AIRCRAFT COLLISION

Interim report

May 8, 2020

Saghir Ilias, Elkadiri Salaheddine

ÉCOLE POLYTECHNIQUE

IP PARIS

CONTENTS

1 INTRODUCTION

In order to avoid conflicts between aircraft, aeronautical regulations provide minimum separation standards to be observed between airliners. At take-off and landing airplanes should be at least two minutes apart. And to avoid any risk of collision while cruising, airplanes operating above 5,000 m must be at least 9 km (5 nmi) horizontally and at least 300 m (1,000 ft) vertically apart.

In this project, we elaborate a probabilistic study of conflicts between planes in the medium term (20 min to 1 hour). A conflict becomes imminent when the prescribed separation distance between two planes is less than a given threshold. In the medium term, the various sources of uncertainty affecting the movement of the aircraft generally cause deviations from the nominal flight path, which cannot be overlooked when detecting conflicts. Calculating the probability of conflicts between aircraft trajectories is generally a delicate task as they are difficult to model, but also because their rarity prevents an exact estimate. In our work, we rely on the Erzberger-Paielli model for the generation of trajectories over durations of 20 minutes. This model involves the different deviations resulting from meteorological circumstances and other uncertainties. Obviously, the more complex the model, and the fewer conflicts, the more costly the generation of simulations in terms of computation and time. Given this constraint, we are examining important sampling solutions to accelerate the convergence of results. These solutions are inspired by rare event simulation methods.

To calculate the probabilities of conflict we used three methods:

- **Naive Monte Carlo**: Estimating probabilities as little as 10^{-5} through simple simulations.
- **Monte Carlo with mean shift**: A change of the mean enables us to decrease the initial distance between the planes and therefore to increase the probability of conflict which should subsequently be calculated with fewer simulations.
- **Monte Carlo with variance shift**: At a given distance, a change of variance makes it possible to increase the disturbances and therefore increase the probabilities of conflict, this method makes it possible to accelerate the convergence of the results and allowed us to approximate probabilities as little as 10^{-10}.

Using these methods we were able to calculate the probability of conflict and observe the impact of the initial distance between planes.

2
MODEL

Our study is based on a conflict probability estimation model suggested in the work of Russel A. Paielli and Heinz Erzberger (1997): "Conflict probability estimation for free flight" In this model, the aircraft's movement is considered plane, and passes through waypoints at regular intervals (20 minutes) during which the nominal trajectory is assumed to be straight. Between two waypoints, the position of an airplane is given by its coordinates in a frame centered on the first waypoint, consisting of two axes a and c : one being parallel to the direction of flight (*along-track* component) and the other perpendicular (*cross-track* component).

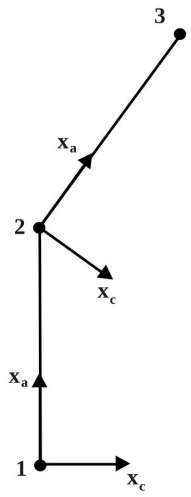

Figure 1: Example of flight plan with 3 waypoints

At first, we will limit ourselves to the study of a trajectory located between two waypoints, simulating 20 minutes of flight time. Thus, reference frame will be fixed by the initial conditions (positions and angles).

Weather hazards (such as wind), piloting disturbances, and measurement errors cause planes to follow different trajectories from the ideal flight plans above. These deviations will therefore be modeled by Gaussian perturbations of the coordinates *along-track* and *cross-track*. So, if $x_t = (x_{a,t}, x_{c,t})$ is the position of the plane over time in a local coordinate system centered on the previous waypoint, we have:

$$x_{a,t} = vt + M_{a,t} \text{ and } x_{c,t} = M_{c,t}$$

where $M_{a,t}$ and $M_{c,t}$ are centered continuous Gaussian processes with the covariance matrix being given by (for $t < s$):

$$\mathbb{C}\text{ov}(M_{a,t}, M_{a,s}) = r_a^2 t^2$$
$$\mathbb{C}\text{ov}(M_{c,t}, M_{c,s}) = \sigma_c^2(1 - e^{-2\frac{r_c}{\sigma_c}vt})e^{-\frac{r_c}{\sigma_c}v(s-t)}$$

For civil turbine aircraft, we have $r_a = 0.25$ nmi/min, $r_c = 1/57$ and $\sigma_c = 1$ nmi in the usual units.

