10-3-2017

Probabilistic Controlled Airspace Infringement Tool and Conflict Resolution

Yousra Almathami
University of Connecticut - Storrs, yousra.almathami@uconn.edu

Probabilistic Controlled Airspace Infringement Tool and Conflict Resolution

Yousra S. Almathami, PhD

University of Connecticut, 2017

Abstract:

A current ground based safety net called Controlled Airspace infringement Tool (CAIT) is used by Air Traffic Controllers (ATCs). It warns them if any aircraft within uncontrolled airspace penetrates the Controlled airspace (CAS) without an advance clearance from the ATC. This penetration or 'Infringement' is considered as a major concern to ATCs where it may cause a possible conflict or mid- air collision. A conflict is an event which one aircraft loses its minimum separation to another. The downside of CAIT is that it only warns ATCs if the aircraft has already infringed CAS, this gives the ATC minimum time to react and avoid any conflict. In this research, we investigate a model which warns ATCs of possible future infringements accurately. We implement two Kalman filters (KF) as our tracking tool, one for each flight mode: constant velocity and constant acceleration each of which has its own state and observation errors. We will implement on-line learning too for these errors at each time step to predict the future uncertainties more accurately. We will then use two methods to estimate the probability of infringements: shortest distance, and complement this method with the Monte Carlo sampling. We will then 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. Our final step will focus on the scenario when multiple aircraft infringe CAS, in case the ATC did not react quickly enough. As of 2019, it is mandatory for all GA to be equipped with a transponder, which sends information such as flight ID, exact location and altitude. Therefore, using this assumption we are investigating an automated method, which alerts and direct multiple GA out of CAS without interfering with commercial traffic. Kinetic triangulation method will be used as an automated manoeuvring tactic, leaving the ATC focusing only on directing commercial flights.

Probabilistic Controlled Airspace Infringement Tool and Conflict Resolution

Yousra S. Almathami

B.S., Indiana University- Purdue University, 2008

M.Phil., University of Exeter, 2013

M.S. , University of Connecticut, 2016

A Dissertation Submitted in Partial Fulfilment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut 2017

APPROVAL PAGE

Doctor of Philosophy Dissertation

Probabilistic Controlled Airspace Infringement Tool and Conflict Resolution

Presented by

Yousra S. Almathami, B.S., MPhil., M.S.

Major Advisor _____

Reda Ammar

Associate Advisor _____

Swapna Gokhale

Associate Advisor _____

Sanguthevar Rajasekaran

University of Connecticut 2017

I would like to dedicate this thesis to my loving parents Jameelah and Saud and the rest of my family ...

Acknowledgement

And I would like to acknowledge and thank Prof. Reda Ammar for his guidance, encouragement and support through out my research. I am lucky to have him as my supervisor for he has been there through out my work, answering questions and taught me everything i needed to know to peruse my career in research and lecturing. I hope I can spread his wisdom and encouragement to other students in the future. I must express my gratitude to three special people my sister Hanan, my best friend Deemah and my brother Majid. Completing this research would have been more difficult were it not for the support and friendship provided by them, I am grateful to them for their help. Finally, I also would like to thank all the members of staff in the computer science and engineering department, who helped me in my supervisor's absence. In particular I would like to thank Ms. Rebecca Randazzo for all her help and kind concern she has provided me since the day I started my program in the department of computer science. Finally, I would like to thank the King Saud University and the Saudi Arabian Cultural mission not only for providing the funding which allowed me to peruse my PhD degree, but also for giving me the opportunity to attend conferences and meet so many experts in this field of science.

Thank you all and god bless you.

Table of contents

List of figures

Chapter 1

Introduction

1.1 Overview

It is the responsibility of air traffic controller (ATC) to monitor and to manage the flow of the aircraft in the airspace. The airspace is divided into seven classes; five of them correspond to controlled airspace (CAS) and two to the uncontrolled airspace (UCAS). These five classes are being controlled by the ATC and therefore, for any aircraft flying within or about to fly into CAS, the communication between that aircraft and the ATC is essential to avoid any conflict. A conflict is an event in which one aircraft loses its minimum separation by another aircraft. The size of the protected zone defined around an aircraft varies but the standard amount of separation each aircraft should have is 5 nautical miles horizontally and no less than 1000ft vertically. ATCs monitor aircraft locations which are collected by radars scattered around the airports every 4 seconds (1 revolution per 4 seconds). The ATC also monitors aircraft flying outside the controlled airspace using another tool, called Controlled Airspace Infringement detection Tool (CAIT) to detect CAS infringements. An infringement occurs when an unauthorised aircraft

penetrates CAS without the ATC's advance clearance. Figure 2.3 shows an example of ATCs' monitor when an aircraft has infringed CAS which is highlighted in magenta.

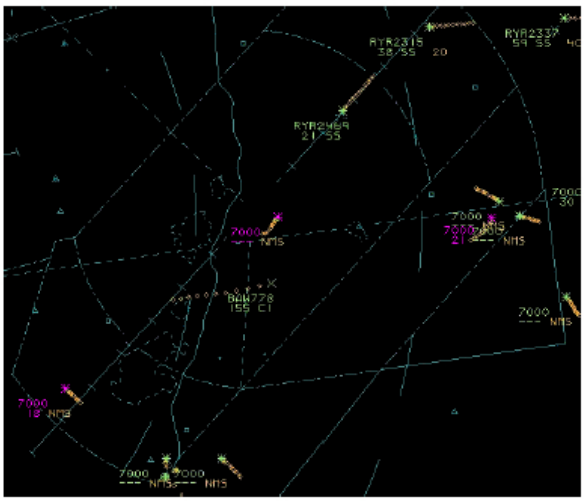


Fig. 1.1 Current controlled airspace infringement too showing three infringements occurred to the controlled airspace highlighted in magenta

1.2 Research Problem

Controlled airspace infringements are major safety concern to ATCs and aircraft pilots which can cause two main issues: one is the possibility of a conflict, where an aircraft loses its minimum separation. Also, these infringements 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 either light weight aircraft that rely on visual flight rules or drones. The infringing aircraft ID and its flight path (way-point) are unknown to ATCs unless the pilot initiates the contact with the tower. A survey was conducted by Eurocontrol [1] to find out why infringements occur more frequently. The three main reasons:

- Pilot is unfamiliar with the Airspace.
- Avoid a bad weather such as cluttered clouds
- Pilot is unsure of airspace or lost

Figure 2 shows the percentage of times these infringements occurred with additional reasons conducted by the survey.

As a result, CAIT was developed to detect any infringements that occured. 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. It would be better for the ATC to have an early warning to resolve a future conflict in advance and maintain the flow of aircraft in the CAS in the same time.

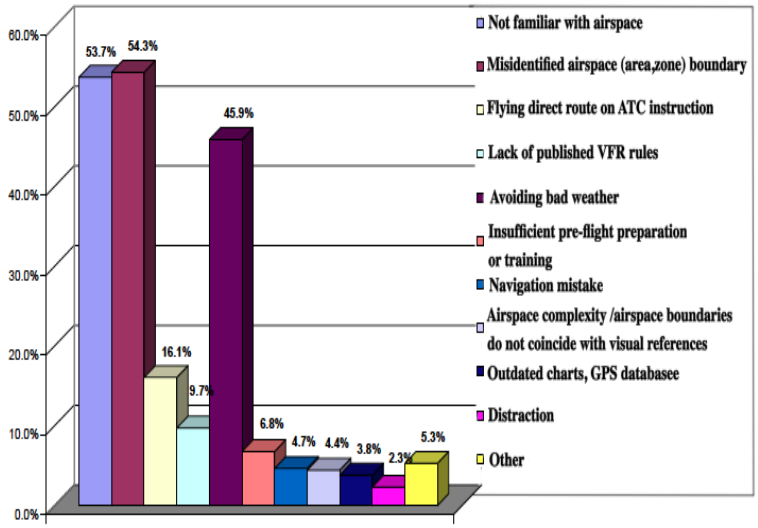


Fig. 1.2 The percentage of infringements caused by these reasons taken from [1]

1.3 Objectives

The focus of this research is to propose a ground based safety net that assists ATC in avoiding mid-air collisions in timely manner. This research is broken onto two categories:

1. Investigate a possible model which warns ATC in advance of any possible future infringements with the assumption of unauthorized pilots do not communicate with ATC.

2. Provide an automated resolution for future conflicts inside CAS using computational geometry method with the assumption pilots communicate with ATC.

In the first part of the research it will focus on examining the following:

1. Implementing a switching Kalman filters that predict future aircraft locations (20 seconds ahead)

2. On-line learning of the Kalman filter's errors to reduce predictions uncertainties

3. Finding the probability of infringement using two different methods: Monte Carlo sampling and shortest distance to CAS.

4. Develop a classifier to enhance our probability of infringement to reduce false alerts.

5. Automated conflict resolution for multiple aircraft infringement using kinetic polygon triangulation.

Figure 1.3 shows the proposed structure of the research model.

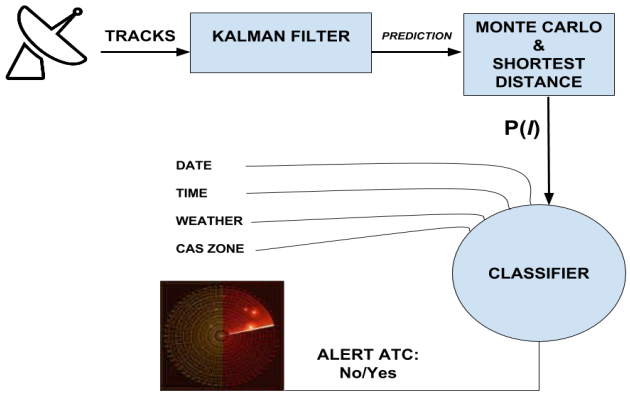


Fig. 1.3 The overall structure of the warning system (aircraft is unidentifed

First, the PSR radars collect and send unidentified aircraft estimated locations and altitudes to the Kalman filter tool and predicts future locations of the aircraft (4 seconds and more in the future). Second, the model will find the probability of infringements *P(I)* for the predicted locations of aircraft using a hybrid method of MC sampling and shortest distance. The probability of infringement will be tuned based on factors used as inputs to the Support vector machine classifier to decrease or increase true positives (infringements occur). A warning is issued to the ATC if the tuned *P(I)* is higher than a specific threshold. The second part of the research proposes an automated way to resolve future conflicts in case multiple aircraft get

inside CAS. Since the Federal Aviation Administration FAA is mandating all aircraft must be equipped with advance transponder (ADS-B) by 2020 that communicates with radar by providing an accurate aircraft location, altitude and ID every 1 second. The research will use this assumption to implement the second part of the research.

1.4 Research Structure

In the next chapters of this thesis, will present the following: Chapter 2 provides an in depth background information related to the airspace, radars, current CAIT and the tools used to help build the probabilistic infringement detection tool. Chapter 3, is the literature review on conflict detections. Chapter 4 shows aircraft location prediction models with on-line learning method. Chapter 5, is finding the probability of infringement. In chapter 6, is the classifier model. Chapter 7 shows the automated conflict resolution and alerting system. Chapter 8, the overall model simulation and analysis. Chapter 9 presents thesis contribution and conclusion.

Chapter 2

Background

The research objective is to implement a probabilistic infringement detection tool which predicts future infringements accurately; therefore, it is essential to give background information related to this research such as what is airspace and who controls it, the types of radars and the current infringement detection tool and its limitation. The tracking tool we will use to predict and estimate the aircraft locations is the Kaman filter which we will review it in this chapter.

2.1 Airspace

Historically as the number of airplanes increased it became essential to manage their flow and maintain their separation from each other in the air, resulting in the establishment of air traffic services. These services are provided by air traffic controllers (ATCs) for aircraft flying within the controlled airspace (CAS). These aircraft are supervised by the ATC and should follow their instructions as long as they are inside the CAS. Any aircraft flying in an uncontrolled airspace (UCAS) is not subject to ATC communication and the pilot should take full responsibility

for his/her safety. Within controlled airspace, the ATCs monitor the air traffic flow and use separation rules to avoid any conflict or mid-air collision. A conflict is an event for which an aircraft loses its minimum separation. Typical minimum separations are 5 nautical miles horizontally and 1000 feet vertically, although these may be modified according to the particular expected behaviour of the aircraft. For example, near a terminal where aircraft are about to land, the separation may be 1 nautical mile (nm) horizontally and 500ft vertically. Airspace is divided into 7 classes; 5 corresponding to CAS and 2 to UCAS. The UCAS classes are available for the general public to use such as light aircraft, helicopters and hot air balloons and drones. This type of airspace extend from the ground up to 3000 ft or to the base of CAS if there is one defined above it.

