A Probabilistic Framework
for Aircraft Conflict Detection*

Maria Prandini, John Lygeros, Arnab Nilim and Shankar Sastry
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley CA 94720
{prandini, lygeros, nilim, sastry}@robotics.eecs.berkeley.edu

Abstract
In this paper we describe a general conflict detection/resolution scheme for a pair of aircraft flying at the same altitude, focusing on the conflict detection component. The proposed approach is formulated in a probabilistic framework, thus allowing uncertainty in the aircraft positions to be explicitly taken into account when detecting a potential conflict. The computational issues involved in the application of the proposed conflict alerting system are solved in a numerically efficient fashion by resorting to appropriate randomized algorithms. Finally, the validation of the proposed detection scheme is performed by Monte Carlo simulation on a stochastic ODE model of the aircraft motion. Our simulations show very promising results. The detection scheme will be used as the basis for our conflict resolution scheme in forthcoming work.

1 Introduction

Safety, of which conflict prediction and resolution form an integral part, is the primary concern of all advanced air traffic management systems. Conflict prediction and resolution are given consideration at three different levels of the air traffic management process:

1. Long range: Some form of conflict prediction and resolution is carried out at the level of the entire National Airspace System (NAS), over a horizon of several hours. It involves composing

*Research supported by DARPA under grant F33615-98-C-3614, by NASA under grant NAG 2-1039 and by ARO under grant MURI DAAH 04-96-I-0341. The authors would like to thank Dr. George J. Pappas for the stimulating discussions.
flight plans and airline schedules (on a daily basis, for example) to ensure that airport and sector capacities are not exceeded. This is typically accomplished using large scale integer and linear programming techniques [1, 2].

2. Mid-range: Conflict prediction and resolution is carried out by air traffic control (ATC), over horizons of the order of tens of minutes. It involves modifying the pre-planned flight plan on-line, to ensure adequate aircraft separation. Semi-automated tools have been developed to assist air traffic controllers with these decisions [3]. The algorithms and techniques of [4, 5] can also operate at this level.

3. Short range: Conflict prediction and resolution is also carried out on board the aircraft, over horizons of seconds to minutes. This is typically considered as a last resort solution. The Traffic Alert and Collision Avoidance System (TCAS) currently operating on all commercial aircraft is such a prediction/resolution algorithm ([6, 7]). The algorithms of [4, 5] also belong to this category.

The work presented here concentrates primarily on the mid-range level of the air traffic management process. It is assumed that the proposed algorithms will eventually be implemented as computational tools to provide assistance to ATC.

Our conflict prediction and resolution framework consists of a number of components (Figure 1). If one thinks of these components as arranged in a feedback loop, the ATC, aircraft, and radar would correspond to the plant (the ATC and radar playing the role of actuators and sensors respectively), while the prediction and resolution components would correspond to the controller. Our goal is to design the two “controller” modules and verify that the closed loop system possesses certain desirable properties (safety, ATC workload reduction, etc.). In the present paper, we restrict our
Figure 1: Outline of the conflict prediction/resolution functionality

attention to specific suggestions for the prediction and validation schemes, describing only in the introduction our current work on conflict resolution. However, in ongoing work, we are investigating a number of design alternatives for prediction, resolution, and validation. The schemes proposed here (highlighted in the remainder of this section and detailed in Section 3) make extensive use of randomized algorithms for estimating integrals and carrying out optimizations (Subsection 3.2). The advantage of randomized techniques is that they tend to be computationally more efficient. They also provide analytical bounds on the accuracy of the approximation involved, provided one makes appropriate design choices (for the cost functions for conflict resolution for example).

Conflict Detection

We consider two aircraft moving on the same horizontal plane, each following its individual flight plan. The flight plan is assumed to consist of a sequence of way points on the plane and a sequence of speeds for moving between them. Based on this information and estimates of the current positions of the aircraft, the models developed by [3, 8] can be modified to provide empirically motivated estimates of the probability distribution for the projected position of the two aircraft in the near future (Subsection 2.1). One can then define the probability of conflict (PC) at a future time
as the probability that at that time the two aircraft will be within an unsafe distance from one another (typically 5 nautical miles outside the TRACON - Terminal Radar Approach Control - and 3 nautical miles inside the TRACON). Conflict detection consists of estimating PC and warranting some action when PC is high.

The design choices that enter into conflict detection are the different ways of computing estimates of PC and the different ways of weighting the value of PC at various times*. The parameters that one needs to set are the prediction horizon (taken to be 20 minutes in this paper as a reasonable look ahead time) and the threshold for declaring a conflict.

Here we use randomized integration to estimate the value of PC at each time instant and declare conflicts based on the maximum value of PC over the prediction horizon (also estimated using randomized techniques). Choices for the prediction cost function include various weighted averages. Other approaches for obtaining an overall measure of the likelihood of a conflict over a finite horizon are proposed in [4] and [3], respectively based on intensive off-line Monte Carlo simulations leading to look up tables and an on-line procedure based on analytical approximation. Systematic guidelines for setting threshold values can be found in [9].

**Conflict Resolution**

Once a conflict has been declared, conflict resolution consists of modifying the flight plans of the two aircraft to ensure that the probability of conflict falls below a specified threshold. We are currently studying a model predictive control approach to achieve this goal. The idea is to use the current radar measurements, the prediction module and an optimization algorithm to minimize an appropriate resolution cost function over all possible flight plans. ATC is notified if changes in the

*This can be thought of as a prediction cost function, assigning a cost to the function of time PC.
upcoming way points are imminent. The process is repeated every time a new radar measurement becomes available.

The design choices that enter into conflict resolution are the different resolution cost functions and the different optimization techniques. The parameters one needs to set are: How far in advance should ATC be notified of flight plan changes (partly determined by human capabilities) and the bounds on the allowable flight plans (partly determined by aircraft capability and ATC protocol). By choosing a finite horizon, discounted weighted average of PC as a resolution cost function, we take into account the fact that the accuracy of our prediction decreases the further we project into the future. Other choices could involve the maximum of PC over the prediction horizon, special cost structures to bias towards not making any changes, etc. The optimization can be carried out either by randomized algorithms or combinatorially, aided by appropriate heuristics.

Validation

The performance of a proposed conflict prediction/resolution scheme is measured in terms of:

1. Safety

2. Impact on ATC workload

3. Efficiency (impact on fuel consumption, deviations from schedule, etc.).

Here we restrict our attention only to the issue of safety\(^1\).

Validation of the prediction scheme requires one to model the radar and the aircraft. Here we use a simple model for the radar (additive white noise) and we introduce a stochastic difference equation

\(^1\)In fact safety is closely coupled to ATC workload, as the controllers may choose to disregard disruptive advisories.
to model the aircraft movement. Validation of the detection scheme is then carried out by Monte Carlo simulation.

At a later stage we hope to be able to provide theoretical bounds on the safety of the proposed design - including the resolution component - and estimate its impact on ATC workload by means of human-in-the-loop simulations.

The paper is organized as follows. In Section 2, we introduce the probabilistic framework in which we develop our prediction scheme (Subsection 2.1) and perform the validation (Subsection 2.2). In Section 3, we introduce our conflict detection scheme and describe how we deal with the involved computational issues in order to formulate the algorithmic version of the proposed scheme (Subsection 3.2). Finally, the Monte Carlo simulation results obtained through the proposed validation model are reported in Section 4.