Interpretation :

- *along-track* component : With time, the along-track coordinate is less and less certain because of the accumulation of navigation hazards, the disturbance is interpreted as an uncertainty about the speed.

- *cross-track* component : On the cross-track coordinate, the phenomenon is different and the variance of the uncertainty stabilizes when t is large. This is interpreted as follows: along the along-track component, the pilot and the on-board instruments seek to correct the deviations from the ideal trajectory inducing a return to 0 phenomenon.

For simulation purposes, we cannot consider trajectories that are continuous in time but only sampled at regular intervals, of one minute in our case. We will say that two planes are in conflict if the distance between them becomes less than a minimum threshold of 0.1 nmi at some point in their trajectories.

3
CONFLICT PROBABILITY

3.1 Simulation

Initially, we will be interested in a portion of trajectory between two waypoints, sampled by intervals of one minute. We will consider two planes with parallel trajectories and neglect the *along-track* perturbation (r_a =0). For each of the planes, the coordinates $(x_{c,t})_{1\leq t\leq 20}$ are given by a random 20 dimensional Gaussian vector whose covariance matrix has been described before, whose mean depends on the nominal distance between the two flight plans. Using the numpy library for Python, we generate simulations of those trajectories, which we then represent using the matplotlib graphics library.

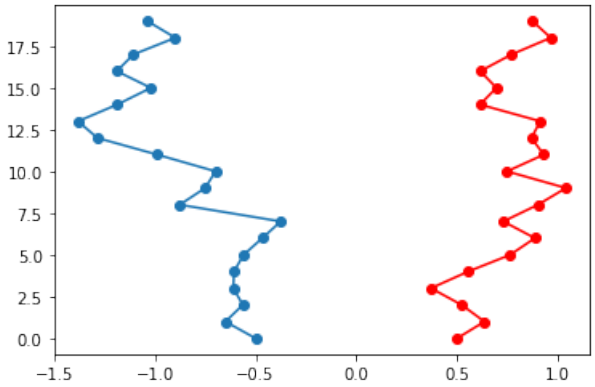

Figure 2: Simulation of two plane trajectories 1 nmi apart

3.2 Estimate

We now seek to estimate, for different flight plans, the probability of conflict. At first, we consider parallel trajectories and try to approach the probability as a function of the nominal distance between the planes d.

3.2.1 • Naive Monte-Carlo

Simulating two trajectories amounts to simulating a couple $Z = (X, Y)$ of independent Gaussian random vectors. The probability of conflict can be seen as the expected value, under the law defined by the model (covariance matrix and distance between trajectories) of a function of the pair of variables $f(Z) = 1$ if $\min_{1 \leq t \leq 20} |X_t - Y_t| < 0.1$ and 0 otherwise.

$$\mathbb{P}(A) \approx \frac{1}{N} \sum_{i=1}^{N} f(Z^{(i)})$$

We then simulate, for different values of the distance d, a large sample of pairs of trajectories ($N = 100{,}000$). We first start with small values of d which we then increase.

