Controlled Airspace Infringements and Warning System

Yousra Almathami
Department of Computer Science and Engineering
University of Connecticut
Storrs, Connecticut 06269
Email: yousra.almathami@uconn.edu

Reda Ammar
Department of Computer Science and Engineering
University of Connecticut
Storrs, Connecticut 06269
Email: reda@engr.uconn.edu

***Abstract*—A major concern for Air traffic controllers (ATC) are facing on a daily basis are controlled Airspace (CAS) infringements. An infringement is when aircraft penetrates CASs without an advanced clearance from the ATC. These infringement may cause a conflict or a mid-air collision with a commercial aircraft flying within CAS. As a result, a ground based safety net called Controlled Airspace infringement Tool (CAIT) is used by ATCs which warns them if any aircraft within uncontrolled airspace (UCAS) has penetrated (CAS). In our previous paper, we developed a probabilistic CAS infringement tool (PCAIT) that predicts future aircraft locations using the Kalman filter and calculate their probability of infringement. In this paper, we review the factors behind CAS infringements and build a classifier based on them to enhance our decision about future infringements. This model "warning system" could provide ATCs with more time to resolve any possible future conflicts.**

***Index Terms*—Kalman filter, Controlled airspace, airspace infringement, safety net, Support vector machines.**

I. Introduction

Most countries have airspace divided into seven classes; five of which are controlled and monitored by ATCs and two uncontrolled airspace (UCAS) where ATC is not required to monitor. However, any pilot flying within UCAS and about to fly into CAS, it is essential for him or her to communicate with ATC and get an advance clearance to avoid possible future conflicts. A conflict is when one aircraft loses its minimum separation to another. The minimum separation for an aircraft is 5 nautical miles horizontally and 1000ft vertically. The separation standard around airport area is 3 nautical miles horizontally.

The ATC monitors the traffic of CAS using information gathered from radars scattered around different CAS grounds. Currently, ATCs are using a ground based safety tool called Controlled airspace infringement tool (CAIT) to monitor infringements. It warns them if one or several aircraft had penetrated CAS zones.

These infringements are a major safety concern to ATCs and every aircraft around the conflict zone. They can cause a possible conflicts with different commercial aircraft. They also cause disruption to flight operations by adding more workload on the pilot and the ATC such as changing the flight routes and finding a safe manoeuvre to avoid a collision.

The majority of these infringements are light weight aircraft which rely on visual flight rules (VFR). Therefore, ATC wont be able to know the infringing aircraft ID or its flight path unless the pilot contacts them. In addition to small aircraft infringements, a number of reported drones were spotted by pilots around airport areas.

To aid ATCs with conflict resolution, CAIT was developed to detect any infringements that occurred. However, it only warns the ATC if it has already infringed the CAS which gives the ATC less time to resolve any possible conflict that may arise. In our previous paper, we examined the follwing:

1) Building two prediction models: Constant velocity and constant acceleration Kalman filters
2) On-line learning of the Kalman filter's errors
3) Reviewed and extended available probability of infringement methods

This will provide advance warning to ATCs for them to resolve it and maintain the flow of aircraft in the CAS in the same time. In this paper, we will examine the factors behind infringements and merge them with our current model to warn ATC with minimal false alerts.

The paper is organised as follows: in Section II is the literature review; in Section III is the probability of infringement using PCAIT; in Section IV is infringement frequency analysis; In Section V classification using flight information; In Section VI is our warning system and results. Finally, the conclusion and the proposed future work are presented in Section VII

