind, but also to errors in tracking, navigation, and control. On a mid-term time horizon uncertainty cannot be neglected.

Paielli & Erzberger [15] conducted an extensive study, comparing track data to initial flight plans and on this basis proposed a set of statistics for flight path deviations. The resulting description of the global effect of the perturbations affecting the aircraft motion over a 20 minutes time horizon is given distinguishing the resultant uncertainty in the *along-track* and in the *cross-track* directions. Specifically, the tracking errors are described as zero mean Gaussian random variables with the variance of the along-track component, $\sigma_a^2(t)$, growing quadratically with time:

$$\sigma_a^2(t) \sim r_a^2 t^2, \tag{28}$$

the variance of the horizontal cross-track component, $\sigma_c^2(t)$, growing quadratically with the traveled distance, $s(t)$, and then saturating at a fixed value:

$$\sigma_c^2(t) \sim r_c^2 s^2(t), \quad sat\{\sigma_c^2(t)\} = \bar{\sigma}_c^2, \tag{29}$$

and, finally, the variance of the vertical cross-track component remaining constant. The instantaneous position in the local coordinate frame of each aircraft (origin at initial position, axes aligned with along-track and cross-track directions) has the following distribution:

$$\chi(t) \sim \mathcal{N}(\mu(t), \bar{V}(t))$$

In which $\mu(t)$ describes the nominal reference position, and $\bar{V}(t)$ the position variance, given by:

$$\bar{V}_i(t) = \begin{bmatrix} \sigma_a^2(t) & 0 \\ 0 & \sigma_c^2(t) \end{bmatrix}.$$

The most widely adopted values for these figures in the conflict detection literature are those for an ensemble of all types of aircraft, with $r_a = 0.25$ nmi/min, $r_c = 1/57$ and $\bar{\sigma}_c = 1$ nmi. In terms of the separation minima these values are substantial, for instance after 20 minutes an aircraft is *expected* to be $5nmi$ off course in the along track direction.

In [6], it is argued that this model is fairly accurate for predicting the position of an aircraft over a mid-term horizon, as it reflects the fact that the flight management systems correct cross-track errors in the short term, while pilots deal with along-track errors in the long term, using small changes in speed.

In this work we consider two models from the literature. The first is a simple model which does not take into account the aircraft dynamics, and has independent wind disturbances acting at each step. The second is a substantially more complex model which attempts to replicate aircraft dynamics as closely as possible, and also includes correlated wind disturbances whereby the wind innovation at a particular time step will tend to have the same magnitude and direction as those at previous time steps. Both models are tuned to fit Paielli & Erzberger statistics on flight plan deviations.

We describe more in detail the first model, since it is the one used for aircraft conflict detection.

The second model is only used for evaluating the performance of the proposed conflict detection method and is the model implemented in the simulator described in [19, 21] and proposed for, amongst other things, the evaluation of conflict detection method. The model implemented in the simulator is more sophisticated in a number of ways than other existing evaluation models. The deterministic dynamics of the model are based on the BADA Total Energy Model (TEM), which has been shown to provide an accurate model of aircraft flight, including procedural factors, such as rate of climb or descent, standard thrusts and airspeeds at different altitudes and realistic modeling of turns. Perhaps a more significant contribution is the inclusion of temporally and spatially correlated wind disturbances affecting the positions of aircraft sharing the same areas of the airspace.
Analysis of data from the rapid update cycle, a weather system operated by t he National Center for Environmental Prediction in the USA, has enabled estimation of a spatial correlation function and its coefficients. Over the range of a few thousand kilometers the following expression gives good approximation of wind correlation in the horizontal plane for aircraft at the same flight level.
For a description of the simulator the reader is referred to [19, 21].

4.1.1 Aircraft model for conflict probability estimation

A simple model we choose to implement here is characterized by the fact that wind disturbances are independent at each time step. In composing this model we assume that we have constant nominal heading angle and velocity. This assumption is maintained throughout this work; while not entirely necessary, it is a realistic reflection of aircraft flight between waypoints. A minimum requirement is some a priori knowledge of nominal heading angle and velocity, such that predictions of nominal position may be made.

The state in the local coordinate frame is given by the along track and cross track positions; $\chi(t) = [\chi_a(t), \chi_c(t)]^T$. Wind is assumed to be independent, which gives rise to the following model:

$$\dot{\chi}(t) = Dv + \eta(t), \tag{30}$$

in which $D = [1, 0]^T$, v is the velocity and $\eta(t)$ represents the wind innovations, acting additively on state; $\eta(t) = [W_a, W_c]^T$. W_a and W_c are noise processes ideally characterized by:

$$\int_0^t W_a dt \sim \mathcal{N}(0, r_a^2 t^2),$$
$$\int_0^t W_c dt \sim \mathcal{N}(0, min\{\bar{\sigma}_c, r_c^2 s^2(t)\}),$$

therefore giving rise to P&E deviation statistics. In general $\eta(t)$ will take the form $\eta(t) = k(t)B(t)$ where $k(t)$ is some time dependent coefficient multiplying $B(t)$; a standard Brownian motion. In order to implement this it is essential that the rate of change of variance is everywhere differentiable ($k(t) = \frac{d}{dt}V(t)$, where V is the time dependent variance). In the case of W_c this is not the case, as there exists some time t at which the variance will saturate. A characterizable approximation may be made, which enables expression of $\eta(t)$:

$$min\{\bar{\sigma}_c, r_c^2 s^2(t)\} \approx \bar{\sigma}_c^2 \left(1 - e^{-2\frac{r_c}{\bar{\sigma}_c}vt}\right),$$

so that

$$\begin{aligned} W_a &= 2r_a^2 t B_a(t), \\ W_c &= 2\bar{\sigma}_c r_c v e^{-2\frac{r_c}{\bar{\sigma}_c}vt} B_c(t), \end{aligned}$$

where $B_a(t)$ and $B_c(t)$ are the two components of the 2-dimensional Brownian motion $B(t)$. The nature of the approximation is such that it initially overestimates the cross track variance (w.r.t. the original model), then underestimates it as time increases. The overall effect is that similar trajectories are produced. The P&E statistics are in any case an approximation. This model is similar to that presented in [14], but includes time varying noise in the cross-track direction, which better reflects the lower posterior uncertainty in predicting cross-track deviations after several observations have been made.

At this point we may observe some of the salient features of this model. Firstly we note the implicit inclusion of the effects of cross-track control in the cross-track variance. The saturation of cross-track error variance occurs due to control effort to bring the aircraft back on course. We also note the implicit inclusion of wind correlation in the across track variances. Were the errors caused by a Brownian motion type of uncertainty the variance would grow in proportion to time. In this case, however, the variance grows quadratically with time as a result of the fact that in the real system disturbances are likely to be similar at consecutive time steps. While this control and correlation are reflected in the structure of the model it is important to note that control and correlated disturbances are not features of the model.

In order to build a two-aircraft system model, we add a subscript i to all the quantities. The reference heading angle of each aircraft, in the global coordinate frame, is defined as θ_i. The position of each aircraft in the global coordinate frame, relative to the last waypoint (also defined in the global coordinate frame), is therefore distributed according to:

$$\begin{aligned} X_i(t) &\sim \mathcal{N}(O_j + R(\theta_i)Dv_i, V_i(t)), \\ V_i(t) &= R(\theta_i)\bar{V}_i(t)R(\theta_i), \\ \bar{V}_i(t) &= \begin{bmatrix} r_a^2 t^2 & 0 \\ 0 & \bar{\sigma}_c^2(1 - e^{-2\frac{r_c}{\bar{\sigma}_c}v_i t}) \end{bmatrix} \\ R(\theta_i) &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix}. \end{aligned}$$

$R(\theta_i)$ is a rotation matrix mapping positions from the local to global coordinate frame, and O_j is the location of the j^{th} waypoint. There is no modeling of the correlation of wind perturbations between aircraft, so the dynamics of the relative separation of two aircraft are given by:

$$\dot{X} = R(\theta_2)Dv_2 - R(\theta_1)Dv_1 + \eta_r(t)$$

Where v_1 and v_2 are the velocities of each aircraft and $\eta_r(t) = R(\theta_2)\eta_2(t) + R(\theta_1)\eta_1(t)$. $\eta_i(t)$ refers to the disturbances acting on each aircraft. From this equation we then have:

$$X(t) \sim \mathcal{N}(\mu_r(t), V_2(t) + V_1(t)),$$
$$\mu_r(t) = O_j^{(2)} - O_j^{(1)} + R(\theta_2)Dv_2 - R(\theta_1)Dv_1.$$

On top of the implicit control of cross track variance discussed above, limited control of cross track deviations is implemented in this model through maintenance of the flight plan. At any time all way points that have been passed are discarded and the current position of the aircraft is encoded as the first waypoint, the heading (θ_i) is set towards the next waypoint. No similar control is implemented in the along-track direction, leading to the unbounded quadratic growth in position variance in this direction.