2 Probabilistic Models

A situation of conflict is encountered when two aircraft come within the minimum allowed distance between each other, where the minimum allowed horizontal separation for en-route airspace is 5 nautical miles (nmi) and the minimum vertical separation is 2000 ft above an altitude of 29000 ft, 1000 ft below an altitude of 29000 ft. Inside the TRACON, the horizontal separation may be reduced to 3 nmi. We now introduce a probabilistic model for the aircraft motion inspired by [3, 8] which allows us to explicitly take into account the uncertainty in the predicted aircraft trajectories when detecting a situation of conflict.

In the following, we shall deal with the 2D case, i.e. the case when the two aircraft fly at the same altitude. This is for ease of explanation, since the generalization to the 3D case is straightforward.
with the only added difficulty consisting of more complex notation.

2.1 Prediction model

Consider two aircraft denoted by 1 and 2, moving on the same plane. We assume that

**Assumption 1** The flight plan of each aircraft \( i \) is known and it is described in terms of a sequence of way points, \( \{ P^i_j \}_{j=0,\ldots,n_i}, P^i_j \in \mathbb{R}^2, i = 1,2 \), where the first way point \( P^i_0 \) represents the current position of aircraft \( i \), and a sequence of speeds \( \{ v^i_j \}_{j=1,\ldots,n_i}, v^i_j \in \mathbb{R}_+, i = 1,2 \), where \( v^i_j \) is the speed of aircraft \( i \) between \( P^i_{j-1} \) and \( P^i_j \).

Let \( \{ T^i_j \}_{j=0,\ldots,n_i} \) denote the nominal time required for aircraft \( i \) to arrive at way point \( P^i_j \). Then, \( \{ T^i_j \} \) can be recursively computed as follows

\[
T^i_j = \frac{\| P^i_j - P^i_{j-1} \|}{v^i_j} + T^i_{j-1}, \quad j = 1, \ldots, n_i
\]

with initial condition \( T^i_0 = 0 \) since \( P^i_0 \) corresponds to the current position of aircraft \( i \).

According to [3, 10, 4], the predicted position of aircraft \( i \) is affected by uncertainty due to wind modeling, prediction errors, tracking, navigation, and control errors. Consequently, the position of an aircraft can be modeled as a multivariate Gaussian random variable. The position of aircraft \( i \) after a time interval \( t \),

\[
x^i(t) := \begin{bmatrix} x^i_1(t) \\ x^i_2(t) \end{bmatrix},
\]

has mean value given by its nominal position in the flight plan and covariance matrix such that the *along-track variance* \( \sigma^2_{AT}^i(t) \) and the *cross-track variance* \( \sigma^2_{CT}^i(t) \) increase with time. Precisely, \( \sigma^2_{AT}^i(t) \) grows linearly with time:

\[
\sigma^2_{AT}^i(t) = c_1 + c_2 t,
\]

(1)
whereas $\sigma_{CT}^{2i}(t)$ grows linearly with the traveled distance $s^i(t)$ until it reaches a saturation value according to equation

$$\sigma_{CT}^{2i}(t) = \min \{c_4, c_1 + c_3 s^i(t)\},$$

(2)

where $s^i(t)$ can be computed by $s^i(t) = v^i_j(t - T^i_{j-1}) + s^i(T^i_{j-1}), \forall t \in (T^i_{j-1}, T^i_j), j = 1, \ldots, n_i$, initialized with $s^i(T^i_0) = 0$.

The values of the initial tracking error variances $c_1$, their growth rates $c_2$ and $c_3$, and the final cross-track error variance $c_4$ are respectively set equal to $c_1 = 50/1850$ nmi$^2$, $c_2 = 0.25$ nmi$^2$/min, $c_3 = 1/57$ nmi and $c_4 = 1$ nmi$^2$, which have been obtained empirically, based on air traffic data with a time horizon of length $T = 20$ min ([3]). In particular, the error covariance associated with the current position $x^i(0)$ represents the effect of the noise affecting the radar measurements.

The distribution of $x^i(t)$ for $i = 1, 2$ is therefore given by

$$x^i(t) \sim N(p^i(t), Q^i(t)),$$

where the mean value $p^i(t)$ is the nominal aircraft position on its flight plan, i.e.,

$$p^i(t) = P^i_{j-1} + v^i_j(t - T^i_{j-1}) \frac{P^i_j - P^i_{j-1}}{\|P^i_j - P^i_{j-1}\|}, \quad t \in [T^i_{j-1}, T^i_j)$$

(3)

for $j = 1, \ldots, n_i$ and the covariance matrix $Q^i(t)$ is obtained through the expression

$$Q^i(t) = R(\theta^i_j) \begin{bmatrix} \sigma_{AT}^{2i}(t) & 0 \\ 0 & \sigma_{CT}^{2i}(t) \end{bmatrix} R(\theta^i_j)^T, \quad t \in [T^i_{j-1}, T^i_j)$$

(4)

where the rotational matrix $R(\theta^i_j) \in SO(2)$ associated with the heading $\theta^i_j$ ($\theta^i_j$ being the angle that vector $P^i_j - P^i_{j-1}$ makes with the $x_1$ axis of the global coordinate frame in which the $P^i_j$’s are given)

is

$$R(\theta^i_j) = \begin{bmatrix} \cos(\theta^i_j) & -\sin(\theta^i_j) \\ \sin(\theta^i_j) & \cos(\theta^i_j) \end{bmatrix}.$$
Figure 2: Probabilistic model for the motion of aircraft 1

Figure 2 gives an overall idea of the probabilistic model for the motion of each aircraft. The two aircraft positions $x^1(t)$ and $x^2(t)$ are assumed to be uncorrelated. It is important to observe that the independence assumption between the tracking noises of the two aircraft is an assumption commonly made ([8]), but it is, in some sense, a strong assumption. The tracking noise is in fact primarily due to wind, which may be strongly correlated between the two aircraft, especially near the conflict point where the aircraft are close one to the other. *Modeling the tracking errors correlation properties is not dealt with in this paper, though it obviously needs to be evaluated in future work.*

### 2.2 Validation model

As we have already mentioned in the previous subsection, the uncertainty in the aircraft position can be described in terms of its “longitudinal” (along-track) and “lateral” (cross-track) components ([8, 4, 3]). At each time instant $t$ up to a time horizon of 20 minutes, these two components can be modeled as independent zero mean Gaussian random variables whose variance is increasing with time. In particular, the variance of the along-track component keeps growing (see equation (1)),
whereas the variance of the cross-track component tends to the saturation value $c_4$ (see equation (2)). We build the model of the aircraft motion for validation on the basis of this distinction, maintaining the characteristics of the along and cross-track uncertainties.

Consider an aircraft moving on a horizontal plane. Let $(x_1, x_2)$ denote its position with respect to an inertial coordinate frame. Assume its velocity has magnitude $v$ and makes an angle $\theta$ with respect to the $x_1$ axis (Figure 3). Consider a body coordinate frame $(\chi_1, \chi_2)$ with $\chi_1$ aligned with the aircraft velocity (along-track) and $\chi_2$ perpendicular to it (cross track). The two frames are related through a coordinate transformation:

$$
\begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} = R(\theta) \begin{bmatrix}
  \chi_1 \\
  \chi_2
\end{bmatrix} + p,
$$

where $R(\theta)$ is the rotation matrix:

$$
R(\theta) = \begin{bmatrix}
  \cos(\theta) & -\sin(\theta) \\
  \sin(\theta) & \cos(\theta)
\end{bmatrix} \in SO(2)
$$

and $p$ denotes the position of the body reference frame origin with respect to the inertial reference frame.

