

UAV Trajectory Prediction for Collision Avoidance

Mayur Ramjee

A Dissertation submitted to the University of Bath in partial
fulfilment of the requirements for the award of
MSc in
Robotics and Autonomous Systems

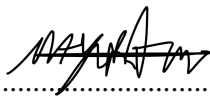
Department of Electronic & Electrical Engineering
University of Bath

September 2021

Keywords: ADS-B, IMM, UAV, Sense and Avoid, Collision Avoidance

Declaration

This dissertation is submitted to The University of Bath in accordance with the requirements of the degree of Master of Science in Robotics and Autonomous Systems, in the Department of Electrical and Electronic Engineering. No portion of the work in this document has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed 

Date 17-09-2021

Abstract

As the use cases of Unmanned Aerial Vehicles (UAVs) are increasing rapidly, there is a growing need for National Airspace (NAS) integration. Before this can happen, UAVs that fly Beyond Visual Line of Sight (BVLOS) require Detect and Avoid (DAA) capabilities to avoid collisions with surrounding aircraft. Automatic Dependent Surveillance-Broadcast (ADS-B) offers data periodically which can be used to create situational awareness and predict the future state of an intruder aircraft. ADS-B data is gathered using a Software Defined Radio (SDR) this information was decoded to create a log of real flight data.

An Interacting Multiple Model (IMM) is used to predict future positions of aircraft by combining linear Kalman Filter models. Three variations of IMM were examined: Constant Velocity Constant Acceleration (CV-CA); Constant Velocity Constant Acceleration with 2D Coordinated Turn (CV-CA-2DCT) and Constant Velocity Constant Acceleration with 3D Coordinated Turn (CV-CA-3DCT). The IMM CV-CA-2DCT has the highest accuracy of prediction when estimating real flight positional data.

Collisions are simulated with both static and dynamic intruders by the propagation of an Intruder Protected Zone (IPZ) using IMM predictions. Look ahead times of 15-25 seconds were found to be feasible for collision detection with times greater than 25 seconds showing instability. This method can be used as a detect component towards DAA requirements.

Acknowledgements

I would like to thank the following persons who made this dissertation possible: my family who have supported me throughout this MSc program by continuously encouraging me to succeed through the ups and downs; Meghani for showing me support throughout my master's journey; Professor Athiemoolam for being a mentor; Karen and her family for their kindness which made Bath feel like home.

Thank you to Dr Rob Wortham, my supervisor, for your guidance, encouragement and mentorship throughout the Robotics and Autonomous Systems course, your enthusiasm towards each module and the future of technology made the program a truly enjoyable and informative experience. I extend my gratitude to all the module convenors and tutors at the University of Bath that have assisted me towards completing the programme. My course-mates, housemates and new friends thank you for making me feel welcome from the first meetings, online and in person.

Finally, thank the Chevening office and FCDO of South Africa, without the award I would not have had the opportunity to undertake this course or form new connections and friendships.

Nomenclature and Abbreviations

UAV	Unmanned Ariel Vehicle
UAM	Urban Air Mobility
eVTOL	Electric Vertical Take-Off and Landing
ATM	Air Traffic Management
NAS	National Airspace
ATC	Air Traffic Control
ICAO	International Civil Aviation Organization
CNS/ATM	Communication, Navigation, Surveillance/Air Traffic Management
GNSS	Global Navigation Satellite System
ADS-B	Automatic Dependent Surveillance-Broadcast
DAA	Detect And avoid
UAVp	UAV pilot
ACAS	Automated Collision Avoidance System
TCAS	Traffic Collision Avoidance System
SDR	Software Defined Radio
IPZ	Intruder Protected Zone
RA	Resolution Advisory
IMM	Interacting Multiple Model
CV	Constant Velocity
CA	Constant Acceleration
CT	Coordinated Turns
CAA	Civil Aviation Authority
VLOS	Visual Line of Sight
BVLOS	Beyond Visual Line of Sight
ATS	Air Traffic Services
UTM	Unmanned Traffic Management
FAA	Federal Aviation Administration
SESAR	Single European SKY ATM Research
PSR	Primary Surveillance Radar
SSR	Secondary Surveillance Radio
UAT	Universal Access Transceivers
FMS	Flight Management System
NIC	Navigation Integrity Category
NAC	Navigational Accuracy Category
SIL	Source Integrity Level
NAC _p	Navigational Accuracy of position
NAC _v	Navigational Accuracy of velocity
HFOM	Horizontal Figure of Merit
VFOM	Vertical Figure of Merit
CPA	Closest Point of Approach
TA	Traffic Advisory
ACASX	Airborne Collision Avoidance System X
KF	Kalman Filter
MMSE	Minimum Mean Squared Error
HTTP	Hypertext Transfer Protocol
ADC	Analog to Digital Converter

ECEF	Earth Centred Earth Fixed
RMSE	Root Mean Squared Error
Ts	Time step
JSON	JavaScript Object Notation
WGS84	World Geodetic System of 1984
PWM	Pulse Width Modulation
EMI	Electro Magnetic Interferences

List of Principal Symbols

<u>Symbol</u>	<u>Description</u>	<u>Units</u>
c	Speed of light in a vacuum	3×10^8 (m.s ⁻¹)
L	Effective length of a dipole antenna	m
f	Frequency	Hz
θ	Aircraft heading angle from True North	Degrees 0-359
\dot{x}	State vector	
w	Gaussian process noise	
y	Output vector	
v	Gaussian observation noise	
x_{k+1}	Discrete state vector at time k+1	
Φ	Discrete transition matrix	
H	Observation matrix	
Q	Process noise matrix	
Δt	Change in time / time step	seconds
σ^2	Variance	
σ	Covariance	
E	Expected value	
μ	Variable mean	
s	Particle displacement	m
\dot{s}	Particle velocity	m/s
\ddot{s}	Particle acceleration	m/s ²
P	Covariance of estimation matrix	
a	Acceleration	m/s ²
Ω	Constant turn rate vector	
ω	Turn rate	Deg./s
$\hat{x}_{k+1 k}$	Future estimate of x at k+1 based on k	
K_k	Kalman gain at time k	
R	Measurement noise matrix	
Λ	Model likelihood	
$\tilde{\mu}$	Conditional model probability	
Π	Markov switching matrix	
ψ	Normalization vector	
Z^j	State innovations	
NM	Nautical Miles	
r_o	Owner positional vector	
r_i	Intruder positional vector	
$d_{separation}$	Minimum separation	
Ω	Ohm	Ohm
I_6	(6x6) identity matrix	
T_d	Time difference between prediction	seconds
T_{col}	Time of collision	seconds
T_{Pred}	Time of predicted collision	seconds

Contents

Declaration.....	i
Abstract.....	ii
Acknowledgements.....	iii
Nomenclature and Abbreviations	iv
List of Principal Symbols.....	vi
Chapter 1: Introduction	1
1.1 Research Aims and Objectives	4
1.2 Report Structure	4
Chapter 2: Background	5
2.1 Regulations	5
2.1.1 Sense and Avoid / Detect and Avoid	6
2.2 ATM (Air Traffic Management).....	7
2.2.1 Air traffic control	7
2.3 ADS-B.....	8
2.3.1 ADS-B Message structure.....	9
2.4 Automated Collision Avoidance System (ACAS).....	12
2.5 Previous research	14
Chapter 3: Theory	15
3.1 Kalman filter	15
3.1.1 Term definitions.....	15
3.1.2 Co-ordinated Turn Model	19
3.1.3 State Prediction	20
3.1.4 Correction Measurement Update	20
3.2 Interacting Multiple Model	20
3.2.1 State interaction	21
3.2.2 Model Probability Update.....	23
3.2.3 State Estimation Combination	24
3.3 Separation volume	25
Chapter 4: Implementation	26
4.1 ADS-B Data acquisition	26
4.1.1 RTL SDR receiver	27
4.1.2 Raspberry Pi.....	27
4.1.3 PiAware - Dump1090	28
4.1.4 Decoded data access	28

4.1.5 Data Extraction	28
4.1.6 Data conversion	29
4.2 Trajectory Prediction Models	30
4.2.1 Constant Velocity Model	30
4.2.2 Constant Acceleration Model	31
4.2.3 Constant Speed Coordinate Turn 3-D.....	32
4.2.4 Constant Speed Coordinate Turn 2-D.....	33
4.2.5 Interacting Multiple Models Variations.....	33
4.2.6 Error evaluation	35
4.3 Collision Detection	35
4.3.1 System overview	38
4.4 Assumptions.....	39
4.5 System Testing.....	39
4.5.1 Prediction system Testing	40
4.5.2 Collision detection system testing	40
Chapter 5: Results	44
5.1 Flight data capture.....	44
5.2 Trajectory prediction results	45
5.2.1 Flight DAL84.....	46
5.2.2 Flight EXS1204	48
5.2.3 Flight EZY89TP.....	50
5.2.4 Flight VLG84NC	52
5.2.5 Prediction Error Averages.....	54
5.3 Collision prediction time results	54
5.3.1 Static Collision results	54
5.3.2 Dynamic Collision results	60
Chapter 6: Discussion	61
6.1 Data acquisition	61
6.2 Trajectory prediction.....	61
6.3 Collision detection	62
6.3.1 Static collisions	62
6.3.2 Dynamic collision	62
6.4 Limitations	62
Chapter 7: Conclusion.....	64
Chapter 8: Future work	65

References.....	66
Appendix A – CAA ATS National Airspace Classification.....	72
Appendix B – Global ADS-B Coverage Map.....	73
Appendix - C Detailed collision timing results.....	74
Figure 2-1:ADS-B Functionality [61].....	9
Figure 2-2: ADS-B signal structure [58]	9
Figure 2-3: TCAS collision time envelopes [45]	13
Figure 3-1: Simplified linear Kalman Filter operation	15
Figure 3-2: Interacting Multiple Model description	21
Figure 3-3: Markov Decision Process of IMM mode switching	22
Figure 3-4: Separation volume intersection creating a collision scenario	25
Figure 4-1: ADS-B data acquisition system	27
Figure 4-2: R820T2 RTL2832U SDR receiver	27
Figure 4-3: Raspberry Pi 4 Single Board Computer.....	28
Figure 4-4: relative track angle of aircraft	29
Figure 4-5: Aircraft separation volume modelled as a collision cylinder.....	36
Figure 4-6: Separation volume propagation for a dynamic intruder.....	36
Figure 4-7: Separation volume used for trajectory owner	37
Figure 4-8: Simplified Flow diagram of collision detection.....	38
Figure 4-9: Flight EXS8L8 static collision points	42
Figure 4-10: Flight EZY89TP static collision points.....	42
Figure 4-11: Flight RYR56UE static collision points	42
Figure 4-12: Flight RYR8213 static collision points.....	42
Figure 4-13: Simulated collision with dynamic intruder	43
Figure 5-1: SDR directionality strength during data capture.....	44
Figure 5-2: Coverage Map of incoming ADS-B information.....	44
Figure 5-3: Range of received ADS-B messages	44
Figure 5-4: Actual trajectory of flight DAL84	46
Figure 5-5: DAL84 Mode switching IMM-CV-CA	46
Figure 5-6: DAL84 Mode switching IMM-CV-CA-CT2D.....	46
Figure 5-7:DAL84 Mode switching IMM-CV-CA-CT3D	47
Figure 5-8: DAL84 RMSE vs Time of predictor methods	47
Figure 5-9:DAL84 trajectory prediction IMM-CV-CA-3DCT	47
Figure 5-10:DAL84 IMM CV-CA-3DCT XY plot	47
Figure 5-11: DAL84 IMM CV-CA-3DCT time vs Z plot.....	47
Figure 5-12: Actual trajectory of flight EXS1204	48
Figure 5-13 EXS1204 Mode switching IMM-CV-CA	48
Figure 5-14: EXS1204 Mode switching IMM-CV-CA-CT2D.....	48
Figure 5-15 EXS1204 Mode switching IMM-CV-CA-CT3D.....	49
Figure 5-16: EXS1204 RMSE vs Time of predictor methods.....	49
Figure 5-17: Best trajectory prediction of EXS1204, IMM-CV-CA-2DCT.....	49
Figure 5-18: DAL84 IMM CV-CA-2DCT XY plot	49
Figure 5-19: DAL84 IMM CV-CA-2DCT XY plot	49
Figure 5-20: Actual trajectory of flight EZY89TP	50

Figure 5-21 EZY89TP Mode switching IMM-CV-CA	50
Figure 5-22: EZY89TP Mode switching IMM-CV-CA-CT2D	50
Figure 5-23 EZY89TP Mode switching IMM-CV-CA-CT3D	51
Figure 5-24: EZY89TP RMSE vs Time of predictor methods	51
Figure 5-25: Best trajectory prediction of EZY89TP, IMM-CV-CA-2DCT	51
Figure 5-26: EZY89TP IMM CV-CA-2DCT XY plot	51
Figure 5-27: EZY89TP IMM CV-CA-2DCT Z plot	51
Figure 5-28: Actual trajectory of flight VLG84NC	52
Figure 5-29: VLG48NC Mode switching IMM-CV-CA	52
Figure 5-30 VLG84NC Mode switching IMM-CV-CA-CT2D	52
Figure 5-31: VLG84NC Mode switching IMM-CV-CA-CT3D	53
Figure 5-32: VLG84NC RMSE vs Time of predictor methods	53
Figure 5-33: VLG84NC Predicted trajectory IMM CV-CA-CT2D	53
Figure 5-34: VLG84NC IMM CV-CA-2DCT XY plot	53
Figure 5-35: VLG84NC IMM CV-CA-2DCT Z plot	53
Figure 5-36: Total RMSE of position observed for each test flight	54
Figure 5-37: EZY89TP Time difference in collision time IMM CV-CA	55
Figure 5-38: EZY89TP Time difference in collision time IMM CV-CA-2DCT	55
Figure 5-39: EZY89TP Time difference in collision time IMM CV-CA-3DCT	55
Figure 5-40: RYR56UE Time difference in collision time IMM CV-CA	56
Figure 5-41: RYR56UE Time difference in collision time IMM CV-CA-2DCT	56
Figure 5-42: RYR56UE Time difference in collision time IMM CV-CA-3DCT	56
Figure 5-43: RYR8213 Time difference in collision time IMM CV-CA	57
Figure 5-44: RYR8213 Time difference in collision time IMM CV-CA-2DCT	57
Figure 5-45: RYR8213 Time difference in collision time IMM CV-CA-3DCT	57
Figure 5-46: EXS8L8 Time difference in collision time IMM CV-CA	58
Figure 5-47: EXS8L8 Time difference in collision time IMM CV-CA-2DCT	58
Figure 5-48: EXS8L8 Time difference in collision time IMM CV-CA-3DCT	58
Figure 5-49: Average Time difference in collision time IMM CV-CA	59
Figure 5-50: Average Time difference in collision time IMM CV-CA-2DCT	59
Figure 5-51: Average Time difference in collision time IMM CV-CA-3DCT	59
Figure 5-52: Dynamic collision time difference Flight VLG8LV IMM CV-CA	60
Figure 5-53: Dynamic collision time difference Flight VLG8LV IMM CV-CA-2DCT	60
Figure 5-54: Dynamic collision time difference Flight VLG8LV IMM CV-CA-3DCT	60

Table 2-1: ADS-B message contents	9
Table 2-2: ADS-B type code data frame contents	10
Table 2-3: ADS-B state vector.....	10
Table 2-4: NACp Values	11
Table 2-5: NACv Category values.....	12
Table 4-1: Technical specifications for RTL-SDR receiver	27
Table 4-2 : filtering fields required for Aircraft trajectory prediction	29
Table 4-3: Initial Process noise covariance constants.....	40
Table 4-4: Simulated aircraft parameters.....	43

Chapter 1: Introduction

Unmanned Aerial Vehicles (UAVs) more commonly known as drones have progressed from specialist military defence and reconnaissance missions, into a widely accessible technology. [1] states that “UAVs incorporate principles of technological modernity such as data processing, autonomy, boundless mobility and mass data collection, in combination are the perfect tools for data-driven highly connected applications”. These benefits have encouraged research into a wide array of applications including agricultural monitoring [2,3]; smart city applications relating to crowd management [4]; Urban Air Mobility (UAM): electric Vertical Take-Off, and Landing (eVTOL) taxis [5,6,7,8]; logistics and last-mile delivery fulfilment [9] etc. have all progressed through drone advancements. These platforms are purpose-built for specific applications in terms of autonomy, operating altitudes, sizes, cruise speeds, manoeuvrability, and payload capabilities. Confining UAVs to a single dedicated airspace would be an arduous task, motivating the growing need for drone integration into National Airspace (NAS).

A study by [10] estimates that there will be at least 76 000 UAVs in the air across the UK by 2030 with 36% used within the public and commercial sectors. In addition, there is a predicted doubling of manned aircraft over the next decade [12], this would strain the current Air Traffic Management (ATM) tools by creating highly dense airspace. As such, the future of aviation will require robust advancements to current Air Traffic Control (ATC) to avoid collisions between air platforms. Due to safety concerns, aviation regulatory bodies apply stringent segregation based on airspace classification which determines when and where aircraft /UAVs may operate [11], detailed in **section 2.1**.

To manage the rapidly growing air traffic into airspace the International Civil Aviation Organization (ICAO) recommends the planning, development, and implementation of Communication, Navigation, Surveillance/Air Traffic Management (CNS/ATM). The basis of CNS/ATM is the Global Navigation Satellite System (GNSS) and datalink technology. A core component of CNS/ATM is the Automatic Dependent Surveillance-Broadcast (ADS-B), ADS-

B OUT, which is already mandated by the Federal Aviation Authority (FAA) for the USA (as of January 2020) [58 - 30] and by European Aviation Space Agency (EASA) for Europe (as of December 2020) [59 -31]. ADS-B has the benefit of being accessible by surrounding aircraft or base stations, which are sufficiently close, with information relating to the owner aircraft's timestamped information as frequently updated data packets.

A principal concern to be addressed before UAV integration into NAS is the requirement for Detect and Avoid (DAA) capabilities for UAVs that fly Beyond Visual Line of Sight (BVLOS). The use of ADS-B towards DAA systems is promising, as the available ADS-B IN messages from surrounding aircraft create situation awareness. Insight into possible loss of separation would warn the UAV pilot (UAVp) or signal an avoidance manoeuvre in the case of an autonomous system. The ADS-B offers information such as the transmitting aircraft's position, speed, altitude, etc detailed in **section 2.3**. ADS-B has the advantages of a high refresh rate, low latency, and high positional accuracy [24] with precise clock synchronization when compared to current radar-based systems.

ADS-B OUT data is easily accessed by decoding the signal transmitted on the 1090 MHz frequency (Mode S Extended Squitter), which is the international standard, with 978 MHz (Universal Access Transceiver) UAT, only used in the USA. The ADS-B signal of surrounding aircraft can be inexpensively accessed with an SDR (Software Defined Radio) receiver with decoding to obtain useful information [57].

Manned aircraft avoid collisions by utilizing Airborne Collision Avoidance Systems (ACAS) such as the Traffic Collision Avoidance System (TCAS) which acts as the last resort safety net regardless of separation standards [42]. A collision or conflict is defined as the loss of minimum separation between two aircraft [51]. The minimum separation between aircraft may be modelled as volume surrounding an aircraft, this is known as the Intruder Protected Zone (IPZ). TCAS II issues voice warnings to pilots when a corresponding pair of aircraft TCAS transponders are in proximity of one another. A collision warning or Resolution Advisory (RA) is issued approximately **15 – 35 seconds** before a collision would occur [45], which is an interaction of aircraft IPZ's, this is the short-term collision avoidance time envelope that gives a pilot sufficient time to react to the warning. The operation of TCAS is detailed within **section 2.4**.

Many studies have been done on predicting aircraft trajectories based on machine learning [31,32] and deep learning [33,34] which require extensive historical training data sets. Additionally, kinematics-based approaches such as [35] use equations of motion, which require specific aircraft data such as mass, thrust and drag coefficients. Alternatively, filter-based algorithms provide a more generalized approach by using transition functions to estimate future states and thereafter recursively update estimates based on real measurements. Multiple dynamic models may be combined with an Interacting Multiple Model (IMM) provides the added advantage of fusing many filter outputs towards a more accurate estimate. This is desirable for tracking applications as multiple linear models such as Constant Velocity (CV) Constant Acceleration (CA) and Coordinated Turns with constant turn rates (CT) are combined to form a higher dimensional prediction [44].

An attractive solution to UAV collision avoidance is the ability to receive ADS-B signals from surrounding aircraft and predicting their future state. A Kalman filter-based predictor may be used to determine the future states of an aircraft. A lookahead period is required such that the UAVp or an embedded guidance system would have sufficient time to manoeuvre the UAV out of harm once a collision is detected on its current path. The bounds of TCAS RA alerts are the current standard for manned aircraft collision avoidance, as such this should be sufficient for UAVs.

1.1 Research Aims and Objectives

This research aims to develop a short-term trajectory prediction system for UAV collision avoidance with aircraft which transmit an ADS-B signal. This system will form a Detect component for integration with a larger DAA system per the requirements for UAV BVLOS operation towards integration into NAS. Components of this study include:

- Recording and extracting real ADS-B (Mode S Extended Squitter) data from flights using an SDR
 - Extraction of position and velocity information of each flight
- Development of linear Kalman predictors for motion models such as:
 - Constant Velocity (CV) – non-manoeuving dynamics.
 - Constant Acceleration (CA) – manoeuvres that maintain a constant acceleration in all axes of motion.
 - Coordinated Turn (CT) – circular turns based on constant turn rates.
- Fusing Multiple linear models using an IMM and evaluating the output prediction accuracy
- Propagation of the separation volume is required for collision detection.
- Simulating collisions to test prediction capabilities of a TCAS RA alert (15-35 seconds) before a collision occurs, would require:
 - Collision simulation using real flight trajectories occurring with static and dynamic intruders.
 - Evaluation of the prediction times produced by a prediction model to a collision vs actual time to a collision.