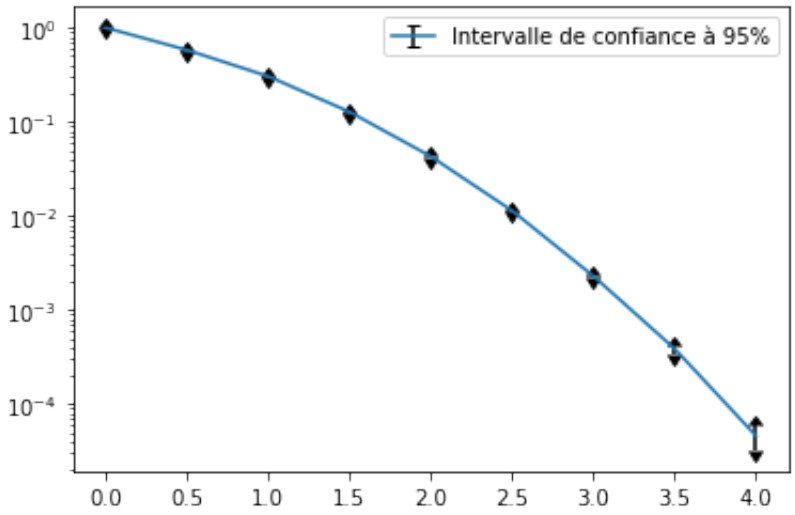

Figure 3: Probability of collision for different values of d, with 95% confidence intervals.

As we would expect, the probability decreases rapidly with distance. We note that for $d = 4$, the probability of conflict is already of the order of 10^{-5} and falls within the field of rare events. For $d = 5$, the estimator is no longer efficient.

3.2.2 • Importance sampling : mean shift

The rarity of the events that we study limits the performance of the classic Monte-Carlo estimator. We therefore use an importance sampling method based on a mean shift :

$$\mathbb{P}(A) = \mathbb{E}_{\mathbb{P}}(1_A) = \mathbb{E}_{\mathbb{Q}}(1_A \frac{d\mathbb{P}}{d\mathbb{Q}})$$

Under $\mathbb{P}$, X and Y are respectively centered on $\mu = (d/2, ..., d/2)^T$ and $-\mu = (-d/2, ..., -d/2)^T$ and $Z = (X, Y)$ is of density :

$$g(Z) = \frac{1}{(2\pi)^n det(C)} e^{-\frac{1}{2}((X-\mu)^T C^{-1}(X-\mu)+(Y+\mu)^T C^{-1}(Y+\mu))}$$

Under $\mathbb{Q}$, X and Y are respectively centered on $\mu^* = (d^*/2, ..., d^*/2)^T$ and $-\mu^* = (-d^*/2, ..., -d^*/2)^T$ and $Z = (X, Y)$ possesses the density :

$$g^*(Z) = \frac{1}{(2\pi)^n det(C)} e^{-\frac{1}{2}((X-\mu^*)^T C^{-1}(X-\mu^*)+(Y+\mu^*)^T C^{-1}(Y+\mu^*))}$$

Thus, we have for N large enough : $\mathbb{P}(A) \approx \frac{1}{N} \sum_{i=1}^{N} f(Z^{(i)}) \frac{g(Z^{(i)})}{g^*(Z^{(i)})}$ where the $Z^{(i)}$ are simulated under $\mathbb{Q}$

Indeed, the more the trajectories are distant, the lower the probability. Reducing the distance allows to observe the realization of the event more often , and a good choice of the mean shift for the Gaussian vectors provides an estimator with a smaller variance. At d fixed, the parameter d^* which minimizes the variance of the estimator is given by: $d^*_{min} = argmin\ \mathbb{E}_{d^*}[(f(Z^{(i)})\frac{g(Z^{(i)})}{g^*(Z^{(i)})})^2]$ whom we approximate through Monte-Carlo simulations of this quantity for a sample of values of d^*. However, we had difficulties implementing this method effectively, the variance of the estimator being very sensitive to the change of mean, it was difficult to determine the optimal change parameter with a limited number of simulations which made the method less efficient than a naive Monte Carlo.