4.2 Sequential Monte Carlo method for conflict probability estimation

We adopt the SMC Algorithm 2 described in Section 3.3 for estimating the probability of conflict for two-aircraft level flight encounters. In this context, the process X represent the relative aircraft position and Y represents the radar measurements of the aircraft relative position.

This algorithm is composed of the following major steps:

1. Sample n trajectories from initial position y_0, estimate $P_A(0)$ accordingly using standard Monte Carlo.
2. On receiving an observation y_t, weight the original sample trajectories such that an estimate of $P_A(t)$ may be obtained using Sequential Monte Carlo.
3. When t is sufficiently large that the accuracy of the estimate $P_A(t)$ is no longer acceptable, resample n new trajectories with initial position y_t.

While other steps are involved, associated with the complexities of handling two particle clouds, the steps above encapsulate the main issues in implementing SMC. We next describe how these issues can be solved with reference to the model described in subsection 4.1.1.

This model in discrete time takes the form

$$X_{k+1} = X_k + (R(\theta_2)Dv_2 - R(\theta_1)Dv_1)\,\Delta t + W_k, \tag{31}$$

with W_k, $k = 0, 1, \ldots$, independent random variables distributed according to:

$$W_k \sim \mathcal{N}(0, V_1((k+1)\Delta t) + V_2((k+1)\Delta t) - V_1(k\Delta t) - V_2(k\Delta t)).$$

The radar sampling interval is typically 12 seconds, although some systems operate with a sampling interval of 6 seconds. in this work we consider it 12 seconds.

In the context of aircraft conflict prediction $P_A(t)$ is the probability of conflict at time t: the Y and X processes represent the aircraft relative position, and $P_A(t)$ is the probability of the relative position becoming smaller than a safety distance. For this reason we shall refer to it as $PC(t)$.

4.2.1 Extracting sample paths

Each time sampling or resampling requires extracting sample paths from the following distribution:

$$\delta_{y_t}(x_t) \prod_{k=t+1}^{t+T} K(x_k, x_{k-1}) \tag{32}$$

The transition kernels are not usually explicitly calculated. In the case of (31), it is easily seen that

$$\begin{aligned} &K(x_{k+1}, x_k) = \mathcal{N}\left(x_k + \left(R(\theta_2)Dv_2 - R(\theta_1)Dv_1\right)\Delta t, V_{r,k}\right), \\ &V_{r,k} = \left(V_1((k+1)\Delta t) + V_2((k+1)\Delta t)\right) - \left(V_1(k\Delta t) + V_2(k\Delta t)\right). \end{aligned}$$

Having formulated the dynamics in this form generation of large numbers of trajectories becomes straightforward. Appropriate application of data assignment structures available in programming tools (e.g. Matlab) enable very fast generation of large numbers of independent trajectories, as required for MC estimation.

4.2.2 Computing the particles weights

When applying weights at time step t we wish to evaluate the expression:

$$\frac{K(x_{t+1}, y_t)}{\mu_{X_{t+1}|Y_{t_0}=y_{t_0})}(x_{t+1}|y_{t_0})}.$$

Now we have that

$$K(x_{t+1}, y_t) = \mathcal{N}\left(y_t + \left(R(\theta_2)Dv_2 - R(\theta_1)Dv_1\right)\Delta t, V_{r,t}\right),$$

whereas

$$\mu(x_{t+1}, y_{t_0}) = \mathcal{N}(\hat{x}_{t+1}, V_r(t+1, t_0)).$$

In the latter expression,

$$\hat{x}_{t+1} = y_{t_0} + \left(R(\theta_2)Dv_2 - R(\theta_1)Dv_1\right)\Delta t(t+1-t_0),$$

where $R(\theta_2)Dv_2 - R(\theta_1)Dv_1$ is the relative velocity and $(t+1-t_0)$ the time over which the distribution has developed. Similarly,

$$V_r(t+1, t_0) = \left(V_1((t+1)\Delta t) + V_2((t+1)\Delta t)\right) - \left(V_1(t_0\Delta t) + V_2(t_0\Delta t)\right),$$

As mentioned before we maintain the assumption of constant velocity and heading. In the general setting this is not necessary, the sole necessity for application of weights using this scheme being knowledge of all the transition kernels over the prediction horizon. The formulation presented above is dependent on the flight plans being straight lines. There is no barrier, however, to the formulation of multi-step transition probabilities for general flight plans. Various authors have tackled conflict detection for more general flight plans, for instance [14].

4.3 Alerting logic and performance measure

In most approaches to conflict detection described in the literature, alerting logic is not given a great deal of attention. In [6, 16] for example, no alerting strategy is given: the probabilistic method concludes with calculation of conflict probability. Alerting logic describes the method used to make the binary decision of whether or not to alert air traffic control to an impending conflict. In more sophisticated future architectures it may be that conflict probabilities are used as a cost in some form of optimal resolution strategy. In the current air space architecture, however, alerting will remain a binary decision, so irrespective of the method behind making the alert decision (geometric, probabilistic etc.) a 1/0 decision must be made at some point.

In [23], the concept of a System Operating Characteristic (SOC) curve is introduced for conflict probability. We now review the SOC trade off between false alerts and successful alerts as a baseline for tuning an alerting logic, pointing out the issue of the geometrical sensitivity. [23] has considered the SOC approach when the criticality measure is the probability of conflict. It is worth noticing that the SOC approach can be combined with other criticality measures such as, for example, collision probability. Actually, the SOC approach can be applied to any kind of well defined statistical decision problem.

4.3.1 SOC curve

An SOC curve is obtained by plotting the probability of false alert against the probability of successful alert, parameterized by the alert threshold (an example of SOC curve is shown in Figure 7). The trade-off between false and successful alerts is suggested in [23] as a method for evaluating the trade-off between safety and workload in conflict alerting systems.

We are interested in four closely-related event classes; conflict, alert, successful alert and false alert. With reference to some flight time horizon $[0, T_f]$, these may be defined as follows:

1. **Conflict.** The aircraft pair under consideration enter conflict at some time $t \in [0, T_f]$.
2. **Alert.** The conflict detection mechanism issues an alert at some time $t \in [0, T_f]$.
3. **Successful Alert.** A conflict alert is issued; the aircraft pair under consideration subsequently enter conflict.
4. **False Alert.** An alert is issued, although the aircraft pair under consideration do not subsequently enter conflict.

We wish to investigate the trade-off between probability of false alert, $\mathcal{P}(FA)$, and probability of successful alert, $\mathcal{P}(SA)$. In order to cast this investigation in a more rigorous setting we introduce the conflict detection time-line in Figure 6. This figure shows window in which conflict alerts may be made. The look-ahead cannot be greater than the overall horizon, and an alert is successful if it is issued at some time t_w before a conflict occurs.

We now define some useful concepts in evaluating a conflict detection scheme (conflict probe) performance. We first define a realization of the observation process trajectory over a flight time

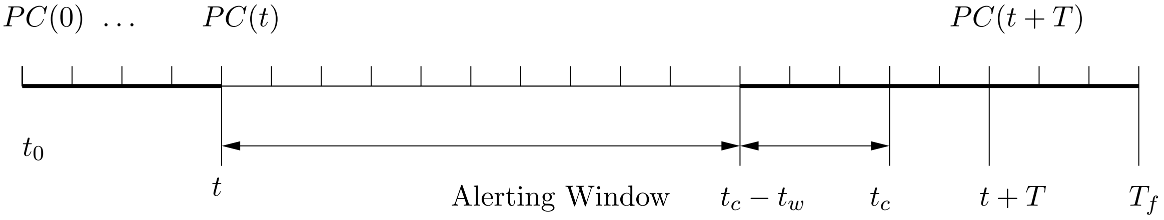

Figure 6: Time-line for reduced horizon conflict detection. Only those probabilities of conflict calculated at times earlier than $t_c - t_w$ are relevant for successful alerts.