1.2 Report Structure

This research is structured as follows: within **Chapter 2** background on UAV regulations, current mid-air collision avoidance systems, the ADS-B message and previous research are explored. **Chapter 3** provides the theory of operation with regards to Kalman predictors, an IMM and collisions are identified geometrically. **Chapter 4** describes the implantation and experimentation implementation to achieve the outlined research objectives, with testing descriptions detailed. **Chapter 5** contains the results based on the testing parameters; a discussion expanded upon in **Chapter 6**. Finally, **Chapter 7** concludes on the findings of this work and **Chapter 8** seeks to highlight future improvements.

Chapter 2: Background

Accompanying the recent increase in drone operations, a comparative increase in regulations has been observed. These regulations are stringent and required to maintain the safety of manned aircraft and the public. To best understand the requirements of NAS integration, the regulations applied by the Civil Aviation Authority (CAA) of the UK is explored.

2.1 Regulations

The CAA stipulates operational limitations within document CAP722 [13], based on the drone's maximum take-off weight. UAVs with a take-off weight of less than 25kg are allowed to operate within the **Open Category**, restricted to 120m above surface level with no requirement for any certification from the CAA besides a Flyer-ID. UAVs that do not meet the requirements of the Open Category are considered to pose a greater operational risk and are subject to the rules of the **Specific Category** which requires a risk assessment and a general Visual Line of Sight (VLOS) Operation certification. VLOS mandates that UAVp should have the capability to see the UAV and the surrounding airspace whilst it is airborne. The VLOS acceptable range is 500m horizontally and 120m above the earth's surface. Comparable to Visual Flight Rules (VFR) for manned aircraft, during VLOS operations the UAV should be able to see/sense an obstacle and avoid it without endangering the surrounding aircraft. The final category is known as the **Certified category** in which the UAV is considered to perform complex operations which would present the equivalent risk to that of specialised manned aircraft. Subsequently, these aircraft are subject to all articles of the Air Navigation Order which requires the operator to obtain a certificate of airworthiness of the UAV; have a permit to fly; have a licenced crew and follow the rules of the air [14]. The UAVs operating within the Certified category require the capability to fly Beyond Visual Line of Sight (BVLOS). BVLOS stipulates that the UAV holds an equivalent level of safety as manned aircraft within the specific ATS (Air Traffic Services) controlled airspace mandated requirements as described within **Appendix A**. The section of regulations within CAP722 which applies to the Certified category is yet to be finalised, with the Future Air Mobility Challenge being a current topic of interest for the CAA innovation hub [15]. Aircraft flying within the Certified category will be

required to adhere to the compulsory use of *Sense and Avoid* (SAA) aka. *Detect and Avoid* (DAA) systems. [16] states that “Sense and Avoid systems must provide for collision avoidance protection between a UAV and other aircraft analogous to the see and avoid the operation of manned aircraft to an acceptable level of safety.”

2.1.1 Sense and Avoid / Detect and Avoid

The CAP722 document outlines the requirements of DAA systems: to maintain separation, prevent aircraft from getting into proximity of one another, and avoid collisions with either have been stated to require the minimum functions including:

- Detect and avoid traffic (aircraft in the air and on the ground) per the rules of the air.
- Detect and avoid all airborne objects, including hang-gliders, paragliders, microlights etc.
- Enable the remote pilot to determine in-flight meteorological conditions.
- Avoid hazardous weather conditions.
- Detect and avoid terrain and other obstacles.

To expand upon this, the CAA has introduced a fundamental principle of UAVs operating BVLOS within document CAP1861, which contains a matrix of potential solutions with 4 types of technology highlighted which might include [18]:

- Ground-based infrastructure – used to detect cooperative and non-cooperative airspace users, supplemented with ground infrastructure including radar.
- Electronic Identification – the Identification, position, speed, heading and altitude of other aircraft which is critical for DAA operations.
- On-Board DAA equipment – includes the detect functions provided by various sensors and the avoid functions provided by flight controllers
- UTM (Unmanned Traffic Management) – utilizing the potential of data collection, processing, and dissemination to the DAA ecosystem and interaction with current ATM (Air Traffic Management).

These guidelines however do not outline a specific regulatory technology for DAA systems when operating in BVLOS conditions which present a barrier to overcome prior UAV integration into NAS [17]. This topic has become the focus of many research collaborations such as the FAA’s (Federal Aviation Administration) NextGen program and SESAR (Single European SKY ATM Research), a collaborative project to completely overhaul the

European airspace and its ATM (Air Traffic Management) [6x]. These initiatives seek to support the ICAO (International Civil Aviation Authority) Global Air Navigation Plan which seeks to increase the capacity, efficiency, and predictability of airspace management.

2.2 ATM (Air Traffic Management)

The International Civil Aviation Organization (ICAO) defines Air Traffic Management as “the dynamic, integrated management of air traffic and airspace including air traffic services, airspace management and air traffic flow management – safely, economically and efficiently — through the provision of facilities and seamless services in collaboration with all parties and involving airborne and ground-based functions” [19]. Air traffic management comprises ground-based services such as Air Traffic Control and airborne transponder systems located on aircraft.

2.2.1 Air traffic control

Air Traffic Control (ATC) is operated in ground-based air traffic control centres and has the function of keeping track of aircraft positions for the safe and efficient management of air traffic. According to ATC regulations, aircraft are not to be flown at altitudes under or over other aircraft for extended periods as controllers only have a two-dimensional view of airspace [51]. ATC systems operate via ground-based radar to locate and detect aircraft. Radar is the sensor used by ATC which allows for the detection of an aircraft’s range and azimuth through the time differences of pulses received [52]. The radar detection system is comprised of Primary Surveillance Radar (PSR) and Secondary Surveillance Radio (SSR) coupled with radio communications.

PSR is a passive echolocation method that does not require the action of the object being tracked. Some disadvantages of PSR include no altitude or identity feedback of the tracked aircraft, false positives such as birds and poor detection performance for flight tangential to the radar in poor weather conditions [53].

SSR is an extension over PSR comprising a ground-based interrogator and an aircraft transponder. Information is gathered under inquiry on the 1030MHz and response on the 1090MHz frequencies. Disadvantages of SSR include the inability to discriminate between two aircraft at the same location due to its dependency on an aircraft's avionics system [53]. Additionally, PSR retains weaknesses from PSR such as poor resolution and various modes of indiscrimination (A/C/S) [55]. Mode A/C provides aircraft identification and altitude while Mode S provides the same information as A/C with the added benefit of unique target

addressing. This is possible with the aid of a unique 24-bit aircraft address through a two-way data link between the ground station and an aircraft. [55] states that “Mode S can resolve two aircraft which are in the same geographical position and can discreetly interrogate single aircraft transponders”.

Radar-based systems are reaching their maximum capacity due to shortcomings concerning visibility propagation, line of sight requirements and lack of digital data links [56]. Therefore, radar-based technology alone would not be feasible for UAV integration.

2.3 ADS-B

ADS-B is a satellite-based surveillance system with parameters such as position, velocity and aircraft identification are transmitted through either Mode S Extended Squitter (1090 MHz) [54] or 978 MHz Universal Access Transceivers (UAT), which is only used in the USA. The mandatory use of ADS-B out systems has already been proposed by the Federal Aviation Authority (FAA) within regulations 14 CFR 91.225 and 14 CFR 91.227 as part of their “NextGen” initiative which seeks to futureproof ATM [64]. Additionally, ADS-B mandates are effect in Australia, Europe, South Africa, India, Malaysia, Taiwan with regulatory activity ongoing in China, Canada, and Saudi Arabia. A global ADS-B coverage map may be seen in Appendix B [50]. ADS-B has the added benefit of a high cycle time of generally 1Hz with a maximum range of 200 NM or 370km.

ADS-B was developed to enhance situational awareness, conflict detection and surveillance in non-radar environments. The automatic component is due to the operation without the intervention of the pilot or ATC. The dependent technique due is to its requirement of positional data from either a GNSS or Flight Management System (FMS). In comparison to PSR/SSR, ADS-B ground-based systems are easier to install and maintain [28]. This technology typically involves aircraft constantly sending in real-time position and flight parameters, broadcast on the same frequency as Mode S messages. ADS-B operation is seen in figure 2-1 whereby an aircraft obtains its location from a GNSS source such as Global Position Satellite (GPS); the ADS-B transponder on the aircraft transmits the signal which contains a state vector of position, velocity and more to surrounding aircraft (Equipped with ADS-B IN); the ADS-B data may be transmitted to ground-based receivers (Equipped with ADS-OUT) via communications channels as frequently updated data packets; data received by ground stations may be stored, used, or shown on real-time maps [62].



Figure 2-1: ADS-B Functionality [61]

2.3.1 ADS-B Message structure

An ADS-B frame consists of 5 main parts and is 112 bits long, encoded using Pulse Width Modulation (PWM) coding with a data rate of 1Mbps. A complete ADS-B signal is composed of 8 μ s preamble pulse followed by 112 μ s data information bit pulse as seen in figure 2-2.

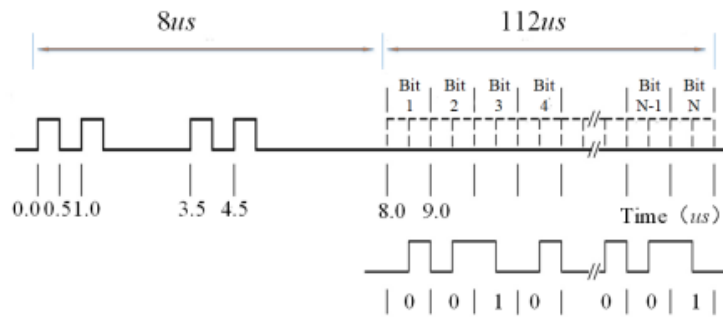


Figure 2-2: ADS-B signal structure [58]

The ADS-B frame contains the 5 main parts as seen in Table 2-1:

Table 2-1: ADS-B message contents

Information	Downlink format	Transponder Capability	ICAO Address	Message extended squitter	Parity /Integrator ID
Number of bits	5	3	24	56	24

The ICAO address is a hex code which allows for a unique identification of aircraft by the Mode-S transponder assigned to it, which is according to the ICAO regulations [60]. The type-code of ADS-B messages is located from bits 33-37 which contain specific information regarding the aircraft, with different transmission rates [54] seen in Table 2-2:

Table 2-2: ADS-B type code data frame contents

Type code	Data frame content	Airborne Transmission rates
1-4	Aircraft identification	0.2 Hz
5-8	Surface position (longitude, latitude)	-
9-18	Airborne position (Barometric Altitude)	2Hz
19	Airborne velocities	2Hz
20-22	Airborne position (GNSS Height)	2Hz
23-27	Reserved	
28	Aircraft status	0.2Hz /1.25Hz (TCAS RA and Squawk code change dependant)
29	Target state and status information	0.8Hz
31	Aircraft operational status	0.4Hz / 1.25 (Dependant on changes in NIC, NAC, SIL)

The state vector of ADS-B messages of interest which would make up a state vector is seen in Table 2-3. The precise representation of all components of the ADS-B message are deemed to be outside the scope of this study but can be found in EUROCONTROL-SPEC-0149-12 specification which documents the ADS-B message in detail.

Table 2-3: ADS-B state vector

Description	Resolution	Notes
Latitude	180/2 ²³ deg.	WGS84
Longitude	180/2 ²³ deg.	WGS84
Altitude	10 ft	Feet
Horizontal Velocity	1/2 or 2 kts (supersonic)	Knots
Vertical Speed	6.25 ft / min	Ft/min
Track heading	360° / 2 ¹⁶	Degrees from Magnetic North
Track angle rate	Deg./s	1/32 deg./s
NACp	-	(0-11) ADS-B version 2
NACv	-	(0-4) ADS-B version 2

Navigation Integrity Category (NIC), Navigational Accuracy Category (NAC) and Source Integrity Level (SIL) are all indicators of position data quality. NIC expresses the integrity of a containment radius of error. [66] defines SIL as “the probability of reported horizontal position exceeding the radius of containment defined by the NIC without an alert, assuming there are no avionics faults”. In ADS-B version 2 messages NAC is divided into Navigational Accuracy of position (NACp) and Navigational Accuracy of velocity (NACv) which are 95% horizontal accuracy bounds. Values of NACp and NACv are seen in Tables 2-4 and 2-5 respectively, values are specified as Horizontal Figure of Merit (HFOM) which specifies uncertainty bounds within the horizontal and Horizontal Figure of Merit and Vertical Figure of Merit (VFOM) which specifies vertical uncertainty bounds.

These are interpreted as measurement error values and will be used for prediction formulation within **section 4.2** of this report.

Table 2-4: NACp Values

NACp	HFOM	VFOM
11	< 3 m	< 4 m
10	< 10 m	< 15 m
9	< 30 m	< 45 m
8	< 0.05 NM (93 m)	N/A
7	< 0.1 NM (185 m)	N/A
6	< 0.3 NM (556 m)	N/A
5	< 0.5 NM (926 m)	N/A
4	< 1.0 NM (1852 m)	N/A
3	< 2 NM (3704 m)	N/A
2	< 4 NM (7408 m)	N/A
1	< 10 NM (18520 m)	N/A
0	>10 NM or unknown	N/A

Table 2-5: NACv Category values

NACv	HFOM rate	VFOM rate
4	< 0.3 m/s	< 0.46 m/s
3	< 1 m/s	< 1.5 m/s
2	< 3m/s	< 4.5m/s
1	<10 m/s	< 15.2 m/s
0	N/A	N/A

2.4 Automated Collision Avoidance System (ACAS)

Traffic Alert and Collision Avoidance (TCAS) which is an example of ACAS is the primary mandated collision avoidance system that operates independently of ATC required for flights with a take-off mass greater than 5700 kg or more than 19 passengers [25]. TCAS operates by monitoring surrounding airspace for aircraft fitted with a corresponding active transponder. The TCAS transponder emits position, altitude, and vertical speed issuing warnings to the pilot if an aircraft enters a virtual unsafe zone. It was designed as a pilot-in-the-loop which issues the pilot with a vertical movement advisory, i.e., a warning to climb, descend or maintain current altitude. The system is not able to generate any advisory for horizontal resolution due to computational requirements and limitations of the directional antennas.

TCAS uses the response signals and altitude of the aircraft to calculate relative movement and determines the Closest Point of Approach (CPA) for a deterministic state prediction. It forms two protected areas around its plane which are Traffic Advisory (TA) with time to CPA in 20 – 48 seconds and Resolution Advisory with time to CPA as 15-35 seconds [26] seen in *figure 2-3*. As, TCAS calculates conflict based on the historical and present position and velocity without any flight intention information, thus with increased traffic flow in dense airspace predictions become more challenging. TCAS is currently limited to large aircraft which can support its hardware (two transponders fitted to the top and bottom of the aircraft's fuselage), power requirements and manoeuvrability demands as set out by the avoidance logic [27]. Due to the size, weight, and power constraints of standard TCAS systems, it would not be an appropriate choice as a transponder for UAV deployment.

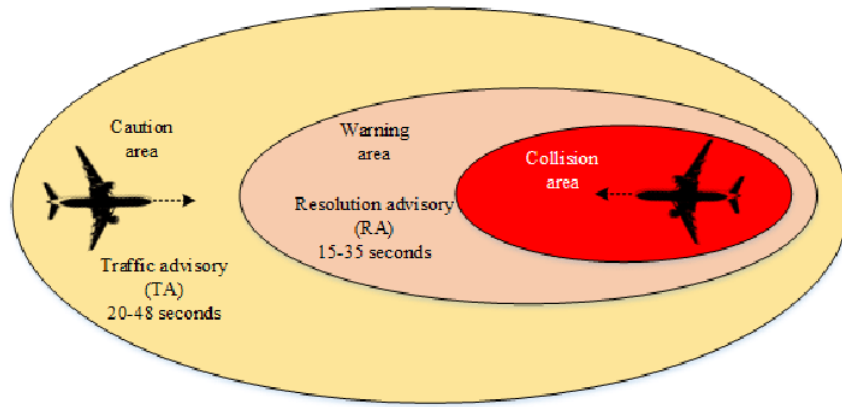


Figure 2-3: TCAS collision time envelopes [45]

[29] has been working on a redesign of the TCAS system, called Airborne Collision Avoidance System X (ACASX) which would support additional surveillance sources. A variation of ACASX known as ACAS Xu is envisioned to accommodate UAVs, based on a Markov Decision Process (MDP) for probabilistic state estimation models. [59] states that “ACAS-Xu is designed to accommodate ADS-B messages, for passive surveillance and coordination with TCAS-equipped aircraft, in addition it will accept radar or electro-optical sensors (EO) to track non-cooperative traffic”. ACAS X is aimed at close range conflicts with a time envelope of 15-48 seconds with vertical advisory only. ACAS-X standards have been submitted to the ICAO but are currently awaiting regulatory approval [33]. ACAS-X alone will not be the best collision avoidance approach as only vertical advisory would not suit UAVs operating at lower altitudes [59].

2.5 Previous research

Research by [28] has investigated an Interacting Multiple Model's (IMM) ability to estimate a conflict probability based on a probabilistic set blocks approximation of the Intruder Protected Zone (IPZ), a cylindrical zone surrounding an aircraft, that if breached signals a collision. This study defined IMM's for vertical and horizontal modes of operation, with models for constant acceleration, constant altitude with constant heading etc. A forecast window of 20 seconds in 1-second intervals with results showing combined linear trajectory predictions function better in predicting the future position of an aircraft in comparison to single linear predictors. The system was tested using linear and circular simulated collision scenarios with an intruder aircraft.

[70] proposed a Sense and Avoid (SAA) system for small UAVs by investigating cooperative and non-cooperative sensor data. This study used an IMM to track intruder aircraft with cooperative ADS-B data available using multiple kinematics behaviours. Results showed ADS-B data was successful in providing situational awareness of intruders towards SAA capabilities. [71] aimed to improve SAA functions by using an IMM to track intruder aircraft, an encounter model was used to generate relisted intruder trajectories. The predicted IMM output was found to reduce estimation errors for a given sensor whilst showing an improvement in detection of the Closest Point of Approach (CPA) horizontally in comparison to linearly extrapolated intruder trajectories.

[65] investigated the use of ADS-B (UAT) and TCAS information fusion towards UAV collision avoidance using an Extended Kalman filter. Results indicated that only a meter of position accuracy is gained, however, if ADS-B data produced accurate velocity and position estimates the incorporation of TCAS range is not worth the computation of data fusion.

Chapter 3: Theory

3.1 Kalman filter

A Kalman Filter (KF) is a popular estimation method that provides a formulation for the Minimum Mean Squared Error (MMSE) for systems with multiple inputs and outputs. KFs have been used widely in the fields of orbit calculations, target tracking, navigation, sensor data fusion and microeconomics [39].

The KF assumes a linear transition and observation model, it requires a zero-mean Gaussian noise distribution. State estimation with the KF is executed recursively via a state prediction via previous state measurements and a state transition function and a state correction based on the actual observed state measurements, a simplified diagram is seen in figure 3-1.