II. Literature Review on Conflict Detection

Several researches were introduced to solve the issue of a conflict between two aircraft. Some have developed new models with their own conflict detection algorithms and others have optimised current models used by ATCs. Different approaches to determine the probability of a conflict and conflict resolution have been introduced: one approach adopted was by Yang and Kuchar [1] who created an alerting system for free flight that uses Monte Carlo simulations (MC) to estimate the probability of a conflict of traffic encounters over time. Because it is a free flight alerting system, they assumed that there is a data-link between aircraft to communicate with each other in the airspace. The idea of the data-link is to collect other aircraft's information in the airspace, such as current state and future trajectory. The current state information for both aircraft contains speed, heading and altitude which are then fed into the MC engine as the initial state. Each MC run projects a path for both aircraft and predicts if a conflict is ahead. The prediction process issues an alert when the host aircraft's protected zone is violated by an intruder aircraft. The protected zone was divided into four stages (where 1 means a remote intruder whereas 4 means nearby intruder and Air Traffic Controllers should take control from here). The size of the protected zone is a trade off between the successful alerts (SA) and unnecessary alerts (UA) and it was examined by using a System Operative Characteristic (SOC). Using Monte Carlo simulations with on-line applications are computationally expensive because prediction models are limited within time constraints. To reduce the amount of computation, Yang and Kuchar [2] proposed in other research incorporating intent information into the Monte Carlo simulation engine. Their method was that by knowing the waypoints of two aircraft, they can create a series of straight segment lines between these waypoints, where each endpoint represents a change of heading or speed. Then check if the host's segment line intersects the intruder's trajectory line.The Monte Carlo simulation engine is fed with intent information, current state, protected zone size and uncertainties such as tracking errors, manoeuvring characteristics then outputs a probability of a conflict $P(conflict)$. Another study was conducted in [3] to predict a conflict in free flight; their method is applied to two aircraft travelling along a straight line with constant errors. They modelled the trajectory prediction errors as randomly distributed based on the live air traffic data and combined covariance error pairs into a single covariance error relative to the position, this was done because common errors cancel each other. The conflict probability prediction is the area under the combined error ellipse within the extended conflict zone. A recent research conducted in [4] under the SESAR WP-E project. Their aim was to build a model which can predict future location of a general aviation aircraft using historical flight paths as an input and produce future paths and the way it is delivered to the ATC management. Their system was meant to help the ATC know about the future volume of flight operations and trigger an alert if the aircraft is approaching controlled Airspace. Their method assumes that the aircraft would be equipped with transceivers and receivers communicating with a ground system. This ground system gathers the information broadcast by the airborne transceivers and predicts the flight path ahead.

Generally, light weight aircraft are not equipped with advanced transponder that a commercial aircraft has which sends their information to the ATC. In order for these aircraft to be detected, the ATC will have to rely on two things: the primary surveillance radar which only detects the location of the aircraft with imprecise altitude; and the pilot communication. Since most aircraft fly under VFR, they do not have specific flight path. A survey was conducted by Eurocontrol [5] to find out why infringements occur more frequently. These are the common reasons:

- Pilot is unfamiliar with the airspace and/or its boundaries.
- Avoid a bad weather such as scattered clouds
- Pilot is unsure of airspace or lost
- Pilot experience
- Lack of published VFR routes
- Outdated maps or GPS database

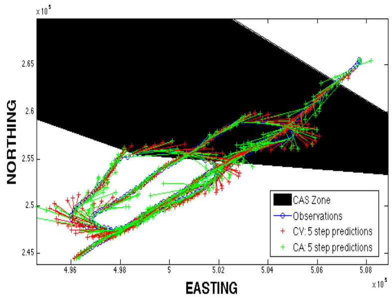
In this research we are aiming to enhance our alerting system (PCAIT) by analysing a collection of past infringements and apply them to our model as warning system as whether to alert ATC 5 locations steps ahead.

III. Probability of infringements using PCAIT

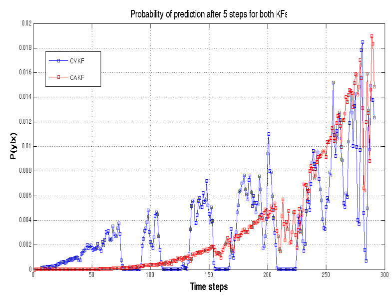
We developed a switching Kalman filters in [6] with an online error learning as our tracking method. We developed two KFs where each has the following parameters:

$$\lambda_m = (A_m, H, Q_m, R)$$

where m is the flight mode (constant velocity and constant acceleration). Both A_m and H are state and observation transition matrices and will be fixed the entire prediction process. Both Kalman filters (KF_{cv}, KF_{ca}) state error covariances (Q_{cv}, Q_{ca}) are being learned using the expectation maximisation algorithm to eliminate them from being propagated with time. Figure 1 shows (a) both KFs prediction locations after 5 steps ahead. Each step is 4 seconds, the time (it takes a radar to make one revolution). In this figure, the aircraft has infringed CAS zone (black polygon). To measure both KFs accuracies in (b), we plotted their probability of the 5th prediction locations given the observation at that time. The blue and red lines represent the probability of the 5th step predictions using CV and CA KFs respectively. We noticed that



(a) 5 step predictions of real observation



(b) Probability of the 5th step prediction for both KFs

Fig. 1: Example of flight path predictions with their probabilities

CA KF predictions has a higher probability than CV when the aircraft makes a turn.

To find their probability of infringement $P_m(i)$, we used two methods: Monte carlo sampling (MC) and shortest distance (SD) in [7]. It uses the aircraft prediction $\widehat{x_t}$ and its error covariance $\widehat{P_t}$ (zone of uncertainty around the prediction) to find $P(I)_t$.

A. Monte Carlo Sampling

The MC sampling draws a random number of samples N from the prediction error covariance $\widehat{P_t}$, then calculates the fraction of samples which fall inside the CAS as follows:

$$P(I) = \frac{\#\text{of samples} \in CAS}{N}$$

B. Shortest Distance Method

Here it calculates the shortest distance d from the prediction to the nearest CAS boundary C to estimate the probability of infringement. Given d,$\widehat{x_t}$ and $\widehat{P_t}$, it uses the error function to find $P(I)$ as follows:

$$P(I) = \frac{1}{2} + \frac{1}{2} erf(\frac{d}{\sqrt{2}}) \ , \ \widehat{x_t} \in CAS$$
$$P(I) = \frac{1}{2} - \frac{1}{2} erf(\frac{d}{\sqrt{2}}) \ , \ \widehat{x_t} \notin CAS$$

In figure 2, it shows the probability of infringement for the 5th steps ahead for (CV,CA) models shown in figure 1 using both methods. Because SD method accuracy around a CAS corner is low, we use MC sampling instead. In the next section we will analyse the factors or infringements and their frequency.

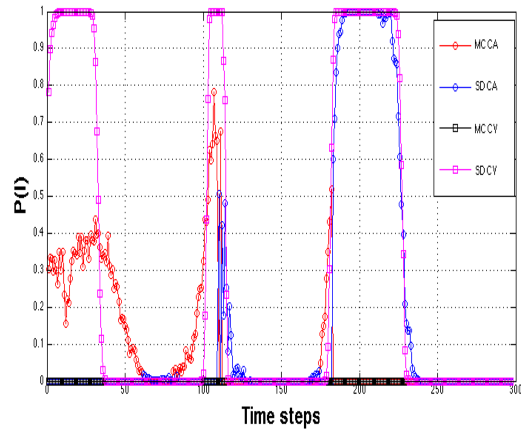


Fig. 2: Probability of infringements for 5th-step predictions using both methods

IV. Infringements Frequency Analysis

We used over 27000 collection of infringements occurred in the UK during the year 2008. They were scattered over 90 CAS zones around London heathrow airport. It contains their location $(Easting, Northing, Altitude)$, time of infringements, duration inside CAS zone and the CAS zone number. Figure 3 shows a geographic view of all aircraft infringed various CAS boundaries over south eastern UK. We calculated the frequency of the information provided below:

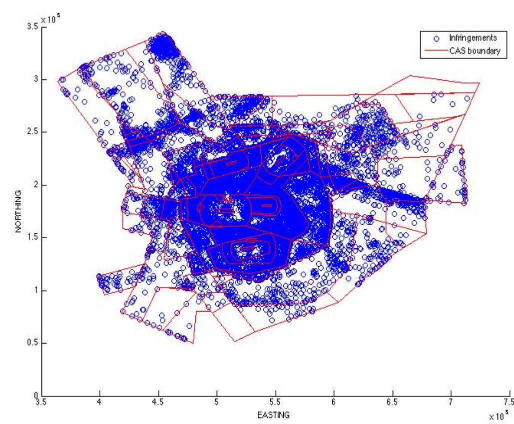
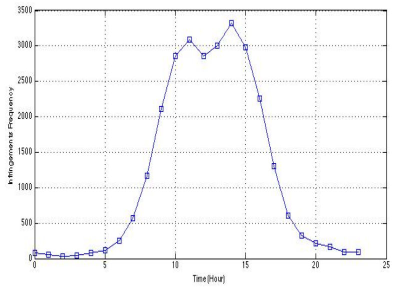


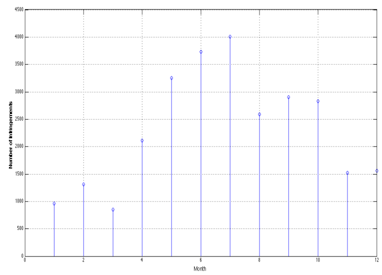
Fig. 3: All infringements occurred during 2008

A. Time, Month and CAS Zone

Since most pilots use VFR, the majority of infringements occurred during the day time from 8 am - 9



(a) Time of infringement frequency



(b) Monthly infringement frequency

Fig. 4: The number of infringements by the time and month

pm. For the months however, the most frequent ones were during the summer time (May-September). Figure 4 below shows the a) frequency of infringement by the hour b) the frequency by the month. We then looked at all of the infringements occurred with respect to CAS zones. Figure 4 shows the frequency of aircraft infringement given time, month and the CAS zone. It appears that CAS zones number from 1-20 had the most infringements.

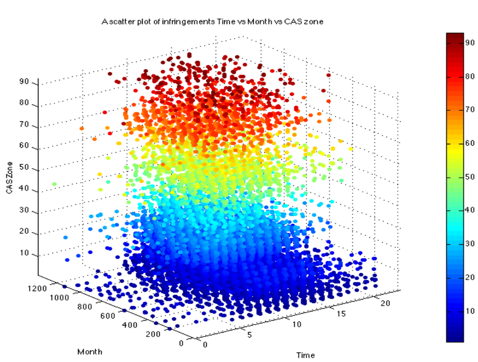


Fig. 5: Most Frequent CAS zone infringements with respect to time and month of the year

By looking at figure 5, it shows a concerning amount of infringements since they arise more frequently during the months and times where major airports such as Heathrow are the busiest. Figure 6 shows our model design where we will use this information as inputs to our classifier and outputs the decision whether to alert ATC of future infringements or not. In the next

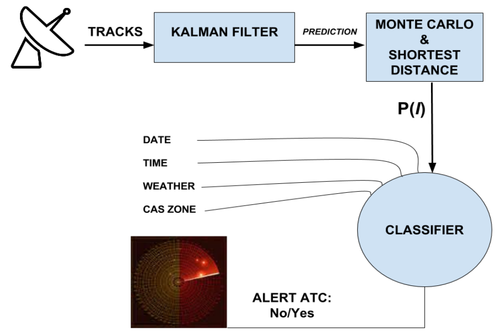


Fig. 6: Overall design of our model

section, will introduce the classifier used to enhance our warning system using the the frequency and data analysis provided in this section.

V. Classification Using Flight Information

Monitoring CAS zones is an ATC top priority, therefore, reducing the amount of false alerts and providing a warning is crucial. As result, given the statistics in figure 4; we will use them as our training set to build a classifier. Since we only have flight information that infringed CAS zones, we will use a one class Support Vector Machine (SVM) as our classifier.