horizon $[0, T_f]$:

$$y_i(\cdot) : \{0, 1, \ldots, T_f\} \longrightarrow \mathbb{R}^2.$$

As a function of this trajectory, we may define an indicator function $I_C(\cdot)$ returning a 1 if the realization enters conflict, 0 otherwise:

$$I_C(y_i(\cdot)) = \{\exists t \in \{0, \ldots, T_f\} : y_i(t) \in A\}.$$

The measure of criticality used in our conflict detection scheme is the estimate of the probability of conflict $PC(t)$ obtained by the SMC method. This will depend on the encounter configuration, say γ, and on the realization of the observation process. This consideration is valid for other measures of criticality. We generally refer to a criticality measure as $C(y_i(\cdot), \gamma, t)$ and to the alert threshold as $\bar{C}$. This enables us to define the indicator function for alert, I_A:

$$I_A(y_i(\cdot); \bar{C}, \gamma) = \{\exists t \in \{0, \ldots, T_f\} : C(y_i(\cdot), \gamma, t) \geq \bar{C}\},$$

where $\bar{C}$ and γ have to be interpreted as parameters. The definitions of indicator functions for successful and false alerts are more complex, as there is a limited range of times over which the alert may be given in order to be successful. As shown in Figure 6 we must alert at some time t_w before the first time at which the realization is in conflict, t_c. This required warning time is a procedural constraint in ATC; alerts issued very shortly before conflict are of no use to ATC as no resolution action may be taken. Given this constraint the indicator function for successful alert is defined as:

$$\begin{aligned} I_{SA}(y_i(\cdot); \bar{C}, \gamma, t_w) = \{\exists t_c \in \{0, \ldots, T_f\} :& (\forall s \in \{0, \ldots, t_c - 1\}(y_i(s) \notin A) \wedge (y_i(t_c) \in A)) \\ & \wedge (\exists \tau \in \{0, \ldots, t_c - t_w\} : C(y_i(\cdot), \gamma, \tau) \geq \bar{C})\}. \end{aligned}$$

Defining the false alert indicator function is more straightforward as there is no restriction on times when a false alert may be issued:

$$\begin{aligned} I_{FA}(x(\cdot); \gamma, \bar{C}) &= \{\exists t \in \{0, \ldots, T_f\} : (C(y_i(\cdot), \gamma, t) \geq \bar{C}) \wedge \forall \tau \in \{0, \ldots, T_f\}(y_i(\cdot) \notin \mathcal{T})\}, \\ &= I_A(y_i(\cdot); \bar{C}, \gamma) \bar{I}_c(y_i(\cdot)). \end{aligned}$$

where $\bar{I}$ represents the event 'not I'. Given these indicator functions we may approximate $\mathcal{P}_{FA}$ and $\mathcal{P}_{SA}$ through MC simulation:

$$\mathcal{P}_{FA}(\gamma, \bar{C}) \approx \frac{\sum_{i=1}^{M} I_A(y_i(\cdot), \bar{C}, \gamma) \bar{I}_c(y_i(\cdot))}{\sum_{i=1}^{M} \bar{I}_c(y_i(\cdot))}, \tag{33}$$

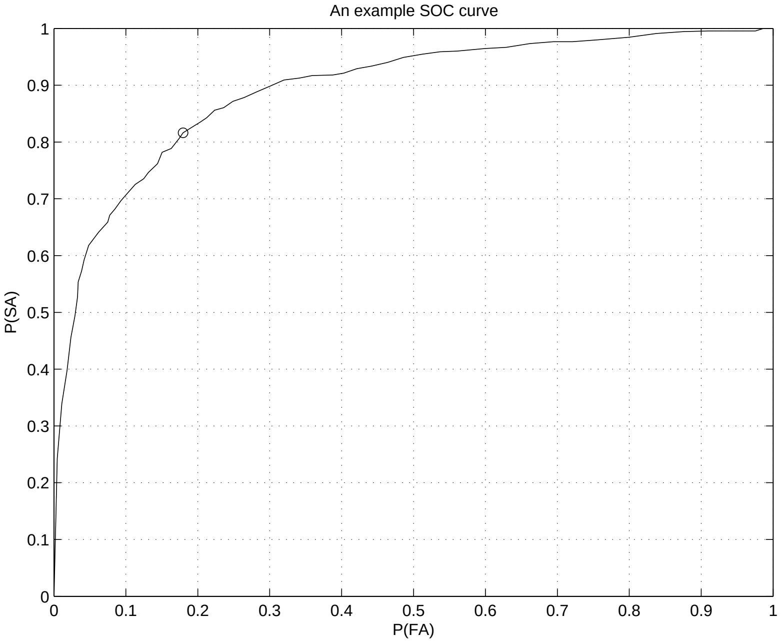

Figure 7: An example SOC curve. The '∘' shows the point closest to [0,1].

$$\mathcal{P}_{SA}(\gamma, t_w, \bar{C}) \approx \frac{\sum_{i=1} I_{SA}(y_i(\cdot), \bar{C}, \gamma)}{\sum_{i=1}^{M} I_C(y_i(\cdot))}. \tag{34}$$

The SOC curve interpretation of these two probabilities investigates the trade-off between the two as a function of $\bar{C}$, the alert threshold. By plotting $\mathcal{P}_{FA}$ against $\mathcal{P}_{SA}$, parameterized by $\bar{C}$, for a fixed γ, an SOC curve is generated. Figure 7 shows an example SOC curve. The curve shown in Figure 7 takes into account only the evaluation of possible conflict given full prior information. Other factors such as human error or changes at the strategic level were included in [23]. Throughout this work however, we shall restrict ourselves to SOC curves dependent only on flight plan deviations, with flight plans known a priori. A variety of factors influence the choice of alert threshold, for instance in certain operational situations it may be preferable to issue more alerts to ensure safety, even though the false alert rate will also rise. These operational factors are, however, deemed to be beyond the scope of this work. We shall concern ourselves principally with the *optimal alert threshold* the point on this curve nearest the ideal operation; the point $(0, 1)$, zero probability of false alert, certainty of successful alert. A measure of success of a conflict probe is given by the minimum distance from this point to the SOC curve:

$$d(\bar{C}, \gamma, t_w) = \sqrt{\mathcal{P}_{FA}(\gamma, \bar{C})^2 + (1 - \mathcal{P}_{SA}(\gamma, t_w, \bar{C}))^2},$$

with the optimal alert threshold defined by:

$$\bar{C}_{opt}(\gamma, t_w) = \arg \min_{\bar{C} \in [0,1]} d(\bar{C}, \gamma, t_w). \tag{35}$$

Note that the optimal alert threshold depends on two factors: the airspace configuration γ and the required warning time t_w.

4.3.2 Sensitivity to the encounter geometry

The sensitivity of alert threshold and therefore probe performance to changes in encounter geometry was first investigated in [46], in which the variation of optimal alert threshold with respect to crossing angle, relative velocity and minimum separation was demonstrated.

This dependence naturally suggests the formulation of a new alerting mechanism taking into account the dependence on encounter geometry (the interested reader is referred to [47]).

Here, we investigate the effects of geometric sensitivity by applying a single, 'globally optimal' alert threshold that gives the best overall performance for a constant threshold. The best performing conflict detection methods will be those that both give low vales of $d(\bar{C}_{opt}, t_w)$ (with no dependence on γ) and have small variations in $\bar{C}_{opt}$ with respect to changing geometry. Setting a single threshold appears appealing: by simply assigning one alert threshold the alerting logic becomes much simpler. It is possible, however, using this method that a probe with excellent performance with a threshold tuned for specific geometries would perform poorly overall.

4.4 Performance of the proposed probabilistic conflict detection method

In this section we compare the proposed SMC method for conflict detection (conflict probe) with other methods proposed in the literature. In particular, we consider the Erzberger and Paielli probe (EP,[6]), the Incrossing probe (IC, [32]) and the Randomized probe (RAN, [14]). The underlying models for evaluating the criticality measures (probability of conflict, incrossing probability, and maximum instantaneous probability, respectively) are tuned to fit Paielli & Erzberger statistics on the tracking errors [15].

It is important to observe that the incrossing method is actually not conceived for conflict detection, but for risk assessment in ATM with reference to aircraft collisions. We shall verify that the other probes will perform better than the incrossing probe under the SOC curve for conflict probability. However, although we do not verify it here, we expect that the incrossing probe would perform better than the other probes under the SOC curve for collision probability. Also, since the incrossing method is designed to assess the low risk levels (10^{-9} incrossings/hour) associated with aircraft collisions, it is expected to outperform the other methods in risk assessment. Verification of this fact is not pursued in this work.

We assess the performance of the considered probes according to the SOC curve, on trajectories obtained by the simple model described in 4.1.1 and by the high complexity model implemented in the simulator developed in [19, 21]. We shall refer to these two models for performance assessment as SDE and HC model, respectively.

We refer to the proposed probe as SMC-SDE.

For assessing the performance, we follow this procedure:

Choose a set of initial conditions similar to those used in studies in the literature. At each point in this set perform sufficient simulations to achieve convergence to the properties of each method for that particular geometry.

The results presented in this section take the form of SOC curves, generated by performing MC simulation of conflict detection on validation data for each of the methods under consideration, as described in equations (33) and (34). The geometries and assessment method are chosen based on other assessments made in the literature (e.g. [14, 6]). In each case the required warning time t_w is set to 1 minute.

The simulations were performed with all combinations of initial geometries with nominal minimum separation of $4, 6, 8$ and 10 nmi, crossing angles of $0, 30, 60, 90, 120$ and 150 degrees and time to minimum separation of $10, 14$ and 18 min, giving 72 simulation settings.

4.4.1 SDE complementary model evaluation

The reported figures show a selection of data gathered from simulations obtained by the model described in 4.1.1. The four conflict probes applied in this case are; SMC probe (SMC-SDE), the Erzberger and Paielli probe (EP), the Incrossing probe (IC) and the Randomized probe (RAN). The probes all perform conflict detection on the same validation process trajectories; the alert state of each of the probes is monitored.