2.2 Radar

Radar is an object detecting system mainly used by ATCs to detect aircraft in air. There are two types of radar that are used by the ATCs, the primary surveillance radar (PSR) which only detects the location of the aircraft with imprecise altitude; and the secondary surveillance radar (SSR) which provides more information to the ATC such as altitude and identity. The SSR however, only operates when the aircraft is equipped with an advanced transponder, which makes it a dependant system that needs to communicate with that aircraft's transponder in order to provide the extra information. Aircraft with no advanced transponders (light aircraft) can therefore only be detected by the PSR. As a result, ATC uses the PSR in conjunction with the SSR to detect and identify the aircraft. Figure 2.1 shows the difference of radars:

Since most aircraft that infringe CAS are not equipped with transponders, they can only be detected by the PSR. This tells the ATC very little about that aircraft altitude, flight plan and

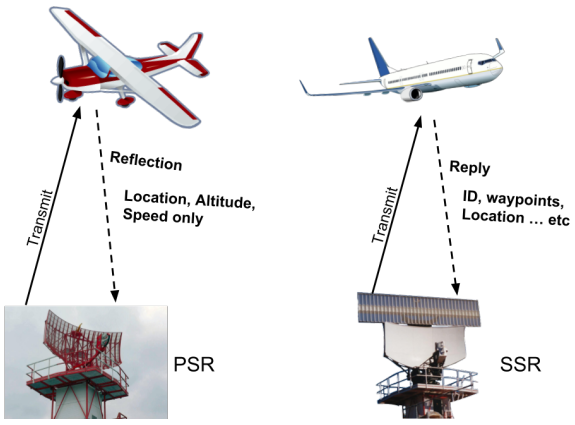


Fig. 2.1 Primary survalliance radar versus secondary survaliance radar

their identity; therefore, ATCs require a safety net that warns them whenever an infringement occurs. This system is called the Controlled Airspace Infringement Tool – CAIT – and is discussed next.

2.3 Current Infringement Detection

Controlled airspace infringements are a major safety concern to ATCs and aircraft pilots which can cause two main issues: one is the possibility of a conflict, where an aircraft loses its minimum separation. Another is that, these infringements 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. Therefore, a ground based safety net called the Controlled Airspace Infringement Tool (CAIT) used by NATS was created in order to warn ATCs of an infringement to the CAS by an unauthorised aircraft. This is achieved by collecting aircraft locations within UCAS by the PSR radar every four seconds, sending these locations to the ATC monitors and then highlighting the aircraft in different colour when the aircraft is

inside the CAS, thus raise a warning to the ATC showing in figure . The downside of CAIT system is that it only warns ATCs if and only if, the aircraft has already infringed the CAS and therefore, could cause a possible conflict with another authorised aircraft flying inside it. To solve this problem is our research objective, which is to investigate a possible model that warns ATC well in advance of any possible infringement in the near future. This will help the ATC to respond to it and resolve it quickly before it could cause further risk. Our last part of the research focuses on the conflict resolution when multiple aircraft infringe CAS simultaneously by finding automated exit routes for them in timely manner.

2.4 Kalman filter and smoother for Aircraft Tracking

2.4.1 Background

In 1960, Rudolf Kalman introduced a a recursive state-estimator of linear dynamical system [2]. It uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone as shown in figure 2.2.

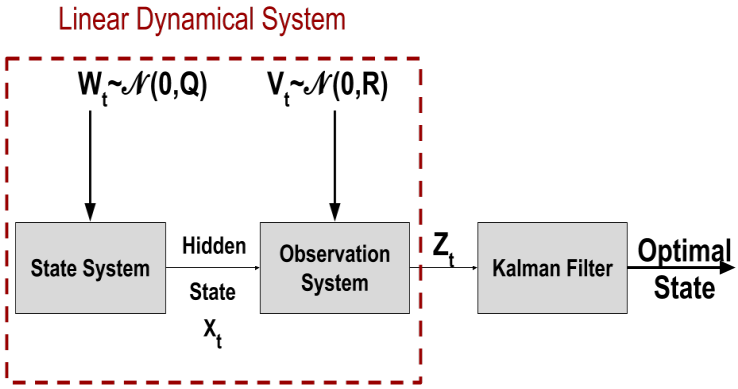


Fig. 2.2 Kalman Filter estimation of linear dynamical system

The algorithm works in a two-step process, *"prediction"* and *"correction"*. In the prediction step, the Kalman filter produces an estimate $\widehat{x}_{t+1}$ of the current state variable $\overline{x}_t$, along with its uncertainty. Once the outcome of the next measurement z_{t+1} (might be corrupted with some noise) is observed, the estimate $\widehat{x}_{t+1}$ is updated using a weighted average, with more weight being given to estimates with higher certainty. The result of the weighted average is a new state estimate $\widehat{x}_{t+1}$ that lies between the predicted and measured state, and has a better estimated uncertainty. Since the algorithm is recursive, it can run in real time; therefore, makes it very useful to predict and track aircraft. By only using the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required. At time t, the KF predicts the next state hidden $\widehat{x}_{t+1}$ and its zone of uncertainty (green ellipse) given $\overline{x}_t$. At time $t+1$, we get a new observation z_{t+1} and corrects the location of the predicted state to get $\overline{x}_{t+1}$. Figure 2.3 shows a sample track and the KF process.

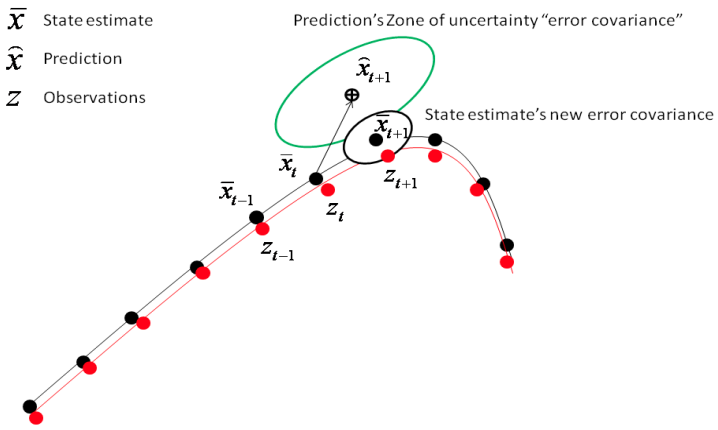


Fig. 2.3 Kalman Filter prediction and correction process on a sample aircraft track.

2.4.2 The Algorithm

First, to use the standard Kalman filter and smoother the observations and the hidden state should be assumed to be linear and normally distributed. Figure 2.4 illustrates the idea of the forward backward inference process.

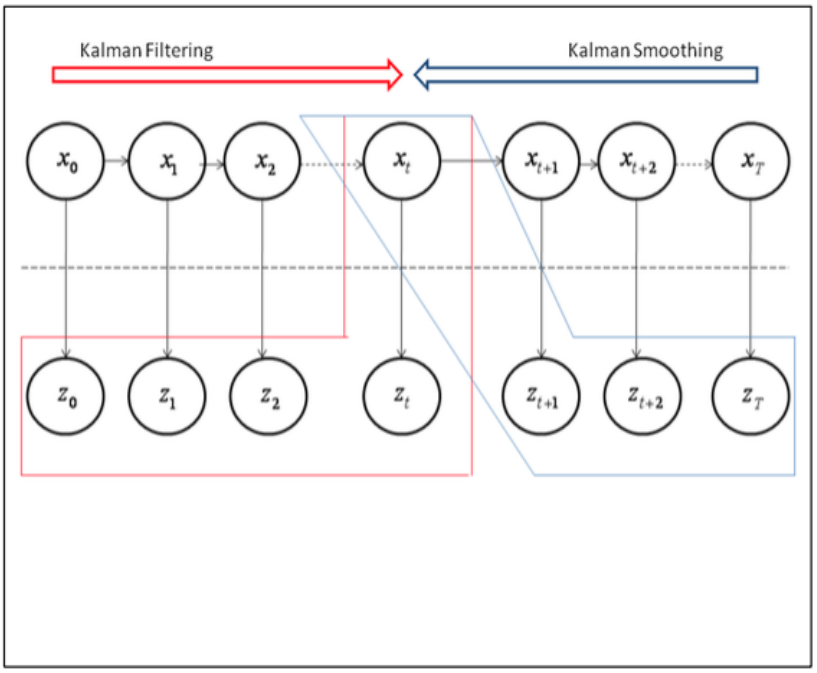


Fig. 2.4 Kalman Filter-smoother given the observations $Z = z_1, z_2, ..., z_T$

The forward pass estimates $P(x|z_{1:t})$ and the backward pass estimates $P(x|z_{1:T})$.

Algorithm 1 Kalman Filter

$\widehat{x}_1 = \overline{x}_1 = z_{t=1}$, $\overline{P}_1 = \widehat{P} = Q_{t=1}$

procedure FORWARD PASS(z_t) ▷ observation z at time t

  for $t \leq T$ **do**

    $\widehat{x}_{t+1} = A\overline{x}_t + w_{t+1}$ ▷ Prediction

    $\widehat{P}_{t+1} = A\overline{P}A^{'} + Q$

    $r = z_{t+1} - H\widehat{x}_{t+1}$

    $S = H\widehat{P}_{t+1} + R$

    $K = \widehat{P}_{t+1}H/S$

    $\overline{x}_{t+1} = \widehat{x}_{t+1} + Kr$ ▷ Correction

    $\overline{P}_{t+1} = (I - KH)\widehat{P}_{t+1}$

  end for

end procedure

where $w_{t+1} \backsim \mathcal{N}(0, Q)$; A and H are the state and observation transition matrices, $\widehat{P}$ and $\overline{P}$ are the predicted and estimated error covariances, Q and R are the system state and observation error covariances; r, S and K are residual, residual covariance and optimal Kalman Gain respectively. Finally, I is the identity matrix. The Kalman smoother (backward pass) follows the kalman filter where it estimates the smoothed state $\tilde{x}_t$ given all the observation from t=T to t=0 [3].

Algorithm 2 Kalman Smoother

1: $t = T-1, \tilde{x}_T = \bar{x}_T$, $\tilde{P}_T = \overline{P}_T$

2: $\widetilde{PP}_T = (I - KH)A\overline{P}_{T-1}$

3: **procedure** BACKWARD PASS($z_{t=T}, \widetilde{PP}_T, \tilde{x}_T, \bar{x}_T, \overline{P}_T$)

4: **for** $t \geq 1$ **do**

5: $J_t = (\overline{P}_t A') / \widehat{P}_{t+1}$

6: $\tilde{x}_t = \tilde{x}_t + J_t(\tilde{x}_{t+1} - A\tilde{x}_t)$

7: $\tilde{P}_t = \overline{P}_t + J_t(\tilde{P}_{t+1} - \overline{P}_{t+1})J_t'$

8: **end for**

9: **end procedure**

In the next section, a literature review on conflict detection and resolution between commercial aircraft is introduced.

Chapter 3

Literature Review on Conflict Detection

In controlled airspace (CAS), each aircraft has a zone of separation that no other aircraft should penetrate. Should this zone be penetrated, the event is termed a “conflict”. The size of the protected zone defined for aircraft varies but the standard amount of separation each aircraft should have is 5 nautical miles horizontally and no less than 1000ft vertically. These conflicts are a major concern for all air traffic controllers for which they have to constantly monitor the airspace and avoid any possible conflict, see figure 3.1 below: Current conflict

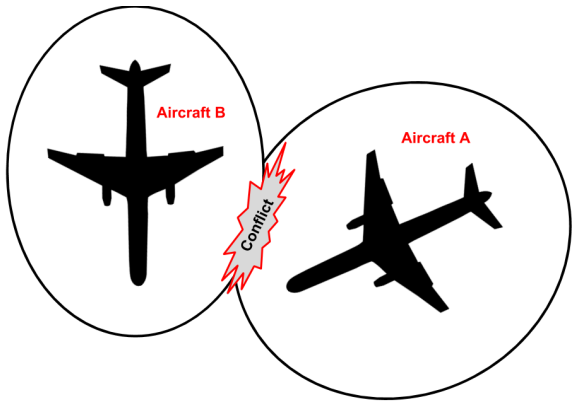


Fig. 3.1 Losing minimum seperation between two aircraft causes a conflict

detection systems cannot warn ATC of a conflict more than 2 minutes away which gives ATCs a small amount of time to resolve it. This issue of conflict detection has been an attractive field of study by researchers, 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 [4] 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 [5] 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 [6] 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 [7] 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.

Over all, the authors were able to present better results in terms of early conflict detection and resolution. They however, assume the availability of intent information and the amount of uncertainties are known. The relationship between conflict detection and infringement detection systems is that they are both a ground based safety nets which warns ATCs for a possible conflict whether it is between two or several aircraft (conflict detection system) or between an unauthorised aircraft and a controlled airspace boundary (infringement detection system). Figure 3.2 breaks the related work in this field, where our focus is highlighted in the red rectangle.