In light of what we discussed so far, equation (5) can be reinterpreted as follows
the origin \( p \) of the body coordinate frame represents the nominal position of the aircraft;

- \( \chi_1 \) and \( \chi_2 \) represent the variation in the aircraft position with respect to the nominal one along the along-track and cross-track directions respectively;

- \( \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \) represents the actual position of the aircraft once one takes into account the uncertainty along the along-track and cross-track directions.

The nominal position \( p \) of the aircraft evolves in time according to the following equation

\[
\dot{p} = R(\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} v,
\]

where \( v \) is the aircraft linear velocity. As for along-track and cross-track errors vector \( \chi := \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} \), it is obtained through the stochastic differential equation:

\[
\dot{\chi} = A \chi + \eta, \chi(0) \sim N(0, V_{\chi}(0))
\]

where \( \eta := \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \) is a white Gaussian noise process with \( \eta(t) \sim N(0, V_{\eta}), \forall t \), and it is independent of \( \chi(0) \). We shall, in fact, show in the paragraph below on the tuning of the validation model parameters that by appropriately selecting matrix \( A \) and the covariance matrices \( V_{\chi}(0) \) and \( V_{\eta} \), we are able to obtain uncorrelated Gaussian long track and cross-track errors whose variance respectively keeps growing with time and saturates according to equations (1) and (2). We now derive the kinematical model of the aircraft trajectory in terms of the inertial frame coordinates. Uniformly with the notation in Section 2.1, we define \( x := \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \). Then, the aircraft position \( x \) evolves in time according to the following equation

\[
\dot{x} = \dot{R} \chi + R \dot{\chi} + \dot{\hat{p}} \\
= (\dot{R} + RA) \chi + R \eta + \dot{\hat{p}}, \quad \text{by equation (5)}
\]

\[
= (\dot{R} + RA) R(\theta)^T (x - p) + R \begin{bmatrix} 1 \\ 0 \end{bmatrix} v + R \eta, \quad \text{by equations (5) and (6)}
\]
\[
\dot{x} = A(\theta)(x-p)+B(\theta)u+C(\theta)\eta \\
\dot{p} = D(\theta)u
\]

Figure 4: Piecewise linear model for the aircraft motion

where we set \( R := R(\theta) \) for simplifying the notation.

By observing that \( \dot{R} = \frac{dR}{d\theta} \), we finally get the equations governing the aircraft motion

\[
\begin{align*}
\dot{x} &= \left[ \frac{dR}{d\theta} R^T \omega + RAR^T \right] x - \left[ \frac{dR}{d\theta} R^T \omega + RAR^T \right] p + R \begin{bmatrix} 0 \\ 1 \end{bmatrix} v + R\eta \\
\dot{p} &= R \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \\
\dot{\theta} &= \omega
\end{align*}
\]

where \((v, \omega)\) are the linear and angular velocities.

Note that the equations describing the aircraft motion are nonlinear. A first simplification may be to set \( \dot{\theta} = 0 \) by assuming in fact that \( \omega = 0 \). In this case, the motion of the aircraft is described by

\[
\begin{align*}
\dot{x} &= A(\theta)(x-p)+B(\theta)u+C(\theta)\eta \\
\dot{p} &= D(\theta)v
\end{align*}
\]

where \( A(\theta) = R(\theta)AR(\theta)^T \), \( B(\theta) = D(\theta) = R(\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \), \( C(\theta) = R(\theta) \) and \( \theta \) is fixed.

In our validation model we model turns as discrete events occurring at the way points. In this way the aircraft dynamics turns out to be piecewise linear with switching times determined by the occurrence of the turn events \( Turn(\bar{\theta}) \), the effect of \( Turn(\bar{\theta}) \) being the addition of \( \bar{\theta} \) to the current value of \( \theta \) (Figure 4).

Given a flight plan, \( i.e. \), a sequence of way points, \( \{P_j\}_{j=0,\ldots,n} \), and a sequence of speeds \( \{v_j\}_{j=1,\ldots,n} \),
our validation model is therefore represented by the stochastic differential equation

\[
\begin{align*}
\dot{x} &= A(\theta_j)(x - p) + B(\theta_j)v_j + C(\theta_j)\eta, \\
\dot{p} &= D(\theta_j)v_j,
\end{align*}
\]

where the switching times \(\{T_j\}\), the nominal arrival times at the corresponding way points \(\{P_j\}\), are computed as follows

\[T_j = \frac{\|P_j - P_{j-1}\|}{v_j} + T_{j-1}, \quad j = 1, \ldots, n,\]

with \(T_0 = 0\). The initial conditions for (8) are given by

\[
\begin{align*}
x(0) &\sim \mathcal{N}(P_0, R(\theta_1) V_\chi(0) R(\theta_1)^T), \\
p(0) &= P_0
\end{align*}
\]

and the white Gaussian noise \(\eta\) determining the tracking errors satisfies \(\eta(t) \sim \mathcal{N}(0, V_\eta), \forall t\), and is independent of \(\chi(0)\).

Moreover, we assume that the position estimates \(y := \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\) available every \(\Delta\) seconds (typically 12 seconds) to the ATC through the radar measurements are affected by some noise, \textit{i.e.},

\[
y(k\Delta) = x(k\Delta) + \xi(k\Delta),
\]

where the measurement noise \(\{\xi(k\Delta)\}_{k \geq 0}\) is described as a sequence of independent identically distributed (i.i.d.) Gaussian random variables with \(\xi(k\Delta) \sim \mathcal{N}(0, V_\xi)\) and it is assumed to be independent of all the other random variables involved in the validation model.

**Tuning of the validation model parameters**

For our stochastic validation model to resemble the statics derived from air traffic data, we need to appropriately tune its parameters, \textit{i.e.},

- the noise \(\chi\) and \(\eta\) covariance matrices \(V_\chi(0)\) and \(V_\eta\) and the matrix \(A\) entering the matrix

  \[
  A(\theta) = R(\theta) A R(\theta)^T
  \]

of the aircraft trajectory model (8);
• the noise covariance $V_{\xi}$ for the radar model (10).

As for the radar model parameters, we set $V_{\xi} = \begin{bmatrix} V_{\xi 1} & 0 \\ 0 & V_{\xi 2} \end{bmatrix}$, where $V_{\xi 1} = V_{\xi 2} = c_1$ so as to reproduce the statistical characteristics estimated in [3] for the uncertainty in the current aircraft position.

The tuning of the aircraft trajectory model is performed in the light of the following proposition, which is based on standard results for the solution of stochastic differential equations ([11]).

**Proposition 1**

Consider the following stochastic differential equation

$$\dot{\chi} = \begin{bmatrix} 0 & 0 \\ 0 & -\alpha \end{bmatrix} \chi + \eta, \quad \chi(0) \sim \mathcal{N}
\begin{bmatrix} V_{\chi 1}(0) \\ 0 \end{bmatrix}, \mathcal{N}
\begin{bmatrix} 0 & 0 \\ 0 & V_{\chi 2}(0) \end{bmatrix}$$

where $\alpha > 0$ and $\eta(\cdot)$ is a white Gaussian noise with $\eta(t) \sim \mathcal{N}
\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathcal{N}
\begin{bmatrix} V_{\eta 1} & 0 \\ 0 & V_{\eta 2} \end{bmatrix}$, $\forall t$, which is independent of $\chi(0)$.