Increasing time to minimum separation: We report in Figures 8, 9, 10 the SOC curves for 90° crossing angle, 6 nmi nominal miss distance and increasing time to minimum separation.

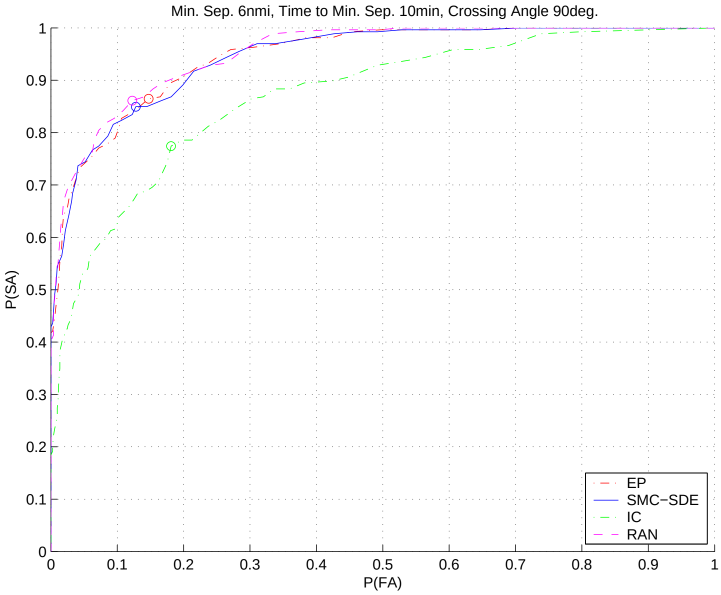

Figure 8: SOC curves for 90° crossing angle, 6 nmi nominal miss distance and 10 min time to minimum separation.

Increasing nominal miss distance: We report in Figures 11, 12, 13 the SOC curves for 90° crossing angle, 14 min time to minimal separation, and increasing nominal miss distance.

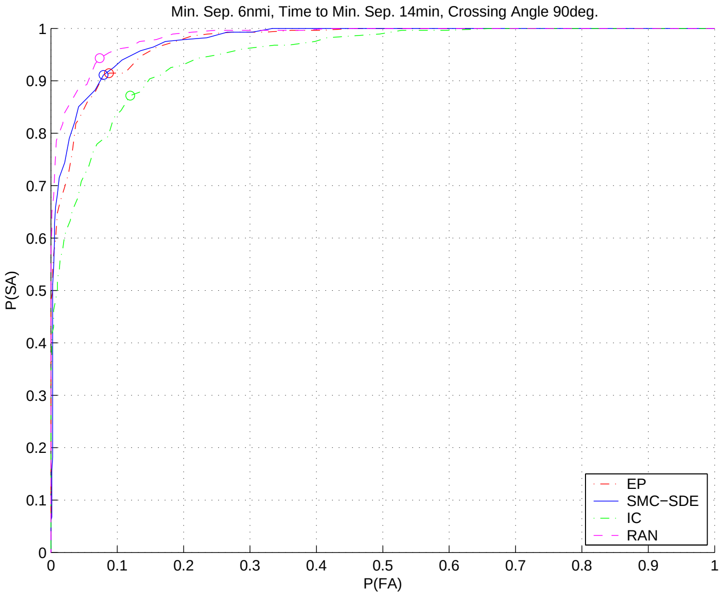

Figure 9: SOC curves for 90° crossing angle, 6 nmi nominal miss distance and 14 min time to minimum separation.

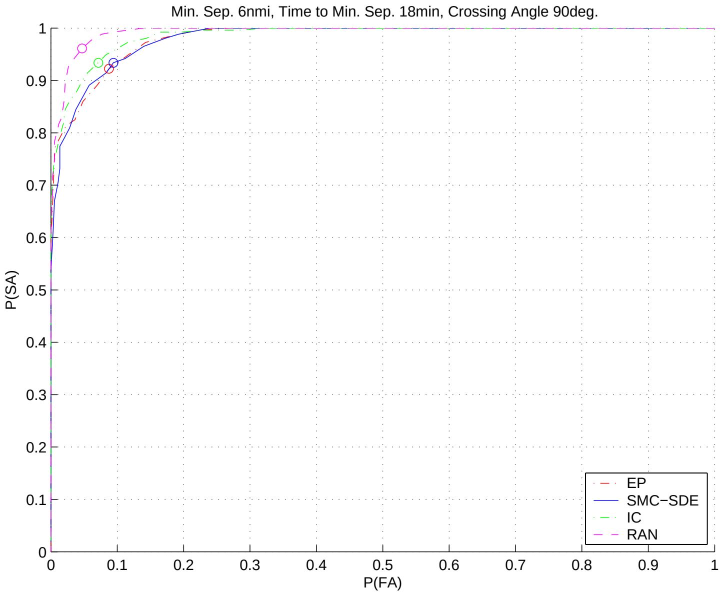

Figure 10: SOC curves for 90° crossing angle, 6 nmi nominal miss distance and 18 min time to minimum separation.

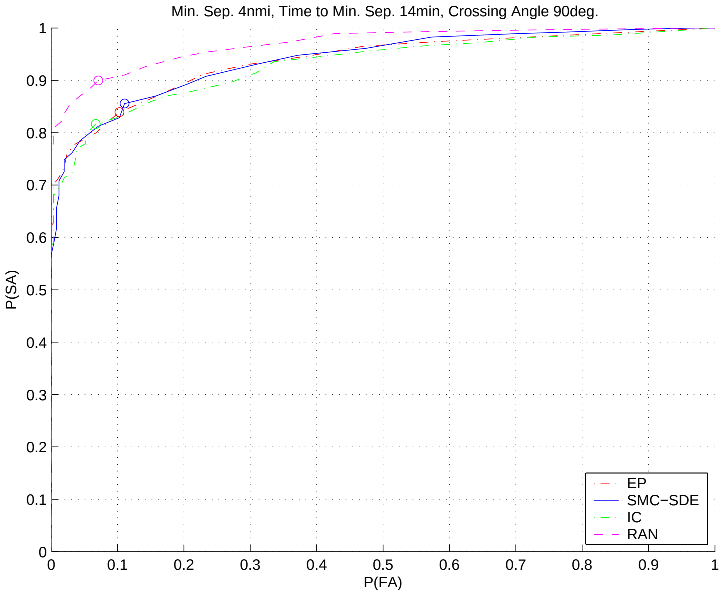

Figure 11: SOC curves for 90° crossing angle, 14 min time to minimal separation and 4 nmi nominal miss distance.

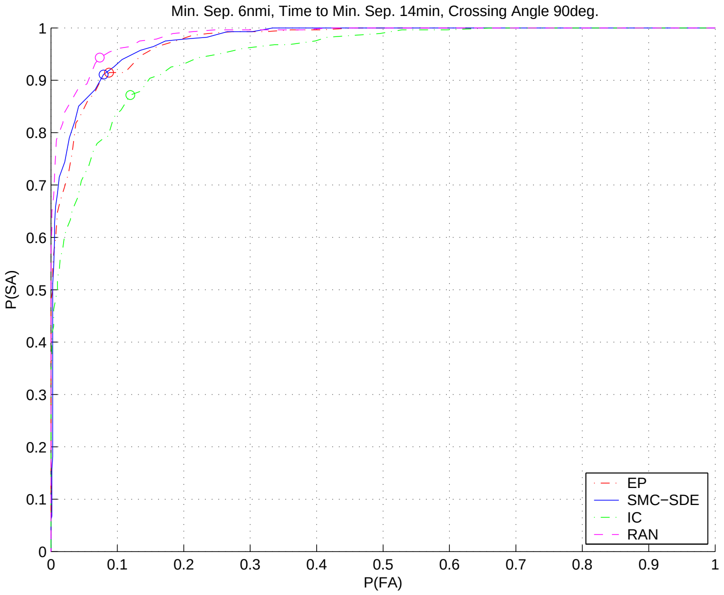

Figure 12: SOC curves for 90° crossing angle, 14 min time to minimal separation and 6 nmi nominal miss distance.