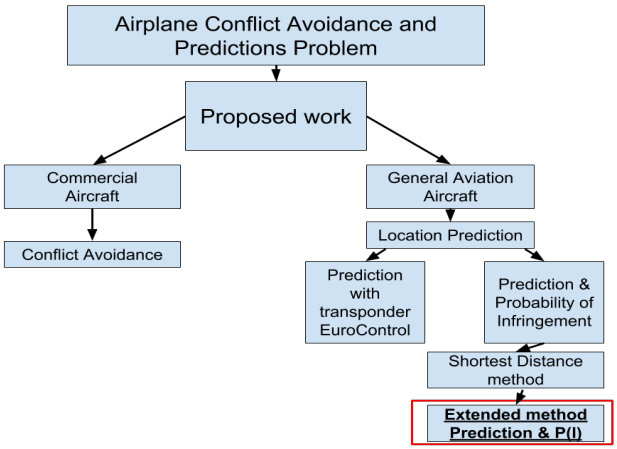


Fig. 3.2 Researches involve aircraft conflict detection

Because the current infringement detection system used by ATCs has a limitation which is the lack of early warning time or predicting future infringements more accurately, our main focus will be to investigating this system to possibly predicting future infringements more accurately.

Chapter 4

Aircraft Location Prediction with On-line Learning

In the previous chapter, we reviewed the basic Kalman filter-smoother tool where this research will use to predict and estimate aircraft location. In this chapter, we will modify the Kalman filter such that it predict 5 steps in the future while learning the model error covariances online. At each time step (4 seconds) and using a history of only 15 steps backwards, the model will learn its error covariances instead of learning using the whole track history steps which its computationally expensive for small window of 20 seconds ahead. This chapter introduces the switching Kalman filters and the algorithm for the on-line learning showing in our model figure 4.1 bellow:

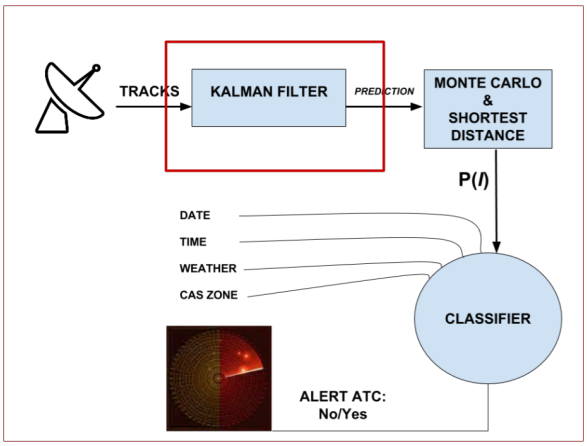


Fig. 4.1 Overall design of the research model

4.1 Switching Kalman Filters

We analysed several real flight paths for aircraft flying in UCAS. They tend to fly in two different modes: with a constant velocity (straight) and constant acceleration (turning) mode. Having one static Kalman filter to predict the next location would be infeasible. This would create large amount of uncertainty making our future prediction inaccurate. Therefore, we implemented two Kalman filters-smoothers for each mode where both are running in parallel. The "*switching*" aspect is based on choosing which of the KF models produces accurate predictions in the future. Both KFs receive the same observation z_t from the observation system, which is in our case the primary radar. The equation of the observation system is defined as:

$$z_t = H\bar{x}_t + v_t, \text{ where } v_t \backsim \mathcal{N}(0,R) \text{ and}$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

4.1.1 Constant Velocity KF

The purpose of the constant velocity KF is to track aircraft travelling in a straight line; therefore, no acceleration is applied. Each Kalman filter has the following parameters $\lambda = (A, H, Q, R)$ where both A and H will be fixed the entire prediction process. The linear equation for the CV-KF state system is defined as the following:

$$A_{CV}\widehat{x} = \begin{pmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} x_{loc} \\ x_{Vel} \\ x_{Acc} \\ y_{Loc} \\ y_{Vel} \\ y_{Acc} \\ z_{Loc} \\ z_{Vel} \\ z_{Acc} \end{pmatrix}$$

Where $T = 4\ sec$, the time interval to observe the next location from the radar $(1RPM = 4\ sec)$.

4.1.2 Constant Acceleration KF

Here when the aircraft makes a smooth turn, it applies a constant acceleration. The linear equation for the CA-KF state system is defined as the following:

$$A_{CA}\widehat{x} = \begin{pmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & T & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x_{loc} \\ x_{Vel} \\ x_{Acc} \\ y_{Loc} \\ y_{Vel} \\ y_{Acc} \\ z_{Loc} \\ z_{Vel} \\ z_{Acc} \end{pmatrix}$$

4.2 On-Line Learning of SKF Errors

Fixing the model with an arbitrary error for the entire flight is not efficient since all aircraft tend to change their heading at any time and travel in different patterns. Therefore, we need to learn each model's changing state and observation error covariances for any flight path. This allows the covariance to change during the flight. And by finding them, it will minimize the errors in the state estimation,prediction and increases the likelihood of the observation. We used the expectation maximization algorithm (EM) method to maximize the likelihood of our observations given the new parameters of the model [8]. It alternates between the E-step which computes the log-likelihood of the observations given the current parameters and the M-step

which maximizes the log-likelihood computed in the E-step by re-estimating the parameters $\theta = (R, Q)$. Here is the steps of the whole process:

E-Step

- Run the Kalman filter (forward pass)
- Run the Kalman Smoother (backward pass)

M-step

- Re-estimate both KFs error parameters $\theta^* = (R, Q)$

In this section, we focus on the M-step which re-estimates the errors of the skf model. The derivation for parameters re-estimation in [9]. However, the final equations are as follows:

$$
\begin{aligned}
R_m^* &= \sum_t^T \frac{(z_t - H\tilde{\mathrm{x}}_t)(z_t - H\tilde{\mathrm{x}}_t)^{'} + (H\tilde{\mathrm{P}}_t H^{'})}{T} \\
D &= \sum_2^T (\tilde{\mathrm{x}}_{t-1}\tilde{\mathrm{x}}_{t-1}^{'} + \tilde{\mathrm{P}}_{t-1}) \\
E &= \sum_2^T (\tilde{\mathrm{x}}_t\tilde{\mathrm{x}}_{t-1}^{'} + \tilde{\mathrm{P}}\mathrm{P}_t) \\
F &= \sum_2^T (\tilde{\mathrm{x}}_t\tilde{\mathrm{x}}_t^{'} + \tilde{\mathrm{P}}_t) \\
Q_m^* &= \frac{F - EA_m^{'} - E^{'}A_m - A_m DA_m^{'}}{T-1}
\end{aligned}
$$

where $\tilde{\mathrm{x}}$ and $\tilde{\mathrm{P}}$ are the smoothed state and error covariances; (R_m^*, Q_m^*) are the new observation and state error covariances for the m-KF; $\tilde{\mathrm{P}}\mathrm{P}$ is the smoothed state error covariance between times $t-1$ and $t-2$. Finally, T is the total number of observation sequence. We have modified

the equations by adding weights defined as the loglikihood of the observations. These weights can reduce or increase error covariance faster than the original equations. In addition to the weights, we defined τ to be the last 15 steps of way-points instead of the whole track. Algorithm 3 shows the algorithm after modification.

Algorithm 3 SKF online learning

$\tau = 15, D = E = F = 0$
procedure LEARNING($W_{T-\tau}, z_{T-\tau:T}, \widetilde{PP}_{T-\tau:T}, \tilde{x}_{T-\tau:T}, \overline{x}_{T-\tau:T}, \overline{P}_{T-\tau:T}$)
    if $T \geq 17$ **then** ▷ Check to see if there is at least a history of 17
        $t = T - \tau$
        $k = \tau$
    else
        $t = 2$
        $k = T$
    end if
    $R_m^* = \sum_t^T W_m \frac{(z_t - H\tilde{\mathrm{x}}_t)(z_t - H\tilde{\mathrm{x}}_t)' + (H\tilde{\mathrm{P}}_t H')}{T}$ ▷ re-estimating R for KF m=CA,CV
    $D = \sum_t^k W_m (\tilde{\mathrm{x}}_{t-1}\tilde{\mathrm{x}}_{t-1}' + \tilde{\mathrm{P}}_{t-1})$
    $E = \sum_t^k W_m (\tilde{\mathrm{x}}_t\tilde{\mathrm{x}}_{t-1}' + \tilde{\mathrm{P}}\mathrm{P}_t)$
    $F = \sum_t^k W_m (\tilde{\mathrm{x}}_t\tilde{\mathrm{x}}_t' + \tilde{\mathrm{P}}_t)$
    $Q_m^* = \frac{F - EA_m' - E'A_m - A_m DA_m'}{\sum_t^k W}$ ▷ re-estimating Q for KF m=(CA,CV)
end procedure

4.2.1 SKF Analysis

Synthetic Tracks

We generated synthetic tracks that contain white noise to represent the actual flight paths. Each track has a series of 400 observed way point (generated from the observation system of linear dynamical system); true tracks (generated from the hidden state system of the linear dynamical system). The true track is the true location of the aircraft, and the observations track is what

the radar receives. Figure 4.2 shows an example of the synthetic generated path. We made the flight travels in a constant velocity for 200 way-points then starts to accelerate the next 200 way-points. Using both models CV and CA, we computed their 1 and 5 steps predictions at

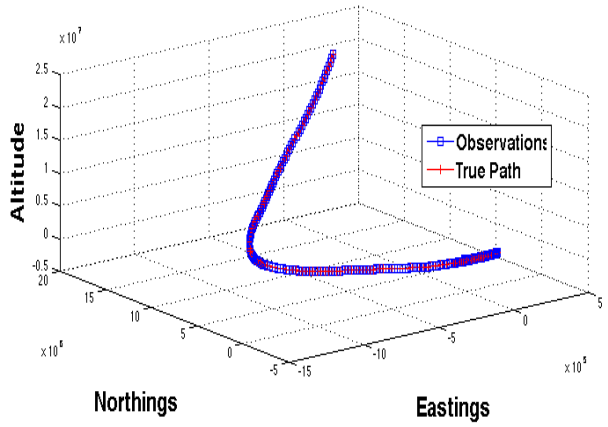


Fig. 4.2 Synthetic flight path with 400 way points. 13 minutes flight time

each step (4 and 20 seconds respectively). We estimated the root mean squared error for each model in separate intervals to see how they perform. We found CA model dominate the CV in terms of predictions during the second interval. Also, both perform similar results during the first interval. We then calculated the log-likelihood of the predictions and the result was similar

4 sec prediction	1-200	200-400
CV RMSE	302 m	3103 m
CA RMSE	306 m	558 m

Table 4.1 Mean Root mean squared error for each model between the way-points 1-200 and 200-400

to table 4.1, figure 4.3 ; shows the log-likelihood of the CV model has dropped after 200 way points,

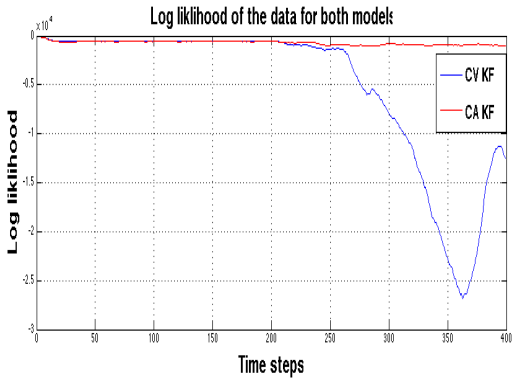


Fig. 4.3 Log-likelihood of the predictions using both models

whereas, CA model has dominated the CV for most of the flight path. When predicting 20 seconds ahead (5 steps), the error covariances "uncertainties" tend to increase which makes it very challenging to predict that far. However, in figure 4.4 shows promising results where the CA picks up the acceleration and makes a better prediction than CV after 20 seconds.