3.2.3 • Monte-Carlo with variance shift

Another way to act on the frequency of collisions is to increase the uncertainty on the trajectories, we tried to implement this in two different ways:

1. Global change :

The idea here is to multiply the covariance matrix of one of the Gaussian vectors (X for example) by a well chosen coefficient $\alpha > 1$, this has the effect of increasing the variance on the coordinates of the trajectory and therefore the uncertainty on them, and thus increase the probability of conflict. The change is as follows:
Under $\mathbb{P}$, X and Y are respectively centered on $\mu = (d/2, ..., d/2)^T$ and $-\mu = (-d/2, ..., -d/2)^T$ and the density of $Z = (X, Y)$ is given by :

$$g(Z) = \frac{1}{(2\pi)^n det(C)} e^{-\frac{1}{2}((X-\mu)^T C^{-1}(X-\mu)+(Y+\mu)^T C^{-1}(Y+\mu))}$$

Under $\mathbb{Q}$, X and Y are respectively centered on $\mu = (d/2, ..., d/2)^T$ and $-\mu = (-d/2, ..., -d/2)^T$ and and the density of $Z = (X, Y)$ is given by :

$$g^*(Z) = \frac{1}{(2\pi)^n \alpha^{\frac{n}{2}} det(C)} e^{-\frac{1}{2}((X-\mu)^T \frac{1}{\alpha} C^{-1}(X-\mu)+(Y+\mu)^T C^{-1}(Y+\mu))}$$

Once again, we have for N large enough : $\mathbb{P}(A) \approx \frac{1}{N} \sum_{i=1}^{N} f(Z^{(i)}) \frac{g(Z^{(i)})}{g^*(Z^{(i)})}$ where the $Z^{(i)}$ are simulated under $\mathbb{Q}$

Then, it's mostly about choosing the right parameter α, we can approach the optimal parameter by Monte-Carlo estimates of the variance for sampled values for α, for example, for $d = 3.5$, and $N = 100{,}000$ simulations, we have the following curve :

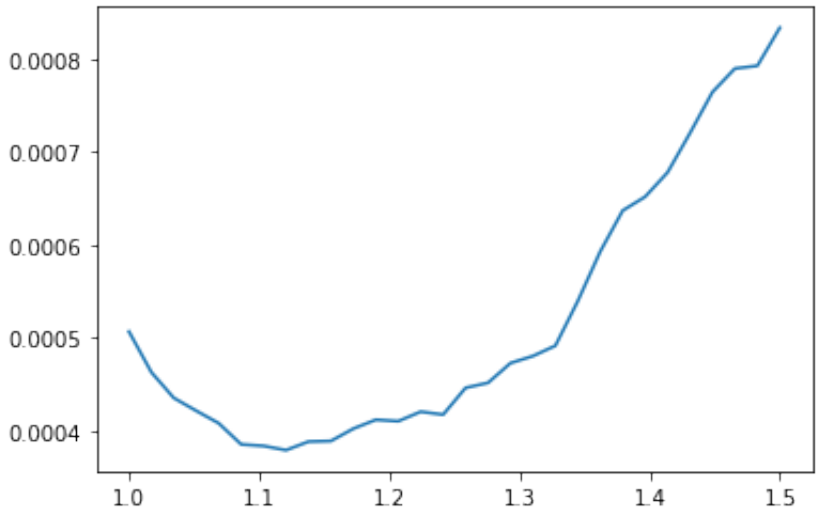

Figure 4: Empirical variance for different values of alpha

We notice that $\alpha \approx 1,1$ is a good parameter choice. When compared with the regular Monte-Carlo method, this estimator is much more efficient :