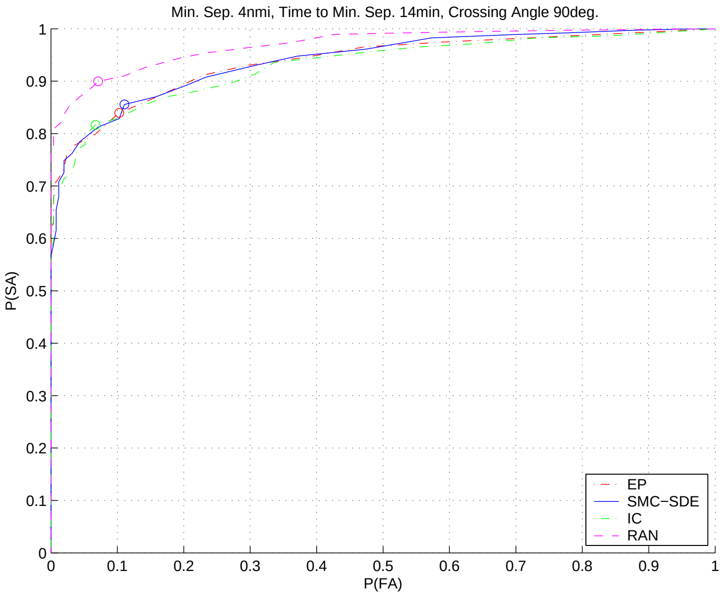

Figure 13: SOC curves for 90° crossing angle, 14 min time to minimal separation and 8 nmi nominal miss distance.

Increasing nominal miss distance: We report in Figures 14, 15, 16, 17, 18 the SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance, and increasing angle of intersection of nominal flight paths.

We now briefly summarize the results gained for this set of experiments.

- The performances of the four probes tested on these data are quite similar. The randomized code performs best overall, providing the best (or equal best) conflict detection in 47 out of the 72 geometries. The SMC-SDE is the next highest performing, outperforming (or matching) all others in 25 geometries. The average value of $d(\bar{C}_{opt}, \gamma, t_w)$ for each probe is shown in Table 3.

Probe	**Mean** $d(\bar{C}_{opt}, \cdot, \cdot)$
Ran	0.0725
SMC-SDE	0.0855
EP	0.097
IC	0.155

Table 3: Ranking of probe performances assessed using SOC curves.

- There is substantial variation in optimal threshold across the tested geometries. This may lead to a degradation in probe quality if a single threshold is adopted for all encounter geometries. A ranking of the conflict probes according to standard deviation of alert threshold is shown in Table 4.

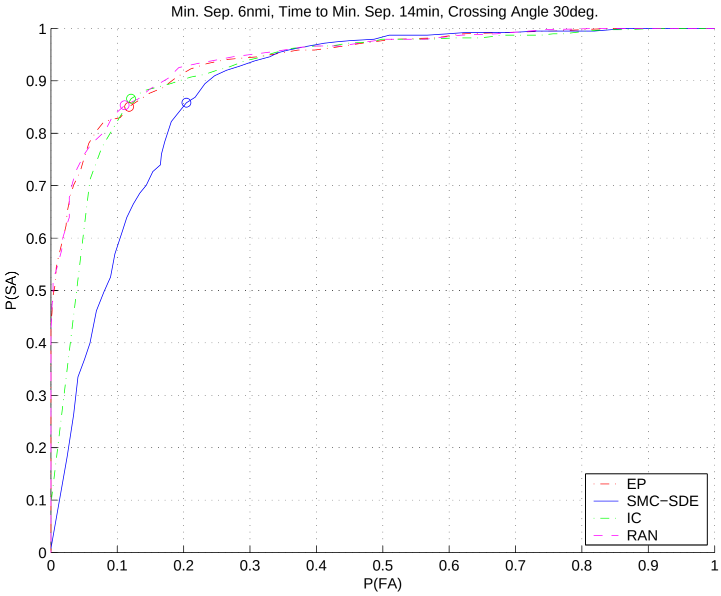

Figure 14: SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance and 30° crossing angle.

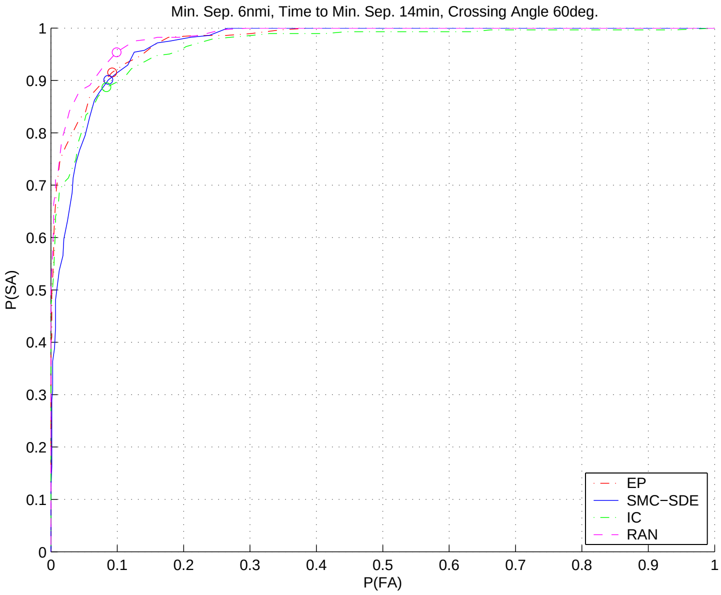

Figure 15: SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance and 60° crossing angle.

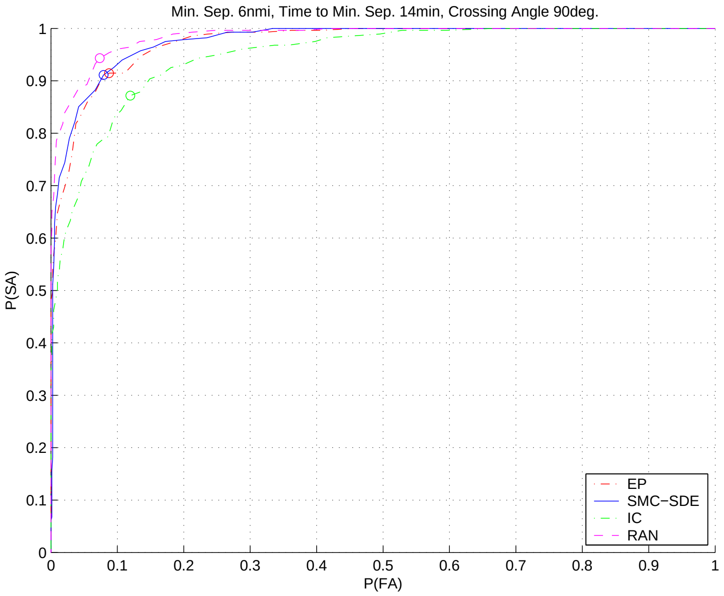

Figure 16: SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance and 90° crossing angle.

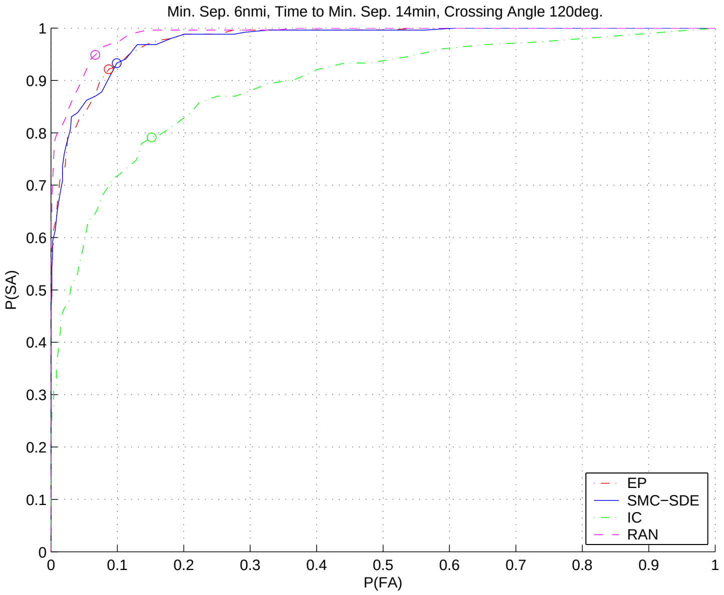

Figure 17: SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance and 120° crossing angle.

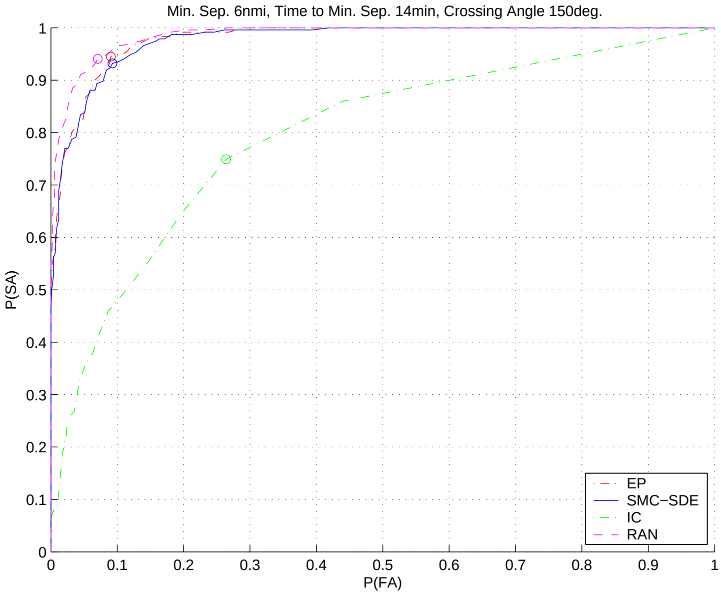

Figure 18: SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance and 150° crossing angle.