Then, the stochastic process $\chi(\cdot)$ satisfies the following properties

1. $\chi_1(\cdot)$ and $\chi_2(\cdot)$ are uncorrelated Gaussian processes;

2. $\chi_1(\cdot)$ is a Gaussian noise with zero mean and variance $\text{var}[\chi_1(t)] = V_{\chi 1}(0) + V_{\eta 1} t$;

3. $\chi_2(\cdot)$ is a Gaussian noise with zero mean and variance $\text{var}[\chi_2(t)] = \frac{V_{\eta 1}}{2 \alpha} [1 - e^{-2 \alpha t}] + V_{\chi 2}(0) e^{-2 \alpha t}$.

**Proof.** Observe that the equations governing the evolution of the two components of process $\chi$ are decoupled since in fact

$$\begin{cases} \dot{\chi}_1 = \eta_1 \\ \dot{\chi}_2 = -\alpha \chi_2 + \eta_2 \end{cases}$$

In particular, we have that

$$\chi_1(t) = \chi_1(0) + \int_0^t \eta_1(\tau) d\tau$$

(11)
and
\[ \chi_2(t) = e^{-\alpha t} \chi_2(0) + \int_0^t e^{-\alpha (t-\tau)} \eta_2(\tau) d\tau. \]  

(12)

Then, point 1 is an immediate consequence of the independence of the processes \( \chi_1(\cdot) \), \( \chi_2(\cdot) \) and the random variables \( \chi_1(0) \) and \( \chi_2(0) \) and of the Gaussian assumption.

As for point 2, from equation (11) we get that
\[ E[\chi_1(t)] = E[\chi_1(0)] + E[\int_0^t \eta_1(\tau) d\tau] = 0 \]

and
\[ E[\chi_1(t) \chi_1(s)] = E[\chi_1(0)^2] + \int_0^t \int_0^s E[\eta_1(u) \eta_1(v)] du dv \]
\[ = V_{\chi_1(0)} + \int_0^{\min\{t,s\}} V_{\eta_1} du \]
\[ = V_{\chi_1(0)} + V_{\eta_1} \min\{t,s\} \]
from which it follows that \( \text{var}[\chi_1(t)] = E[\chi_1(t)^2] = V_{\chi_1(0)} + V_{\eta_1} t. \)

Analogously, from equation (12) we get that
\[ E[\chi_2(t)] = E[e^{-\alpha t} \chi_2(0)] + E[\int_0^t e^{-\alpha (t-\tau)} \eta_2(\tau) d\tau] = 0 \]

and
\[ E[\chi_2(t) \chi_2(s)] = E[\chi_2(0)^2] e^{-\alpha (t+s)} + \int_0^t \int_0^s e^{-\alpha (t-u)} e^{-\alpha (s-v)} E[\eta_2(u) \eta_2(v)] du dv \]
\[ = V_{\chi_2(0)} e^{-\alpha (t+s)} + e^{-\alpha (t+s)} \int_0^{\min\{t,s\}} e^{2\alpha u} V_{\eta_2} du \]
\[ = V_{\chi_2(0)} e^{-\alpha (t+s)} + \frac{V_{\eta_2}}{2\alpha} [e^{-\alpha (t+s)} - e^{-\alpha (t+s)}] \]
from which it follows that \( \text{var}[\chi_2(t)] = E[\chi_2(t)^2] = V_{\chi_2(0)} e^{-2\alpha t} + \frac{V_{\eta_2}}{2\alpha} [1 - e^{-2\alpha t}]. \)

This concludes the proof. \( \Box \)
On the basis of this result, for the noise \( \chi(\cdot) \) to reproduce the empirically observed characteristics of the along-track and cross-track errors, we set

\[
V_{\chi_1}(0) = V_{\chi_2}(0) = c_1
\]

\[
V_{\eta_1} = c_2
\]

\[
\frac{V_{\eta_2}}{2\alpha} = c_4
\]

\[
A = \begin{bmatrix} 0 & 0 \\ 0 & -\alpha \end{bmatrix}, \text{ where } \alpha = \frac{c_3}{2(c_4 - c_1)} v(0).
\]

With such positions we get from Proposition 1 that

\[
\text{var}[\chi_1(t)] = c_1 + c_2 t
\]

and

\[
\text{var}[\chi_2(t)] = c_4 + [c_1 - c_4] e^{-\frac{c_3}{(c_4 - c_1)} v(0)t}.
\]

Thus, \( \text{var}[\chi_1(t)] \) is equal to the along-track variance \( \sigma^2_{AT}(t) \) in equation (1), whereas \( \text{var}[\chi_2(t)] \) is monotonically increasing from the initial value \( c_1 \) to the saturation value \( c_4 \) as the cross-track variance \( \sigma^2_{CT}(t) \) in equation (2). Moreover, the initial growing rate of \( \text{var}[\chi_2(t)] \) is the same as that of \( \sigma^2_{CT}(t) \). By the Taylor formula applied to expression (2), we in fact get that for \( t \) small enough

\[
\sigma^2_{CT}(t) = c_3 v(0)t + c_1,
\]

\( s(t) = v(0)t \) for \( t \) small), which agrees up to linear terms with the Taylor series for (14).

When validating a conflict prediction/resolution scheme, we consider two aircraft coming close one to the other. This means that we should characterize also the correlation properties of the stochastic processes describing each aircraft position and the corresponding radar measurements.

In this paper, we suppose that the noises related to the two aircraft are independent.
3 Conflict Detection

3.1 General conflict detection

Consider two aircraft, 1 and 2, flying at the same altitude. Let \( x^1 \) and \( x^2 \) denote their positions in the inertial reference frame. Given that a conflict occurs when the aircraft are closer than 5 nmi to one another, according to the probabilistic model for the conflict prediction in Subsection 2.1, the probability of conflict at time instant \( t \), i.e., the probability that \( \|x^1(t) - x^2(t)\| \leq 5 \), can be computed as follows

\[
PC(t) = \text{Prob}(\|x^1(t) - x^2(t)\| \leq 5) = \int_{\|x\| \leq 5} f(x, t) dx,
\]

where

\[
f(x, t) = \frac{1}{2\pi\sqrt{\det(Q(t))}} e^{-\frac{1}{2}(x - \mu(t))^T Q(t)^{-1} (x - \mu(t))}
\]

is the probability density function of the distance \( d(t) := x^1(t) - x^2(t) \). As a matter of fact, given the probabilistic model described in Subsection 2.1, \( d(t) \) turns out to be a Gaussian random variable with mean \( \mu(t) := p^1(t) - p^2(t) \) and variance \( Q(t) := Q^1(t) + Q^2(t) \), since \( x^i(t) \sim \mathcal{N}(p^i(t), Q^i(t)) \), \( i = 1, 2 \), and \( x^1(t), x^2(t) \) are independent.

In the conflict prediction algorithm proposed in this contribution, the detection of a situation of conflict in the near future is based on the maximum value assumed by the probability of conflict over a certain time horizon \( T \). An “alert” is issued when the maximum probability of conflict exceeds a certain threshold \( \widehat{PC} \). The value for the time horizon \( T \) is chosen to be equal to 20 minutes following [3, 8]. As for the threshold on the probability of conflict, it can be set either by heuristic arguments such that used in [8] or by referring to the procedure suggested in [9], where the main objective is that of striking a compromise between the false alarms and missed detections. Other aspects that should be taken into account for the tuning of the threshold are the detection...
of conflict a certain amount of time before it occurs and the prediction of its occurrence time (so as to avoid ‘last minute’ maneuvers). These aspects, in fact, highly influence the efficiency of every prediction/resolution scheme involving the human-in-the-loop component. For a discussion on the computation of the threshold see Section 4.