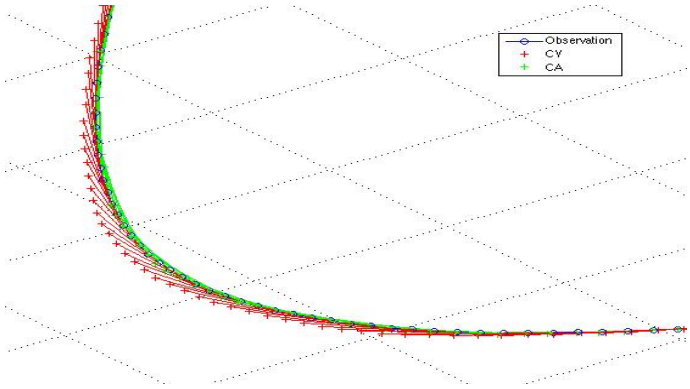
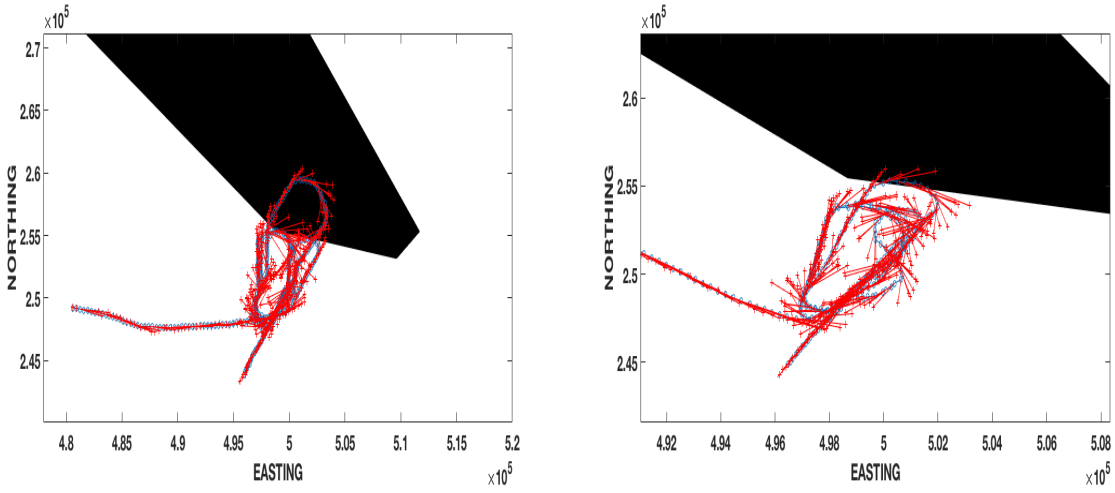


Fig. 4.4 A 5 step predictions from the observation at each time. Green lines are the projections of the predictions using CA model; the red are the CV model

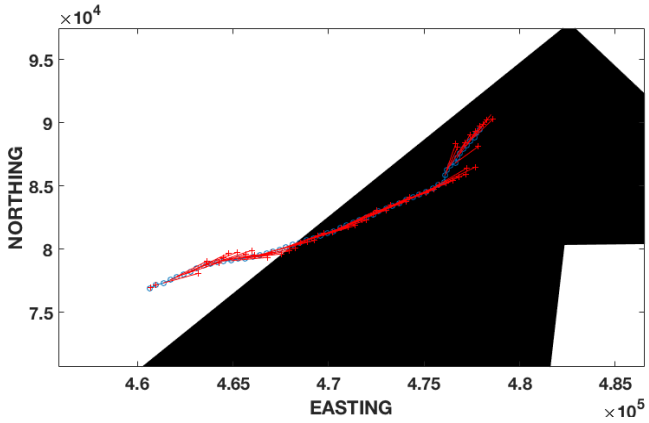
Real Tracks

After testing on synthetic tracks, the SKF is used on real tracks. These tracks contain their (X,Y,Z) locations in eastings, northings and altitudes in feet. Figure 4.5 shows an example of three tracks and their 5 steps prediction locations infringing two different CAS zones in 2D. Figure 4.6 shows a zoomed in version of both (CV,CA) and SKF 5 steps predictions and their



(a) Example 1: track infringed CAS zones multiple times

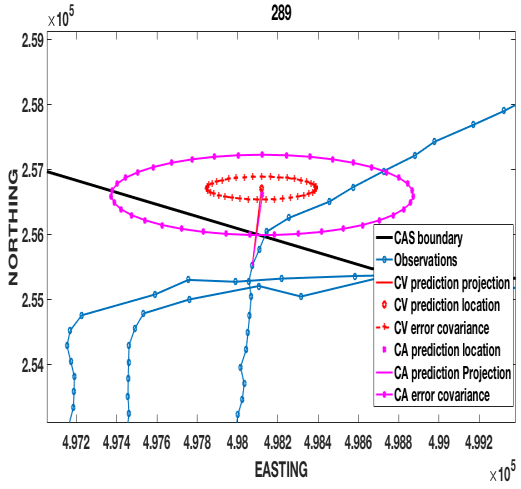
(b) Example 2: track infringed CAS zone temporary and turned away



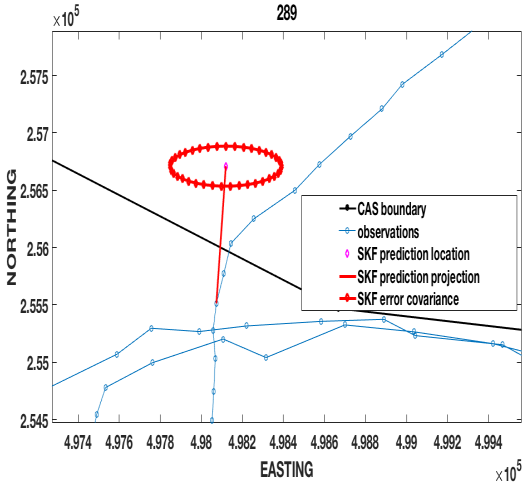
(c) Example 3: track infringed CAS zone in a straight line

Fig. 4.5 Real tracks examples with 5 steps prediction projections

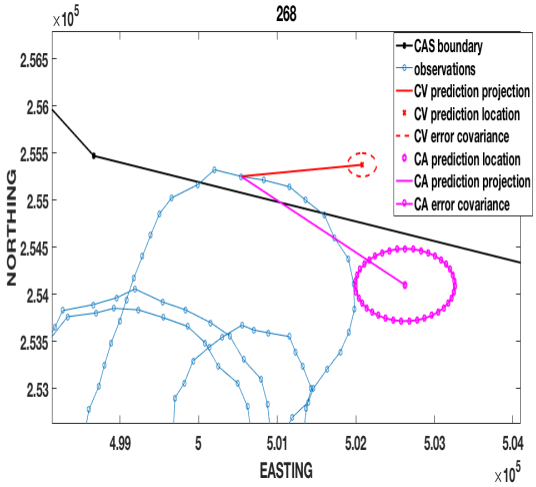
zone of uncertainties "Prediction error covariances".



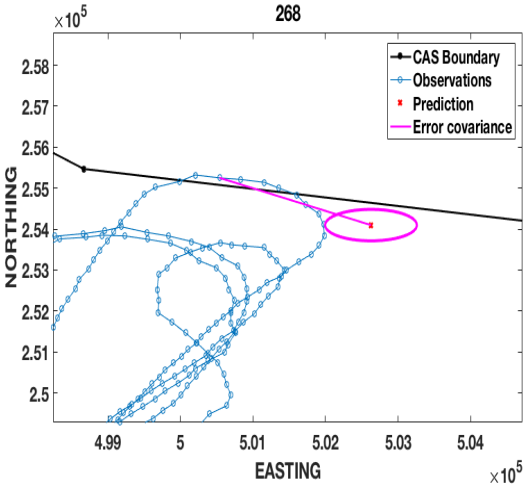
(a) zoomed in version of track 4.5a for both CV,CA



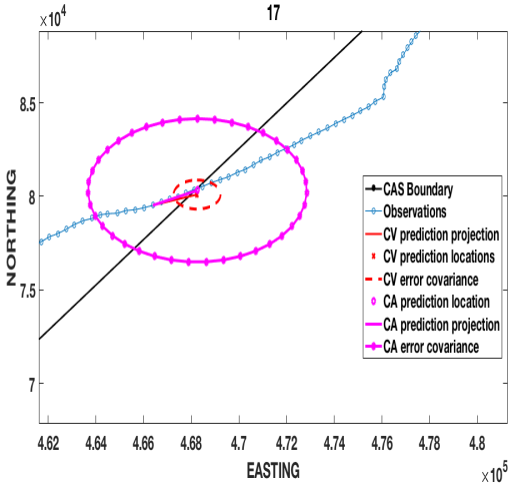
(b) zoomed in version of track 4.5a using SKF



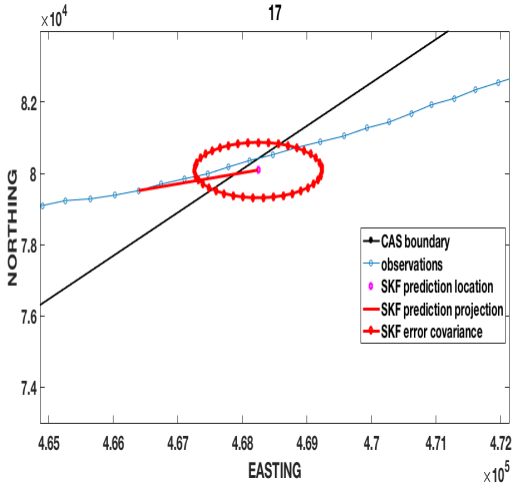
(c) zoomed in version of track 4.5b for both CV,CA



(d) zoomed in version of track 4.5b using SKF



(e) zoomed in version of track 4.5c for both CV,CA



(f) zoomed in version of track 4.5c using SKF

Fig. 4.6 zoomed in images of the 5 steps prediction projections and their uncertainty zones (ellipses)

We estimated the 5 steps predictions RMSE for the SKF model on 34 different tracks. The mean RMSE for all of them was 543 m and 142 standard deviation. Figures 4.7 and 4.8 shows the 1 and 5 step predictions RMSE for 34 real tracks.

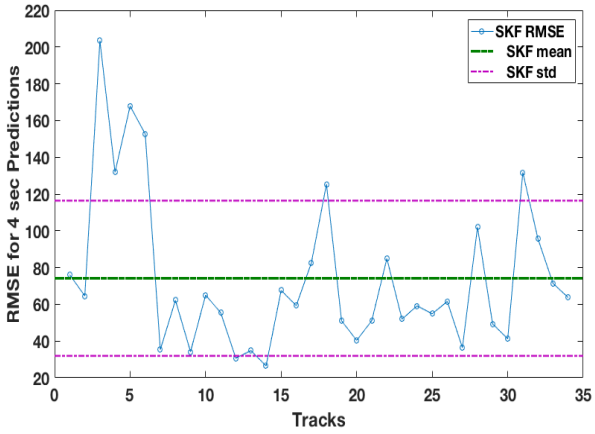


Fig. 4.7 A 1 step predictions RMSE for the switching Kalman filter

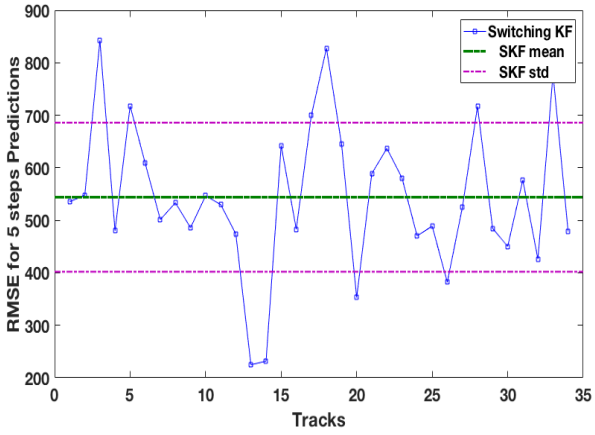


Fig. 4.8 A 5 step predictions RMSE for the switching Kalman filter

The average cruise speed of GA aircraft ranges from 53-62 m/s. Since radars collect aircraft location every 4 seconds, the aircraft can travel within a range of 212-248m. The mean RMSE

for the model in figure 4.8 is for 20 seconds range of error. It shows promising results, since the RMSE for every 4 second (1 step) is 74m range of prediction location shown in figure 4.7.

4.3 Summary

Our objective is to build a model that can predict future infringements accurately. We combined two models constant velocity and constant acceleration to track the aircraft while learning their error covariances. Both models switch to create an optimal prediction track based on the probability of the observation given the current prediction location and observation. As the prediction increases the errors increase as well, as a result, we implemented a new modified version of the EM algorithm used in [9] to either decrease or increase the prediction error covariances quicker. The reason to do so was because the SKF used the last 15 steps of history instead of the whole track. Given the 5 steps predictions as means with their learned error covariances, chapter 5 shows how to find their probability of infringements by taking advantage of those two variables (mean and covariance) as inputs.