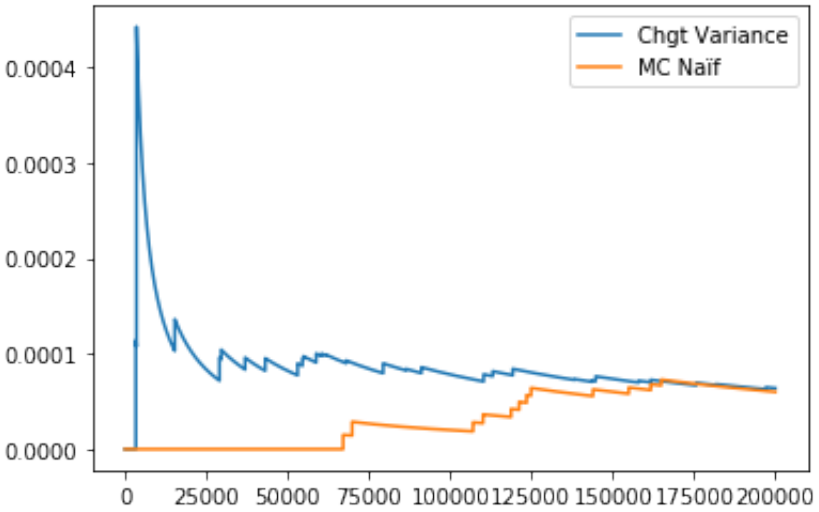

Figure 5: Naive MC vs. Variance shift

We do this for different values of d in order to obtain acceptable confidence intervals for the probabilities of conflict :

Distance	4	4,5	5	5,5	6
α	1,4	2,25	2,8	3,4	4
CI 95%	$[6, 7.10^{-5}, 7, 8.10^{-5}]$	$[4, 8.10^{-6}, 8.10^{-6}]$	$[2, 1.10^{-7}, 6, 7.10^{-7}]$	$\leq 1, 3.10^{-7}$	$[8, 1.10^{-11}, 7, 1.10^{-10}]$

2. Local variance shift :

Instead of modifying the entirety of the coefficients of the covariance matrix, one could notice that, under this model, the dependence between close coordinates $textitcross - track$ is more and more important for large times. Indeed, at $s - t$ fixed, covariance increases with t. This is reflected in the fact that the path of an aircraft, over time, has less and less chance of suddenly deviating from its last positions. Thus, we could give more chance to planes to conflict if we act on the last portion of the trajectory. We translate this by a change of probability acting on the last block of the matrix C.

Under $\mathbb{Q}$ the density of $Z = (X, Y)$ is given by :

$$g^*(Z) = \frac{1}{(2\pi)^n det(C^*)} e^{-\frac{1}{2}((X-\mu)^T (C^*)^{-1}(X-\mu)+(Y+\mu^*)^T (C^*)^{-1}(Y+\mu))}$$

avec $C^*_{ij} = \alpha C_{ij}$ pour $k+1 \leq i, j \leq 20$ et $C^*_{ij} = C_{ij}$ sinon

$\theta = (k, \alpha)$ will be a parameter to determine, which minimizes the quantity $\mathbb{E}_\theta[(f(Z^{(i)})\frac{g(Z^{(i)})}{g^*(Z^{(i)})})^2]$.

4
CONCLUDING REMARKS

The magnitude of the probabilities of the events studied means that their statistical analysis becomes very delicate and requires a careful and appropriate approach. At this stage of the study, we have managed to approach probabilities as little as 10^{-10} but at the cost of a very large number of simulations (up to around 10 million). It is obvious for the rest of our project that we will have to build more efficient estimators in terms of convergence time. This will be achieved either by the refinement of the methods already proposed (local change of the terms of covariance), or by the exploration of other techniques (splitting ...), but also by the optimization of the numerical calculation that we have tried to achieve by exploiting parallel computing and vectorization provided by scientific libraries as much as possible. Our objectives thereafter will be to simulate the distribution of the time of conflict as well as the conditional distribution of the position of conflict, and this for different types of trajectories and speeds.