**Conflict detection algorithm**

Given the flight plans of aircraft 1 and 2: \( \{P^i_j\}_{j=0,...,n_i}, \{v^i_j\}_{j=1,...,n_i}, i = 1, 2, \)

1. compute the maximum value of the probability of conflict in the incoming time horizon of length \( T, \) i.e.,

\[
PC_{\text{max}} := \sup_{t \in [0,T]} PC(t),
\]

where \( PC(t) \) is given by equation (15);

2. if \( PC_{\text{max}} \geq PC, \) a prescribed threshold, then declare that a situation of conflict has been detected.

This algorithm is expected to be executed at every time instant when a new radar measurement is available on the updated flight plans of the aircraft. The idea is that only when an alert is issued should a conflict resolution algorithm be executed to compute a modification of the two aircraft flight plans such that the probability of conflict is appropriately reduced.

The major obstacle in the implementation of the proposed prediction scheme is represented by the computation of \( PC_{\text{max}} \) in equation (17), since \( PC(\cdot) \) cannot be expressed in analytical form and even the evaluation of \( PC(t) \) for a given \( t \) is time consuming. It should also be noted that this obstacle turns out to be overwhelming in the on-line application of the prediction scheme where computations are subject to time constraints. A way out to these complicated implementation
issues is to abandon the ambitious objective of finding an exact solution and instead compute an
approximate solution. In order to provide an effective method, however, one should not only work
out an algorithm to find the approximate solution but also set up a methodology able to provide
quantitative information on the level of approximation introduced. A significant help is provided
from using some results of the theory of empirical processes, whose main aim is to study how to
estimate unknown quantities through experimentation ([12]).

3.2 Randomized algorithms for the conflict detection

Suppose for the time being that given a \( t \in [0, T] \), we are able to compute \( PC(t) \) with no error.

Let us introduce the uniform probability distribution \( Q \) on the time interval \([0, T]\). In order to
maximize \( PC(t) \), we can resort to the following optimization algorithm.

Randomized optimization algorithm

1. extract at random \( N \) independent values \( \tau_1, \ldots, \tau_N \in [0, T] \) according to probability \( Q \);

2. choose \( PC_{\text{max}} = \max_{\tau \in \{\tau_j\}_{j=1}^N} PC(\tau) \).

Clearly, since we are testing just \( N \) values of \( \tau \), we cannot expect that \( PC_{\text{max}} \) is the global maximum
of \( PC(t) \) over \([0, T]\). In addition, the quality of the result is random due to the stochastic selection of
\( \tau_j \)'s. Nevertheless, a quantitative statement can be proven showing that \( PC_{\text{max}} \) can be considered
optimal in some probabilistic sense.

Theorem 1 (Sampling Theorem)

Let \( f : Z \to \mathbb{R} \) be a measurable function on the probability space \((Z, F, Q)\). Select \( N \) independent
\( z_1, z_2, \ldots, z_N \in Z \) according to \( Q \) and set \( \bar{f} = \max_{z \in \{z_j\}_{j=1}^N} f(z) \). Fix an arbitrary real number \( \beta > 0 \).
Then,

$$Q\{z \in Z : f(z) > \bar{f} \} \leq \beta,$$

(18)

with probability greater than $1 - (1 - \beta)^N$.

**Proof.** See Lemma 11.1 in [12].

In the previous statement, $\bar{f}$ is a random variable defined on the product space $Z^N$ with the product probability measure $Q^N := Q \times \ldots \times Q$ hosting the random extraction $(z_1, z_2, \ldots, z_N)$. Thus, $Q\{z \in Z : f(z) > \bar{f} \} \leq \beta$ may or may not be satisfied according to such a random extraction and, so, $Q\{z \in Z : f(z) > \bar{f} \} \leq \beta$ defines a probabilistic event in the space $Z^N$. Theorem 1 says that probability $Q^N$ of this event is greater than $1 - (1 - \beta)^N$.

A possible interpretation of equation (18) is as follows: Suppose that an opponent would like to determine a $\bar{z}$ such that $f(\bar{z}) > \bar{f}$ and she uses the same probabilistic strategy as in the randomized algorithm: she randomly selects a $\bar{z}$ in $Z$ according to $Q$. Then, her probability of “beating” $\bar{f}$ is not greater than $\beta$. In Theorem 1, parameter $\beta$ is arbitrary and, therefore, the probability of success left to the opponent can be reduced at will. On the other hand, (18) is not a deterministic statement and it only holds true with probability $Q^N$ at least equal to $1 - (1 - \beta)^N$. As $\beta$ approaches 0, $1 - (1 - \beta)^N$ tends to 0 and so the statement becomes evanescent. In addition, we observe that statement (18) only quantifies the probability that one can improve result $\bar{f}$ by randomly selecting a new parameter $\bar{z}$. On the other hand, Theorem 1 says nothing on how large such an improvement can be.

In our case, we are interested in maximizing the probability of conflict $PC(t)$ over the time interval $[0, T]$. Observe first that given the flight plans $\{P^i_j\}_{j=0, \ldots, n_i}$, $\{v^i_j\}_{j=1, \ldots, n_i}$, $i = 1, 2$, the corresponding probability of conflict $PC(\cdot)$ is a piecewise smooth function. As a matter of fact, the only points
where $PC(\cdot)$ loses smoothness are those time instants belonging to the finite set $\{T_j \in \mathcal{T} : T_j \leq T\}$, where $\mathcal{T}$ denotes the sequence generated by $\{T^1_j\}_{j=0,\ldots,n_1} \cup \{T^2_j\}_{j=0,\ldots,n_2} \cup \{T^3_1, T^3_2\}$ after linear ordering and eliminating multiple entries, $T^1_j$ and $\{T^1_j\}_{j=0,\ldots,n_1}$ respectively being the time instant when the cross-track error variance saturates $(c_1 + c_3 s^i(T^1_j) = 1)$ and the nominal arrival times at the way points.

This implies that if $Z$ is taken to be the interval $[0, T)$, $F$ the Borel $\sigma$-algebra on $[0, T]$ and $Q$ the uniform probability distribution, Theorem 1 can be applied with function $f(z)$ given by the probability of conflict $PC(z)$. Then the approximate optimization result can be interpreted as follows: There exists an exceptional $S \subset [0, T]$ of Lebesgue measure at most $\beta T$ such that

$$\sup_{[0,T]\setminus S} PC(t) \leq \overline{PC}_{\text{max}} \leq \sup_{[0,T]} PC(t).$$

In other words, $\overline{PC}_{\text{max}}$ is bracketed by the supremum of $PC(t)$ over all $[0, T]$ and the supremum of $PC(t)$ over ‘nearly’ all of $[0, T]$.

We now introduce a method to approximately compute function $PC(t)$ - which is the probability measure of the set $\mathcal{C} = \{z \in \mathbb{R}^2 : \|z\| \leq 5\}$ - given in equation (15). This method allows us to circumvent the difficulty of integrating the probability density function $f(x, t)$ in (16) over the circle $\mathcal{C}$ of radius 5, since such an integral cannot in fact be computed in closed-form.