One Class Support Vector Machine

A support vector machine is a type of a learning method in machine learning which uses different types of algorithms (depends on the application) for classification or regression. In classification it distinguishes a given data set from one class or another in higher dimension. In our case, we will use the one-class classification to identify one class from another. Here it is assumed that only data of one of the classes is available, i.e data that infringed CAS zone. This means that just this flight information can be used as training set and that no information about the other class that is "flight information with no infringements" is present. The distinction between classes here, is done by defining a boundary around the trained one class, such that it allows as much of the target object as possible "infringement occurred", while minimizing the chance of accepting outlier objects "no infringement". The separation boundary between target objects and the outliers is constructed by a set of support vectors in a higher dimension using a hyperplane. When providing OCSVM a new set of objects $X_{new} = \{x_1, x_2, ..., x_n\}$, it classifies them as similar or different to the training set and outputs y_i such that $y_i \in \{-1, 1\}$. Because the objects can not be separated in current space A they will be lifted to higher space B and separated by hyperplane defined as:

$$\mathbf{w}^T x + b = 0$$

where $\mathbf{w} \in A$ and $b \in B$. This hyperplane determines the boundary of classes; such that objects $y_i < 0$ are in one class; whereas $y_i > 0$ are in another class. Objects that lie one the boundary have zero values. The objective of the hyperplane construction is to create the maximum separation between two classes while mimizing over-fitting with noisy objects. To avoid over-fitting, the SVM introduces a slack variables ξ_i to allow some objects to fall around the boundary. Finding the trade off between the maximum boundary margin and objects lie within the margin; a constant variable $C > 0$ is introduced. Given these values, the objective function of the SVM classifier is defined as:

$$\min_{w,b,\xi_i} = \frac{||\mathbf{w}||^2}{2} + C\sum_{i=1}^{n} \xi_i$$

This minimization problem will be solved using Lagrange multipliers α_i, the decision function rule for this classifier is defined as :

$$f(x) = sgn(\sum_{i=1}^{n} \alpha_i y_y \mathbf{K}(\mathbf{x}, \mathbf{x}_i) + b)$$

The projection of the objects to higher space is done by non-linear kernel function $\mathbf{K}(\mathbf{x}, \mathbf{x}_i)$. The Kernel function we will use is the Gaussian radial basis "RBF".

In the next section, we will present our warning system algorithm using the one-SVM as a threshold with our probability of infringement.

VI. Warning System and Results

The performance of our model and probability methods can be found in [6]. Here we will show the analysis of results of our classification model after training and testing. Since we are given a set of flight information "objects" X from only one class, we will use this set to train our classifier. The number of objects to be trained is 27654 such that $X_{train} = [x_{date}, x_{time}, x_{cas}]$. Using the classifier output range $\{-1, 1\}$ as our alerting threshold, we look at the $P(I)$. In algorithm 1, we show that If $P(I) \geq 0.5$ for some prediction $\widehat{X}_i$ with its information $\widehat{X}_{info} = [x_{time}, x_{date}, x_{CAS}]$, the classifier ONSVM takes it in as input and outputs its similarity y_i with the trained data. If the similarity is high we will warn the ATC otherwise wait for the next time step.

By using the track in figure 1 above, for example and its probability of infringement results in figure 2; we

Data: WarnSysFunc($P_t(I), \widehat{X}_t^{info}$)
Result: Alert ATC or Not

```
if P_t(I) ≥ 0.5 then
    y_t ← ONSVM(X̂_t^info);
    if y_t ≥ 0 then
        Alert ATC;
    else
        WarnSysFunc(P_t + 1(I), X̂_{t+1}^info);
    end
end
```

Algorithm 1: Warning system function

tested the classifier using a randomly generated test set of size 1000. It contains times and dates with the CAS number being fixed to the one infringed in figure 6. It shows that 38 were classified as similar to the infringed ones. Figure 7 shows the classifier output of similar objects which is the score of the classifier.