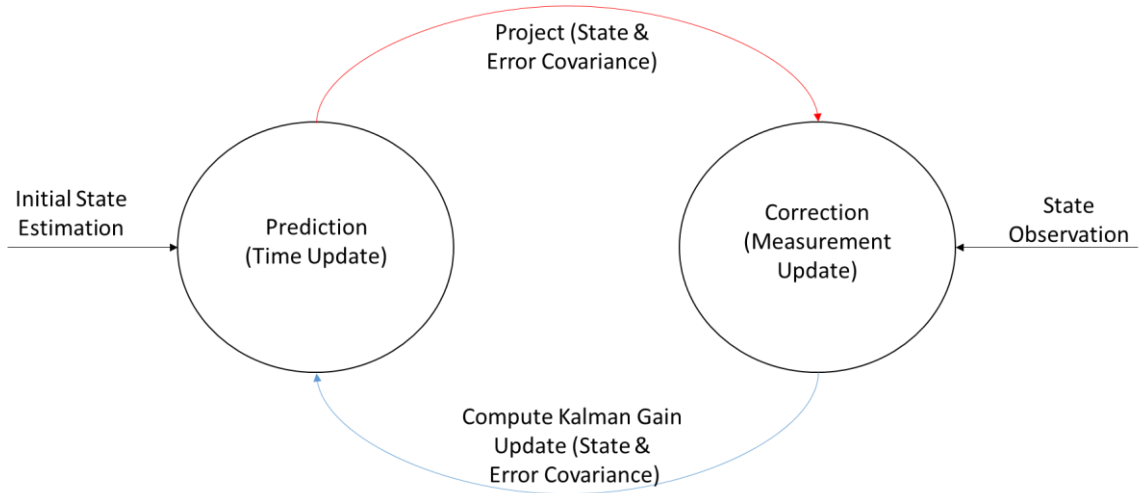


Figure 3-1: Simplified linear Kalman Filter operation

3.1.1 Term definitions

For a continuous process without a deterministic input and no modelled coupling between stochastic inputs and measurements the system can be represented by a system of equations:

$$\begin{aligned}\dot{x} &= Ax + w \\ y &= Cx + v\end{aligned}\tag{3.1}$$

Where A is the continuous process dynamics matrix, x is the state vector, and w is the continuous process noise vector. The output vector is y with observations C and v is the

continuous observation noise vector. As the filters will not be used on a continuous process but rather updated by discrete measurements, a discrete model must be used:

$$\begin{aligned} x_{k+1} &= \Phi x_k + w_k \\ y_k &= H x_k + v_k \end{aligned} \quad (3.2)$$

The discrete model uses a sample time k and discrete versions of the transition matrix Φ , observation matrix is H and w_k , v_k are the process noise and measurement noise at time step k respectively. In the case of the discrete linear KF, the measurement noise and process noise are uncorrelated. Additionally, the process and measurement are modelled as zero-mean Gaussian distributions, which can be modelled as below:

$$\begin{aligned} E[v_k] &= 0 \\ E[w_k] &= 0 \end{aligned} \quad (3.3)$$

The measurement and process covariance matrices are as follows:

$$\begin{aligned} R_k &= E[v_k v_k^T] \\ Q_k &= E[w_k w_k^T] \end{aligned} \quad (3.4)$$

The derivation of Q_k can be explained by the example of a particle moving at a constant velocity with displacement s and velocity v , a process covariance matrix Q is defined as:

$$Q = \begin{bmatrix} \sigma_x^2 & \sigma_{xv} \\ \sigma_{vx} & \sigma_v^2 \end{bmatrix} \quad (3.5)$$

Where σ^2 is the variance of a variable and σ is the covariance between two variables, this may be expressed as equations below:

$$\begin{aligned} \sigma_v^2 &= E(v^2) - \mu_v^2 = E((a\Delta t)^2) - (\mu_a \Delta t)^2 \\ &= \Delta t^2 (E(a^2) - \mu_a^2) \\ &= \Delta t^2 \sigma_a^2 \end{aligned} \quad (3.6)$$

$$\begin{aligned}
\sigma_x^2 &= E(x^2) - \mu_x^2 = E\left(\left(\frac{1}{2}a\Delta t^2\right)^2\right) - \left(\frac{1}{2}\mu_a\Delta t^2\right)^2 \\
&= \frac{\Delta t^4}{4}(E(a^2) - \mu_a^2) \\
&= \frac{\Delta t^4}{4}\sigma_a^2
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
\sigma_{vx} = \sigma_{xv} &= E(xv) - \mu_x\mu_v = E\left(\frac{1}{2}a\Delta t^2a\Delta t\right) - \left(\frac{1}{2}\mu_a\Delta t^2\mu_a\Delta t\right) \\
&= \frac{\Delta t^3}{2}(E(a^2) - \mu_a^2) \\
&= \frac{\Delta t^3}{2}\sigma_a^2
\end{aligned} \tag{3.8}$$

Where Δt is the change in time, E is the expected value of a variable and μ is the mean of each variable respectively, terms are expressed as a normally distributed standard deviation of acceleration σ_a with a mean of zero. When substituting these values into equation 3.5 the resultant covariance matrix is:

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \sigma_a^2 \tag{3.9}$$

For continuous acceleration applications this covariance may be projected by using a state transition matrix. Consider a particle moving at constant acceleration, with states seen in equation 3.10 the transition matrix Φ may be defined as equation 3.11

$$\dot{x} = [s \quad \dot{s} \quad \ddot{s}]' \tag{3.10}$$

$$\Phi_{CA} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

And since the acceleration term is required, the matrix Q_a is defined as:

$$Q_a = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \sigma_a^2 \quad (3.12)$$

By using the transition model Φ a projection of the process covariance matrix Q may be determined by the formulation:

$$\begin{aligned} Q &= \Phi Q_a \Phi^T \\ &= \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \Delta t & 1 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} \sigma_a^2 \\ &= \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} \sigma_a^2 \end{aligned} \quad (3.13)$$

This is known as the Wiener-process acceleration process model [41]. Once the process noise matrix Q and state transition matrix Φ are known a covariance matrix P is required. This matrix is known as the estimated uncertainty or covariance of the current state, matrix P . A particle considered to be moving at constant acceleration as described in equation 3.10 would have a covariance matrix as seen below

$$P = \begin{bmatrix} p_s & p_{s\dot{s}} & p_{s\ddot{s}} \\ p_{\dot{s}s} & p_{\dot{s}\dot{s}} & p_{\dot{s}\ddot{s}} \\ p_{\ddot{s}s} & p_{\ddot{s}\dot{s}} & p_{\ddot{s}\ddot{s}} \end{bmatrix} \quad (3.14)$$

Where p_s is the variance of displacement, p_v is the variance of velocity and p_a is the variance of acceleration. Additionally, all off diagonal entries are covariance values between displacement, velocity, and acceleration respectively.

Matrix R is defined as the measurement uncertainty and is proportional to the amount of noise observed during a state measurement. Matrix H is defined as the state observation, this relates the state vector x to which values are measured during each time step.

3.1.2 Co-ordinated Turn Model

The constant speed Coordinate turn model in 3-Dimensions (3D-CT) model, assumes a target moves at a constant turn rate in a circular motion. When an object is moving at constant speed motion, the acceleration vector is orthogonal to the velocity vector [40]. For an arbitrary plane of motion, the acceleration may be described by:

$$a = \Omega \times v \quad (3.15)$$

Where Ω is the constant turn rate vector such that $\dot{\Omega} = 0$ and v is the velocity vector such that taking the derivative of 3.15 leads to

$$\dot{a} = (\Omega \cdot v)\Omega - (\Omega \cdot \Omega)v \quad (3.16)$$

As v is orthogonal to Ω , equation 3.16 may be reformulated as

$$\dot{a} = -\omega^2 v \quad (3.17)$$

Where ω is defined as:

$$\omega \triangleq \|\Omega\| = \frac{\|a\|}{\|v\|} \quad (3.18)$$

Thus, for a particle undergoing a co-ordinated turn with states as seen in equation 3.10 the discretised transition function in cartesian coordinates may be defined as:

$$\Phi_{CT} = \begin{bmatrix} 1 & \frac{\sin\omega\Delta t}{\omega} & \frac{1 - \cos\omega\Delta t}{\omega^2} \\ 0 & \cos\omega\Delta t & \frac{\sin\omega\Delta t}{\omega} \\ 0 & -\omega\sin\omega\Delta t & \cos\omega\Delta t \end{bmatrix} \quad (3.19)$$

3.1.3 State Prediction

The state prediction step seeks to predict the model's future behaviour based on the state transition matrix Φ :

$$\hat{x}_{k+1|k} = \Phi x_k \quad (3.20)$$

Additionally, the estimated covariance matrix P may be predicted for next time step $k+1$:

$$P_{(k+1|k)} = \Phi P_{(k|k)} \Phi^T + Q \quad (3.21)$$

3.1.4 Correction Measurement Update

After the predictions are done, measurements of the systems actual observations occur as described in equation 2 of 3.2. After an observation has occurred a Kalman gain is calculated, this value is used as weighting between measurement and estimates of the system itself. The Kalman gain is calculated by:

$$K_k = P_{(k|k-1)} H^T (H P_{(k|k-1)} H^T + R_k)^{-1} \quad (3.22)$$

After the Kalman gain is calculated for timestep k a state update sequence of the Kalman filter is done by comparing the predicted state matrix from equation 3.15 to a state observation:

$$y_k = H x_k \quad (3.23)$$

The state update equation, therefore, uses the previous prediction with a Kalman gain weighting of accuracy related to the difference of prediction and measurement

$$\hat{x}_{k|k} = \hat{x}_{(k|k-1)} + K_k (y_k - H \hat{x}_{(k|k-1)}) \quad (3.24)$$

Thereafter the covariance matrix P requires an update such that useful information with regards to each measurement is updated for the current state of the system.

$$P_{k|k} = (I - K_k H) P_{(k|k-1)} (I - K_k H)^T + K_k R_k K_k^T \quad (3.25)$$

The state prediction and state measurement update calculations are performed recursively till the end of the tracked objects trajectory.

3.2 Interacting Multiple Model

The Multiple Model (IMM) Approach assumes that the system of a tracked object obeys a finite set of linear/nonlinear modes of operation [36]. The process can be defined as a discrete set of differing system dynamics [37] throughout its trajectory. This can be viewed as a bank of parallel state estimators which each produce a state hypothesis for each time step. The management of which filter model is used during the object's trajectory is governed by an underlying Markov chain [46]. The IMM algorithm is a method of combining these hypotheses

to form an improved combined state estimate of targets with dynamics that can be segmented. The IMM has been stated to be an ideal suboptimal Pseudo-Bayes state estimation algorithm with low computational cost having historical success in its application to trajectory prediction [38].

The IMM algorithm steps are seen in *figure 3-2*, there are three distinct steps required for a combined estimate output from the system. An initial step of state interaction considers mode probabilities to produce relative state estimates and covariances. Each of the N Kalman filters then produce state estimates $\hat{X}^N(k)$ and an additional mode likelihood output Δ_k^N . The mode likelihood output is used to update the probability of which of the N filters is most likely to be relevant to the state observed at time k . Finally, the state combination produces a combined hypothesis based on a combined filter output $\hat{X}(k)$.

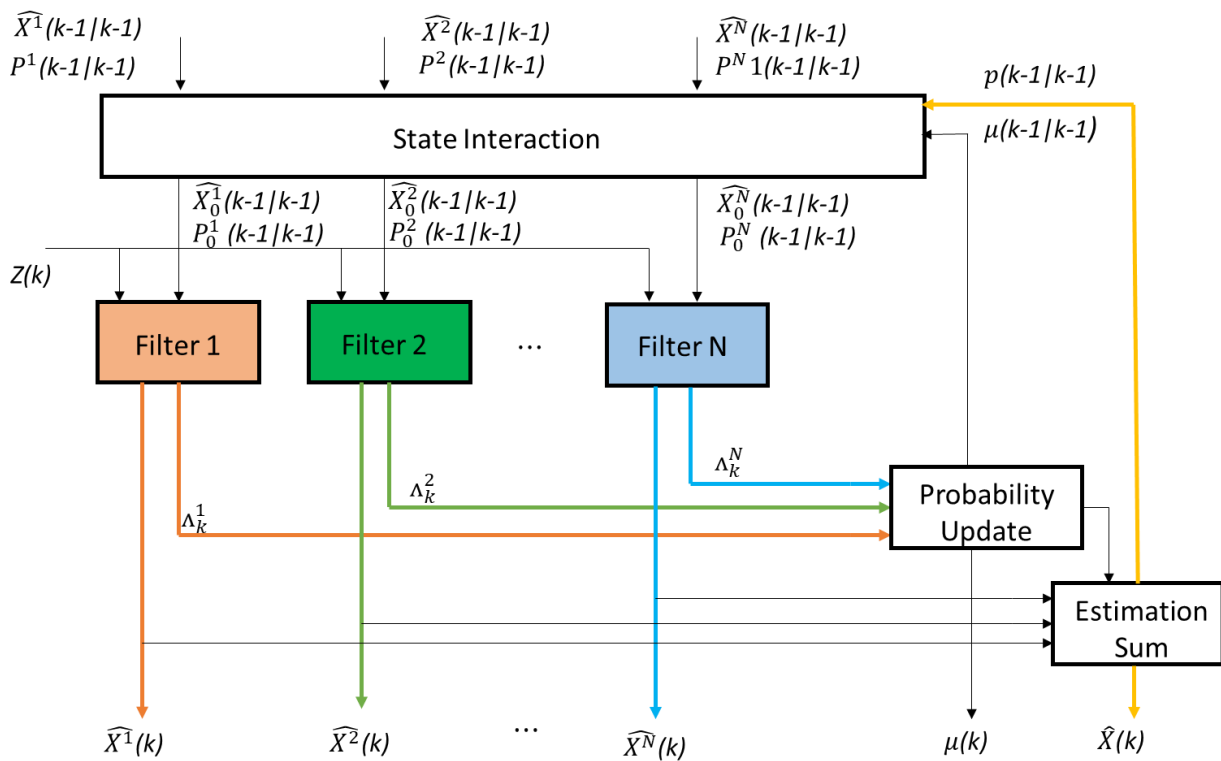


Figure 3-2: Interacting Multiple Model description

3.2.1 State interaction

Prior to any state updates via the bank of Kalman filters the model state vectors estimates $\hat{X}^N_{(k-1|k-1)}$ are combined this is done by summing the initial states \hat{X}^i up to N for j filter modes with the conditional model probabilities $\tilde{\mu}^{ij}$ of the previous state

$$\hat{X}_0^J = \sum_{i=1}^N \hat{X}^i \tilde{\mu}^{ij} \quad (3.26)$$

Additionally, the covariance matrices $P_{0(k-1|k-1)}^N$ are combined by summing the product of $\tilde{\mu}^{ij}$ and the difference of the selected state vector \hat{X}^i and the combined state vector for the j^{th} filter model \hat{X}^{0j} to find the combined covariance matrix.

$$\hat{P}_0^J = \sum_{i=1}^N \tilde{\mu}^{ij} [\hat{P}^i + (\hat{X}^i - \hat{X}^{0j})(\hat{X}^i - \hat{X}^{0j})^T] \quad (3.27)$$

The conditional model probability $\tilde{\mu}^{ij}$ is calculated by considering the specific elements of the switching matrix that models the probability of switching between modes as a Markov Decision Process (MDP) of an IMM with 3 modes seen in figure ... The probability of switching between model i to model j is defined as p^{ij}

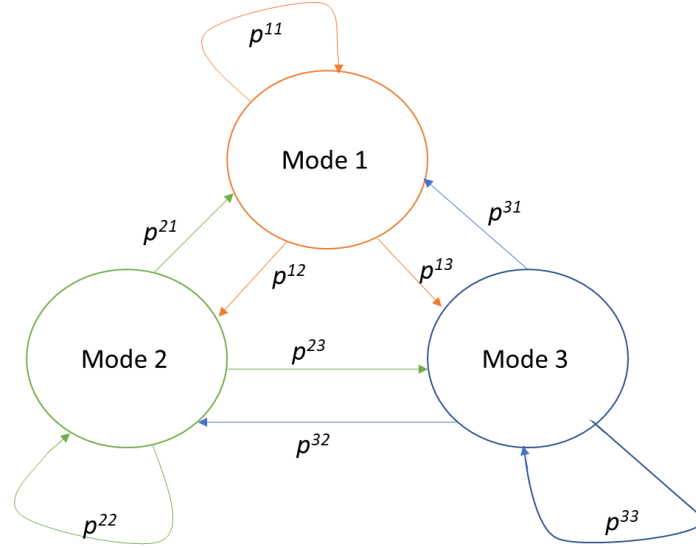


Figure 3-3: Markov Decision Process of IMM mode switching

The switching matrix Π is, therefore:

$$\Pi = \begin{bmatrix} p^{11} & p^{12} & p^{13} \\ p^{21} & p^{22} & p^{23} \\ p^{31} & p^{32} & p^{33} \end{bmatrix} \quad (3.28)$$

With $\hat{\mu}^i$ being the mode selection probability, which is initialized as equal, as each mode could be correct when tracking begins without prior knowledge. The conditional model probability is calculated as:

$$\begin{aligned}\hat{\mu}^{ij} &= \frac{1}{\bar{\psi}^j} p^{ij} \hat{\mu}^i \\ \bar{\psi}^j &= \sum_{i=1}^N p^{ij} \mu^i\end{aligned}\tag{3.29}$$

Where $\bar{\psi}^j$ is a normalization vector to ensure that the total model probability is 1. After the state interaction of each mode is complete, each filter produces a state estimation. The probability of which filter is the most accurate is determined.

3.2.2 Model Probability Update

The likelihood of each model Λ^j is computed by using innovations Z^j during each state update, the innovation covariance matrix S^j computed within the Kalman gain of each filter. Gaussian distribution is assumed for the likelihood of each mode is the correct therefore

$$\Lambda^j = \frac{1}{\sqrt{|2\pi\tilde{S}^j|}} \exp \left[-0.5(Z^j)^T (\tilde{S}^j)^{-1} (Z^j) \right]\tag{3.30}$$

Innovations for each filter is calculated by:

$$Z^j = m_o - \tilde{m}^j\tag{3.31}$$

Where m_o is a vector of observations for the time step update and \tilde{m}^j is the predicted track state for filter model j transformed into the correct form of observations using the observation matrix H . The innovation covariance matrix S^j is calculated by using the predicted covariance matrix \hat{P}_0^j and summing the measurement noise variance matrix R

$$\tilde{S}^j = H^j P^{0j} (H^j)^T + R\tag{3.32}$$

Once the likelihood of each mode has been updated the new model probabilities may be calculated:

$$\begin{aligned}\hat{\mu}^j &= \frac{1}{c} \Lambda^j c^j \\ c &= \sum_{i=1}^N \Lambda^i c^i\end{aligned}\tag{3.33}$$

Where $\hat{\mu}^j$ is the filtered model probability for model j and c is a normalization constant to maintain the summation of mode probabilities to 1. The model probabilities are updated after

each cycle to ensure the most relative model is used for following update. States and covariance are then combined to form a combined filtered output of the IMM.

3.2.3 State Estimation Combination

When combining the state estimates and covariance matrices, the weighting of each model output is determined by the updated model probabilities.

$$\hat{\mathbf{X}} = \sum_{i=1}^N \mathbf{X}^i \hat{\mu}^i \quad (3.34)$$

The combined IMM filtered output is $\hat{\mathbf{X}}$ and \mathbf{X}^i is filtered output from each of the models. Finally, the covariance matrix is combined.

$$\hat{\mathbf{P}} = \sum_{i=1}^N \hat{\mu}^i \left[\hat{\mathbf{P}}^i + (\hat{\mathbf{X}}^i - \hat{\mathbf{X}})(\hat{\mathbf{X}}^i - \hat{\mathbf{X}})^T \right] \quad (3.35)$$

This process is executed recursively throughout the objects motion to compute state interactions, probability updates and estimation combinations

3.3 Separation volume

A collision between aircraft is defined as a loss of minimum separation between the IPZ. For manned aircraft, this separation volume is 5NM (9.26 km) horizontally and 1000ft vertically (304.8m). This volume however is not practical when considering dense airspace with UAVs flying closer together, research by [69] suggests the use of separation minima as defined by Eurocontrol for Remote Piloted Aircraft Systems (RPAS) within specification RPA11 [68]. This specification states that “a Remote Pilot should, except for aerodrome operations, maintain a minimum distance of 0.5 NM (926 m) horizontally or 500 ft (152.4m) vertically between their RPA and other airspace users, regardless of how the conflicting traffic was detected and irrespective of whether or not he was prompted by a Sense and Avoid system”. Thus, this minimum separation volume is implemented for this study

A collision occurs when:

$$\|r_o - r_i\| < d_{separation} \quad (3.36)$$

Where r_o and r_i are the positional vectors of the owner and intruder respectively, $d_{separation}$ is the minimum separation. A conflict scenario is seen in *figure 3-4*

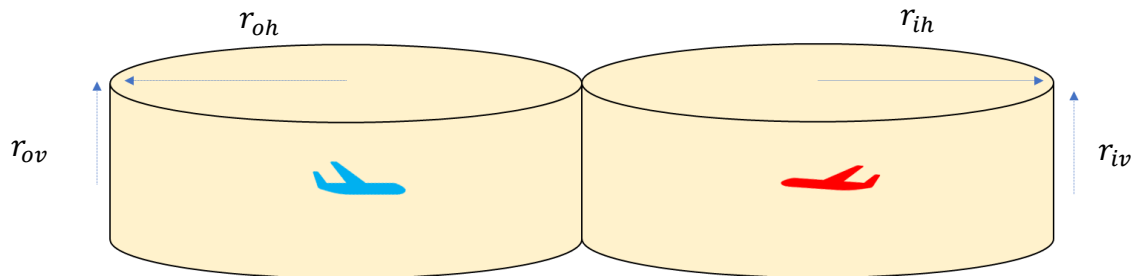


Figure 3-4: Separation volume intersection creating a collision scenario

Chapter 4: Implementation

The implementation of the collision prediction system is divided into subsystems which include:

- Acquiring and decoding ADS-B data for each flight detected.
- Building state prediction models based on linear state dynamics and combining linear models with an IMM.
- Simulation of static and dynamic intruders on the flight path that breaches the defined IPZ to determine the time to a collision.

Each of these subsystems is detailed within this chapter

4.1 ADS-B Data acquisition

A simplified version of the data acquisition system is seen in *figure 4-1*, the periodically broadcast ADS-B signal of surrounding aircraft is received via a telescopic dipole antenna connected to the SDR receiver. The SDR communicates with the Raspberry PI for radio wave access, decoding is done via Dump1090, a Mode S ES decoder, operating on PiAware client software. The decoded ADS-B data is pushed to an ADS-B database after each read cycle, this information is made available on port 8080 of the Raspberry Pi's local IP address. The decoded data which is shared on port 8080 is pulled via an HTTP read within MATLAB. The port is periodically read at a rate of 1Hz to flight data contained in each time frame. This data is converted into appropriate units and is stored as a flight object for each unique ICAO callsign. Each flight object contains information regarding the aircraft's real measured trajectory and propagation of uncertainty values regarding position and velocity of measurement.

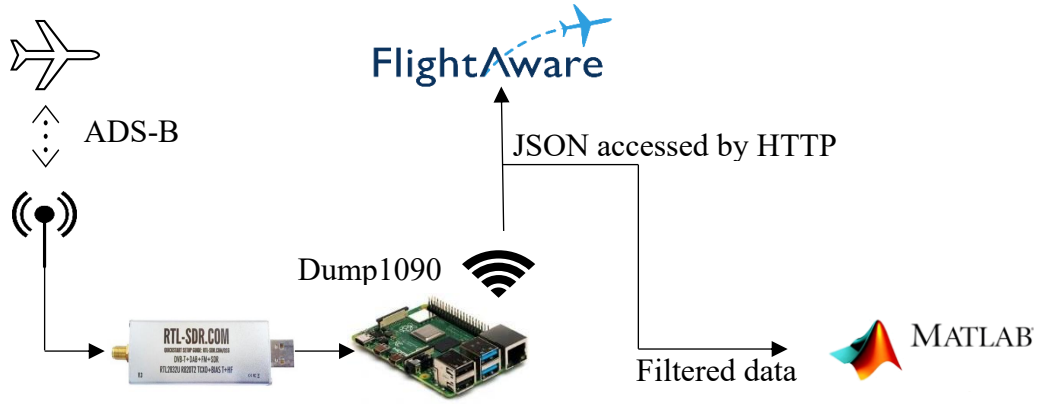


Figure 4-1: ADS-B data acquisition system

4.1.1 RTL SDR receiver

To acquire ADS-B data, an RTL SDR receiver is required to access, record, and extract useful information from the incoming surrounding aircraft's ADS-B OUT 1090 MHz frequency. The selected receiver used for this research, seen in *figure 4-2* makes use of an RTL2832U Analog to Digital Converter (ADC) with technical specifications seen in *Table 4-1*. To access the 1090 MHz frequency an appropriate antenna is required, a telescopic halfwave dipole antenna has is used to adjust to the correct effective length. The effective length of 137.614 mm (68.8 mm per leg) is calculated by equation 4.1. The antenna is mounted as high as possible (3m) to ensure that received radio signals are not interrupted by surrounding obstructions.