Given the measurable space $(Z, F)$ and a probability measure $P$ on $(Z, F)$, suppose that we want to compute the measure of $A \in F$

$$P(A) := \int_A P(dz).$$

Then, according to a probabilistic approach, we pick up some points $z_1, \ldots, z_M$ at random from $Z$ according to the distribution $P$ and form the estimate

$$\hat{P}(A) := \frac{1}{M} \sum_{i=1}^{M} I_A(z_i),$$

where $\mathcal{C}$ is the finite set $\{T_j \in \mathcal{C}, \mathcal{C} : T_j \leq T\}$, and $\mathcal{T}$ is the sequence generated by $\{T^1_j\}_{j=0,\ldots,n_1}$ after linear ordering and eliminating multiple entries, $T^1_j$ and $\{T^1_j\}_{j=0,\ldots,n_1}$ respectively being the time instant when the cross-track error variance saturates $(c_1 + c_3 s^i(T^1_j) = 1)$ and the nominal arrival times at the way points.
being \( I_A(z) \) the indicator function of \( A \), that is

\[
I_A(z) = \begin{cases} 
1, & \text{if } z \in A, \\
0, & \text{otherwise}.
\end{cases}
\]

On the one hand, this makes the problem computationally tractable. On the other hand, one can resort to results from the theory of empirical processes to quantify the level of approximation introduced by substituting the integral with a finite sum.

**Theorem 2 (Estimation of Probability Measures)**

Fix an accuracy parameter \( \epsilon \in (0, 1) \). Suppose that \((Z, F)\) is a measurable space and \( P \) is a probability measure on \((Z, F)\). Fix a set \( A \in F \). Extract \( z_1, \ldots, z_M \) i.i.d. samples drawn from \( Z \) in accordance with \( P \) and define

\[
\hat{P}(A) := \frac{1}{M} \sum_{i=1}^{M} I_A(z_i).
\]

Then,

\[
P^M \left\{ (z_1, \ldots, z_M) \in Z^M : |\hat{P}(A) - P(A)| > \epsilon \right\} \leq 2e^{-2M\epsilon^2}.
\]

**Proof.** By Chernoff bound. \( \square \)

Suppose now that we are given, not just a single set \( A \), but a collection of sets \( \mathcal{A} \subset F \). By drawing \( M \) i.i.d. samples \( z_1, \ldots, z_M \) from \( Z \) in accordance with \( P \), we can form empirical estimates for each \( P(A), A \in \mathcal{A} \) as before, i.e.

\[
\hat{P}(A) := \frac{1}{M} \sum_{i=1}^{M} I_A(z_i), \quad \forall A \in \mathcal{A}.
\]

If the set \( \mathcal{A} \) has finite cardinality \(|\mathcal{A}| < \infty\), from Theorem 2 we then get that

\[
q(M, \epsilon) := P^M \left\{ (z_1, \ldots, z_M) \in Z^M : \sup_{A \in \mathcal{A}} |\hat{P}(A) - P(A)| > \epsilon \right\} \leq 2|\mathcal{A}|\epsilon^{-2M\epsilon^2}, \quad (19)
\]

which means that each collection of sets \( \mathcal{A} \) has the property of uniform convergence of empirical probabilities (UCEP) since \( q(M, \epsilon) \to 0 \) as \( M \to \infty \), for each fixed \( \epsilon \), i.e., the empirical probabilities
converge uniformly to their true values as the number of samples $M$ goes to infinity.

Our objective is computing $\widehat{PC}_{\text{max}} = \max_{\tau \in \{\tau_j\}_{j=1}^N} PC(\tau)$. Once we obtain a uniform approximation of $PC(\tau)$ over the finite set $\{\tau_j\}_{j=1}^N$, we can use it to get an estimate of the desired $PC_{\text{max}}$. Such a uniform approximation is obtained next by

1. reducing the problem of computing $PC(t)$ to that of computing the probability measure of a time dependent set according to a fixed probability distribution;

2. applying result (19).

The probability of conflict $PC(t)$, i.e., the measure of the fixed set

$$\mathcal{C} = \{z \in \mathbb{R}^2 : \|z\| \leq 5\}$$

according to the probability distribution $\mathcal{N}(\mu(t), Q(t))$, can be viewed as the measure of an appropriate time dependent set $E_t$ according to the fixed (standard) probability distribution $\mathcal{N}(0, I)$, being $I \in \mathbb{R}^2$ the identity. This is easily shown by operating an appropriate change of variables for the computation of the integral of the normal distribution $\mathcal{N}(\mu(t), Q(t))$ over the set $\mathcal{C}$:

$$PC(t) = \int_{\mathcal{C}} \frac{1}{2\pi \sqrt{\det(Q(t))}} e^{-\frac{1}{2}(x-\mu(t))^T Q(t)^{-1} (x-\mu(t))}.$$

As a matter of fact, by computing the Cholesky factorization of the covariance matrix $Q(t)$

$$Q(t) = L(t)L(t)^T$$

and setting

$$v = L(t)^{-1} [x - \mu(t)],$$

\footnote{A similar procedure is followed in [3].}
we get
\[
PC(t) = \int_{E_t} \frac{1}{2\pi} e^{-\frac{1}{2}v^Tv} dv,
\]
where the set
\[
E_t = \{ v \in \mathbb{R}^2 : [v + L(t)^{-1}\mu(t)]^T L(t)^T L(t)[v + L(t)^{-1}\mu(t)] \leq 25 \},
\]
is obtained by
\[
\|x\| = \|L(t)v + \mu(t)\| \leq 5.
\]
The estimate \( \widehat{PC}(t) \) can therefore be calculated by extracting \( M \) i.i.d. samples \( (v_1, v_2, \ldots, v_M) \) from \( P \sim \mathcal{N}(0, I) \) and computing
\[
\widehat{PC}(t) := \frac{1}{M} \sum_{i=1}^{M} I_{E_t}(v_i), \tag{20}
\]
since \( PC(t) \) is in fact the measure of \( E_t \) with respect to the standard normal distribution in the 2-dimensional Borel space.

According to equation (19), we therefore obtain that by setting \( M = \left\lfloor \frac{1}{2\|\mu\|^2} \ln \frac{2N}{\delta} \right\rfloor \) - where the symbol \( \left\lfloor z \right\rfloor \) denotes the smallest integer greater than \( z \) - \( \widehat{PC}(t) \) in equation (20) is a uniform approximation of \( PC(t) \) to accuracy \( \epsilon \) with confidence \( 1 - \delta \) over each finite set \( t \in \{\tau_1, \ldots, \tau_N\} \).

Observe that maximizing even the modified function \( \widehat{PC}(t) \) is not easy, since there is not a closed-form expression for \( \widehat{PC}(t) \) without which it is not possible to use efficient gradient-based optimization methods. This is the reason why we shall finally resort to the randomized optimization method described before, thus obtaining a fully randomized procedure for the approximate evaluation of \( PC_{\text{max}} \). As a matter of fact, a joint use of the randomized maximization algorithm with the sample estimate of \( PC(t) \) leads to the following fully randomized algorithm for the maximization of \( PC(t) \) over \([0, T]\).
Algorithm for estimating $\text{PC}_{\text{max}}$

Given $\epsilon \in (0, 1)$, $\beta \in (0, 1)$ and $\delta \in (0, 1)$, do the following:

1. extract $N = \left\lceil \frac{\ln(\beta)}{\ln(1-\beta)} \right\rceil$ independent time instants $\tau_1, \tau_2, \ldots, \tau_N \in [0, T]$ according to the uniform probability distribution $Q$ on $[0, T]$;

2. extract $M = \left\lfloor \frac{1}{2\epsilon} \ln \frac{4N}{\delta} \right\rfloor$ independent vectors $v_1, v_2, \ldots, v_M \in \mathbb{R}^2$ according to the normal probability distribution $P \sim \mathcal{N}(0, I)$;