Chapter 5

Probability of Infringement

This chapter shows finding the probability of infringement methods for the 5 step prediction locations received from the switching kalman filter showing in figure 5.1. To find $P(infringement)$

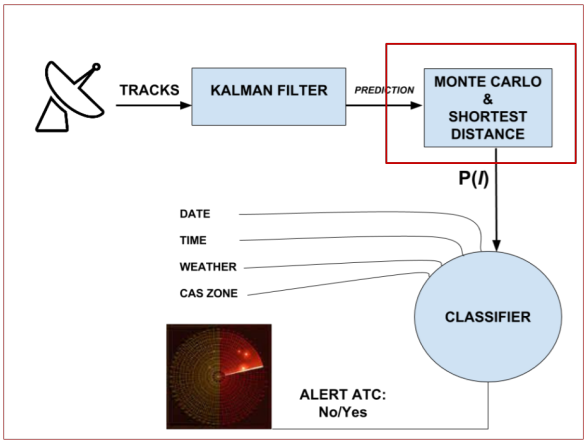


Fig. 5.1 Overall design of the research model

we used two probabilistic methods: one the shortest distance to the nearest CAS line proposed in [10]; and the Monte Carlo sampling.

5.1 Shortest Distance

This method is used to find the probability of an infringement at a given instant. It uses the aircraft prediction $\widehat{x}_t$, its error covariance $\widehat{P}_t$ (zone of uncertainty around the prediction) and calculates the shortest distance d from this prediction to the nearest CAS boundary C to estimate the probability of infringement. See figure 4: Here given $d, \widehat{x}_t$ and $\widehat{P}_t$, it uses the error function

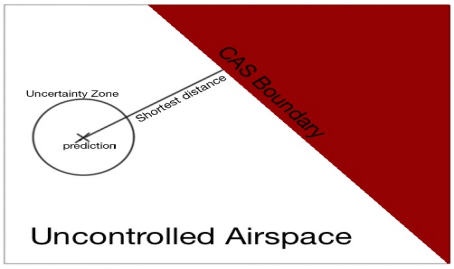


Fig. 5.2 A prediction near CAS showing the shortest distance d to its boundary

to find $P(I)$ as follows:

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

where d is the shortest distance from the prediction after standardising the prediction error covariance (ex: Sphering) $\widehat{P}_t$.This method will not predict $P(I)$ properly if the prediction is approaching a CAS corner (vertex). Figure 5.3 shows thus scenario, where it will provide higher probability of infringement since it included area outside CAS zone. Therefore, we introduced MC sampling method in case the Aircraft is approaching one.

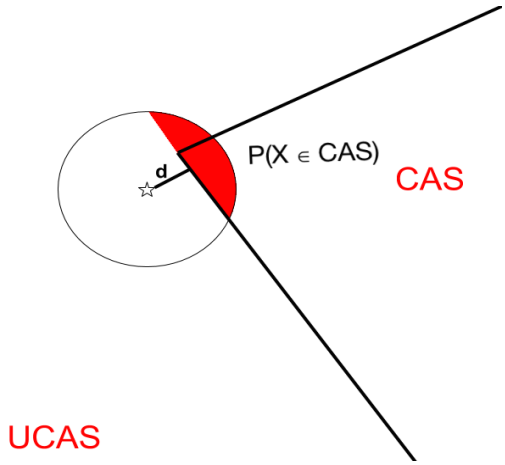


Fig. 5.3 Aircraft approaching CAS vertex it will provide higher or smaller P(I)

5.2 Monte Carlo Sampling

MC sampling method uses the prediction and its error covariance in the same way as the shortest distance method to find the $P(I)$; however, here 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:

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

To find the optimal number of samples N and calculation time, we estimated the $P(I)$ and calculation time using this method on a Mac machine with OSX 2.6 GHz intel Core i5 where $50 \leq N \leq 2000$ and found that 900 samples can be optimal, shown in figure 5. The amount of time required to estimate $P(I)$ using 900 samples is the same or less than the amount of time a radar takes to observe the next location which is 4 seconds. Given both methods, we developed a hybrid model which switches between Monte Carlo sampling and the shortest distance method to find the probability of future infringements. The reason for this is because MC sampling

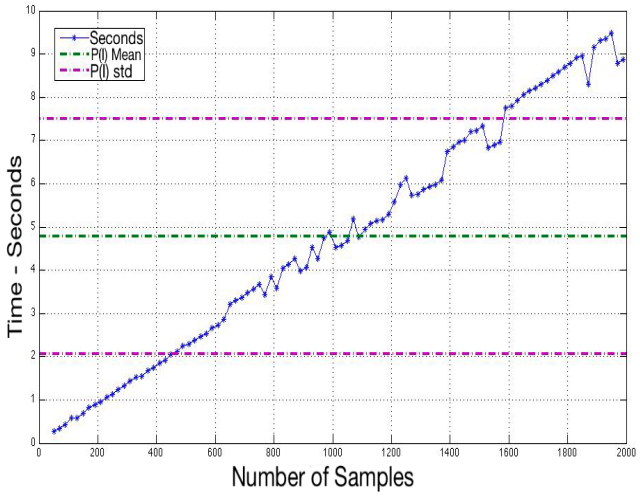


Fig. 5.4 Calculation time for a different N sample sizes, where the mean calculation time is 4.7 seconds with 2.7 standard deviation

is an effective way of finding the P(I) when the prediction is near the CAS boundary corner. However, because the MC sampling can be slow (computationally expensive), the model should switch to the shortest distance method for finding P(I) as long as it is away from CAS corner (vertex).

5.3 P(Infringements) Analysis

When to switch between both methods is defined by a threshold. We propose flexible conditions which can be changed to determine when to use one method or the other in order to find the probability of infringement accurately, they are the following:

1. If the distance from the prediction to any vertex of the CAS boundary is greater than $3\sqrt{\sigma_{max}}$, where σ_{max} is the largest variance from the prediction error covariance in all

directions($\sigma_x, \sigma_y, \sigma_z$) then $P(I) = 0$. This means the prediction is very far away from any CAS zone; therefore, the probability of infringement will be effectively zero.

2. If the distance from the prediction to any vertex of the CAS boundary is greater than $2\sqrt{\sigma_{max}}$. This condition means the prediction is not near a CAS vertex, therefore; the model will switch to shortest distance.

3. Switch to MC sampling otherwise

Finding the P(I) under these conditions on real tracks and CAS zone locations is done in two ways:

1. Estimate the P(I) using CV and CA models prediction locations separately

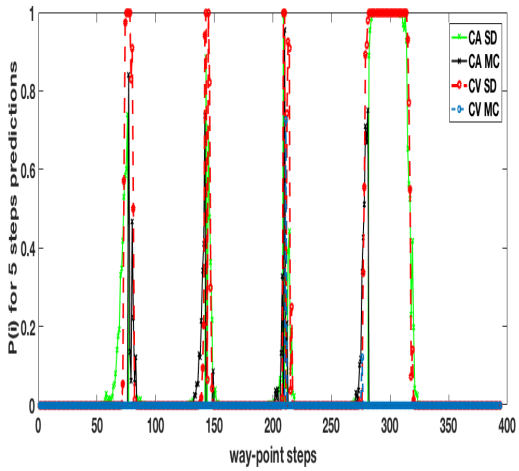
2. Estimate the P(I) using the SKF model prediction locations.

Given the conditions defined previously and the two methods (MC sampling, shortest distance), the next two sections present the estimated P(I) using both CV and CA Kalman filters separately and the switching kalman filters introduced in chapter 4.

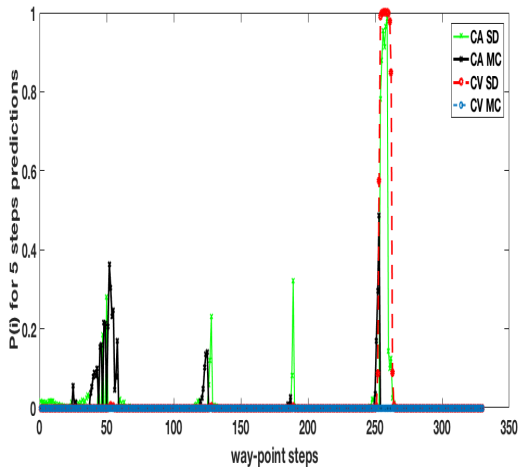
P(I) using both CV and CA models predictions

The probability of infringements for the 5 steps predictions for tracks in figure 4.5 are shown in figure 5.5. In both figures 4.5a and 4.5b, the aircraft have approached and infringed CAS zone. The probability of infringement increases until its $P(I) = 1$ as shown in figures 5.5a and 5.5b. When the aircraft, however, turns back to UCAS the predictions get decreased for both models but gets picked up faster by the CA since this model identifies the constant acceleration applied. The $P(I)$ using the CV model was higher than the CA model, because the model predictions assumes the aircraft still flies in a straight line as it approaches the CAS boundary. In figure

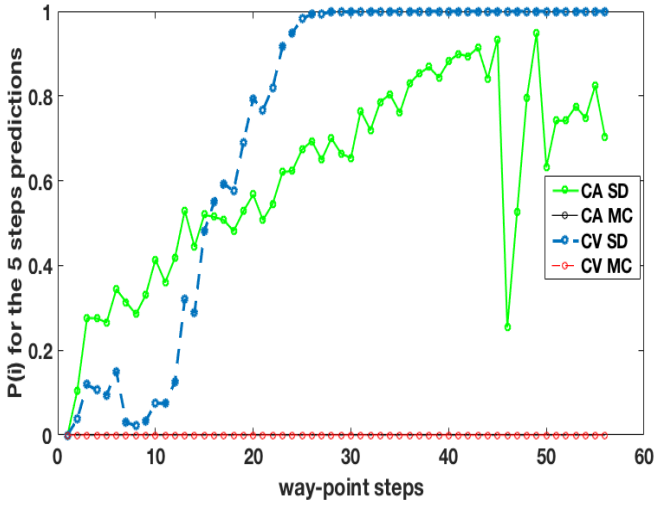
4.5c the aircraft has no intention to turn away which makes the probability of infringement increase significantly until the predictions falls inside CAS zone as shown in 5.5c.



(a) The P(i) for 5 steps predictions for figure 4.5a



(b) The P(i) for 5 steps predictions for figure 4.5b



(c) The P(i) for 5 steps predictions for figure 4.5c

Fig. 5.5 The probability of infringements using both methods on CV and CA models predictions (Not Switching)

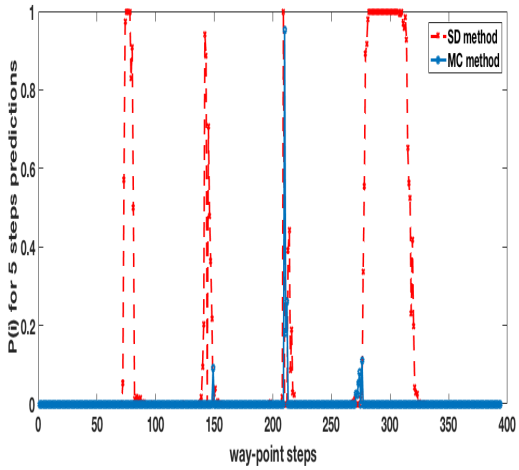
The average run time at each step including (prediction, learning and finding P(i)) on both (CV,CA) models is shown in table 5.1:

Table 5.1 Run time for tracks using all methods on both models

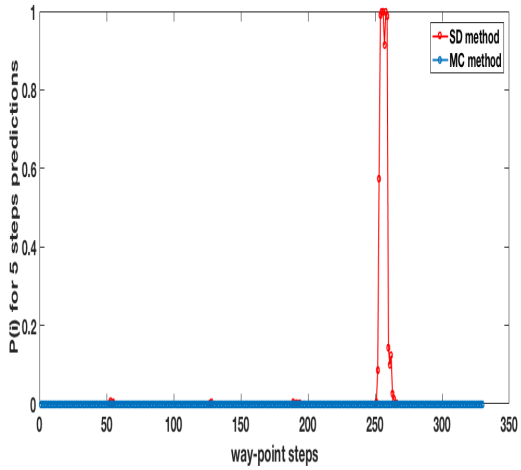
Average run time (seconds)	
Track 4.5a	0.3024
Track 4.5b	0.3023
Track 4.5c	0.0736

P(I) using SKF model predictions

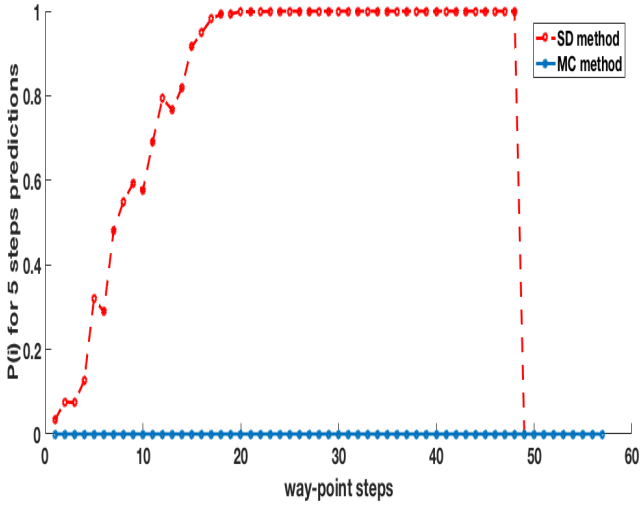
After estimating the probability of infringement for both CV and CA models separately; here, we estimate the probability of infringement using the switching Kalman filter (the optimal 5 steps prediction track). To stay conisistant through out our examples, figure 5.6 shows the P(i) using the same tracks in figure 4.5.