Probe	$\sigma\left(\bar{C}_{opt}\right)$
Ran	0.22
EP	0.273
SMC-SDE	0.289
IC	0.361

Table 4: Ranking of standard deviations of alert thresholds.

4.4.2 HC complementary model evaluation

The sets of Figures 19, 20 and 21 show a selection of data gathered from simulations of the high-complexity simulator. The same four conflict probes were used as for the SDE model-based simulation, as well as an additional probe; the high-complexity model SMC probe (SMC-HC) developed in [47], which is obtained by applying the SMC algorithm to the high complexity model implemented in the simulator. This involves some approximation procedure when applying the SMC algorithm, since the state of the process has to include wind samples to be Markov, but then the full information assumption fails.

The same geometries and assessment method were used.

A brief summary of the results is given below:

- In parallel flight (0° crossing angle), either very few conflicts occurred or every simulation resulted in conflict. Which of these occurred is predicated on the initial separation. Due to the strong wind correlation those trajectories which started far apart tended not to converge. Those which started close together were, or course, already in conflict. Mean-

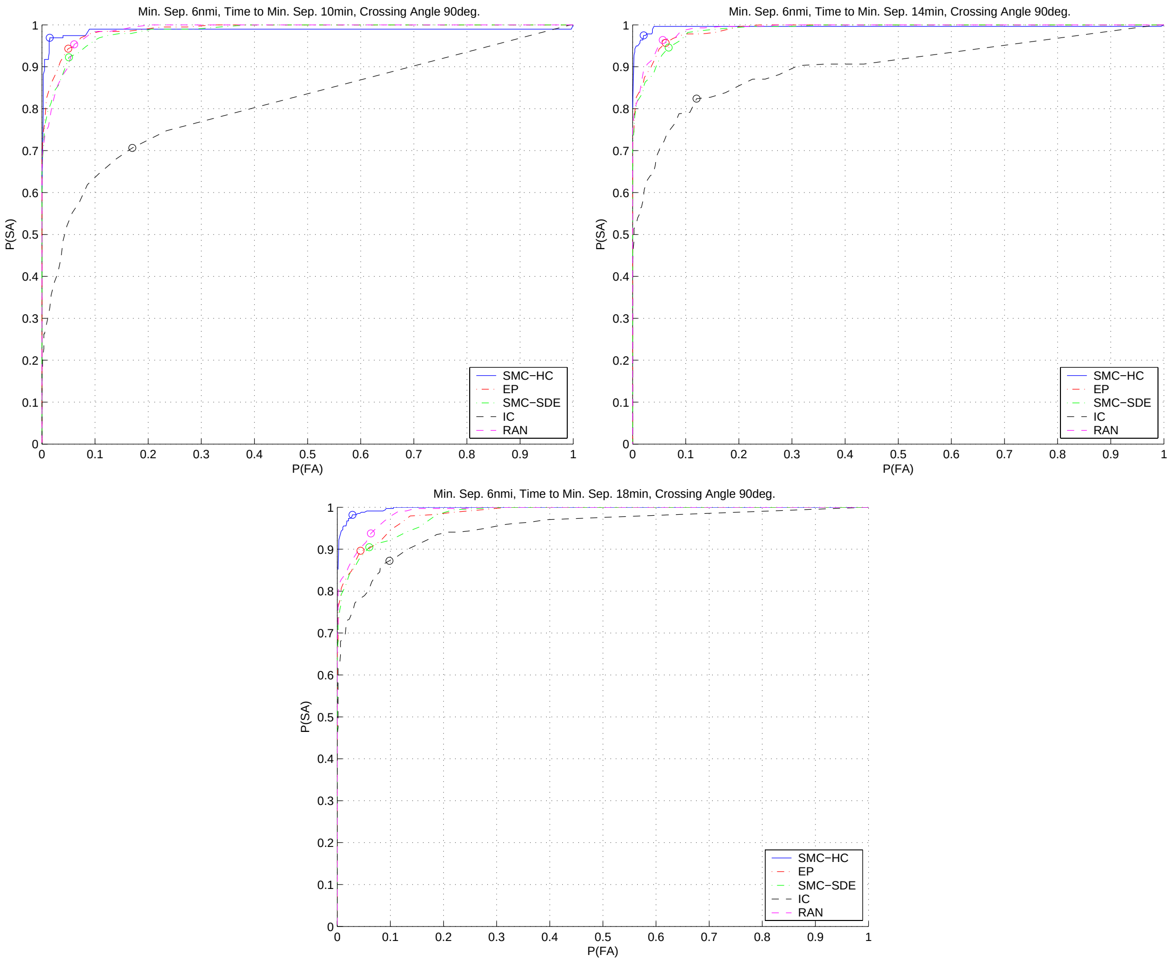

Figure 19: SOC curves for 90° crossing angle, 6 nmi nominal miss distance and increasing time to minimum separation.

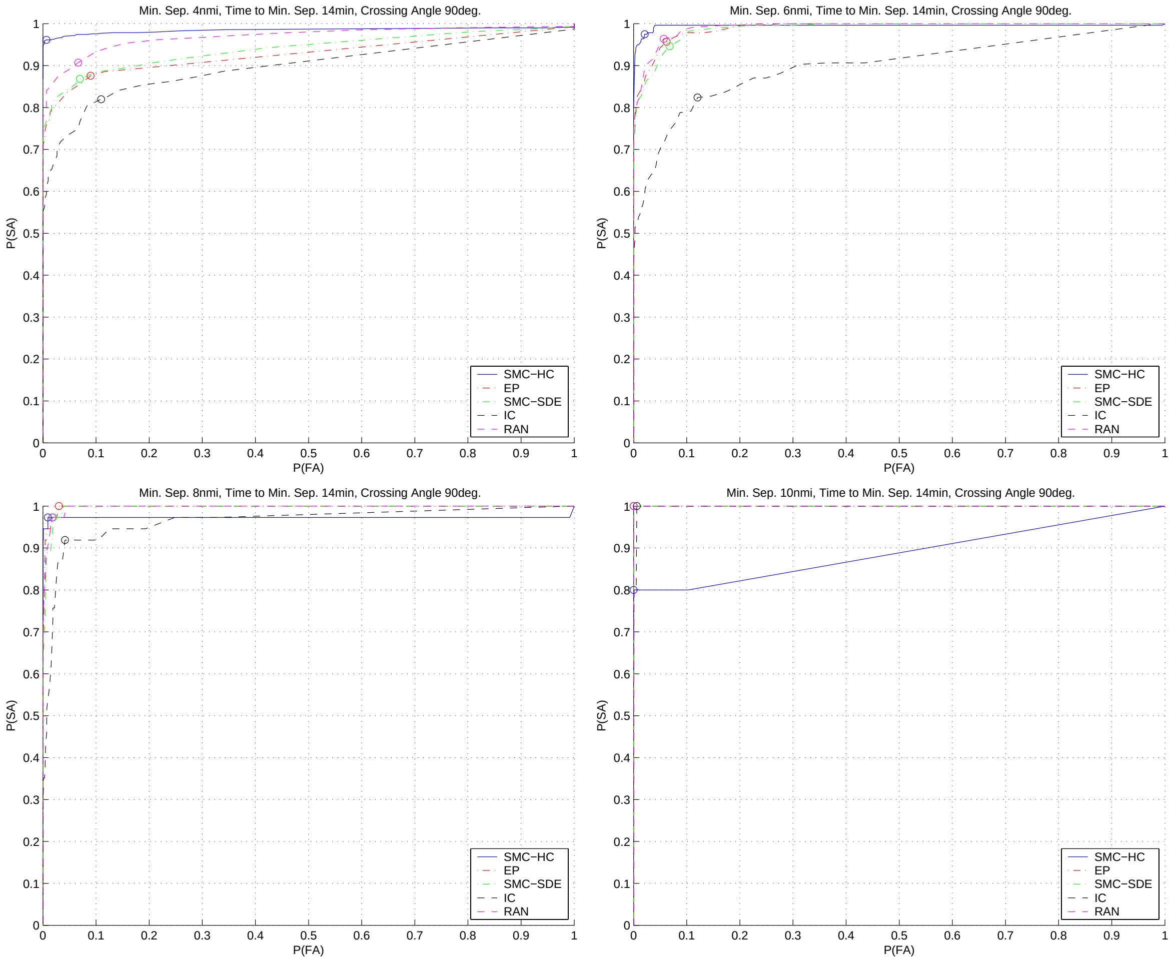

Figure 20: SOC curves for 90° crossing angle, 14 min time to minimal separation and increasing nominal miss distance.