3. for $j = 1, 2, \ldots, N$ compute
   \[
   \hat{\text{PC}}(\tau_j) := \frac{1}{M} \sum_{i=1}^{M} I_{E_{\tau_j}}(v_i),
   \]
   where the ellipse $E_t$ is given by
   \[
   E_t = \{ v \in \mathbb{R}^2 : [v + L(t)^{-1}\mu(t)]^T L(t)^T L(t) [v + L(t)^{-1}\mu(t)] \leq 25 \},
   \]
   being $L(t)$ the lower triangular matrix obtained through the Cholesky decomposition of $Q(t)$;

4. choose $\hat{\text{PC}}_{\text{max}} = \max_{\tau \in \{\tau_j\}_{j=1}^{N}} \hat{\text{PC}}(\tau_j)$. \hfill \Box

**Theorem 3 (Approximate estimation of $\text{PC}_{\text{max}}$)**

Given $\epsilon \in (0, 1)$, $\beta \in (0, 1)$, and $\delta \in (0, 1)$, $\hat{\text{PC}}_{\text{max}}$ is an approximate maximum of $\text{PC}(t)$ over $[0, T]$ to accuracy $2\epsilon$ and level $\beta$ with confidence $1 - \delta$ in the sense that

\[
Q \{ t \in [0, T] : \text{PC}(t) > \hat{\text{PC}}_{\text{max}} + 2\epsilon \} \leq \beta,
\]

with probability greater than $1 - \delta$.

**Proof.** $\hat{\text{PC}}_{\text{max}}$ is a random variable on the probability space $[0, T]^N \times \mathbb{R}^{2M}$. As parameters $\tau_j$’s are independent of one another and the same holds for $v_i$’s and also considering that the extraction of
set \( \{\tau_j\}_{j=1}^N \) is independent of the one of set \( \{v_i\}_{i=1}^M \), we can conclude that the probability measure in \( [0, T]^N \times \mathbb{R}^M \) is just the product probability measure \( Q^N \times P^M \). Now, according to Theorem 1 relation

\[
Q\{t \in [0, T] : PC(t) > \bar{PC}_{\text{max}}\} \leq \beta,
\]

holds true with probability \( Q^N \) greater than or equal to \( 1 - (1 - \beta)^N \). Set

\[
q(M, \epsilon) := P^M \{ (v_1, \ldots, v_M) \in \mathbb{R}^M : \sup_{\tau \in \{\tau_1, \ldots, \tau_N\}} |\bar{PC}(\tau) - PC(\tau)| > \epsilon\}.
\]

Putting together these two results we conclude that the following holds with a product probability \( Q^N \times P^M \) not less than \( 1 - [(1 - \beta)^N + q(M, \epsilon)] \)

\[
Q\{\tau \in [0, T] : PC(\tau) > PC(\arg\max_{\tau \in \{\tau_1, \ldots, \tau_N\}} \bar{PC}(\tau)) + 2\epsilon\} \\
\leq Q\{\tau \in [0, T] : PC(\tau) > PC(\arg\max_{\tau \in \{\tau_1, \ldots, \tau_N\}} PC(\tau))\}
\]

(using (22))

\[
\leq \beta.
\]

(using (21))

Thus, maximizing \( \bar{PC}(\tau) \) over \( \{\tau_j\}_{j=1}^N \) leads to an approximate maximum of \( PC(\tau) \) to accuracy \( 2\epsilon \) and level \( \beta \) with confidence \( 1 - \delta \) where

\[
\delta = (1 - \beta)^N + q(M, \epsilon).
\]

If we now use estimate \( q(M, \epsilon) \leq 2N e^{-2M \epsilon^2} \) in equation (19) (\( A \) and \( |A| \) being respectively \( E_{\tau_j} \) and \( N \)), we can easily conclude that using

\[
N = \left\lfloor \frac{\ln(\frac{\delta}{\ln(1 - \beta)})}{\ln(2)} \right\rfloor
\]

time instants \( \tau \)'s and

\[
M = \left\lfloor \frac{1}{2\epsilon^2} \ln \frac{4N}{\delta} \right\rfloor
\]

vectors \( v \)'s suffices to approximately maximize \( PC(t) \) over \([0, T]\) to accuracy \( 2\epsilon \) and level \( \beta \) with confidence \( 1 - \delta \).
We are now in the position to precisely formulate our conflict detection scheme.

**Algorithm for the aircraft conflict detection**

Fix the accuracy parameter $\epsilon \in (0, 1)$, the level parameter $\beta \in (0, 1)$ and the confidence parameter $\delta \in (0, 1)$. At each time instant when a new radar measurement becomes available, do the following:

1. compute the mean and the covariance matrix of the probability distribution of the aircraft distance on the basis of their current flight plans $\{P_{j}^{i}\}_{j=0,\ldots,n_i}$ (where the first way point $P_{0}^{i}$ represents the current radar measurement of the aircraft $i$ position) and $\{v_{j}^{i}\}_{j=1,\ldots,n_i}, i = 1, 2,$

$$\mu(t) := p_{1}(t) - p_{2}(t)$$

$$Q^{i}(t) := Q^{1}(t) + Q^{2}(t)$$

where $p_{j}^{i}(t)$ and $Q^{i}(t)$, $i = 1, 2$, are given by equations (3) and (4);

2. detect an incoming situation of conflict:

   a. extract at random $N = \left\lceil \ln \left( \frac{1}{\epsilon} \right) \right\rceil$ independent time instants $\tau_1, \tau_2, \ldots, \tau_N \in [0, T]$ according to the uniform probability distribution $Q$ on $[0, T]$;

   b. extract at random $M = \left\lceil \frac{1}{2} \ln \frac{4N^2}{\delta} \right\rceil$ independent vectors $v_1, v_2, \ldots, v_M \in \mathbb{R}^2$ according to the normal probability distribution $\mathcal{N}(0, I)$;

   c. for $j = 1, 2, \ldots, N$ compute

$$\overline{PC}(\tau_j) := \frac{1}{M} \sum_{i=1}^{M} I_{E_{\tau_j}}(v_i),$$

where $E_t = \{v \in \mathbb{R}^2 : [v + L(t)^{-1}\mu(t)]^T L(t) L(t)[v + L(t)^{-1}\mu(t)] \leq 25 \}$, being $L(t)$ the lower triangular matrix obtained through the Cholesky decomposition of $Q(t)$;
3. if \( \bar{C}_{\text{max}} \geq \bar{C} \), issue an alert signal, otherwise do nothing.

It is interesting to observe that the number of samples needed to achieve a certain approximation level in terms of accuracy \( \epsilon \), confidence \( \delta \) and level \( \beta \), is independent of the nature of the sample space and of the probability distribution. In particular, this means that it does not depend on the dimension of the Euclidean space from which the \( v \)’s are extracted. Thus, by extending the proposed approach to the 3D case, the computational load does not significantly increase. This will not be the case if one would resort to numerical methods based on gridding.