Figure 4-2: R820T2 RTL2832U SDR receiver

Table 4-1: Technical specifications for RTL-SDR receiver

Bandwidth	ADC Resolution	Frequency Range	Input Impedance	Current Draw
2.4 MHz	8-bit	500 kHz – 1766 MHz	50Ω	270 – 280 mA

$$L = \frac{c}{2f} \quad (4.1)$$

4.1.2 Raspberry Pi

The RTL SDR raw data would need to be decoded to obtain useful information required for trajectory prediction, this process is done by using a Raspberry Pi 4B (4 GB) seen in *figure 4-2*. The Raspberry Pi is an inexpensive rapid prototyping computer capable of running its own Operating System (OS) such as Linux with I/O such as USB ports, Ethernet input, GPIO and

HDMI out. The RTL-SDR receiver communicates with the Raspberry Pi via the onboard USB 3.0 port.



Figure 4-3: Raspberry Pi 4 Single Board Computer

4.1.3 PiAware - Dump1090

Dump1090 is an open-source command-line Mode-S ES decoder specifically designed for RTL-SDR devices. Written in C it is an optimized method of decoding raw data received into useful flight information for trajectory prediction. The version of Dump1090 used is Dump1090-fa [20], the PiAware platform which is built as a Debian package that functions as a client to the FlightAware website. The Raspberry Pi continuously uploads data to the FlightAware server via the onboard Wi-Fi module. FlightAware is a flight tracking and data analytics company that allows individuals to feed real-time flight ADS-B data to their data repository.

4.1.4 Decoded data access

The PiAware client ADS-B data is accessible as a JavaScript Object Notation (JSON) file via HTTP (Hypertext Transfer Protocol) from the Raspberry Pi. The JSON file contains the flight information stored as a struct which contains all reachable aircraft ADS-B data. Flight data is accessed by periodically reading port 8080 of the Raspberry Pi's local IP address <http://localhost:8080/data/aircraft.json> which is read at 1Hz.

4.1.5 Data Extraction

Not all ADS-B data that is emitted from each aircraft is useable for predicting flight trajectories, thus incoming data requires filtration. The fields which are deemed important for input to the system dynamics models are seen in *Table 4-2*. Only ADS-B data that contain these 13 fields are stored as a flight table of occurrence for experimentation with prediction algorithms.

Table 4-2 : filtering fields required for Aircraft trajectory prediction

Field	Description	Data Type
Flight Identification	Hexadecimal 24-bit ICAO identifier of the aircraft, as 6 hex digits	String
Flight Callsign	Flight name/aircraft registration	Character Array
Geometric altitude	Feet (GNSS / INS) ref. WGS84 ellipsoid	Double
Ground speed	Knots	Double
Track heading	Heading in degrees from magnetic north 0° - 359°	Double
Track rate	Rate of change of heading in degrees/second	Double
Geometric altitude rate of change	Feet/min	Integer
Latitude	degrees	Double
Longitude	degrees	Double
NACp	Navigation Accuracy Category for Position	Integer
NACv	Navigation Accuracy Category for velocity	Integer
Seen	Time since last message was received	Integer
Version	ADS-B version number 0,1,2	Integer

4.1.6 Data conversion

As the incoming data in in terms of feet, knots and minutes a conversion to SI units is required to input data into a more useable frame of reference. The geometric altitude, longitude and latitude of the aircraft is converted to an Earth Centred Earth Fixed (ECEF) coordinate frame of reference based on the World Geodetic System of 1984 (WGS84) ellipsoidal model. Additionally, Ground speed is decomposed into its x and y components of velocity (equations 4.6 and 4.7) by virtue of the aircraft heading θ (0-360 from true north) as seen in figure 4-4.

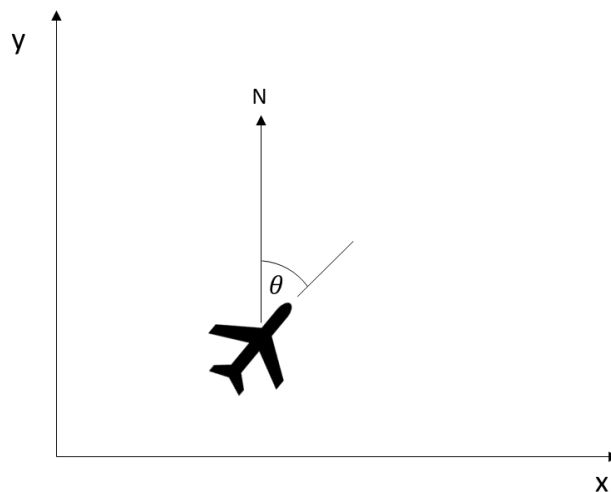


Figure 4-4: relative track angle of aircraft

$$V_x = V \sin(\theta) \quad (4.2)$$

$$V_y = V \cos(\theta) \quad (4.3)$$

The velocity of aircraft movements within the z dimension is extracted from the geometric altitude rate of change. The filtered and converted data is then stored as a flight object which contains the flights the required information to construct a state vector of position and velocity in 3 dimensions.

4.2 Trajectory Prediction Models

To predict the trajectory of an aircraft moving with linear dynamics, 3 motion models were considered for testing, the first model Constant Velocity (CV) considers the aircraft moving at a constant speed in the X, Y and Z-axis. The second model Constant Acceleration (CA) considers the aircraft manoeuvring about all axes. Two variations of turn models are constructed, a Coordinated Turn (CT) 2D which assumes the aircraft executes turns at a constant angular rate of change about the Z-axis and a CT 3D model in which turns are executed about all axes

Variations of these models are combined using an IMM. The IMM variations are tested to determine which produces the most accurate predictions as described in **section 4.4**.

4.2.1 Constant Velocity Model

The CV model assumes that the aircraft would be moving at a constant velocity with no manoeuvres occurring in either axis x, y, and z. This model would best allow for predicting aircraft trajectories when at cruising altitude. The state vector for the CV:

$$x_{CV} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]' \quad (4.4)$$

With the state transition function Φ_{CV} obtained from standard relation of displacement to constant velocity:

$$s = \dot{s}t \quad (4.5)$$

As motion in the axes are orthogonal the transition function is:

$$\Phi_{CV} = \begin{bmatrix} I_3 & I_3 \cdot \Delta t \\ 0_3 & I_3 \end{bmatrix} \quad (4.6)$$

As state positions and velocities accessible from the decoded ADS-B signal the measurement matrix H is defined as:

$$H_{CV} = I_6 \quad (4.7)$$

Initial Covariance matrix PCV is configured with arbitrary high initial values for both velocity and position as the system is initialised only with information of the initial position before tracking begins.

$$PCV = I_6 \cdot 200 \quad (4.8)$$

Where I_6 is a (6x6) identity matrix.

The process noise matrix QCV is calculated as:

$$QCV = \begin{bmatrix} I_3 \cdot \frac{\Delta t^4}{4} & I_3 \cdot \frac{\Delta t^3}{2} \\ I_3 \cdot \frac{\Delta t^3}{2} & I_3 \cdot \Delta t^2 \end{bmatrix} \cdot \sigma_a^2 \quad (4.9)$$

where I_3 is a (3x3) identity matrix and σ_a is random variation in acceleration.

The measurement covariance matrix is derived from uncertainty measurements obtained from NACp and NACv values as identified within **section 2.3**. It is assumed that these values do not fluctuate throughout the flight cycle and that initial values are consistent throughout the flight segment. Assuming that the measurement error in each position is independent of the errors within the velocity measurement results in:

$$R = \begin{bmatrix} \sigma_{x_m}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{x}_m}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{y_m}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}_m}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{z_m}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{z}_m}^2 \end{bmatrix} \quad (4.10)$$

Once these matrices are known the CV prediction model is constructed using the Kalman filtering process as defined in section

4.2.2 Constant Acceleration Model

The constant acceleration model attempts to predict an aircraft's trajectory when manoeuvres are required in either a change of direction or a change in altitude. This model would be used to predict the aircrafts position and velocity when under non constant velocity. The state vector is:

$$x_{CA} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \ddot{x} \quad \ddot{y} \quad \ddot{z}]' \quad (4.11)$$

With a state transition matrix Φ_{CA} obtained from the relation of displacement and velocity as a function of constant acceleration.

$$\begin{aligned} s &= \dot{s}t + \frac{1}{2}\ddot{s}t^2 \\ \dot{s} &= \ddot{s}t \end{aligned} \quad (4.12)$$

As motion in the axes are orthogonal the transition function is:

$$\Phi CA = \begin{bmatrix} I_3 & I_3 \cdot \Delta t & I_3 \cdot \frac{\Delta t^2}{2} \\ 0_3 & I_3 & I_3 \cdot \Delta t \\ 0_3 & 0_3 & I_3 \end{bmatrix} \quad (4.13)$$

The Constant acceleration model requires a different measurement matrix HCA as the state vector is equation 4.12.

$$HCA = \begin{bmatrix} I_3 & 0_3 & 0_3 \\ 0_3 & I_3 & 0_3 \end{bmatrix} \quad (4.14)$$

The Process covariance matrix is set similarly to equation 4.10 as an arbitrarily high value of 200 for each diagonal entry (variance) as per matrix (3.14) as it is assumed that positional, velocity and acceleration estimation errors are not correlated.

$$PCA = 200 \cdot I_9 \quad (4.15)$$

Where I_9 is a (9x9) identity matrix.

The measurement noise matrix R is the same as that used within the CV filter, described by equation 4.11. The process noise matrix QCA is obtained by equation 3.13 and relates process noise to random acceleration measurements σ_a

$$QCA = \begin{bmatrix} I_3 \cdot \frac{\Delta t^4}{4} & I_3 \cdot \frac{\Delta t^3}{2} & I_3 \cdot \frac{\Delta t^2}{2} \\ I_3 \cdot \frac{\Delta t^3}{2} & I_3 \cdot \Delta t^2 & I_3 \cdot \Delta t \\ I_3 \cdot \frac{\Delta t^2}{2} & I_3 \cdot \Delta t & I_3 \end{bmatrix} \cdot \sigma_a \quad (4.16)$$

4.2.3 Constant Speed Coordinate Turn 3-D

This model describes the motion of an aircraft moving in a circular trajectory in the X, Y and Z axes. The state matrix, measurement matrix, measurement noise matrix and process noise matrix are the same as those used within the CA model. By adapting the transition matrix described 3.19 for 3 dimensions the CT-3D transition matrix $\Phi CT3D$ is defined as:

$$\Phi CT3D = \begin{bmatrix} I_3 & \frac{\sin(\omega \Delta t)}{\omega} \cdot I_3 & \frac{1 - \cos(\omega \Delta t)}{\omega^2} \cdot I_3 \\ 0_3 & \cos(\omega \Delta t) \cdot I_3 & \frac{\sin(\omega \Delta t)}{\omega} \cdot I_3 \\ 0_3 & -\omega \sin(\omega \Delta t) \cdot I_3 & \cos(\omega \Delta t) \cdot I_3 \end{bmatrix} \quad (4.17)$$

Where I_3 is the (3x3) identity matrix and 0_3 is a (3x3) zero matrix,

4.2.4 Constant Speed Coordinate Turn 2-D

This model describes the motion of an aircraft moving in a circular trajectory about the Z-axis such that a yaw change occurs whilst constant altitude is maintained. The state matrix, measurement matrix, measurement noise matrix and process noise matrix are the same as those described by the CV model. A variation of the state transition matrix 4.16 is used to define the state transition matrix Φ_{CT2D} :

$$\Phi_{CT2D} = \begin{bmatrix} 1 & 0 & 0 & \frac{\sin(\omega\Delta t)}{\omega} & \frac{\cos(\omega\Delta t) - 1}{\omega} & 0 \\ 0 & 1 & 0 & \frac{1 - \cos(\omega\Delta t)}{\omega} & \frac{\sin(\omega\Delta t)}{\omega} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & \cos(\omega\Delta t) & -\sin(\omega\Delta t) & 0 \\ 0 & 0 & 0 & \sin(\omega\Delta t) & \cos(\omega\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

4.2.5 Interacting Multiple Models Variations

The IMM models constructed for this research include:

- IMM-CV-CA fuses the modes of CV and CA to produce position predictions when an aircraft switches between non- manoeuvring and manoeuvring states.
- IMM-CV-CA-CT2D, which adds to the IMM-CV-CA model by adding a turn component about the Z-axis with a constant turn rate.
- IMM-CV-CA-CT3D, which adds tracking for circular motion models in X, Y and Z with a constant turn rate.

4.2.5.1 IMM CA-CV

The initial state probabilities of for the CV and CA modes are set to a 0.5 probability each such that initial state probability matrix is:

$$\hat{\mu}^0 = [0.5 \ 0.5] \quad (4.19)$$

The mode switching matrix π has the following mode switching probabilities:

$$\pi = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix} \quad (4.20)$$

With each entry p^{ij} describing the probability of moving between modes. The prediction and filtering process is completed as described in **section 3.2** the prediction models for both CV and CA predictors run in parallel forming a fused output estimation of the aircrafts position.

The combined state vector size is that of a 6x1 column vector, the same as CV mode, as such the combined state output does not output a predicted acceleration. This is done to ensure that the interaction step may combine states and covariance, the state and covariance vectors of the CV mode are padded with zeros to allow for matrix arithmetic. From testing, it was found that process noise values through experimentation it was found that the best process noise constants for CV and CA set to the specifications in **Table 4-3** (smaller process noise produced better results when combined in an IMM in comparison to single modes).

4.2.5.2 IMM CA-CV-CT2D

The IMM CA-CV-CT2D model adds a third dynamics model to the IMM-CA-CV predictor, the CT2D model expects an aircraft to follow a circular turn in the horizontal plane about the Z-axis. To maintaining a linear model a known turn rate is applied to the prediction algorithm as described in **section 4.24**. Since it is not known whether this turn may be applied in a clockwise or anti-clockwise direction a CT mode is created for either direction. The same transition matrix Φ_{CT2D} is used as defined in matrix 4.18 with each mode having a differing sign of turn rate ω . Through experimentation, it was found that large turn rates greater than 5 deg./s would cause overestimation with the aircraft overshooting the actual trajectory when moving within a CT mode. resulting in erroneous predictions. A turn rate of 2 deg./s is applied for both clockwise and anti-wise modes which cause more steps of the CT mode to be active but produces better results.

The initial state probabilities are set to 0.25 each such that the initial state probability is:

$$\hat{\mu}^0 = [0.25 \ 0.25 \ 0.25 \ 0.25] \quad (4.21)$$

With a mode switching matrix having the following probability entries:

$$\Pi = \begin{bmatrix} 0.6 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.15 & 0.15 \\ 0.15 & 0.3 & 0.4 & 0.15 \\ 0.15 & 0.3 & 0.15 & 0.4 \end{bmatrix} \quad (4.22)$$

Similarly, to the IMM-CV-CA predictor the mixed state produced is a 6x1 column vector which produces future positions and velocities. The IMM filtering process thereafter completed as described within **section 3.2**

4.2.5.3 IMM CA-CV-CT-3D

The IMM CA-CV-CT-3D prediction filter uses a 3D turn motion model as described by matrix 4.17 and extends on the functionality of the IMM CA-CV predictor. Similarly, to **section 4.2.5.2** modes are created for both Clockwise and Anti-Clockwise directions. An angular turn rate of 1.5 deg./s is used to construct the state transition function Φ_{CT3D} as per matrix 4.17. The initial state probabilities are set to

$$\hat{\mu}^0 = [0.25 \ 0.25 \ 0.25 \ 0.25] \quad (4.23)$$

With mode switching probabilities:

$$\Pi = \begin{bmatrix} 0.6 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.15 & 0.15 \\ 0.1 & 0.1 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix} \quad (4.24)$$

The prediction process conducted in the same manner as **section 4.2.5.2** with state vector interaction, probability updates and estimations combined.

4.2.6 Error evaluation

Error in positional predictions is evaluated by considering the Root Mean Squared Error (RMSE) per time step, this is calculated by equation 4.20. This serves as metric when evaluating the positional prediction output.

$$RMSE = \sqrt{\sum_{i=1}^N \frac{1}{N} ((\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2 + (\tilde{z}_i - z_i)^2)} \quad (4.25)$$

Where \tilde{x}_i , \tilde{y}_i and \tilde{z}_i are predicted position. RMSE is used as a metric of accuracy for each of the prediction algorithm's output.

4.3 Collision Detection

The IPZ of an owner aircraft is defined as a cylindrical volume surrounding a point in space. This is modelled within a MATLAB simulation as a collision cylinder centred at a specified position. By assigning the length and radius of the cylinder as the separation volume is generated as a convex mesh geometry as seen in *figure 4-5* is created.

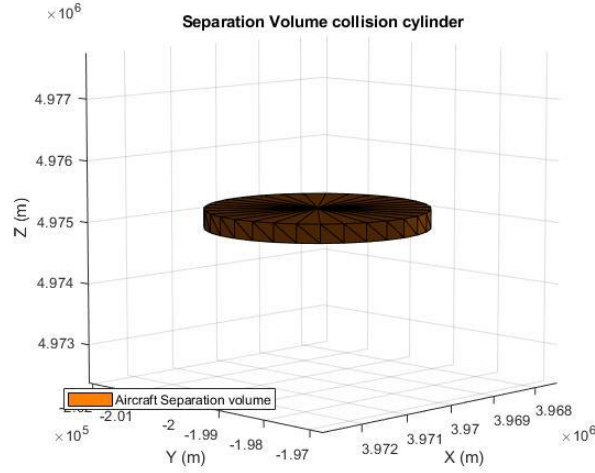


Figure 4-5: Aircraft separation volume modelled as a collision cylinder

This mesh is transformed by an offset in the X, Y and Z axes, to model the movement of an aircraft. A collision cylinder is used to model the predicted position output by a prediction algorithm, which is transformed by a predicted offset output at each Timestep during program runtime. In the case of a static intruder collision zone test, a collision cylinder is positioned at the specific coordinate with a prediction cylinder propagated towards it. When simulating a dynamic collision, two collision cylinders are propagated at every time step as seen in *Figure 4-6*. The shortest distance between the two collision cylinders is measured at each time step and a Boolean condition is set if an intersection of the two geometries occurs which is deemed to be a collision.

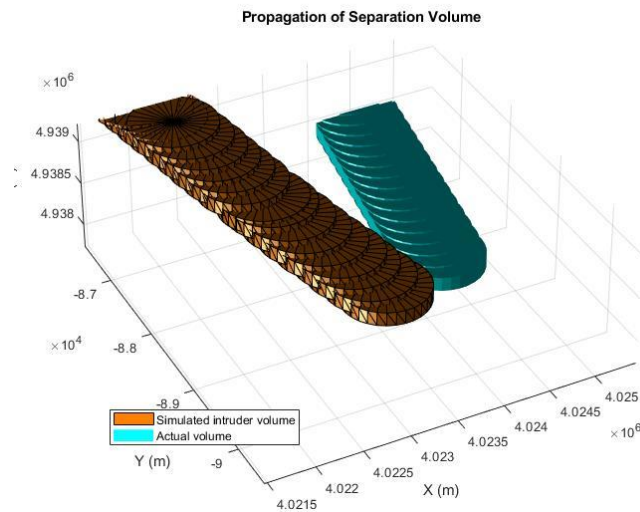


Figure 4-6: Separation volume propagation for a dynamic intruder

To account for stochasticity in real environments such as atmospheric disturbances and weather conditions, the size of the IPZ of the predicted state is augmented such that the size is seen in

Figure 4-7. This is the size of separation volume that is propagated for the trajectory owner when predicting collisions. The intruder IPZ however is not augmented for the testing simulation.

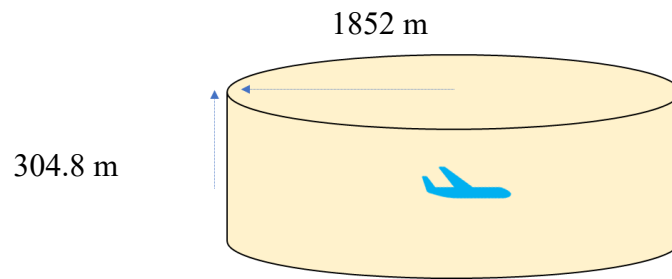


Figure 4-7: Separation volume used for trajectory owner

4.3.1 System overview

A simplified flow diagram of the trajectory prediction system is seen in *figure 4-8*. The ADS-B information received requires decoding as outlined in **section 4.1**, the recorded decoded flight information is then saved within a database of single flight trajectories. Longitude, Latitude and Altitude are converted into an ECEF based co-ordinate system with velocities making up a state vector for prediction purposes, additionally the uncertainty in position and velocity is recorded during the start of each flight's trajectory.

The state vector and measurement uncertainty are used as an input into a prediction algorithm as mentioned in **section 4.2** this allows for trajectory predictions to be made at Time step (T_s) intervals for the aircraft, which is recursively corrected by means of the Kalman filtering process. The position output for each future T_s is used to transform the location of a separation volume surrounding the aircraft (owner) which is modelled as a collision cylinder geometry within the MATLAB simulation. The collision cylinder is transformed/propagated to the predicted location of the aircraft output at each timestep of the prediction algorithm. If the owner surrounding aircraft IPZ is found to intersect with an intruder IPZ this is detected as a collision and the time to collision at the closest point of approach is recorded.