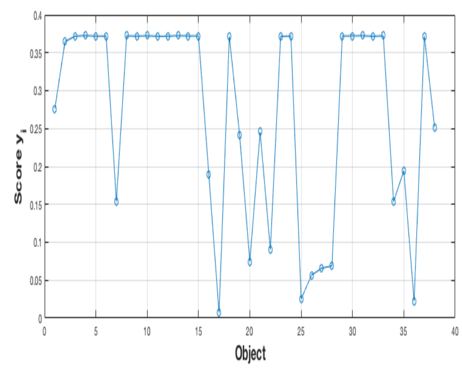
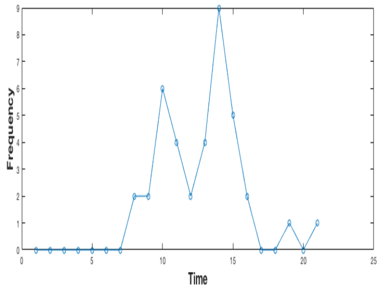


Fig. 7: Classifier output score of the similar objects in the test set to that in the training set

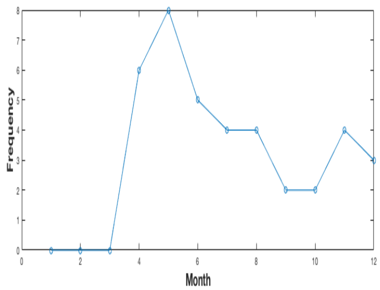
After analysing these specific objects (times and dates), we see in figure 8; were the majority of the infringements occurred in the CAS shown in figure 2. Therefore, if the aircraft in figure 6 was flying during these times or months, our alerting system function would trigger a warning to the ATC. Otherwise, it would wait for the next prediction location at time $t + 1$. Combining the classifier with the probability of infringement can help reduce false alerts to the ATC. Finally, we are looking to obtain more real VFR flight tracks that **did not** infringe CAS to help us test the reliability of the model combined. Since creating a random synthetic tracks and flight information (ex:time, date, CAS zones, weather) to measure the performance of our model as a whole will not represent pilots intentions, therefore; we will have to use real VFR flight tracks.

VII. Conclusion

In our previous paper, we developed an aircraft tracking tool with online learning called "probabilistic



(a) Time frequency



(b) Month frequency

Fig. 8: The time and month frequencies, where the classifier produced them as similar given a random set

controlled airspace tool" which predicts 5-steps ahead and provide the probability of their infringements to controlled airspace zones at each time step. After reviewing it in this paper, we introduced the factors behind the infringements, after a survey conducted by eurocontrol. We then presented the one-class support vector machine and trained it on the data provided which infringed CAS zones. We later, combined it with PCAIT to enhance our warning system by developing a function that uses the classifier's score as threshold to whether it warns ATC or wait given $P(I) > 0.5$. We tested this classifier on a track that infringed specific CAS zone but with unknown date and time. Therefore, we generated a random set of times and dates along with the known CAS number and applied it the classifier. It provided similarity to other infringed aircraft information such that if this specific aircraft was to fly during these times and dates, a warning will be issued to the ATC. Our future work will focus on solving the problem of multiple infringements in the same time. We will investigate the possibility of using computational geometry methods to reroute heavy traffic safely in the case of multiple infringements.

References

[1] L. C. Yang, J. K. Kuchar, L. C. Yang, and J. K. Kucharf, "Prototype conflict alerting system for free flight," *Journal of Guidance, Control, and Dynamics*, pp. 768–773, 1997.

[2] L. C. Yang and J. K. Kucharf, "Using intent information in probabilistic conflict analysis," in *in AIAA Guidance, Navigation, and Control Conf*, 1998.

[3] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *AIAA JOURNAL OF GUIDANCE CONTROL AND DYNAMICS*, vol. 20, pp. 588–596, 1997.

[4] C. L. Tallec, D. Taurino, C. Lancia, and J. Verstraeten, "Predicting the future location of a general aviation aircraft," tech. rep., NLR Air Transport Safety Institute and Onera and Deep Blue, Nederlands, 2014.

[5] "General aviation airspace infringement survey."

[6] Y. S. Almathami and A. Reda, "Probabilistic controlled airspace infringement tool," in *in Signal Processing and Information Technology (ISSPIT) 2015*, 2015.

[7] K. Mcdonald-Wallis, "An approach into the probabilistic prediction of the movement of uncontrolled aircraft to improve uk aviation safety," master's thesis, 2009.