4 Validation

In order to validate the conflict detection scheme introduced in Section 3, we use Monte Carlo simulations based on the probabilistic model of the aircraft trajectories and the radar described in Section 2.2. Precisely, the protocol for evaluating the detection scheme performance consists of the following steps:

1. Given the flight plans of the two aircraft, generate pairs of aircraft trajectories over a 20 minutes time horizon according to the discretized version of the stochastic differential equation (8) initialized with (9) (sample time set equal to 1 second), and the corresponding radar measurements according to equation (10) (radar measurement time set equal to 12 seconds);

2. For each pair of simulated trajectories, execute the detection algorithm at every radar measurement time, each time using the updated flight plans and time horizon. These are respectively obtained by removing the way points which have been surpassed and setting the first way point equal to the current radar measurement, and by subtracting the elapsed time from
the 20 minutes initial horizon. A conflict is declared as soon as the estimated value of the maximum probability of conflict exceeds the prescribed threshold;

3. Compute the probability of false alarm $P(FA)$ and the probability of successful alert $P(SA)$, i.e., the ratio between the number of alerts issued when there was no conflict and the total number of cases when there was no conflict ($P(FA)$), and the ratio between the number of alerts issued before a situation of conflict effectively happens and the total number of conflicts ($P(SA)$).

It is evident that the computed $P(FA)$ and $P(SA)$ estimates depend on the value of the threshold used to decide whether an alert should be issued, being both decreasing as functions of the threshold. The System Operating Characteristic (SOC) curve drawn in Figure 5 represents the probability of successful alert versus the probability of false alarm, parametric in the threshold. An ideal conflict detection scheme should operate at point $(0,1)$ where, in fact, there are no false alarms and all the situation of conflicts are detected. On the other hand, a real conflict detection scheme cannot operate at this point - irrespective of the threshold chosen - due to the uncertainty affecting the aircraft positions. However, the more the SOC curve approaches the point $(0,1)$, the better is the performance of the system. The threshold can, therefore, be defined on the basis of this consideration thus striking an “optimal” compromise between the number of false alarms and successful alerts.

We now describe the results obtained by Monte Carlo simulation in the two cases of a $90^\circ$ path crossing angle configuration and a zig-zag flight paths configuration. For these two cases, we draw the SOC curve and compare the performance obtained by the proposed detection algorithm with those obtained by implementing the detection algorithm described by Erzberger et al. in [8].
The algorithm in [8] is, in fact, based on the same description of the along-track and cross-track errors we made in our model for the conflict prediction. Differently from our approach, the value of the probability of conflict on which the detection is based on is the one corresponding to the minimum deterministic (computed on the nominal trajectories) point. Moreover, such a value of the probability of conflict is computed by an analytical approximation of the integral of the distance probability density function obtained by extending the circular domain of integration to a whole strip (see [3] for more details).

**Example 1 (90° path crossing angle configuration)**

Consider the case when the two aircraft are flying straight at the same altitude along paths whose crossing angle is 90° at speeds of 485 nmi/h and 496 nmi/h. For sake of clarity, in Figure 6 we have drawn a realization of the trajectories obtained by performing step 1 of the described simulation protocol. By running our detection algorithm and the one proposed in [8] on 1000 Monte Carlo samples, we estimate the probability of successful alert and false alarm and draw the SOC curve for both the algorithms (see Figure 7, where the solid line curve corresponds to the detection algorithm proposed in this paper). We can then compute the “optimal” threshold

---

1. In the final version of the paper we will increase the number of Monte Carlo samples to 10000.
Figure 6: Sample pair of simulated trajectories for Example 1. The ○ stand for way points.

Figure 7: SOC curves for our algorithm (solid) & Erzberger algorithm (dotted). The ○ stand for the “optimal” threshold points (Example 1: $\epsilon = 0.025; \delta = 0.1; \beta = 0.05; 1000$ simulations)

Figure 8: Plots of $P(FA)$ and $P(SA)$ vs threshold (Example 1)
value $\bar{PC}$ as the one corresponding to the point of the SOC curve nearest to ideal operating point $(0,1)$. This leads to $\bar{PC} = 0.82$ (P(FA)=0.072, P(SA)=0.871) and $\bar{PC} = 0.87$ (P(FA)=0.357 P(SA)=0.848) respectively for our detection procedure and for the one proposed in [8]. Thus, to make the comparison our algorithm has a slightly higher probability of successful detection and a false alarm probability of 7.2%, which is 1/5 that of [8].

Example 2 (zig-zag flight paths configuration)

In this example, we consider the case when the sequence of way points in the flight plans describes a zig-zag configuration at a fixed altitude with the speeds being $v_1^1 = 505$ nmi/h and $v_1^2 = 455$ nmi/h, $v_2^1 = 480$ nmi/h and $v_2^2 = 470$ nmi/h. Figure 9 represent a realization of the trajectories at the fixed altitude.

The estimates of the probability of successful alert and false alarm obtained by 1000 Monte Carlo samples are used to plot the SOC curves in Figure 10, where the solid line SOC curve corresponds to the detection algorithm proposed in this paper. In this case, the “optimal” threshold values for the introduced detection procedure and the one proposed in [8] are respectively $\bar{PC} = 0.52$, corresponding to P(FA)=0.168 and P(SA)=0.852, and $\bar{PC} = 0.63$, corresponding to P(FA)=0.346 and P(SA)=0.799. Thus, our algorithm has a slightly higher probability of successful detection and
Figure 10: SOC curves for our algorithm (solid) & Erzberger algorithm (dotted). The ◦ stand for the “optimal” threshold points (Example 2: $\epsilon = 0.025; \delta = 0.1; \beta = 0.05; 1000 simulations$)

Figure 11: Plots of $P(FA)$ and $P(SA)$ vs threshold (Example 2)
a false alarm probability of 16.8%, which is 1/2 that of [8].

In these two examples, the SOC curve obtained by the algorithm of [8] is lower than the one obtained by the algorithm proposed in this paper. In our opinion, the reason for this rests on the fact that the parameter used for detecting a situation of conflict in [8] is an over approximation of the probability of conflict at the time of minimum deterministic distance (which is assumed to be the most critical time instant over the considered time horizon). As a matter of fact, this is reflected in the higher value assumed by the probability of false alarm in the algorithm of [8] with respect to our approach, irrespective of the chosen threshold. On the other hand, this increase of $P_{FA}$ is not compensated by an adequate increase of the probability of successful alert as it is shown in Figure 8 and 11 representing $P_{FA}$ and $P_{SA}$ as a function of the threshold for the two examples, with the solid line corresponding to our detection scheme and the dotted line corresponding to the detection scheme in [8].

It is important to observe that, as expected, different flight plans lead to different SOC curves. A sensitivity analysis of the dependence of the threshold on the flight plans should be performed by parameterizing them as a function for example of the crossing angles, minimum deterministic distance, and time of minimum distance, in order to set a value for the threshold which is appropriate for the typically encountered path configurations. This is currently under study.

5 Conclusions

We outlined a framework for conflict prediction and resolution for pairs of aircraft moving on the same horizontal plane, and specifically dealt with the prediction component. We are currently working on formulating algorithms for resolution and investigating the different design alternatives listed in the introduction. Further objectives are extending our approach to the three dimensional
case and multiple aircraft. The extension to three dimensions is straightforward, while the correct way of extending this approach to multiple aircraft is less clear. It may turn out to be adequate to resolve conflicts pairwise (as proposed in [3]), but this approach has the potential for combinatorial explosion and it is unclear whether it will always lead to overall resolution.

Once the prediction/resolution algorithm has stabilized, we hope to be able to test it in human-in-the-loop simulations. This will allow us to tune the various parameters, and assess its impact on ATC workload. In parallel we are working towards a methodology for formally evaluating the safety properties of the proposed algorithm. This will hopefully lead to a more general probabilistic verification methodology for hybrid systems.

References