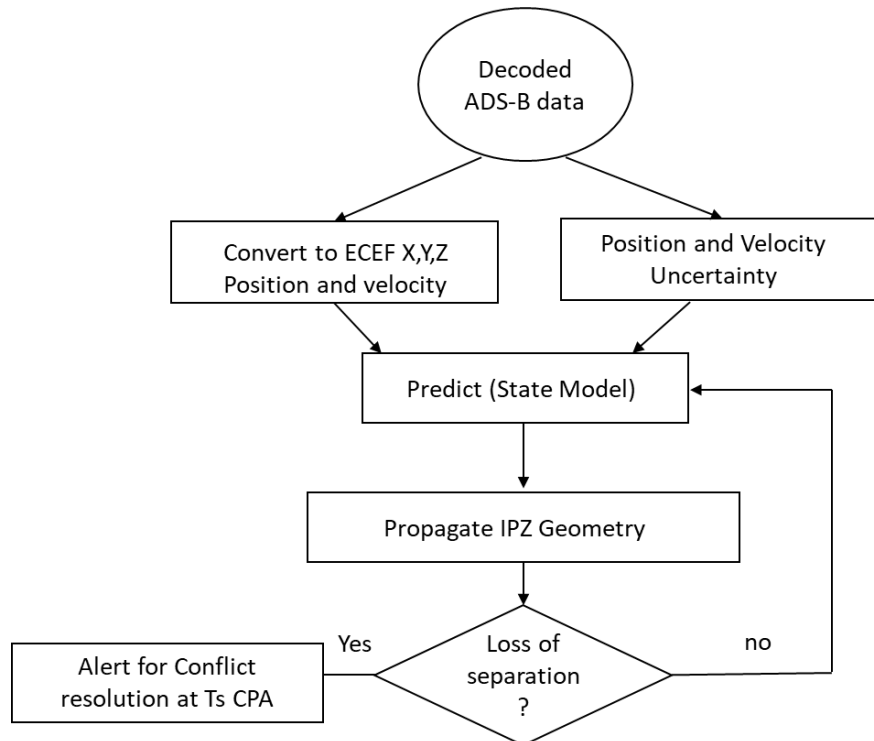


Figure 4-8: Simplified Flow diagram of collision detection

4.4 Assumptions

When Testing the Collision prediction system in MATLAB the following assumptions were made:

- All ADS-B data received from the SDR is accurate as reported, no spoofed or erroneous data was received.
- ADS-B data is accessible at 1Hz.
- ADS-B received is Version 2 compatible.
- Data produced by the dump1090 decoder produces negligible errors without propagation delay affecting the received flight data
- There are no signal interferences such as Electro-Magnetic Interferences (EMI) influencing the recorded data.
- Aircraft ADS-B OUT signals propagate at the speed of light in a vacuum.
- The initial values recorded per flight object positional (NACp), and velocity (NACv) uncertainties do not change during the received flight recordings.
- Measurement errors between all positions and velocities are uncorrelated.
- Intruder aircraft begins its trajectory at the same time as the real tracked aircraft.
- The prediction algorithms know the tracked flights initial coordinates (X_0 , Y_0 , Z_0) before prediction.
- When testing the prediction to collision case the IPZ owner aircraft does not know the intruder aircraft trajectory thus state variables are initialized to 0.
- The intruder aircraft does not perform any quick manoeuvring actions during the simulated loss of separation and follows a Constant Velocity during simulation.

4.5 System Testing

The collision prediction system is separated and validated in sections, firstly, Kalman prediction methods are evaluated with regards to their trajectory prediction accuracy. Flight trajectories are selected to test the adaptability of the IMM systems to manage generalized flight dynamics. An appropriate Time Step (T_s) is selected for future state predictions based on the TCAS RA lower bounds of 15s thereafter RMSE outputs are compared. Once prediction methods are validated the collision detection system is tested. Flights Trajectory Change Points (TCPs) are selected to test the capability of the various prediction systems ability to detect a definite collision as a flight's IPZ is propagated through an ECEF frame of coordinates. Flight trajectories used for testing are selected from a saved database of recorded flights over a 6-hour period.

4.5.1 Prediction system Testing

To test the accuracy of predictors as introduced in **section 4.2**, 4 unique flight trajectories are selected with the following ICAO callsign identifiers:

- EZY89TP
- DAL84
- VLG48NC
- EXS1204

These specific trajectories are selected due to their differing dynamics, to test the accuracy and adaptability of predictor algorithms. The predictor systems which are tested include a CV Kalman filter, a CA Kalman Filter, an IMM with CV-CA data fused (IMM-CV-CA), an IMM with CV-CA modes including a CT within 2 dimensions about the Z-axis (IMM-CV-CA-CT2D), and finally an IMM CV-CA with CT about X, Y and Z dimensions (IMM-CV-CA-CT3D). Each of the 4 selected flight trajectories are tested with look-ahead Time Step (Ts) of 15 seconds for each prediction algorithm. Predictors are evaluated by measuring the RMSE of predicted position vs the actual position measured for the timestep (k+1). A plot of Root Mean Squared Error (RMSE) vs time is produced to evaluate the predicted position vs actual position during the recorded flight trajectory.

After calculating the RMSE of each test flight the RMSEs are averaged to determine which filter produces the most accurate estimations of position. The initial process noise acceleration and process noise jerk constants were selected for testing are seen in Table 4-3. These values are used as starting points and were tuned during testing.

Table 4-3: Initial Process noise covariance constants

Component	Value
Process noise Acceleration constant CV	$15 (m/s^2)^2$
Process noise Acceleration constant CA	$10 (m/s^3)^2$

4.5.2 Collision detection system testing

The collision system is evaluated for two separate cases: the first case seeks to test the system's ability to detect a collision occurring with static collision zones; the second seeks to test the system with a moving intruder flight, which has an intersecting trajectory with a test flight.

The time difference of prediction is used to evaluate the effectiveness of collision predictions this can be expressed as

$$T_d = T_{col} - T_{pred} \quad (4.23)$$

Where T_{col} is the observed time to collision, when propagating the IPZ of the trajectory owner vehicle towards an intruder IPZ and T_{pred} is the predicted time to collision when using a predictor. A positive result for T_d indicates the collision is predicted ahead of time in the simulation, with a negative result indicating the collision was detected after the actual observed time. If the collision is not detected by the system a penalty of -35 seconds is applied as T_d (the upper limit of TCAS RA). This is done to properly evaluate the lookahead timestep (Δt) for each prediction algorithm.

4.5.2.1 Static Collision prediction testing

The static collision zones are positioned at points of interest such as Trajectory Change Points (TCPs) of the flight path to test the predictors' ability in collision detection when a flight is switching between dynamics modes. The collision zones are offset in the X and Y axes whilst maintaining the same altitude as the real flight trajectory. The actual collision time is determined by propagating the true flight trajectory's Intruder Protected Zone (IPZ) through the ECEF frame at 1 second intervals and recording the time of CPA between the two collision volumes. The IMM variations are tested by producing predicted times to a collision with varying prediction time steps (Ts). The flights with the following ICAO callsigns are selected for static collision prediction testing:

- EXS8L8, seen in *figure 4-7*
- EZY89TP, seen in *figure 4-8*
- RYR56UE, seen in *figure 4-9*
- RYR8213, seen in *figure 4-10*

Each owner trajectory is seen in blue with intruder conflict zones shown in red.

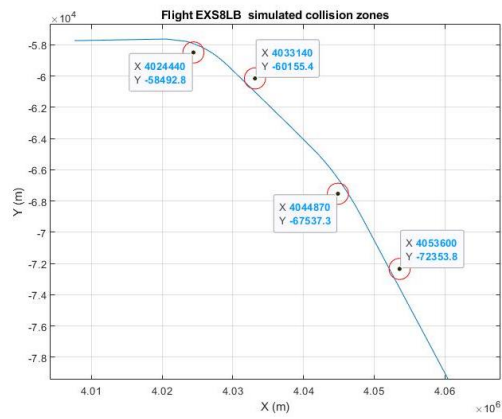


Figure 4-9: Flight EXS8L8 static collision points

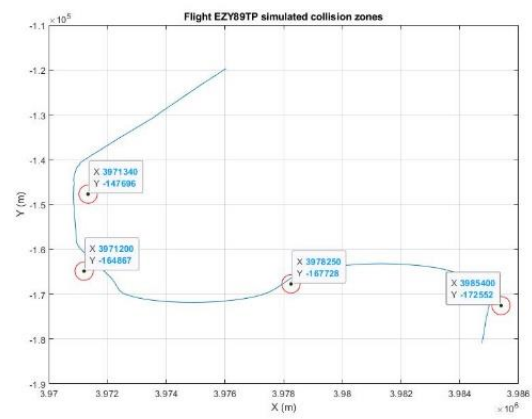


Figure 4-10: Flight EYZ89TP static collision points

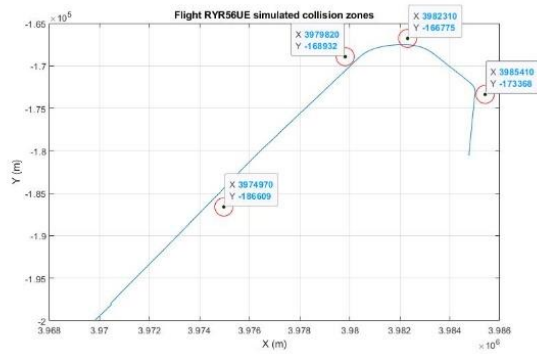


Figure 4-11: Flight RYR56UE static collision points

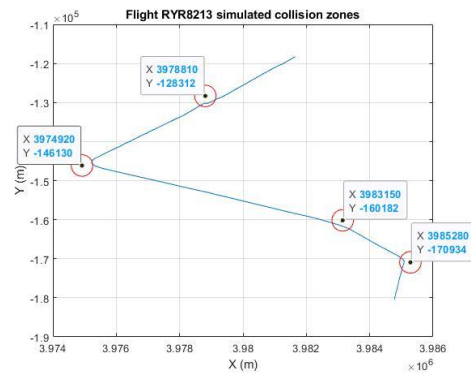


Figure 4-12: Flight RYR8213 static collision points

4.5.2.2 Dynamic intruder collision test

The dynamic collision simulation tests the viability of detecting a collision with a target moving with Constant Velocity (CV) towards the real flight trajectory. The flight callsign VLG8LV is selected to test a simulated intruder aircraft. The simulated collision has specifications detailed in *Table 4-4*. The simulated collision is seen in *figure 4-11* with the ownership trajectory being flight VLG8LV in blue and the simulated intruder aircraft seen in red.

Table 4-4: Simulated aircraft parameters

Simulation Parameters	
Component	Value
Flight initial Speed	193.63 m/s
Intruder Speed	185.093 m/s
Flight initial position	X: 4020820.59 Y: -75090.97 Z: 4939240.11
Intruder initial position	X: 4028820.59 Y: -77090.97 Z: 4939240.11
Flight initial heading (linear)	276.4
Intruder heading (linear)	86.6
Flight initial vertical speed	14.14 m/s
Intruder vertical speed	-14.24 m/s
CPA within simulation	X: 4026720.56 Y: -93853.82 Z: 4938110.23

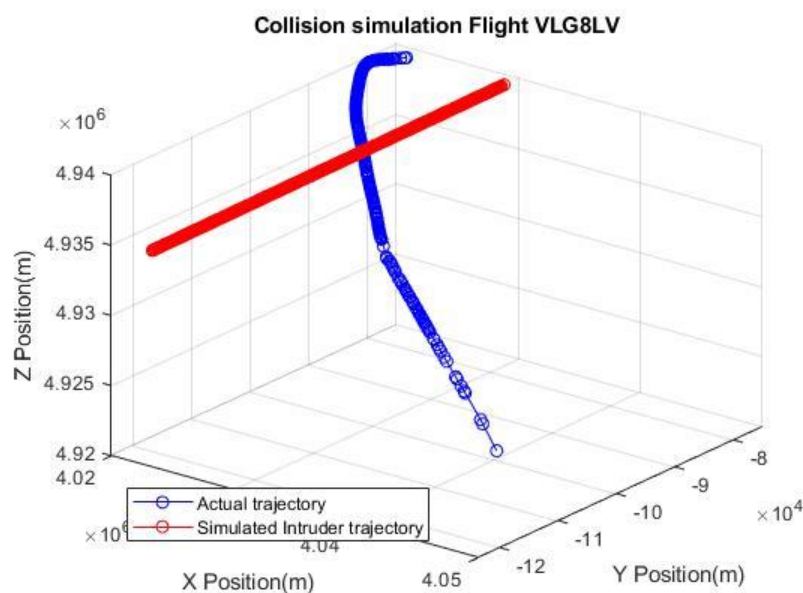


Figure 4-13: Simulated collision with dynamic intruder

Chapter 5: Results

5.1 Flight data capture

The recorded flight data used for testing was recorded on 11/08/2021 during a 6-hour period sampled at 1 second intervals which resulted in 279 unique flight trajectories. The signal directionality strength is seen in figure 5-1 output from FlightAware's data feeder statistics.

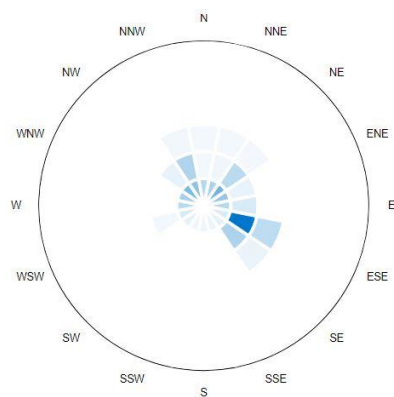


Figure 5-1: SDR directionality strength during data capture

A flight coverage map is seen in figure 5-2 the radius of coverage is seen centred at the location of the SDR additionally range statistics based on several messages received seen in figure 5-3. Very few messages received are beyond the 100-mile radius from the receiver, which is limited by the signal to noise ratio of the antenna used for this experiment.

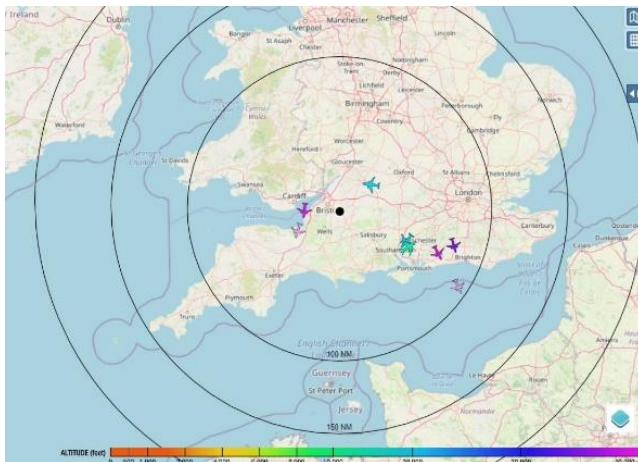


Figure 5-2: Coverage Map of incoming ADS-B information

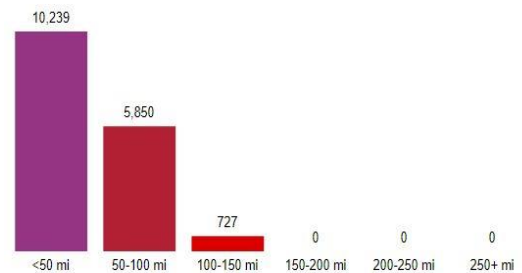


Figure 5-3: Range of received ADS-B messages

The recorded flight data would serve as the basis for testing each component of the flight predictor with favourable trajectories used as described in **section 4.4.1**.

This data is converted and stored as individual flight objects as mentioned in **section 4.1**. Data is accessed by the predictor by specifying the flights unique to ICAO callsign within the MATLAB driver program. Additionally, the user may define the Timestep (T_s), Prediction filter type and the type of collision (dynamic or static).

5.2 Trajectory prediction results

The results from tests described by **section 4.4.1** are detailed below, each flight trajectory is seen with corresponding probability switching outputs from each IMM prediction variation. The RMSE vs time plot the accuracy predictors. The predictor with the lowest total RMSE is shown in a 3D plot with corresponding XY and Z plots to analyse their accuracy to changes predictions in direction and altitude change.

5.2.1 Flight DAL84

The actual flight trajectory of flight DAL84 is seen in figure 5-4, with reading ADS-B position shown as points, the flight undergoes a decrease in altitude with slight changes in direction. The switching probabilities of the IMM predictors are seen in figures 5-5 to 5-7, these indicate the change in modes during the flight path. The RMSE observed as a function of the time of each predictor is seen in Figures 5-8, used to evaluate the accuracy of each predictor. The best performing trajectory is seen in Figures 5-9 with X-Y and Z plots shown in Figures 5-10 and 5-11.

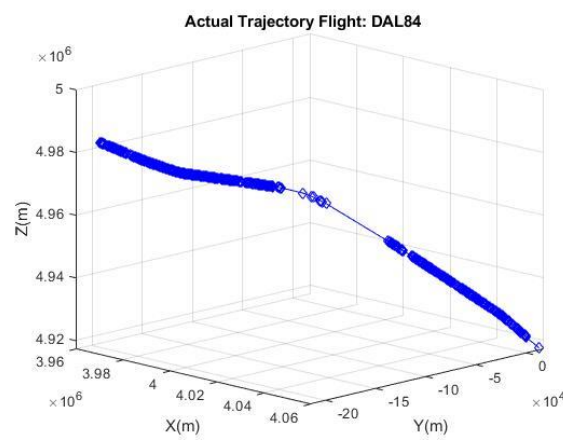


Figure 5-4: Actual trajectory of flight DAL84

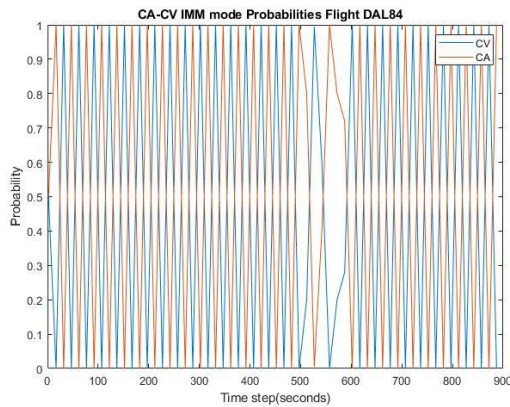


Figure 5-5: DAL84 Mode switching IMM-CV-CA

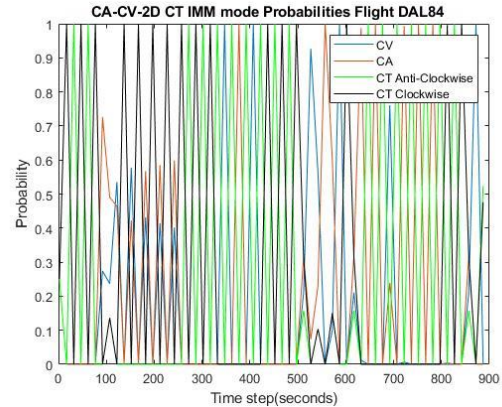


Figure 5-6: DAL84 Mode switching IMM-CV-CA-CT2D

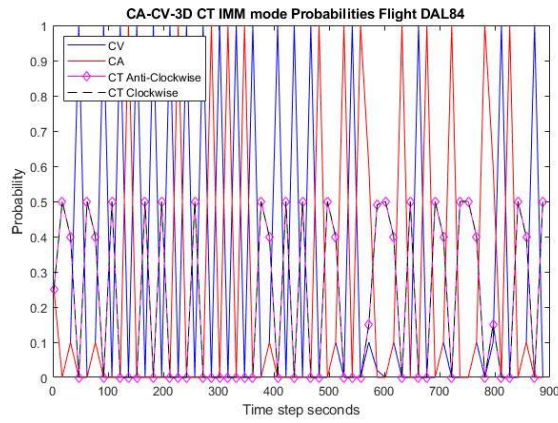


Figure 5-7: DAL84 Mode switching IMM-CV-CA-CT3D

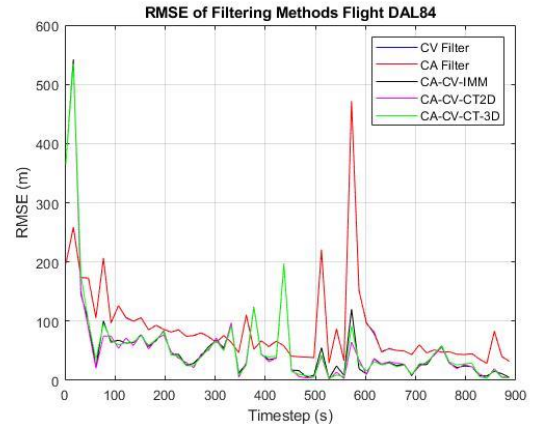


Figure 5-8: DAL84 RMSE vs Time of predictor methods

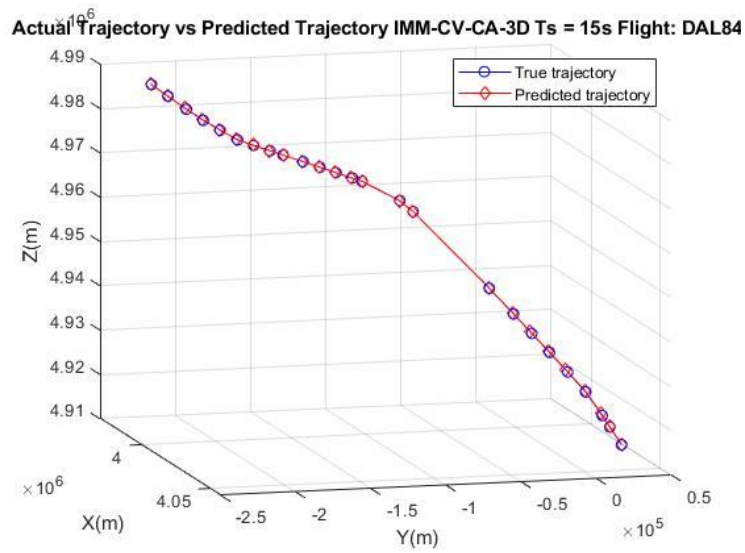


Figure 5-9: DAL84 trajectory prediction IMM-CV-CA-3DCT

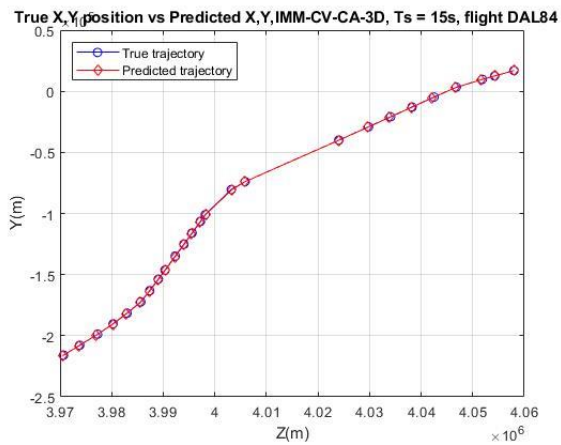


Figure 5-10: DAL84 IMM CV-CA-3DCT XY plot

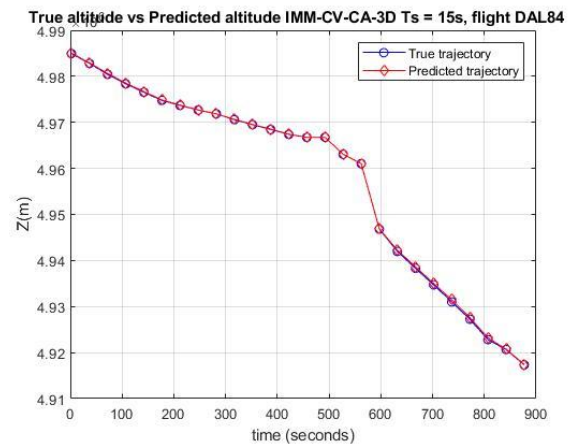


Figure 5-11: DAL84 IMM CV-CA-3DCT time vs Z plot

5.2.2 Flight EXS1204

The actual flight trajectory of flight EXS1204 is seen in figure 5-12, this flight has an upward trajectory with a turn. This flight was selected as the turn occurs with minimal information and a change in direction occurring rapidly. The variations of IMM predictors switching probabilities are seen in figures 5-13 to 5-15. A comparison of the RMSE of each filter as a function of time is seen in figure 5-16. The best performing flight trajectory method being the IMM-CV-CA-3D CT seen in figure 5-17.