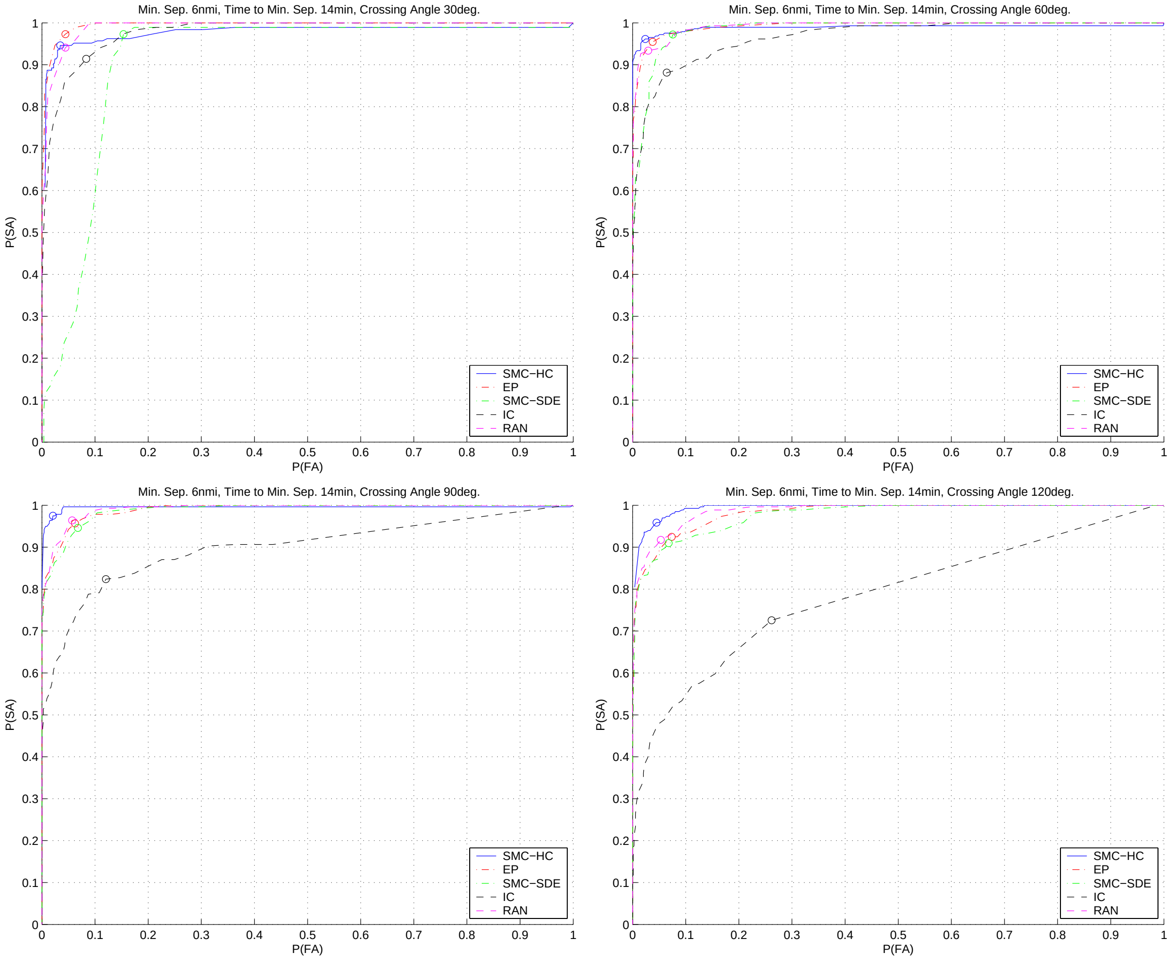

Figure 21: SOC curves for 14 min time to minimal separation, 6 nmi nominal miss distance and increasing angle of intersection of nominal flight paths.

ingful results could not therefore be generated for any of the probes as there did not exist a sufficient balance between conflict and no-conflict to achieve passable convergence to $\mathcal{P}_{SA}(\cdot,\cdot,\cdot)$ and $\mathcal{P}_{FA}(\cdot,\cdot,\cdot)$.

- The SMC-HC probe outperformed all others in all but 7 of the 72 test geometries; the performance was only marginally poorer in those three. The 'ranking' of probes according to minimum average $d(\bar{C}_{opt},\gamma,t_w)$ is shown in Table 5.

Probe	**Mean** $d(\bar{C}_{opt},\cdot,\cdot)$
SMC-HC	0.0325
Ran	0.053
EP	0.0577
SMC-SDE	0.0778
IC	0.232

Table 5: Ranking of probe performances assessed using SOC curves.

- There is substantial variation of alert threshold for some probes, pointing to a degradation in performance where uniform thresholds were used for all geometries. The ranking of probes according to minimum standard deviation of alert threshold is shown in Table 6

Probe	$\sigma\left(\bar{C}_{opt}\right)$
Ran	0.2006
EP	0.228
SMC-SDE	0.276
IC	0.326
SMC-HC	0.404

Table 6: Ranking of standard deviations of alert thresholds.

It is important to note that Table 6 does not necessarily relate the ranking of sensitivity to alert threshold. To understand this statement, suppose that a probe is such that a large range of threshold values provides performance close to the optimal one, uniformly over the different encounter geometries, but that the optimal thresholds are different. Then, in this case, it is actually the probe low sensitivity to the encounter geometry that causes a high optimal alert threshold variance.

This appears to be the case for the SMC-HC method, as shown in the plot of Figure 22 representing the performance index $d(\bar{C},\gamma,t_w)$ as a function of the threshold $\bar{C}$.

In general it appears that SMC methods are comparable or superior to existing methods for a low-complexity validation model, and superior for a higher complexity model.

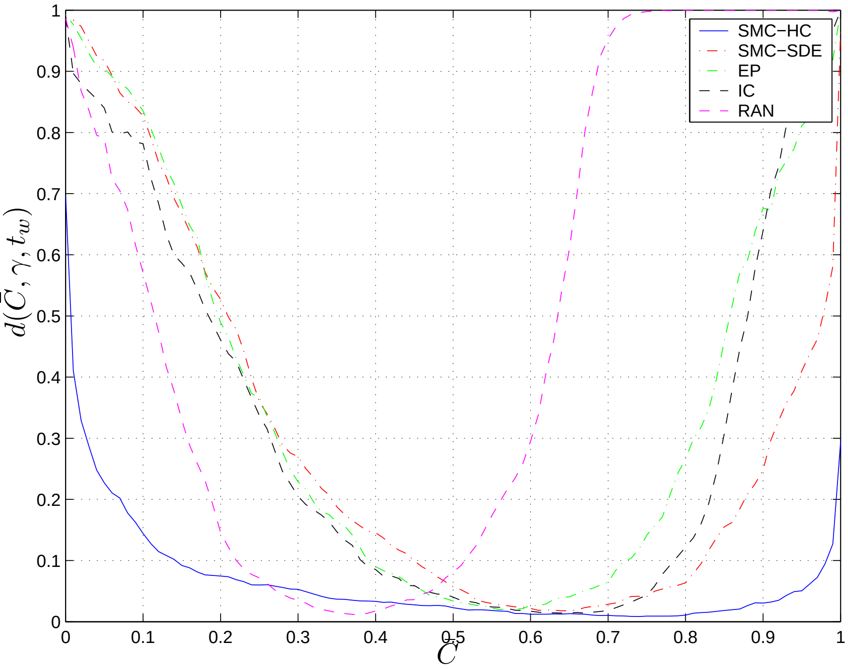

Figure 22: Performance of the different probes as a function of the alerting threshold.

5 Conclusions

In this deliverable we have studied probabilistic mid-term aircraft conflict detection. We have considered, in particular, the probability of conflict as criticality measure for a two-aircraft encounter.

Computing the probability of conflict is generally a difficult task. Different methods have been proposed in the literature to estimate it. Methods based on analytic approximation require simple models for predicting the aircraft positions. Numerical methods based on gridding techniques do not require the prediction model to be simple, but become unpracticable as the dimensionality of the problem grows. The numerical evaluation of the probability of conflict by Monte Carlo methods does not requires specific assumptions on the prediction model and does not suffer of the curse of dimensionality issue. However, the standard Monte Carlo approach can be computationally demanding when the prediction model is complex because of the time required to draw simulations of the aircraft trajectories. This can make it impracticable in the ATM application of interest, where the probability of conflict has to be repeated estimated every time a new radar measurement becomes available.

These considerations have motivated our study of techniques to speed-up the Monte Carlo method. The considered solutions are inspired by the multi-level approach to rare event probability estimation and the sequential Monte Carlo (SMC) approach to nonlinear filtering.
Simulation results show that both methods are promising. Theoretical results have been proven, but only under some restrictive assumptions.
Further work is needed to determine the finite sample properties of these methods under more

realistic assumptions than those adopted here.