(a) The P(i) for 5 steps predictions for figure 4.5a



(b) The P(i) for 5 steps predictions for figure 4.5b



(c) The P(i) for 5 steps predictions for figure 4.5c

Fig. 5.6 The probability of infringements using both methods on the SKF predictions

The average run time at each step including (prediction, learning and finding P(i)) on the SKF is shown in table 5.2:

Table 5.2 Average run time for tracks using the SKF

Average run time (seconds)	
Track 4.5a	0.118
Track 4.5b	0.0489
Track 4.5c	0.0609

Table 5.2 shows the reduction time at each time step when using the skf predictions. Given the prediction locations and their probability of infringements, it is necessary that our model generates minimal false alerts to the ATC. Therefore, our model needs an assurance system such as a classifier that could help either warn or not given its output result.

5.4 Summary

This chapter focused on calculated the $P(I)$ using two probabilistic methods: the MC sampling and the shortest distance to CAS boundary. Both methods are switching according to pre-defined threshold which can be manually tuned. The Monte carlo sampling method was added because it makes up for the weakness the shortest distance method has, which is the case when aircraft approaching a CAS vertex (corner). WE then tested these method separately on both models (CV,CA) to show how the P(I) behaves with them. We then tested these methods on the optimal 5 steps prediction (switching kalman filter prediction). So far we have seen promising results in the field of GA aircraft conflict detection with no assumptions made in place such as the availability of intent information, communication and transponders. However, the problem lies within when to raise an alarm to ATC of future infringements with a reduced false alarms. In

the next chapter, will show the proposed classifier used for reducing false alerts issued to the ATC by inserting factors that helped cause infringements in the past.

Chapter 6

Classification Model

The previous chapter shows the probability of infringement for 5 steps predictions where the show promising results. However, it is important that the model will not raise the alarm to ATC based on these probabilities alone. Since the majority of infringements occurred in the past were based on human errors; this made the infringing aspect unpredictable. However, Eurocontrol [1] conducted a survey on GA pilots inquiring the reason why do they infringe CAS zones frequently. As presented in chapter 1 1.2, there were several factors that caused them to infringe CAS zones. In this chapter, it will introduce the classifier used that takes infringement factors as inputs and makes a decision that define the threshold to whether or not warn the ATC. First, we will provide the frequency analysis of infringements and then introduce the support vector machine as the classifier for our model showing in figure 6.1 bellow:

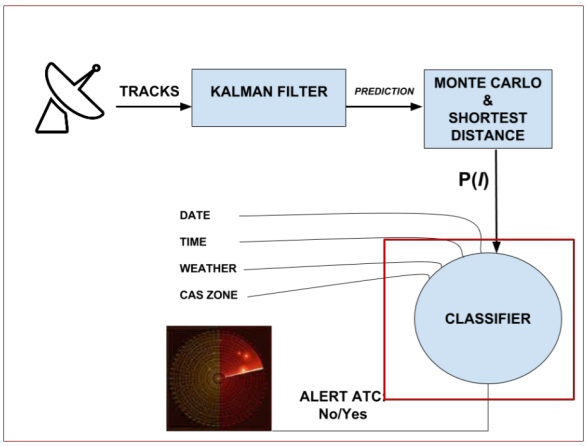


Fig. 6.1 Overall design of the research model

6.1 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 6.2 shows a geographic view of all aircraft infringements (o) inside various CAS boundaries (|) over south eastern UK.

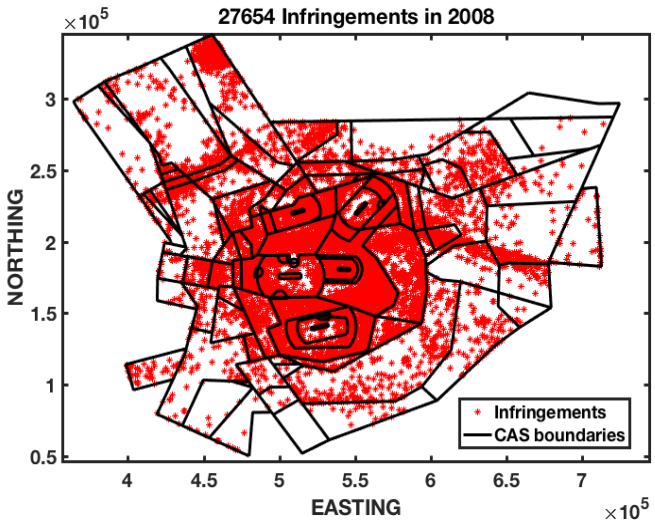
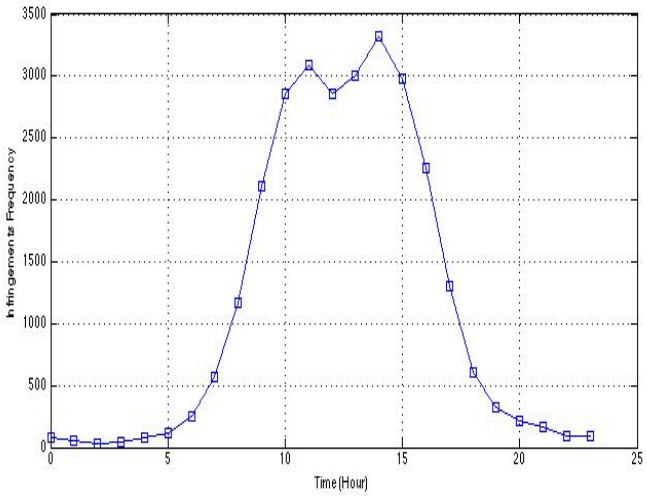
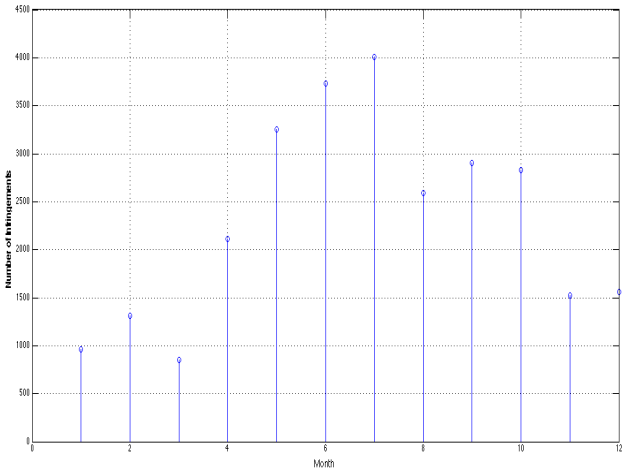


Fig. 6.2 All infringements occurred during 2008

Since most pilots use VFR, the majority of infringements occurred during the day time from 8 am - 9 pm. For the months however, the most frequent ones were during the summer time (May-September). Figure 6.3 shows the a) frequency of infringement by the hour b) the frequency by the month.



(a) Time of infringement frequency



(b) Monthly infringement frequency

Fig. 6.3 Most Frequent CAS zone infringements with respect to time and month of the year

We then looked at all of the infringements occurred with respect to CAS zones. Figure 6.4 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. Figure 6.3 shows a concerning amount of infringements since they more frequent during the months and times where major airports such as Heathrow are the busiest. The number of infringement in the summer (June-

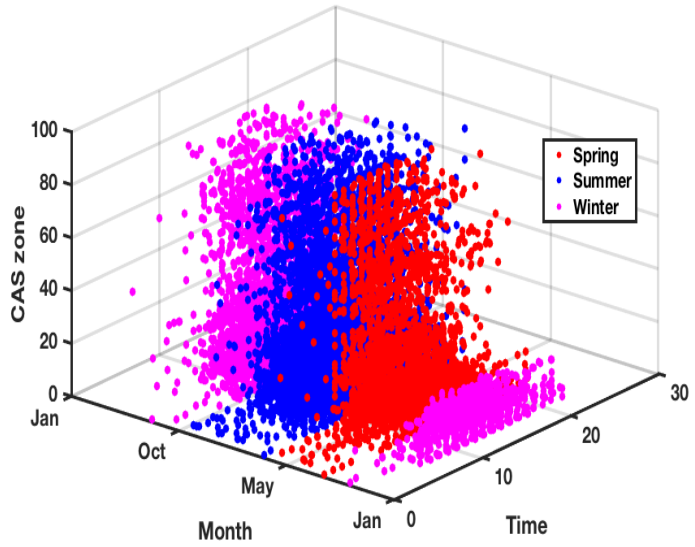


Fig. 6.4 Most Frequent CAS zone infringements with respect to time and month of the year

September) is 13244, spring (March-May) is 7171 and in the winter (October-February) is 8190. Given these factors as variables (day, month, cas zone, weather) we will train a binary support vector machine given a history data of tracks information that infringe and did not infringe (1 or 0). The next section presents a brief description of the SVM.

6.2 Support Vector Machine

It's 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 binary classification to identify one class from another (infringement or not). The distinction between classes in SVM, is done by defining a boundary around the trained class. Because the objects

can not be separated in current space A they will be lifted to higher space B. The separation boundary between target objects and the outliers is constructed by a set of support vectors in a higher dimension using a hyperplane. Given the trained classifier, we will use its output as our

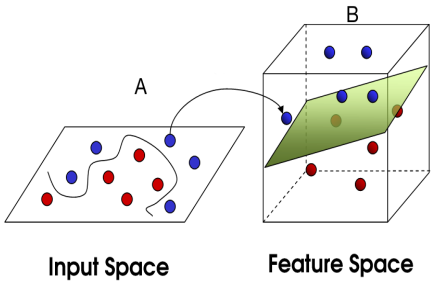


Fig. 6.5 Lifting objects into higher dimension

decision whether to alert the ATC or not. If the $P(I) > 0.5$ for a specific prediction $\widehat{x}_t$, we will use its information (date,time,CAS zone number its approaching) and predict the possibility of infringement. If the classifier scores larger than specific threshold $\tau = 0$ which can be manually tuned, it warns ATC, otherwise wait for the next prediction location $\widehat{x}_{t+1}$. Algorithm 4 shows the whole process:

Algorithm 4 Infringement Warning

Ensure: Alert ATC or Not

procedure WARNSYSFUNC($P_t(I), \widehat{X}_t^{info}$)
  if $P_t(I) \geq 0.5$ **then**
    $y_t = SVM(\widehat{X}_t^{info})$
    if $y_t \geq \tau$ **then**
      Alert ATC
    else
      WarnSysFunc($P_{t+1}(I), \widehat{X}_{t+1}^{info}$)
    end if
  end if
end procedure

Classifier performance

Given the infringed data from figure 6.4 as our one class of data "infringed CASs" and another data class of 900 that "did not infringe" as our training data, we developed a Support Vector Machine (SVM) as our classifier. After training the SVM with a total number of training data 29505 (infringement class=28605, no infringement class=900) and 10 folds cross validation using the Gaussian radial bases function (RBF) as our kernel function with a scale of 0.43. It took 25 seconds for the training. We also compared our classifier with different learning algorithms such as trees and logistic regression and showed that the SVM dominated the accuracy and TP rates shown in the figure 6.6

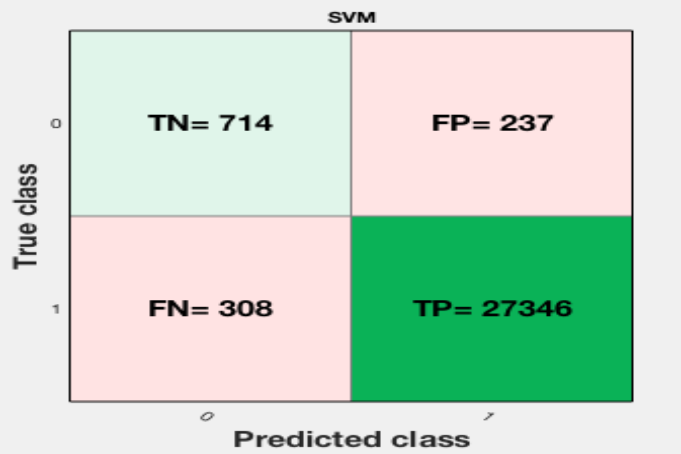
Where class "0" is re-presents no infringements and class "1" for infringements. The accuracy for the SVM was 98.1% where the accuracy of tree and LR were 97.4% and 97.1% respectively. Even though the tree was the fastest classifier which took 2 seconds and the SVM was the slowest at 22 seconds; the accuracy is what this research focus on (less false positives). Given figure 6.6 and the accuracy is of the SVM was the highest, therefore, as this this research classifier.