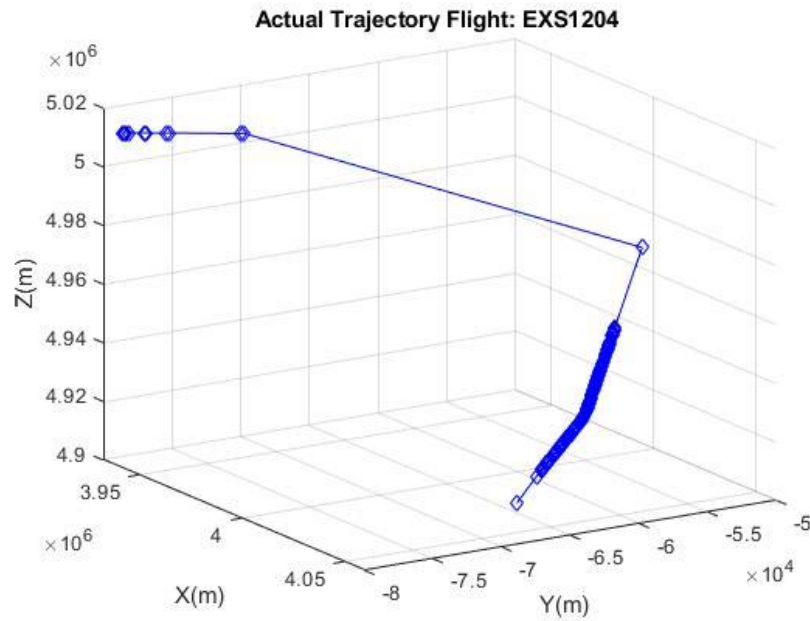


Figure 5-12: Actual trajectory of flight EXS1204

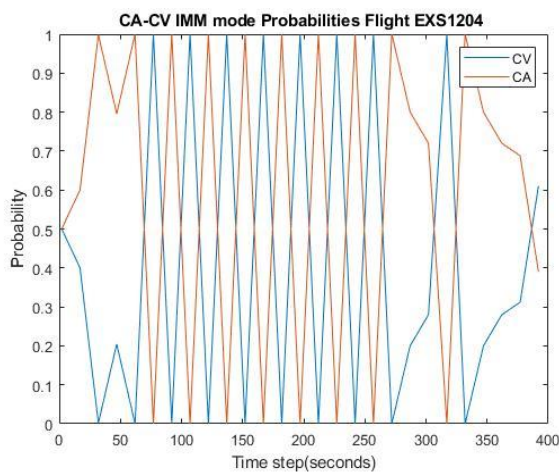


Figure 5-13 EXS1204 Mode switching IMM-CV-CA

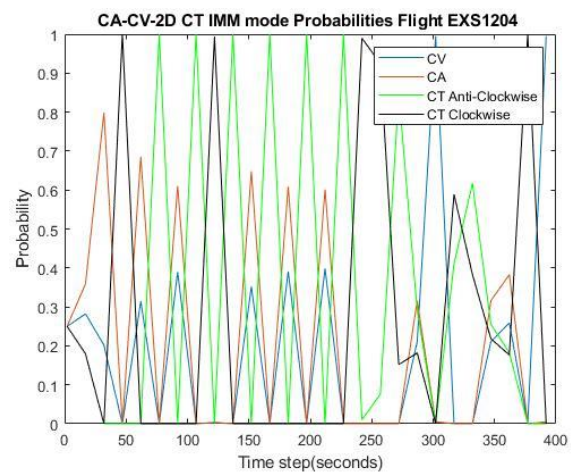


Figure 5-14: EXS1204 Mode switching IMM-CV-CA-CT2D

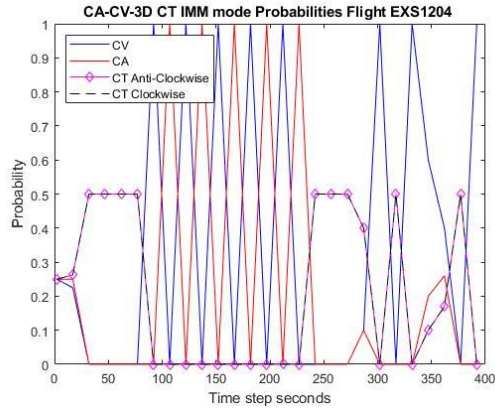


Figure 5-15 EXS1204 Mode switching IMM-CV-CA-CT3D

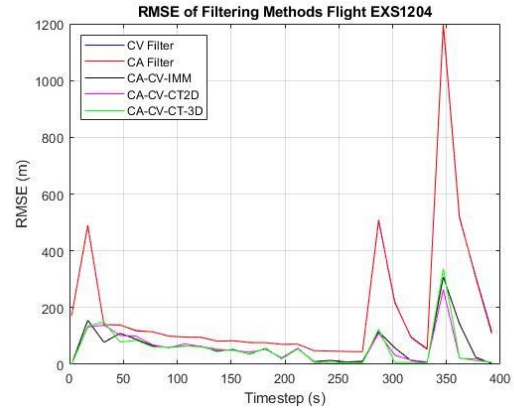


Figure 5-16: EXS1204 RMSE vs Time of predictor methods

Actual Trajectory vs Predicted Trajectory IMM-CV-CA-2D Ts = 15s Flight: EXS1204

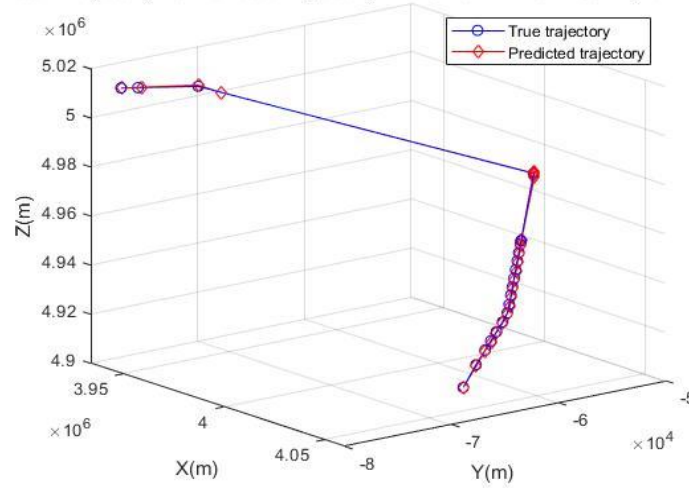


Figure 5-17: Best trajectory prediction of EXS1204, IMM-CV-CA-2DCT

True X,Y position vs Predicted X,Y, IMM-CV-CA-2D, Ts = 15s, flight EXS1204

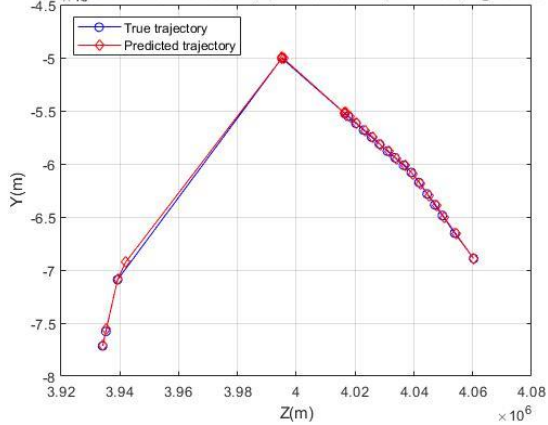


Figure 5-18: DAL84 IMM CV-CA-2DCT XY plot

True altitude vs Predicted altitude IMM-CV-CA-2D Ts = 15s, flight EXS1204

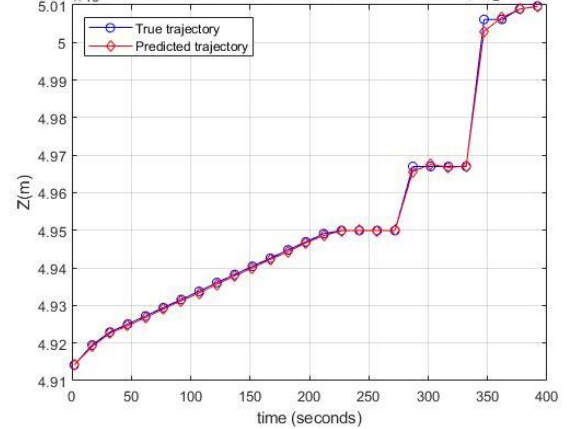


Figure 5-19: DAL84 IMM CV-CA-2DCT XY plot

5.2.3 Flight EZY89TP

The flight trajectory of flight EZY89TP is seen in figure 5-20, the flight is seen to be performing a landing operation with a gradual decreasing altitude with clockwise and anticlockwise manoeuvres. This flight has sufficient information relating to TCPs throughout the flight path as seen as points on the figure. The IMM switching probabilities are seen in figures 5-21 to 5-24 to indicate mode switches during trajectory prediction. The RMSE of each predictor is seen as a function of time in figure 5-24.

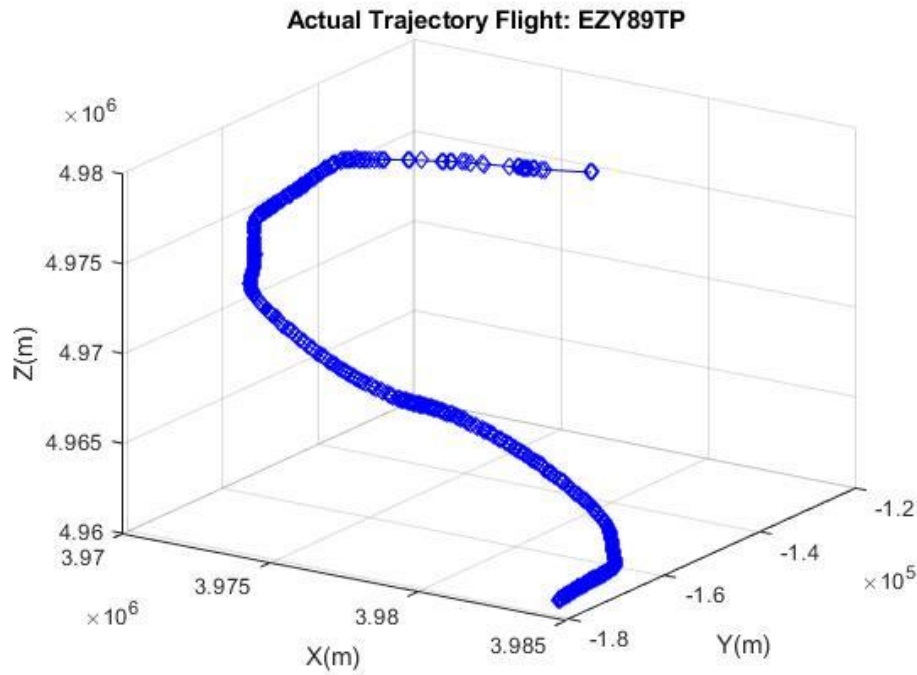


Figure 5-20: Actual trajectory of flight EZY89TP

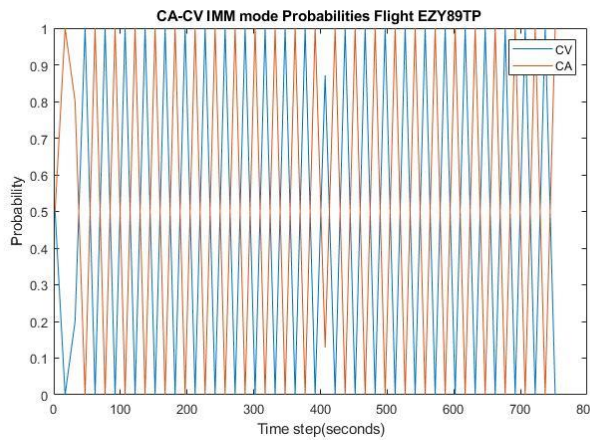


Figure 5-21 EZY89TP Mode switching IMM-CV-CA

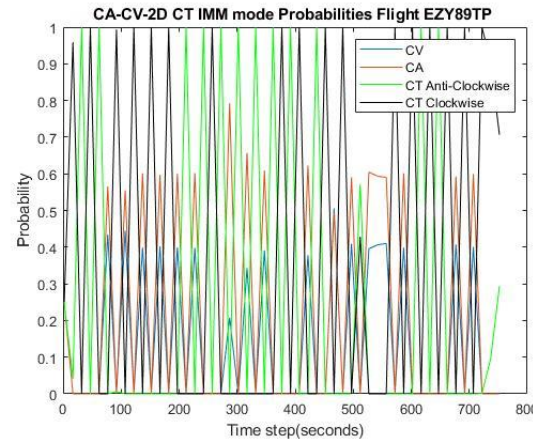


Figure 5-22: EZY89TP Mode switching IMM-CV-CA-CT2D

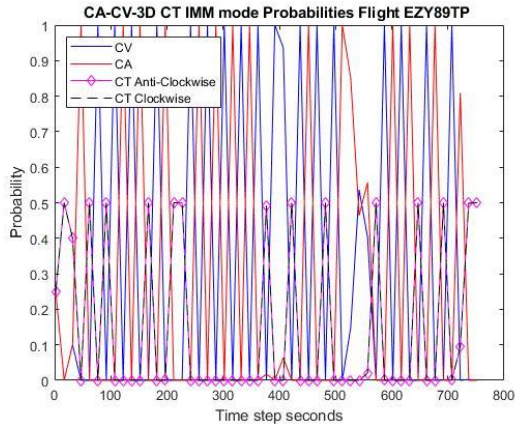


Figure 5-23 EYZ89TP Mode switching IMM-CV-CA-CT3D

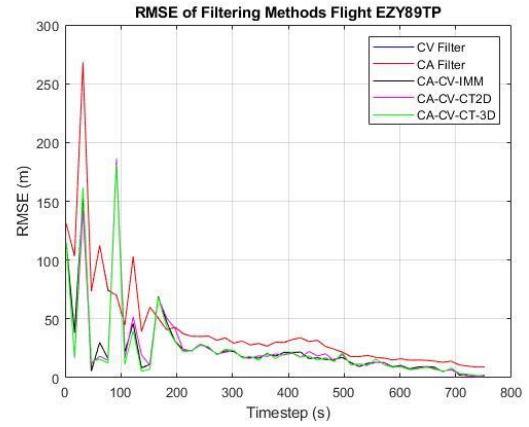


Figure 5-24: EYZ89TP RMSE vs Time of predictor methods

Actual Trajectory vs Predicted Trajectory IMM-CV-CA-2D Ts = 15s Flight: EYZ891

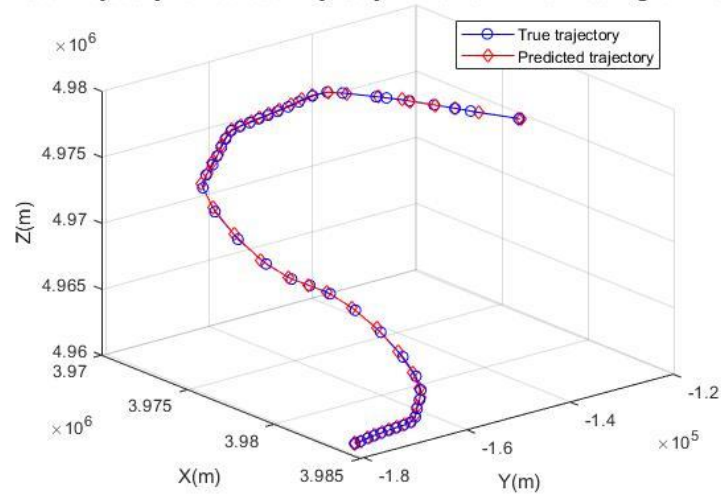


Figure 5-25: Best trajectory prediction of EYZ89TP, IMM-CV-CA-2DCT

True X,Y position vs Predicted X,Y,IMM-CV-CA-2D, Ts = 15s, flight EYZ891

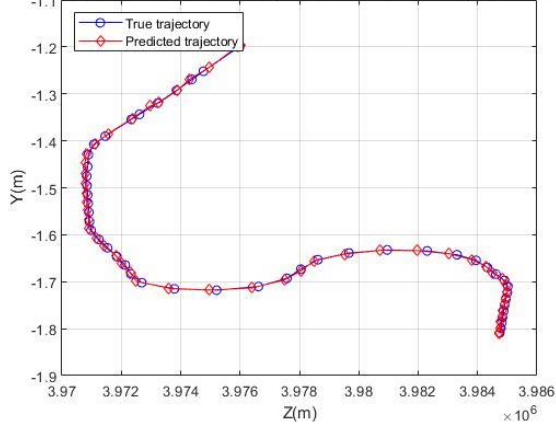


Figure 5-26: EYZ89TP IMM CV-CA-2DCT XY plot

True altitude vs Predicted altitude IMM-CV-CA-2D Ts = 15s, flight EYZ891

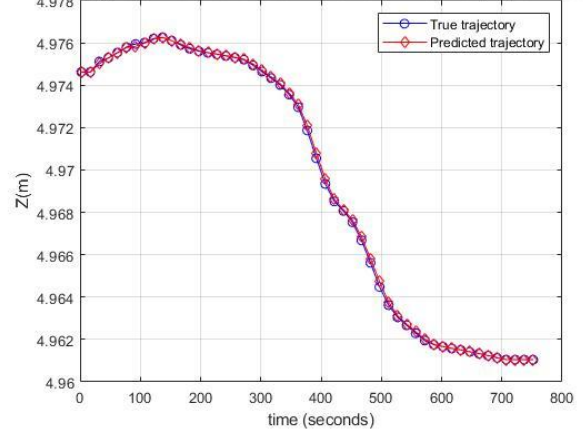


Figure 5-27: EYZ89TP IMM CV-CA-2DCT Z plot

5.2.4 Flight VLG84NC

The actual trajectory of flight VLG84NC is seen in *figure 5-28*, this flight undergoes positive altitude change with a change in direction, ADS-B read positional messages are indicated by the points indicated on the plot. *Figures 5-29 to 5-31* indicate the switching probabilities for the IMM predictors. The RMSE in meters as a function of time is seen in *figure 5-32*, with the best performing prediction algorithm being the IMM-CV-CA-2DCT trajectories seen in *figures 5-33 to 5-35*.