The performance of an alerting system based on the conflict probability SMC estimate has been assessed based on the SOC curve, without taking into account the human-in-the-loop component. Air traffic controllers are facing a dynamical decision problem under uncertainty. Their perception is a fundamental aspect that has to be taken into account when designing support tools. This requires further investigation.

References

[1] D. Bertsimas and S. Stock Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46:406–422, 1998.

[2] P.B.M Vranas, D. Bertsimas, and A.R. Odoni. Dynamic gound-holding policies for a network of airports. *Transportation Science*, 28:275–291, 1994.

[3] Christos G. Panayiotou and Christos G. Cassandras. A sample path approach for solving the ground-holding policy problem in air traffic control. In *Proc. 38th IEEE Conference on Decision and Control*, Phoenix, AZ, 1999.

[4] Radio Technical Commission for Aeronautics. Minimum operational performance standards for traffic alert and collision avoidance system (TCAS) airborn equipment. Technical Report RTCA/DO-185, RTCA, September 1990. Consolidated Edition.

[5] J. Lygeros and N. Lynch. On the formal verification of the TCAS conflict resolution algorithms. In *Proc. of the* 36^{th} *Conf. on Decision and Control*, pages 1829–1834, San Diego, CA, December 1997.

[6] R.A. Paielli and H. Erzberger. Conflict probability estimation for free flight. *Journal of Guidance, Control, and Dynamics*, 20(3):588–596, 1997.

[7] H. Erzberger, R.A. Paielli, D.R. Isaacson, and M.M. Eshow. Conflict detection and resolution in the presence of prediction error. In *Proc. of the 1st USA/Europe Air Traffic Management R & D Seminar*, Saclay, France, June 1997.

[8] L.C. Yang and J. Kuchar. Prototype conflict alerting system for free fligh. In *Proc. of the AIAA 35th Aerospace Sciences Meeting, AIAA-97-0220*, Reno, NV, January 1997.

[9] M.G. Ballin and H. Erzberger. An analysis of landing rates and separations at Dallas/Ft. Worth Airport. Technical report, NASA Technical Memorandum, TM 110397, July 1996.

[10] R.A. Paielli. Empirical test of conflict probability estimation. In *2st USA/Europe Air Traffic Manegement R & D Seminar*, 1998.

[11] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems, Special Issue on Air Traffic Control - Part I*, 1(4):179–189, 2000.

[12] European Experimental Centre. *User manual for the base of aircraft data (BADA) revision 3.3*, 2002. Available from http://www.eurocontrol.fr/projects/bada/.

[13] G.B.M. Heuvelink and H.A.P. Blom. An alternative method to solve a variational inequality applied to an air traffic control example. In A. Bensoussan and J.L. Lions, editors, *Analysis and Optimization of Systems*, pages 617–628. Springer, 1988.

[14] M. Prandini, J. Hu, J. Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems, Special Issue on Air Traffic Control - Part I*, 1(4):199–220, 2000.

[15] R.A. Paielli. Empirical test of conflict probability estimation. Technical report, NASA Ames Research Center, Moffett Field, CA, USA, 1998. Available from: http://www.ctas.arc.nasa.gov/publications/paper/.

[16] K. Blin et al. A stochastic conflict detection model revisited. In *AIAA Guidance, Navigation and Control Conference*, Denver, August 2000.

[17] O.J. Watkins and J. Lygeros. Stochastic reachability for discrete time systems: An application to aircraft collision avoidance. In *Proc. of the* 42^{nd} *Conf. on Decision and Control*, Maui, USA, December 2003.

[18] R.S. Schild. *Rule optimization for airbone aircraft separation.* PhD thesis, Wien Technical University, 1998. Available from http://www.eos.tuwien.ac.at/Oeko/RSchild/Rules/.

[19] W. Glover and J. Lygeros. A stochastic hybrid model for air traffic control simulation. In *HSCC*, pages 372–386. 2004.

[20] J. Hu and M. Prandini. Aircraft conflict detection: a method for computing the probability of conflict based on Markov chain approximation. In *European Control Conf.*, Cambridge, UK, September 2003.

[21] W. Glover and J. Lygeros. A multi-aircraft model for conflict detection and resolution algorithm validation, July 2003. HYBRIDGE Project IST-2001-32460, Work Package WP1, Deliverable D1.3.

[22] Mariken Everdij and Henk Blom. Bias and uncertainty modelling in accident risk assessment, February 2005. HYBRIDGE Project IST-2001-32460, Work Package WP8, Deliverable D8.4.

[23] J.K. Kuchar. *A unified methodology for the evaluation of hazard alerting systems.* PhD thesis, MIT, 1995.

[24] L.C. Yang and J. Kuchar. Using intent information in probabilistic conflict analysis. In *AIAA Guidance, Navigation, and Control Conf.*, Boston, MA, 1998.

[25] B.van Doorn, B. Bakker, and C. Meckiff. Evaluation of advanced conflict modelling in the Highly Interactive Problem Solver. In *Proc. 4th USA/Europe ATM R&D Seminar*, December 2001.

[26] M. Innocenzi, P. Gelosi, and L. Pollini. Air traffic management using probability function fields. In *AIAA Guidance, Navigation, and Control Conf.*, Portland, USA, 1999.

[27] R. Irvine. A geometrical approach to conflict probability estimation. Technical report, Eurocontrol Experimental Centre, Bretigny-sur-Orge, France, 2001. Available from http://atm2001.eurocontrol.fr.finalpapers/pap137.

[28] J. Lygeros and M. Prandini. Aircraft and weather models for probabilistic conflict detection. In *Proc. of the* 41^{st} *Conf. on Decision and Control*, Las Vegas, NV, December 2002.

[29] W. Glover and J. Lygeros. Simplified multi-aircraft models for conflict detection and resolution algorithms, 2004. HYBRIDGE Project IST-2001-32460, Work Package WP1, Deliverable D1.4.

[30] J. Hu, M. Prandini, and S. Sastry. Probabilistic safety analysis in three dimensional aircraft flight. In *Proc. of the* 42^{nd} *Conf. on Decision and Control*, Maui, USA, December 2003.

[31] J. Hu, M. Prandini, and S. Sastry. Aircraft conflict detection in presence of spatially correlated wind perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 2005. To appear.

[32] G.J. Bakker and H. Blom. Air traffic collision risk modelling. In *Proc. of the* 32^{nd} *Conf. on Decision and Control*, December 1993.

[33] H.A.P. Blom and G.J. Bakker. Conflict probability and incrossing probability in air traffic management. In *CDC02*, Las Vegas, Nevada, U.S.A., December 2002.

[34] H.A.P. Blom, G.J. Bakker, M.H.C. Everdij, and M.N.J. van der Park. Collision risk modelling of air traffic. In *Proc. European Control Conf.*, Cambridge, U.K., September 2003.

[35] H.A.P. Blom, G.J. Bakker, M.H.C. Everdij, and M.N.J. van der Park. Stochastic analysis background of accident risk assessment in Air Traffic Management, 2004. HYBRIDGE Project IST-2001-32460, Work Package WP2, Deliverable D2.2.

[36] J. Krystul and H. Blom. Monte Carlo simulation of rare events in hybrid systems, 2004. HYBRIDGE Project IST-2001-32460, Work Package WP8, Deliverable D8.3.

[37] F. Cerou, P.Del Moral, F. Le Gland, and P. Lezaud. Genetic genealogical models in rare event analysis. Draft paper available at http://www.nlr.nl/public/hosted-sites/hybridge, December 2002.

[38] J. Krystul, A. Bagchi, and H. Blom. Risk decomposition and assesment methods, 2003. HYBRIDGE Project IST-2001-32460, Work Package WP8, Deliverable D8.1.

[39] P. Lezaud, J. Krystul, and H. Blom. Accident risk assesment and Monte Carlo simulation methods, 2002. HYBRIDGE Project IST-2001-32460, Work Package WP8, Deliverable D8.2.

[40] M. Villn-Altamirano and J. Villn-Altamirano. RESTART: A method for accelerating rare event simulation. In *13th Int. Teletraffic Congress, ITC 13 (Quequing, Performance and Control in ATM)*, Copenhagen, Denmark, 1991.

[41] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for pratictioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

[42] *Sequential Monte Carlo methods in practice.* Springer-Verlag, 2001.

[43] P. Del Moral and L. Miclo. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to nonlinear filtering. In *Seminaire de Probabilites XXXIV*, volume 1729 of *Lecture Notes in Mathematics*, pages 1–145. Springer-Verlag, 2000.

[44] A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.

[45] P. Del Moral, A. Doucet, and G.W. Peters. Sequential Monte Carlo samplers. 2004.

[46] S.W. Ong. Probabilistic methods for conflict detection in air traffic control. Technical report, University of Cambridge, 2002.

[47] O. Watkins. *Stochastic reachability, conflict detection, and air traffic management.* PhD thesis, University of Cambridge, 2005.