So far the focus was on the scenario when one aircraft infringes CAS zone. In the next chapter, we will present the scenario when multiple aircraft infringe (the advance warning issued) and how to aid the ATC resolve it.

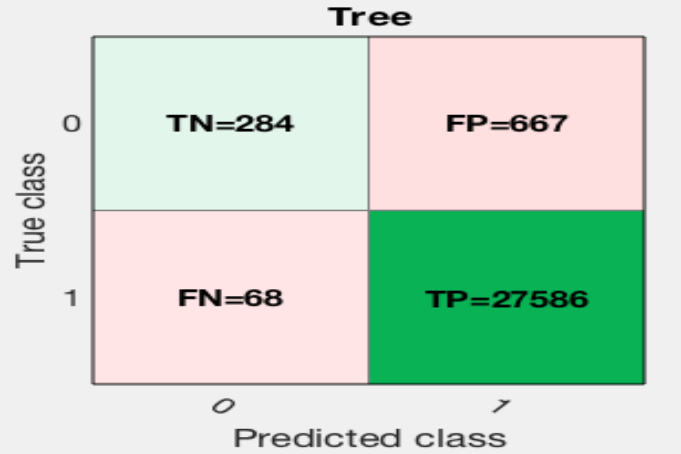
6.3 Summary

This chapter have shown the chosen classifier for the proposed model of aircraft tracking, prediction and finding the probability of infringement. The reason for adding a classifier was to reduce false alerts given the probability is greater than or equal to 50% chance of infringement.

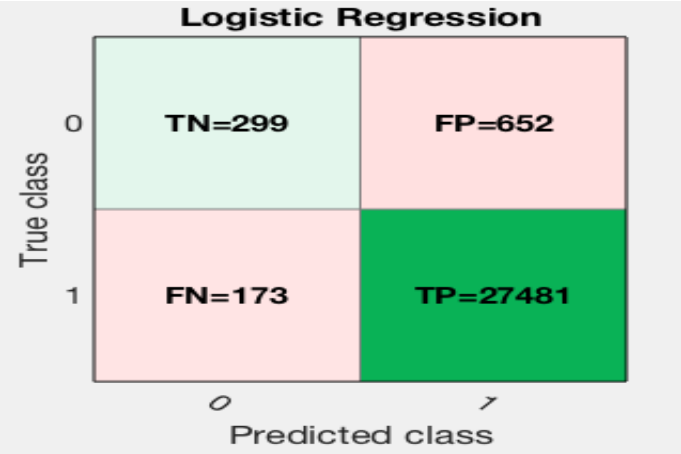
Training the SVM was the slowest compared to the rest of the classifiers (tree and logistic regression); however, the accuracy was what the focus on and it was the highest. This classifier will be given information of the current prediction such as time, date, CAS zone its approaching and the local weather around it. It will then issue a score range between [-1,1] where -1 means its not infringing and 1 otherwise. We defined τ to be 0 by default which can be tuned in the future.



(a) SVM



(b) Tree



(c) Logistic regression

Fig. 6.6 Confusion matrices of the three classifiers: SVM, Tree and Logistic Regression

Chapter 7

Automated Conflict Resolution

In reality multiple infringements occur frequently in several CAS zones given the factors in the previous chapter. Some trying to avoid a cloud or not aware of the CAS boundary or even mistaken someone else clearance as his or hers. Either ways, even with the advance warning to ATCs these aircraft might end up inside CAS zones and ATCs will not be able to handle all of them at once and might redirect commercial aircraft to another one by mistake. The Federal Aviation Administration has made it mandatory by 2019 for all GA aircraft to be equipped with a Automatic Dependent Surveillance transponder (ADS-B) that communicates automatically with the ATC. It will send the flight ID, exact location and altitude once every 1 second to the ATC. According to FAA cite [11], ADS-B uses satellites instead of radars, since radars are primitive in tracking GA where it relies on estimated radio signals and antennas to determine an aircraft's location. Whereas ADS-B uses satellite signals to track aircraft movements. However, the missing output of this transponder is that it still can not determine GA future locations and way-points. We would like to use the first part of the research which is the aircraft future prediction and find exit routes from CAS that would be sent to the GA aircraft automatically. Therefore, we would like to develop a routing plan, which generate an automated exit routes

for all unauthorized GA aircraft while not disturbing commercial aircraft original routes. In this chapter, a brief introduction to polygon triangulation and the Delaunay triangulation (DT) are presented, and the implementation of DT on CAS zone P as the polygon/plane and the GA aircraft as indices inside inside P.

7.1 Polygons and Triangulations

In this section, it will provide a brief overview of polygon triangulation and apply a computational geometry method called "Delaunay Triangulation" on a set of points in the plane.

7.1.1 Polygon Triangulation

A polygon is a shape which has line segments as "edges" connected via end points called "vertices". Figure 7.1 below shows different simple polygon shapes:

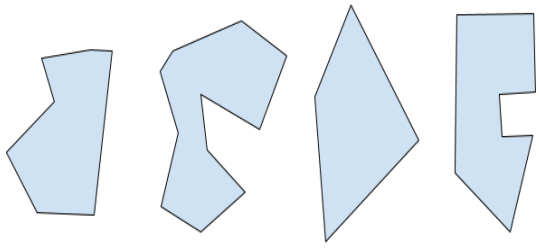


Fig. 7.1 Different shapes of simple polygons

A polygon triangulation (PT) is the process of partitioning a polygon P into non-overlapping triangles. Here are the basic properties for simple polygons:

1. A simple polygon is a closed polygonal curve without self-intersection.

2. Every simple polygon admits a triangulation.

3. There is exactly n-2 triangles in n-gon

Figure 7.2 shows a triangulation of a simple polygon (real CAS zone), it has 24-gon, therefore, has 23 triangulates:

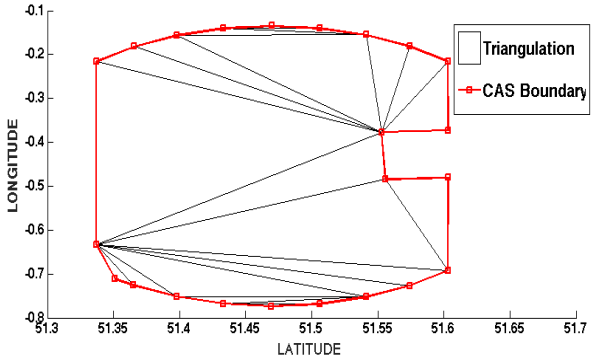


Fig. 7.2 Different shapes of simple polygons

Since CAS zones maintain different simple polygon shapes and volume sizes which have all three basic properties. Figure 7.3 shows real 105 different CAS zones around London and south east coast of UK.

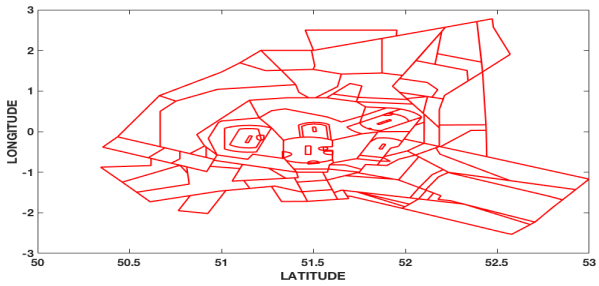


Fig. 7.3 Different CAS polygon shapes over UK southeast side

7.1.2 CAS Triangulation Using Delaunay Triangulation

In the previous subsection, we showed an example of triangulations of simple polygon. Here, each polygon or CAS zone have vertices inside them (multiple GA aircraft). We will apply a computational geometry method called "Kinetic Delaunay Triangulation" (KPT) on a set of points on a plane $P = \{p_1, p_2, ..., p_n\}$, where P is the CAS zone vertices and the aircraft predictions inside it. KDT(P) is triangulations of P such that no point in P is inside the circumcircle of any triangle in KDT(P). Figure 7.4 shows an example of a triangle and its circumcenter.

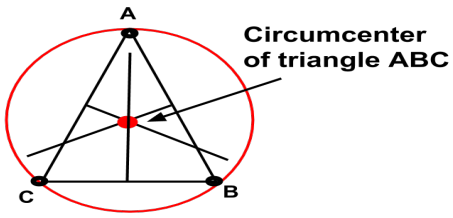


Fig. 7.4 Circumcenter of a triangulation in a plane

The circumcircle always passes through all three vertices of a triangle and its center is at the point where all the perpendicular bisectors of the triangle's edges cross. When triangulating a set of points, no point should lie inside the circle except the set of triangle points (ABC) that defined this circle. The kinetic aspect in the triangulation involves real time moving of the vertices and therefore, the triangulation changes with time.

7.2 Multiple Infringements and Resolution

In this section, we will show a real scenario where multiple GA aircraft had got inside CAS zone where there is a commercial aircraft in route for landing. The KDT will be applied on the infringed CAS zone P and the GA aircraft as vertices V_{GA} inside P at specific time t. Figure 7.5 show a KDT(P) of V_{GA} inside P .

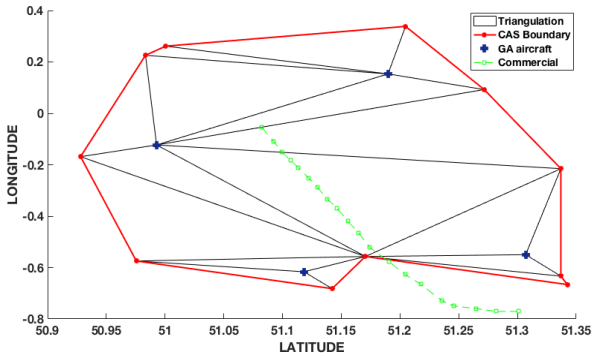


Fig. 7.5 KDT(P) of CAS zone with multiple GA aircraft inside at time t

The method proposed in this chapter is to take advantage of the KDT(P) and use edges of the triangles as way-points to exit CAS. To choose the exit route edge after triangulating $P = CAS$ it should follow two simple rules:

- No aircraft share the same exit route (edge)
- Each edge that intersect commercial aircraft way-point will not be removed from the exit route (not usable edge).

7.2.1 Conflict Routes

Before determining which edges GA aircraft should take as way points, we first need to eliminate edges that intersect commercial aircraft way-points segments "edges" inside P. Figure 7.6 shows which edges are flagged "Not an exit route" or "conflict route":

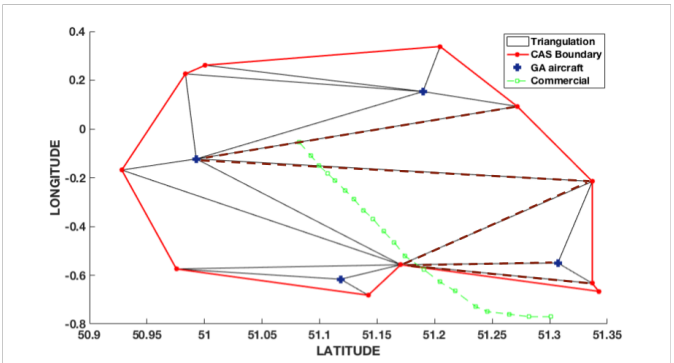


Fig. 7.6 KDT(P) of CAS zone with edges marked in "- -" as not suitable for exit route

To find "conflict edges" we will use the line segment intersection test between these edges and all other edges. We first need to assume that the line segments to be tested are in general position, meaning the following:

- No three endpoints are collinear (three points lie on same straight line).
- No two endpoints have the same x-coordinate. Specifically, no segment should be absolute vertical, no segment with no two points "just a point", and no two segments share an endpoint.

The naive way to find intersections is brute force where the algorithm checks every segments against all of them which takes $O(n^2)$ where n is the total number of line segments in polygon P. The algorithm used here for the line segment intersection test is called *sweep line algorithm*

which takes $O((n+k)logn)$, where k is the number of intersections (output) which in this case is conflict edges. This algorithm is the faster way to find intersections between multiple line segments instead of only comparing two segments [12]. The line sweep algorithm begins by sorting all segment endpoints 2n (2 endpoints for each segment) along x-axis. It then passes a sweeping line from left to right, checking at each endpoint (event) and compares above and bellow the current even.It takes O(n log n) in time. Figure 7.7 shows the sweep line process on a polygon (real CAS zone):

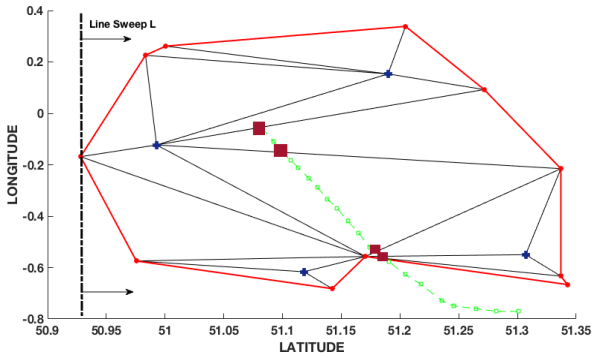


Fig. 7.7 Sweep line l passes through the plane to find intersections marked in ■

This algorithm uses Binary search tree and a queue to store segment labels. That what made it attractive since the running time will depend not only on the number of vertices but also on the output (number of intersections).