5 ANNEX

```
#MC Naïf

d=3
N=100000

m=np.zeros(20)
cov=np.zeros((20,20))
for s in range(20):
    cov[s,s]=(sigma_c**2)*(1-np.exp(-2*r_c*v*(s)/sigma_c))
    for t in range(s):
        cov[t,s]=cov[s,t]=(sigma_c**2)*(1-np.exp(-2*r_c*v*(t)/sigma_c))*np.exp(-r_c*v*(s-t)/sigma_c)

X = np.random.multivariate_normal(m, cov,size=N)-d/2
Y = np.random.multivariate_normal(m, cov,size=N)+d/2
Z=np.min(abs(Y-X),axis=1)<0.1
print(np.mean(Z))
```

```
0.00242
```

Figure 6: Naive Monte Carlo

```
def xtcx(x,C):
    return(np.apply_along_axis(lambda x:np.dot(x.T,np.dot(C,x)),0,x))
```

Figure 7: This function allows to calculate $x^T Cx$ for two given matrices x and C by exploiting the vector representation numpy and avoiding to calculate an excessively large matrix

```
#Changement de moyenne :

d=2
N=100000
deltas = np.linspace(0,2,30)
l=[]
mu=np.zeros(20)+d
X = np.random.multivariate_normal(mu, cov,size=N)
Y = np.random.multivariate_normal(np.zeros(20), cov,size=N)
inv_C=np.linalg.inv(cov)
delta=1.08
mu_=(mu-delta).reshape(20,1)
X_=(X-delta).T
Y_=Y.T
Z=np.min(abs(X_-Y_),axis=0)<0.1
mu=mu.reshape(20,1)
r=np.exp(0.5*(-xtcx(X_-mu,inv_C)+xtcx(X_-mu_,inv_C)))
print(np.mean((Z*r)))
```

Figure 8: Monte Carlo with mean shift

```python
# Choix de alpha :


d=4
N=2000000
alphas = np.linspace(1,2,30)
l=[]
mu=np.zeros(20)+d/2
X = np.random.multivariate_normal(mu, cov,size=N)
Y = np.random.multivariate_normal(-mu, cov,size=N)

for alpha in alphas:
    mu=np.zeros(20)+d/2
    X_=(((alpha)**0.5)*X+(1-(alpha)**0.5)*d/2).T
    Y_=Y.T
    mu=mu.reshape(20,1)
    Z=np.min(abs(X_-Y_),axis=0)<0.1
    inv_C=np.linalg.inv(cov)
    r=np.exp(0.5*(xtcx(X_-mu,(1/alpha-1)*inv_C)+20*np.log(alpha)))
    l.append(np.mean((Z*r)**2))

plt.plot(alphas,l)
plt.yscale('log')
plt.show()
```

Figure 9: Simulations in order to approximate the optimal value of *alpha* for our change in variance

```python
#Chgt de variance :
N=1000000
d=4
mu=np.zeros(20)+d/2
cov=np.zeros((20,20))
for s in range(20):
    cov[s,s]=(sigma_c**2)*(1-np.exp(-2*r_c*v*(s+1)/sigma_c))
    for t in range(s):
        cov[t,s]=cov[s,t]=(sigma_c**2)*(1-np.exp(-2*r_c*v*(t+1)/sigma_c))*np.exp(-r_c*v*(s-t)/sigma_c)

inv_C=np.linalg.inv(cov)
alpha=1.4
mu=np.zeros(20)+d/2
X = np.random.multivariate_normal(mu, cov,size=N)
Y = np.random.multivariate_normal(-mu, cov,size=N)
X_=(((alpha)**0.5)*X+(1-(alpha)**0.5)*d/2).T
Y_=Y.T
mu=mu.reshape(20,1)
Z=np.min(abs(X_-Y_),axis=0)<0.1
r=np.exp(0.5*(xtcx(X_-mu,(1/alpha-1)*inv_C)+20*np.log(alpha)))
print(np.mean(Z*r))
```

Figure 10: Monte Carlo with variance shift

REFERENCES

[1] Maria Prandini; Oliver J. Watkins : Probabilistic aircraft conflict detection. 30 May 2005.

[2] Heinz Erzberger Russel A.Paielli : Conflict probability estimation for free flight. *NASA Ames Research Center.*

[3] Edward Nelson : Dynamical theories of brownian motion. *Department of Mathematics, Princeton University.*

[4] Jianghai Hu; Maria Prandini et Shankar Sastry : Aircraft conflict prediction in the presence of a spatially correlated wind field.