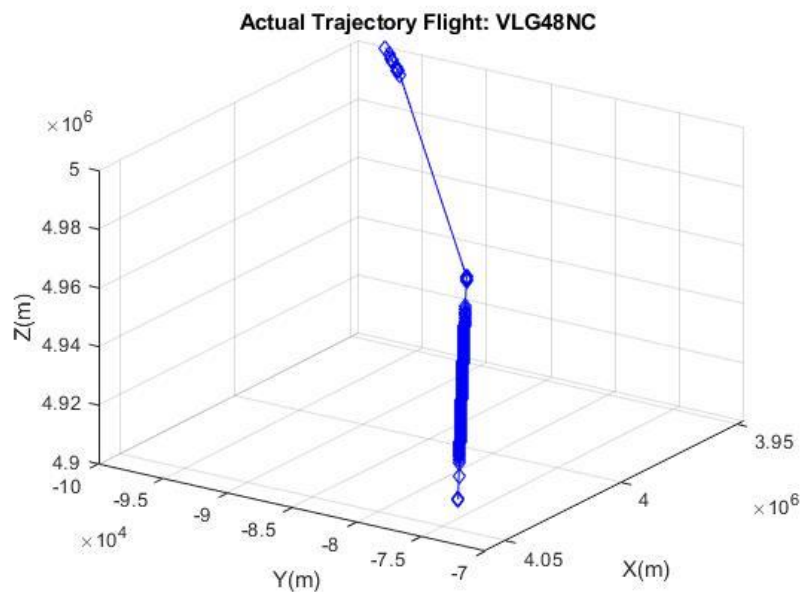


Figure 5-28: Actual trajectory of flight VLG84NC

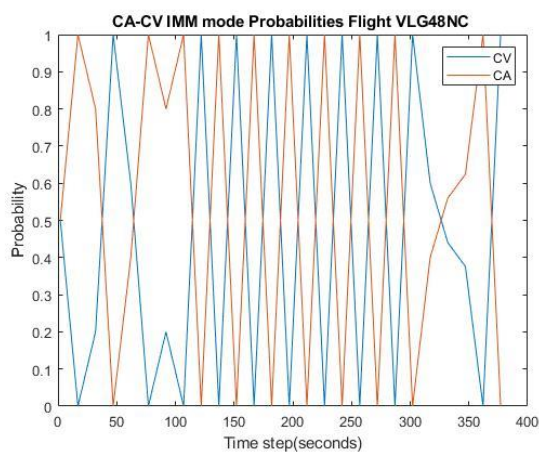


Figure 5-29: VLG48NC Mode switching IMM-CV-CA

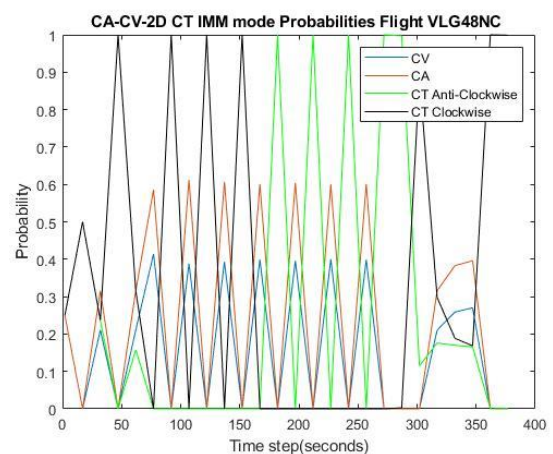


Figure 5-30 VLG84NC Mode switching IMM-CV-CA-CT2D

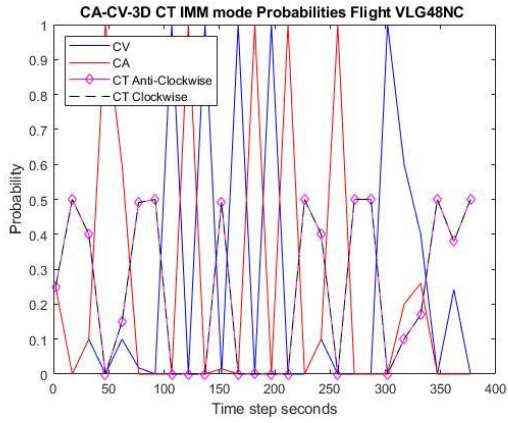


Figure 5-31: VLG84NC Mode switching IMM-CV-CA-CT3D

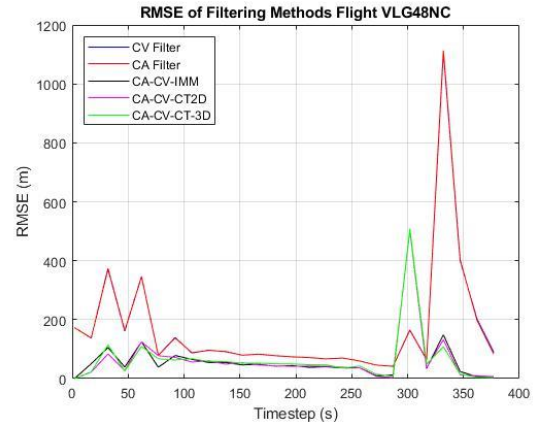


Figure 5-32: VLG84NC RMSE vs Time of predictor methods

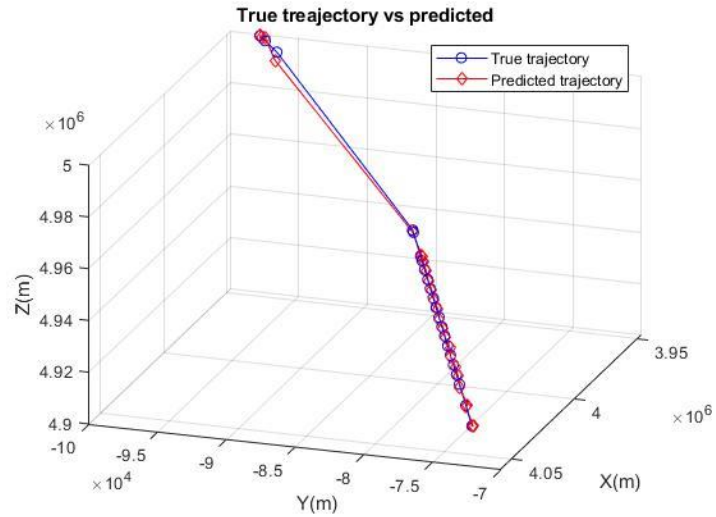


Figure 5-33: VLG84NC Predicted trajectory IMM CV-CA-CT2D

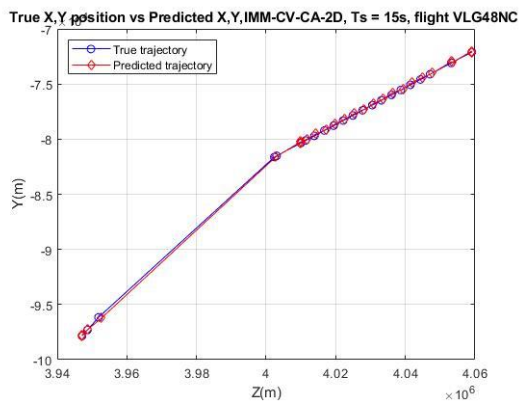


Figure 5-34: VLG84NC IMM CV-CA-2DCT XY plot

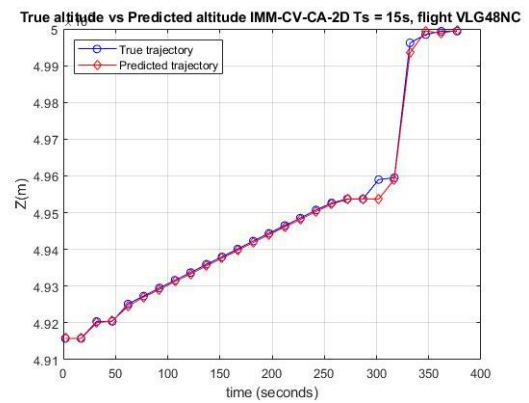


Figure 5-35: VLG84NC IMM CV-CA-2DCT Z plot

5.2.5 Prediction Error Averages

The total RMSE observed for test flight trajectory is calculated for each of the trajectory prediction methods. The total error for each flight and trajectory prediction is seen in figure 5-36. Additionally, the average error for each predictor is calculated to understand which predictor produces would be best suited for generalized cases.

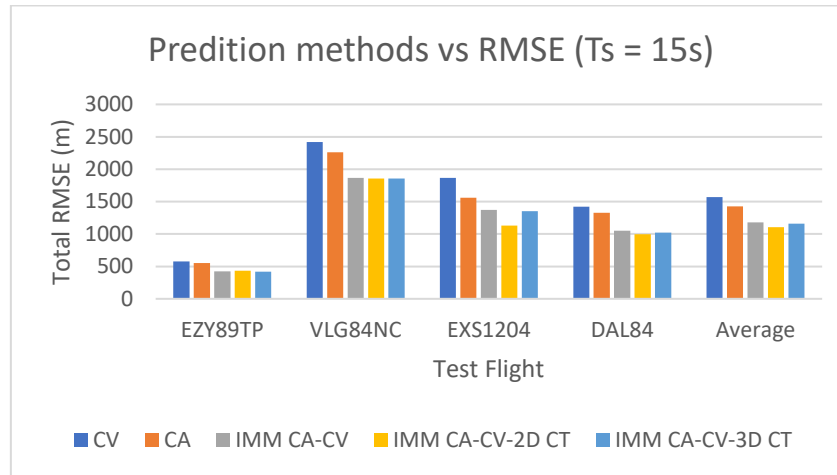


Figure 5-36: Total RMSE of position observed for each test flight

5.3 Collision prediction time results

The collision prediction system is tested as described in **section 4.4.2**; results are in the form of time to a collision. The real collision time is determined within the simulation by propagating the IPZ of each flight along its true trajectory at 1 second intervals. In the case of static collision zones, a collision detection time is found for each of the 4 collision zones generated by the IMM predictor variations, these times are compared with the actual collision times as seen in **section 5.3.1**. In the case of a dynamic intruder collision test, a simulated intruder is positioned at an offset as described by **section 4.4.2.2** and begins its trajectory at the same time as the real aircraft. The actual time to collision is determined and compared to the time to collision with positions output from the IMM variations.

5.3.1 Static Collision results

The static collision results are obtained by measuring the time difference T_d between the predicted collision time and actual collision time as described in equation 4.23. This is done in 5 second intervals from the lower bounds, 15 seconds to the upper bounds of 35 seconds relating to a TCAS RA.