7.2.2 Automated Re-routing

After identifying the conflict routes, the proposed model in this paper should transmit exit routes to the GA aircraft inside CAS zone. Since each vertex in P has least two edges connected to it. The decision rules on which edge the aircraft should follow depend on the following:

1. The chosen exit route will be the shortest distance path to CAS boundary vertex.

2. The exit route "edge" should be used only by one GA aircraft (no two or more aircraft share the exit route)

Figures 7.8 and 7.9 shows the direction of the chosen exit routes represented by the arrows "$\longrightarrow$".

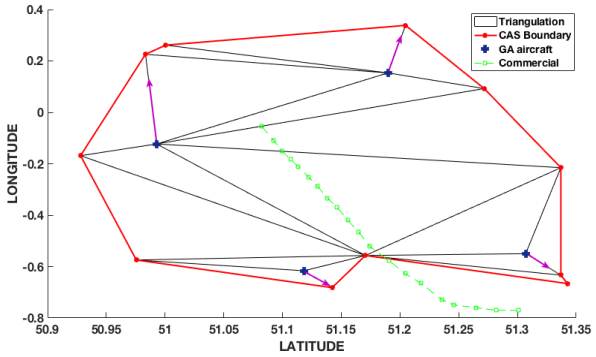


Fig. 7.8 Exit routes for 4 GA aircraft inside CAS

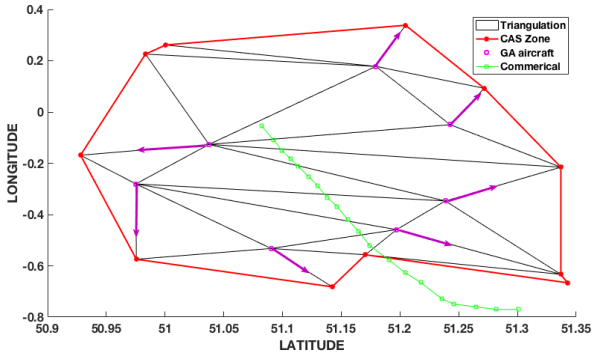


Fig. 7.9 Exit routes for 7 GA aircraft inside CAS

If there are multiple commercial and GA aviation in one CAS zones, we then need to be flexible with the first decision rule and choose the one that is not conflict with other GA aircraft.

7.3 Analysis

The method will work if the three basic properties in section 7.1.1 hold and no three GA aircraft are collinear in the space (all three in the same line segment). Fortunately, we are dealing with large geographical map locations it would be rare to find all three aircraft would overlap each other *exactly* in either (latitude or longitude). Furthermore, with the use of satellites instead of radars for tracking, the prediction accuracy in the model would increase, therefore, we use the GA aircraft prediction location at $t = 4$ to 20 sec ahead as vertices in P and not the current location at time t. This will reduce the frequency of triangulation to every 4 to 20 seconds instead of every 1 second. And by doing so it will give more time for the model to estimate and generate routes accordingly.

7.4 Summary

The objective of this research is to investigate and build a possible hybrid model that can predict future infringements accurately. Therefore, a hybrid model was implemented to track aircraft while learning their error covariances. In the first two parts of this research [13], [14] they have shown promising results so far in the field of conflict detection for GA with no assumptions such as the lack of transponders and communication and intent information. They were dealing with with one GA aircraft infringing CAS zones without warning or authorization. In reality however; there could be multiple infringements occur simultaneously. The ATC will not be able to resolve them on time, therefore, in this paper we studied this scenario and proposed a method were it can simplify the ATC manoeuvring tasks by finding an automated way to re-route GA aircraft towards the UCAS or to another CAS zone with less aircraft density within

it. This method was proposed under the assumption that the FAA mandating every GA aircraft must be equipped with an ADS-B transponder by year 2019. In this paper, it took advantage of the geographical map of CAS zones and computational geometry algorithms such as polygon triangulations and line intersection tests using line segment sweep algorithm. This algorithm takes $O((n+k)logn)$ in running time which gave the model time to implement the exit routes. The method in this paper "polygon triangulation" and finding routes in space can be applied on commercial drones as well such as shipping companies (not personal drones). For instance, Amazon inc is looking to switch their delivery system from commercial air and ground shipping to drones, however, they are looking to implement a grid in the space dedicated for drones to travel with no conflicts with other flying objects such as GA aircraft, commercial aircraft and other drones.

Chapter 8

Conclusion and Contribution

In this chapter, we summarise our research problem, objectives and the methods used to solve the problem. In addition, it will present the list of contributions in this research and finally, recommendations for future work.

8.1 Conclusion

It is the responsibility of air traffic controller (ATC) to monitor and to manage the flow of the aircraft in the airspace. The airspace is divided into seven classes; five of them correspond to controlled airspace (CAS) and two to the uncontrolled airspace (UCAS). These five classes are being controlled by the ATC and therefore, for any aircraft flying within or about to fly into CAS, the communication between that aircraft and the ATC is essential to avoid any conflict. A conflict is an event in which one aircraft loses its minimum separation by another aircraft. The size of the protected zone defined around an aircraft varies but the standard amount of separation each aircraft should have is 5 nautical miles horizontally and no less than 1000ft vertically.

ATCs monitor aircraft locations which are collected by radars scattered around the airports every 4 seconds (1 revolution per 4 seconds). An infringement occurs when an unauthorised aircraft penetrates CAS without the ATC's advance clearance. The research problem showed that the controlled airspace infringements were a major safety concern to ATCs and aircraft pilots which can cause two main issues: one was the possibility of a conflict, where one aircraft loses its minimum to another. Also, these infringements known to 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 tactics to avoid a collision. The majority of these infringements were either light weight aircraft that rely on visual flight rules or drones. Several reported incidents occurred around John F. Kennedy airport where pilots spotted drones or GA aircraft flying around the airport area during departure or landing. The infringing aircraft ID and its flight path (way-point) are unknown to ATCs unless the pilot initiates contact with the tower. For that reason, controlled airspace infringement tool (CAIT) was developed to warn ATC of CAS infringements. 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. ATCs as a result, needed a safety net that could warn them in advance to be able to maintain the flow of aircraft in the CAS. Developing a probabilistic model which predicts future infringements can be very helpful and effective if it can increase the warning time and reduces the safety threat ATCs face every day. Our research objective was broken into two parts: one was examining the possibility of building a hybrid model which can predict future infringements and assign a probability of infringement to them P(I). The probability of infringements were combined with the support vector machine classifier output which took infringement factors as inputs. Second was providing an automated way to resolve multiple conflicts inside CAS given the assumption that all GA must be equipped with advance transponders.

In this first part of the research, we combined two models constant velocity and constant acceleration for aircraft tracking while learning their error covariances. Both models switch during flight time creating a hybrid model called "switching Kalman filter". This created an optimal prediction track based on the probability of the observation given the current prediction location and observation. As the prediction increases the errors increase as well, as a result, we implemented a new modified version of the EM algorithm to either decrease or increase the prediction error covariances faster. The reason for this was because the implemented SKF used the last 15 steps of history instead of the whole track. Given the 5 steps predictions as means with their learned error covariances, we calculated the $P(I)$ using two probabilistic methods: the shortest distance to CAS boundary and MC sampling. The shortest distance method used both aircraft prediction $\widehat{x_t}$ and its error covariance $\widehat{P_t}$ (zone of uncertainty around the prediction) and calculated the shortest distance d from this prediction to the nearest CAS boundary C to estimate the probability of infringement. This method however; showed that it can only work effectively if the aircraft is approaching a straight CAS boundary. We extended this study by removing the assumption that CAS boundary can only be a straight line in order to find the P(I) which reflects the reality. The MC sampling method used the prediction and its error covariance to find the $P(I)$ as well but it drew a random number of samples N from the prediction error covariance $\widehat{P_t}$, then calculated the fraction of samples which fall inside the CAS. Given both methods, they were combined to create a hybrid method which uses both to find the probability of future infringements. The reason for combining the two methods is because MC sampling was an effective way of finding the P(I) when the prediction is near the CAS boundary vertex. However, because the MC sampling can be slow, the shortest distance method is used long as it is away from CAS corner which was defined by a flexible threshold. We then tested these methods separately on both models (CV,CA) to show how the P(I) react with both. Later, we tested these methods on the optimal 5 steps prediction (switching kalman

filter prediction). So far we have seen promising results in the field of GA aircraft conflict detection with no assumptions made in place such as the availability of intent information, communication and transponders. When to warn ATC of future infringements with reduced false alerts was the next to step to resolve. Since the majority of infringements occurred in the past were based on human errors; this made the infringing aspect unpredictable. Given history data of past infringements and their information such as dates, times and controlled zones that were infringed, we analysed them and implemented a support vector machine as this research classier. The SVM was favoured among other classification methods such as trees and logistic regression because it provided a better accuracy result of 98%. This classifier was trained using 29505 data set containing time,date and controlled airspace zone number. The output provided a score between [-1,1] where -1 means least likely to infringe and 1 infringement will occur. This score is then compared to manually defined threshold τ. If the probability of infringement is over 50%, then we check the score of the classifier given the current information of the prediction. If it is larger than τ, then warn ATC otherwise wait for the next time step.

The next part of the research studied the scenario of having multiple infringements occurred simultaneously. The ATC will not be able to resolve them on time or could resolve one but lead it to another conflict inside CAS. We proposed a method were it can simplify the ATC manoeuvring tasks by finding an automated way to re-route GA aircraft towards the UCAS or to another CAS zone with less aircraft density within it. This method was proposed under the assumption that the FAA mandates for every GA aircraft must be equipped with an ADS-B transponder by year 2019. Here, we took advantage of the geographical map of CAS zones and applied computational geometry algorithms on it such as polygon triangulations and line intersection tests using line segment sweep algorithm. The line segment sweep algorithm took $O((n+k)logn)$ in running time which gave the model time to implement the exit routes. The method in this part

is the "polygon triangulation" and finding routes in space can be applied on commercial drones as well such as shipping companies (not personal drones) which could be a possible future work.

Overall, the model can be used as a ground base safety net that can aid ATC of possible future infringements. The warning can be provided in two ways: can be a one time alarming ring or it can show a future intruder aircraft as red color in the monitor. Predicting further than 5 steps ahead could lead to larger uncertainty in predictions, therefore, more false alerts. Even though the ADS-B transponder will not provide all the data the ATC needs such as intended way-points, however; tracking aircraft every 1 seconds can help produce a longer better prediction in the future.

8.2 Contributions

In this research the list of contributions were the following:

1. Developed a switching Kalman filter with weighted on-line learning to reduce future uncertainty using a history window of 15 steps.

2. Extended the probability of infringement method in [10] so that our model can handle the case when aircraft is approaching CAS boundary corner where common infringements occur.

3. Developed an SVM classifier to classify factors whether they contribute to infringements or not to reduce false alerts.

4. Implemented an automated manoeuvring plan for multiple conflict using computational geometry methods such as Kinetic triangulation and line segment intersections test.

8.3 Future Work Recommendation

To fly a general aviation aircraft after 2019, it must be equipped with ADS-B transponder as per FAA mandatory request. With the advanced tracking satellites in place, drones became a major concern now a days. Several companies specially that works in shipping are looking to expand or replace their shipping methods with drones to reduce costs and speed up their deliveries. These drones whoever can cause conflict just as a general aviation aircraft. A possible future work recommendation would be creating an agreed network of drone traffic in the airspace using the polygon triangulation method mentioned in the previous chapter. Another possible work would be predicting more than 5 steps ahead using the 1 second ahead intervals instead of 4 seconds. Since ATC will rely on satellites as an observation system, the observation error covariance will be smaller compared to the primary radar error covariance.

References

[1] "General aviation airspace infringement survey."

[2] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[3] B. M. Yu, K. V. Shenoy, and M. Sahani, "Derivation of kalman filtering and smoothing equations," tech. rep., 2004.

[4] 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.

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

[6] 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.

[7] 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.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

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

[11] F. A. Administration, "Nextgen equip ads-b," 2015.

[12] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Santa Clara, CA, USA: Springer-Verlag TELOS, 3rd ed. ed., 2008.

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

[14] Y. Almathami and R. Ammar, "Controlled airspace infringements and warning system," pp. 45–50, 2016.