5.3.1.1 Flight EY89TP

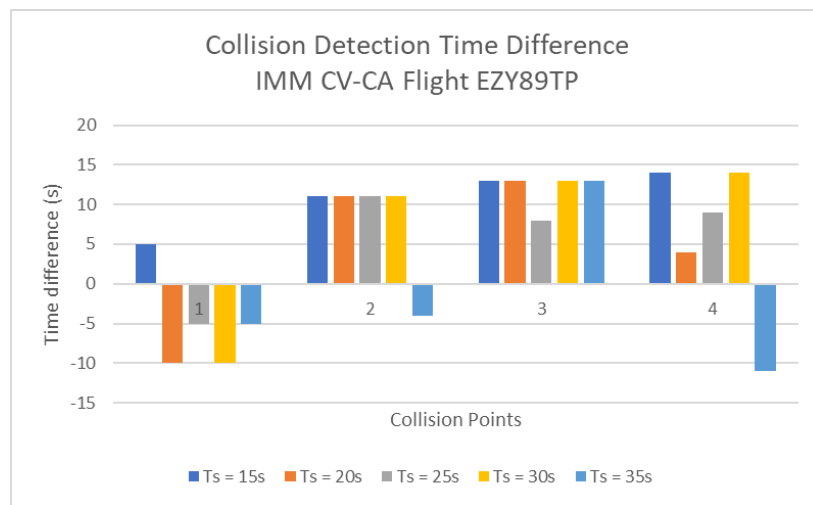


Figure 5-37: EY89TP Time difference in collision time IMM CV-CA

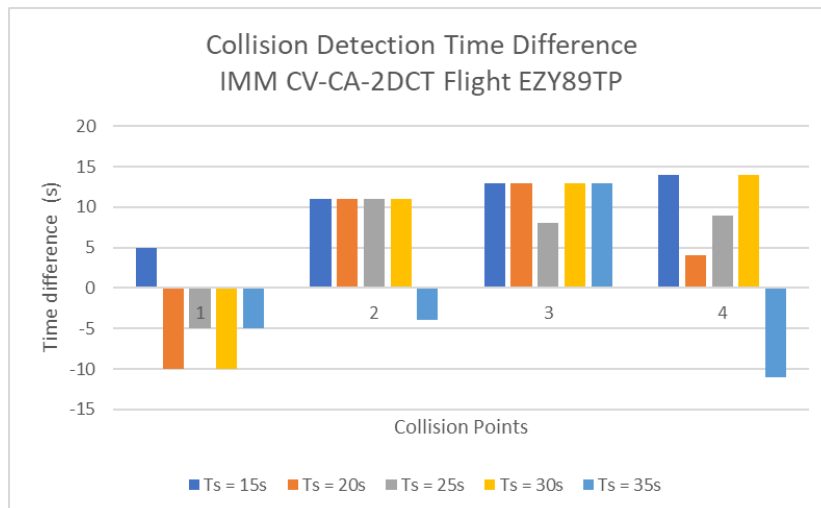


Figure 5-38: EY89TP Time difference in collision time IMM CV-CA-2DCT

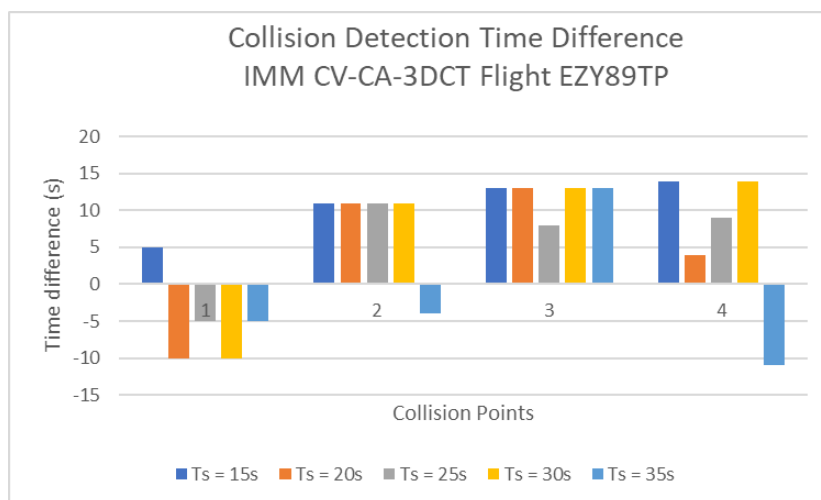


Figure 5-39: EY89TP Time difference in collision time IMM CV-CA-3DCT

5.3.1.2 Flight RYR56UE

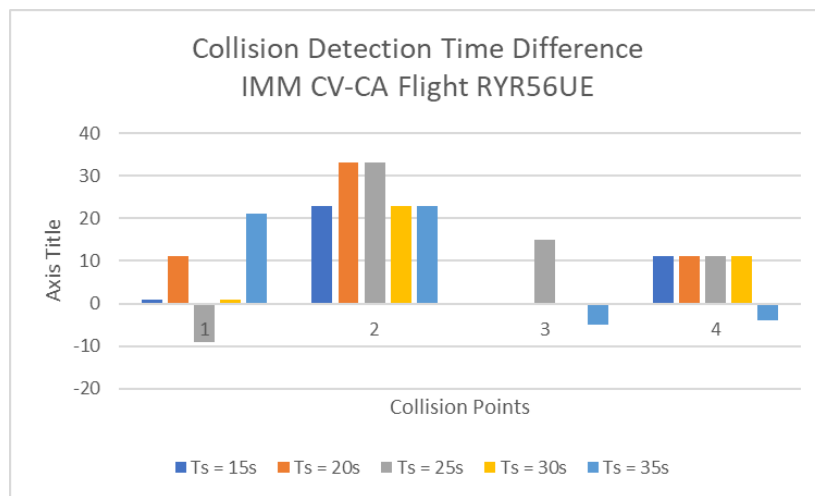


Figure 5-40: RYR56UE Time difference in collision time IMM CV-CA

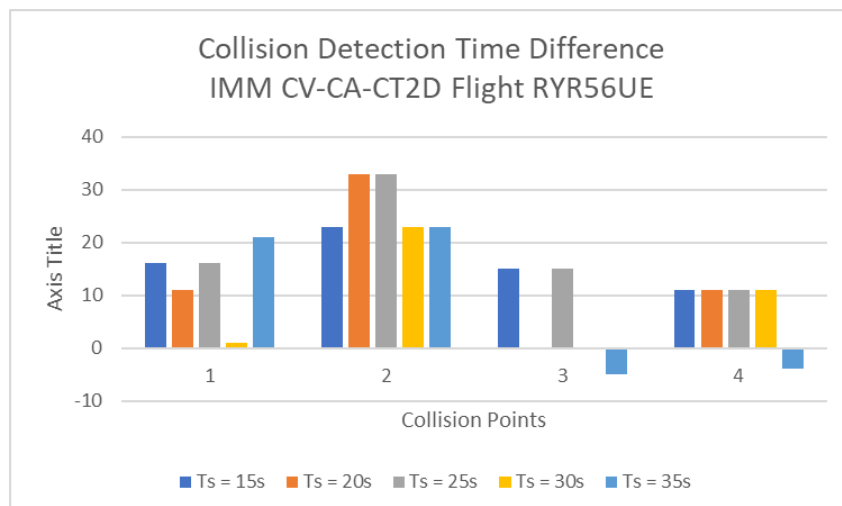


Figure 5-41: RYR56UE Time difference in collision time IMM CV-CA-2DCT

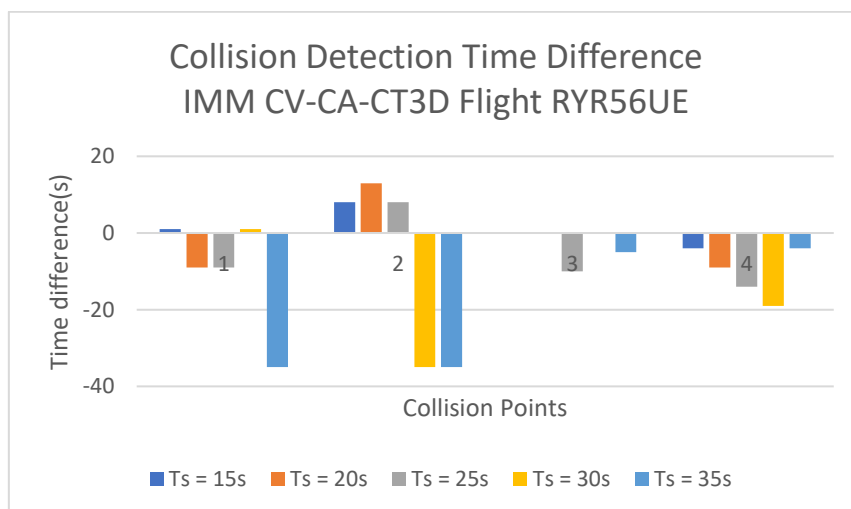


Figure 5-42: RYR56UE Time difference in collision time IMM CV-CA-3DCT

5.3.1.3 Flight RYR8213

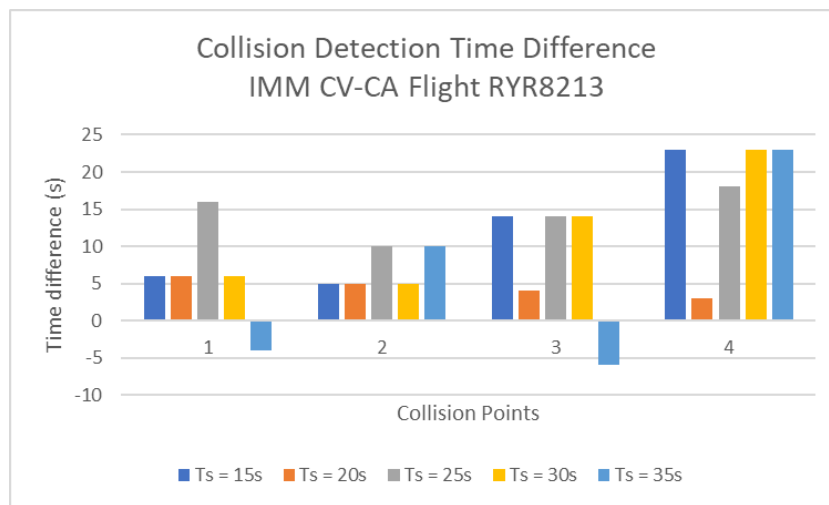


Figure 5-43: RYR8213 Time difference in collision time IMM CV-CA

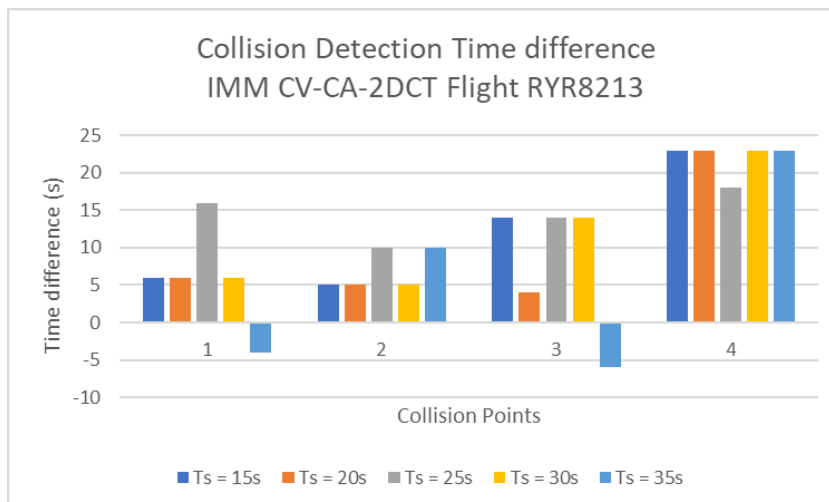


Figure 5-44: RYR8213 Time difference in collision time IMM CV-CA-2DCT

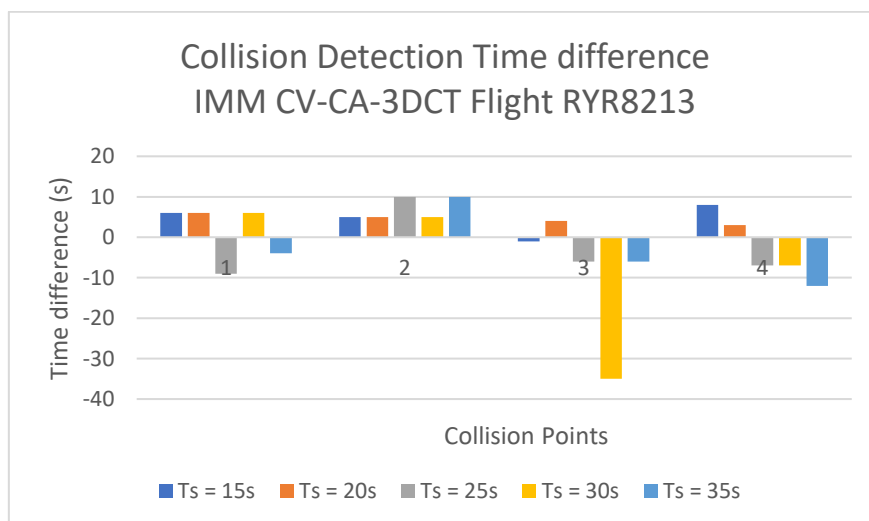


Figure 5-45: RYR8213 Time difference in collision time IMM CV-CA-3DCT

5.3.1.4 Flight EXS8L8

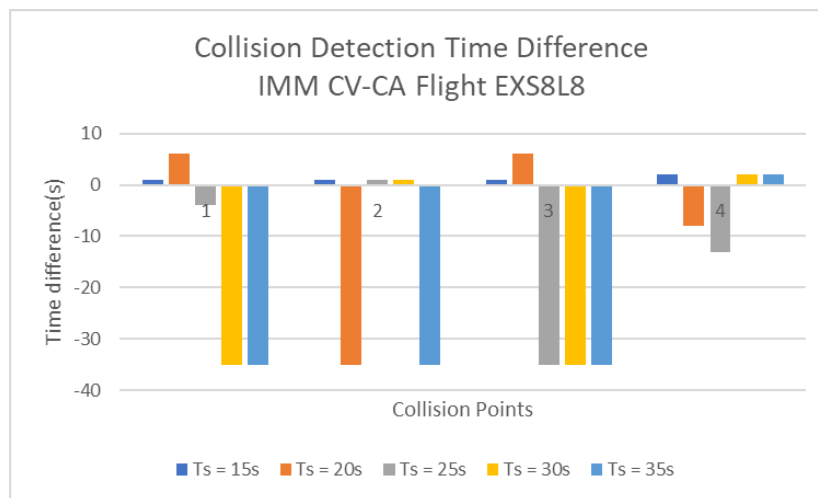


Figure 5-46: EXS8L8 Time difference in collision time IMM CV-CA

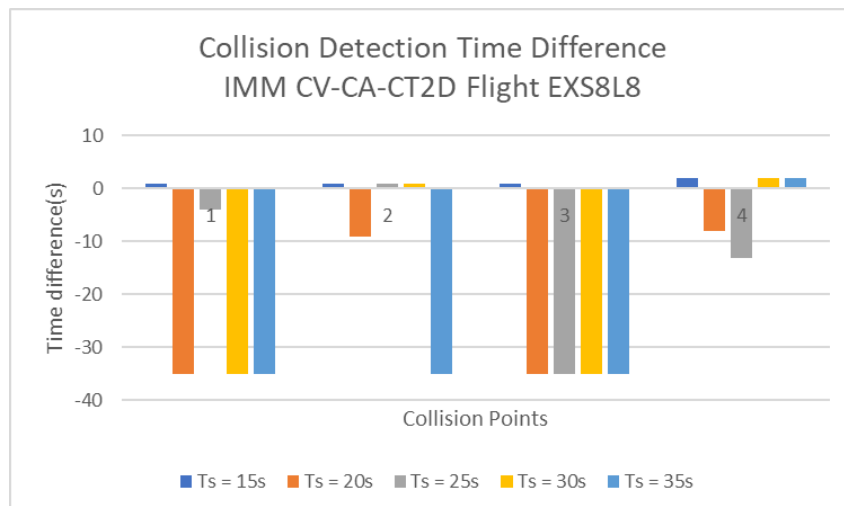


Figure 5-47: EXS8L8 Time difference in collision time IMM CV-CA-2DCT

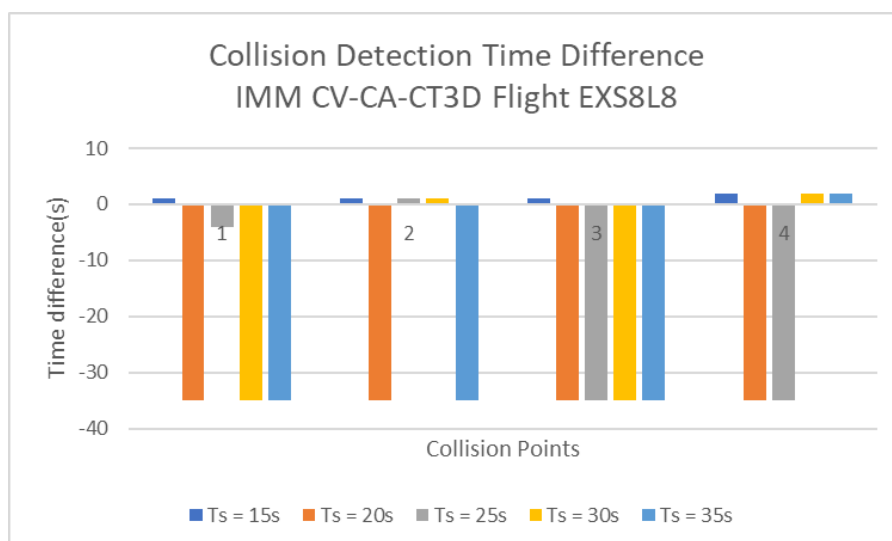


Figure 5-48: EXS8L8 Time difference in collision time IMM CV-CA-3DCT

5.3.1.5 Average time difference

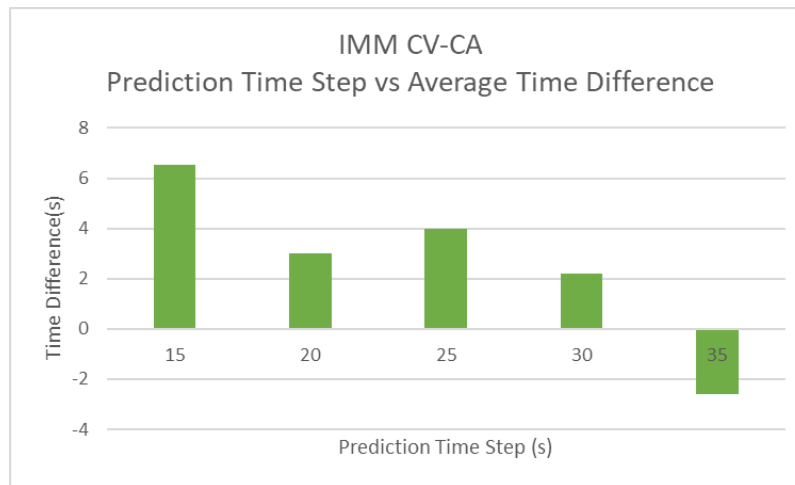


Figure 5-49: Average Time difference in collision time IMM CV-CA

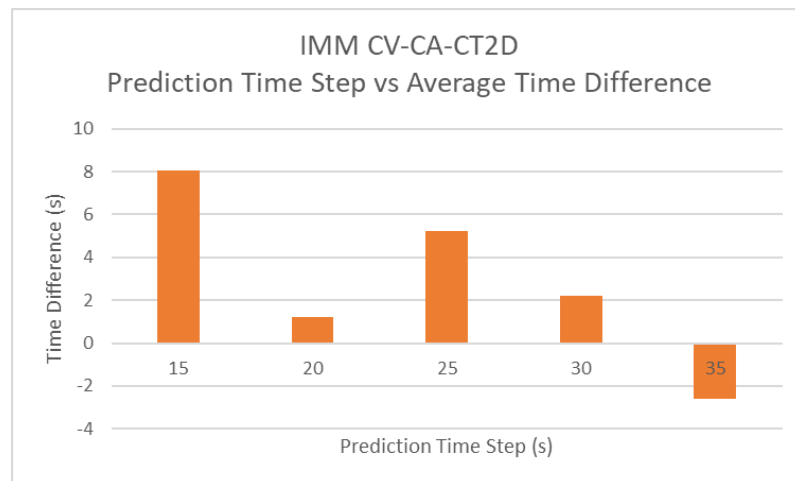


Figure 5-50: Average Time difference in collision time IMM CV-CA-2DCT

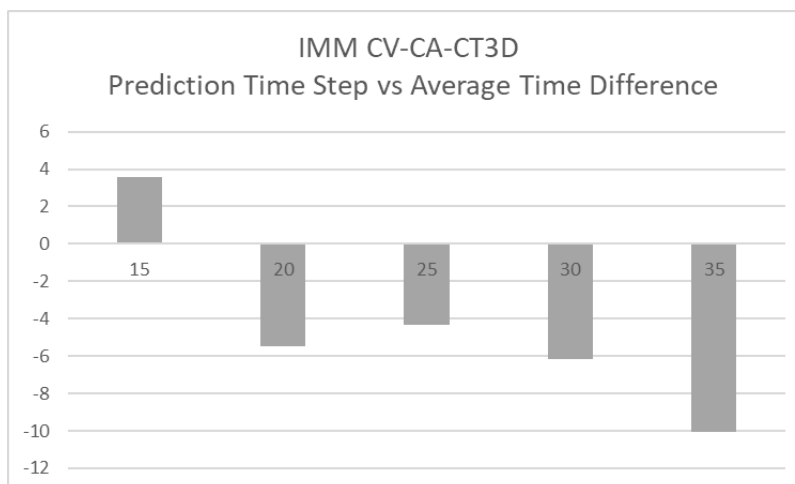


Figure 5-51: Average Time difference in collision time IMM CV-CA-3DCT

5.3.2 Dynamic Collision results

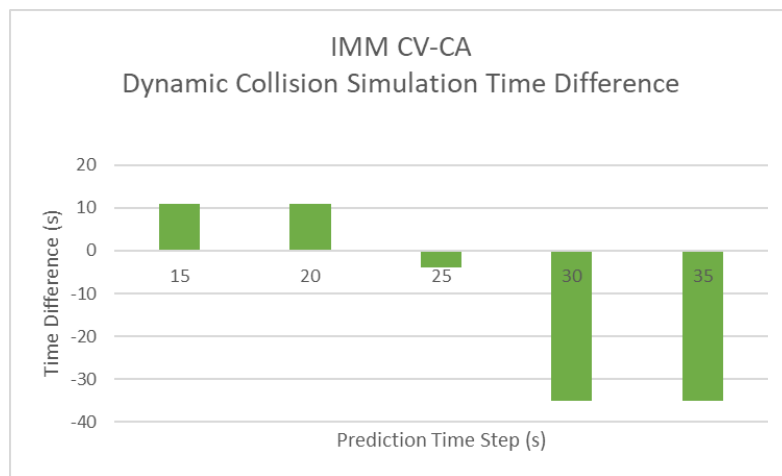


Figure 5-52: Dynamic collision time difference Flight VLG8LV IMM CV-CA

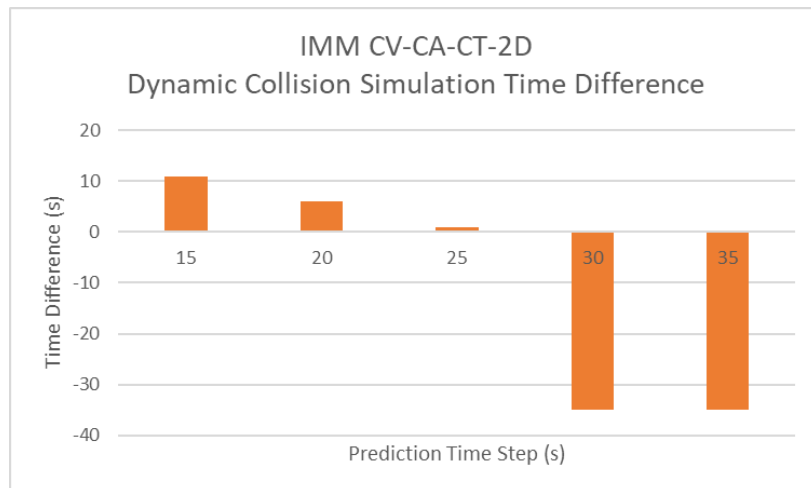


Figure 5-53: Dynamic collision time difference Flight VLG8LV IMM CV-CA-2DCT

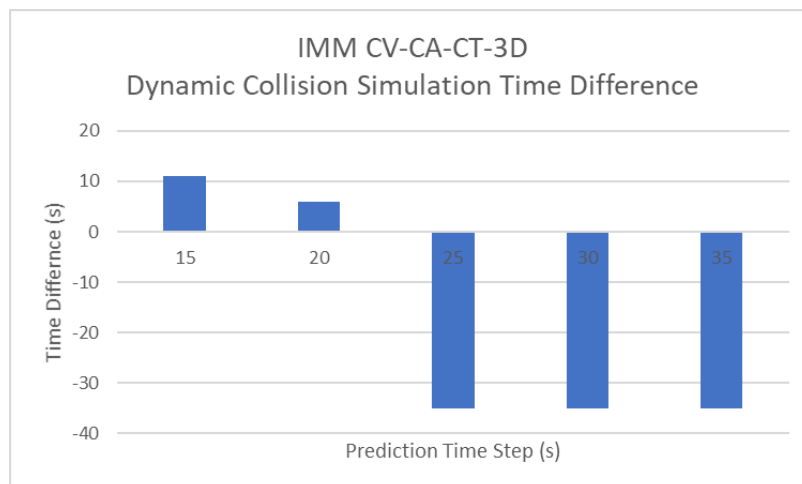


Figure 5-54: Dynamic collision time difference Flight VLG8LV IMM CV-CA-3DCT

Chapter 6: Discussion

6.1 Data acquisition

The ADS-B data acquired to form the database of flight trajectory testing is seen to show a weakness in the south by southwest direction as seen in figure 5-1, this is attributed to the position and mounting height of the antenna during the experiment. Reading data at a higher level with less interference to the radio line of sight would solve this problem. Additionally, in figure 5-3 the ADS-B data received is in a range predominantly less than that of 100 miles away from the receiver. The limitations of the range are due to the Signal to Noise (S/N) ratio of the telescopic halfwave dipole antenna used during the data capture process.

6.2 Trajectory prediction

The results support the research that the use of an IMM such as an IMM-CV-CA-CT2D produces predictions of aircraft future state that is more accurate than single-mode linear Kalman filters. On average cumulative RMSE of position between the 4 test flights used, the IMM-CV-CA-CT2D filter produced results that are 21.1% more accurate than a CV filter alone and 15.6% more accurate than a CA filter alone when using a 15 second T_s of prediction. The results from flight trajectories which had a high number of actual state vectors such as flight EZY89TP (761 actual state positions), seen by blue markers in *figure 5-20* performed the best during the trajectory prediction testing as trajectory predictions are corrected more often during the total flight path. Flight VLG84NC has fewer state positions (378), seen in blue markers in *figure 5-28*, and performed the worst across all the state prediction models tested. Results show in figure 5-35 that the flight “jumps” in altitude after a manoeuvre takes place, the IMM-CV-CA-CT2D filter output lags this motion as no for a rapid increase in altitude such as exponential climb exists within this study.

The RMSE vs time plot is seen in *figure 5-32* shows that the lack of real measurements surrounding this increase causes a spike in errors in each of the prediction methods.

As none of the test flights follows a fully circular trajectory the IMM switching probabilities never stay within a single CT mode for extended periods during the trajectory of any flight path with flights instead of oscillating between Clockwise and Anticlockwise turns as seen in figure 5-14. However, the mode switching does align with expectations as the CA mode and CT anticlockwise motion occurs when flight EXS1204 experiences a turn with an increase in

altitude seen in *figure 5-12*. An unexpected output is that of *figure 5-15* where the anticlockwise and clockwise motion follows the same switching probabilities. This is due to the transition function Φ_{CT3D} producing the same output regardless of the sign of constant angular velocity the mode probabilities are seen as the same for both Clockwise and Anticlockwise manoeuvres. The 3D Coordinated turn is rarely executed with a probability of 1 as the test flights do not adhere to 3D circular movements. The IMM CV-CA-2DCT with a turn rate of 2 deg./s was found to produce the best results among the four test flights used.

6.3 Collision detection

6.3.1 Static collisions

The method of collision detection was found to produce cases whereby a collision was not detected as seen in *figure 5-42* whereby collision points 1 and 2 are seen to not be detected by the IMM CV-CA-CT3D prediction algorithm. This is seen to have the worst performance on flight EXS8L8 in *figures 5-46 to 5-48*, which is attributed to the trajectory prediction methods not producing accurate predictions of the flight's actual trajectory across all time steps. Additionally, the IMM CV-CA-CT3D is seen to produce poor predictions on Flight RYR8213 which is attributed to the actual trajectory of the flight seen in *figure 4-12*. The IMM CV-CA-CT3D is seen however to have the worst performance among the predictors in *figure 5-51* due to poor performance at prediction steps greater than 15s

The IMM CV-CA predictor is seen to have performed the best amongst the 3 IMM's when comparing average performance as seen in *figure 5-49* producing stable predictions with look ahead times up to 30 seconds into the future. The IMM CV-CA-CT2D is seen to have the best performance at 15 and 20 second look ahead times, with a drop off at 20 seconds which is unexpected in *figure 5-50* this is attributed to the transition function producing better estimates of actual positions with a larger magnitude time step. Based on the results, collision detection for static conditions were found to be stable between 15 and 30 seconds except for the IMM CV-CA-CT3D which shows poor performance beyond a 15 second look ahead time

6.3.2 Dynamic collision

Dynamic collisions were only found to be suitable for readings up to 25 seconds look ahead time for the IMMCV-CA-2DCT.

6.4 Limitations

As the testing was done on a limited number of flights (9) these results are specific to the flight trajectories used within testing. The assumptions made in **section 4.4** are applied throughout this study.

The flights used for testing the IMM variations did not have an equal amount of state readings. Thus, some flights had the added benefit of having many Kalman correction steps in comparison to others. To properly test this, flight data with a consistent number of state readings to fairly test the prediction algorithms defined.

The trajectories used for static collision testing had collision zones defined at TCP defined prior to testing. To properly test a static collision zone, more than the four used for testing would need to be evaluated with attitudes which vary from the actual trajectory of the flight.

Only one flight trajectory was used to test a simulated dynamic collision, to best validate the results produced by the system multiple trajectories are required. Additionally, the simulated intruder aircraft was limited to constant velocity dynamics, to properly test a dynamic collision an element of stochasticity from the environment is required.

Chapter 7: Conclusion

This research paper investigated the requirements of BLVOS operation towards drone integration into NAS. The requirement for SAA ability was found to be critical prior to UAV integration. An objective was set to perform trajectory prediction for collisions avoidance with a lookahead time of 15-35 seconds corresponding to the Resolution Advisory issued by TCAS.

ADS-B data was recorded over a 6-hour period to produce a data set of real flight trajectories for testing by recording and decoding SDR data using a Raspberry Pi running a version of Dump1090, an opensource ADS-B decoding software. Three variations of IMM predictors were constructed and tested on 4 Actual flight paths. It was that the IMM CV-CA-2DCT produced the most accurate trajectory predictions on average when using a 15 second look ahead time.

A collision detection system was produced by transforming a collision volume in space, which is periodically transformed by co-ordinates output by the trajectory prediction algorithms. This method collision detection was used in testing Static and Dynamic collision simulations. Results indicate that the use of a prediction algorithm such as an IMM with look ahead times beyond 25 seconds is not feasible as predictions become unstable. The instability in predicted positions caused drift in the collision geometry positions resulting in false negatives (no collision detected).

The objectives outlined for this research were achieved by creating a collision prediction system that can detect collisions at the lower bounds of a TCAS RA 15 to 25 look-ahead time.

Chapter 8: Future work

The data recorded within this study was shown to be skewed towards terminal space due to the proximity to Bristol International Airport most flight data is seen to be either performing take-off/landing manoeuvres. The variance of data would be improved by using an ADS-B data set over a longer period. The data received may be improved by utilizing an antenna with a better SNR / using an ADS-B bandpass filter such that flight data over beyond the 100-mile radius may be obtained. By using the current system an improvement would be testing on more flight trajectories than the 9 in total used to produce results. When testing on more flight trajectories would produce a more generalized and adaptable flight prediction model. The current system may be vastly improved by adding the functionality to predict the position of aircraft in real time.

The IMM's accuracy is dependent on the relevance of prediction model used to the actual trajectory of the object being tracked. As such, this study will be improved by combining non-linear modes such as a current, Singer model with an adaptive mean [73], and non-constant turn rate turn models. The use of a constant turn rate CT model is not practical as none of the flights observed during testing maintained their turn rate. This would require the use of an Extended Kalman filter (EKF). Additionally, the method of collision detection used within this study would be improved by using a probabilistic method. This would model the intruder protected zone as the integral of a probability density function which may be adapted to the UAVs relative velocity. This approach would alter the size of the IPZ depending on the reaction time required like the research done by [72]. A hybrid collision detection model may be constructed by including weather report information and Trajectory Change Points (TCPs) which would be available within the ADS-B data in the future. TCPs would give insight into future movement of an aircraft allowing for intent based trajectory which could be fed into an IMM giving insight into the aircraft's future dynamics.

The study would be improved by simulating the intruder aircraft within a Software-In-the-Loop method. The ADS-B data may be implemented within a JSB-Sim, which is an open-source Flight Dynamics Model which has a MATLAB API. Simulation real flight dynamics would allow for an added stochasticity when testing collision detection. Additionally, this would allow for testing on varied sizes of aircraft to validate the IPZ selected within this study.

Finally, the code produced in MATLAB could be converted to C++ code using the MATLAB Coder. This would allow for the software to be loaded onto a Pixhawk controller on a physical drone with ADS-B IN functionality to test the viability of receiving flight information for real time collision avoidance.

References

- [1] Y. Lin and S. Saripalli, “Sense and avoid for unmanned aerial vehicles using ADS-B,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015, pp. 6402–6407, Jun. 2015, doi: 10.1109/ICRA.2015.7140098.
- [2] S. Yang, X. Yang, and J. Mo, “The application of unmanned aircraft systems to plant protection in China,” *Precision Agriculture*, vol. 19, no. 2, pp. 278–292, Mar. 2017, doi: 10.1007/s11119-017-9516-7.

- [3] S. Manfreda *et al.*, “On the Use of Unmanned Aerial Systems for Environmental Monitoring,” *Remote Sensing*, vol. 10, no. 4, p. 641, Apr. 2018, doi: 10.3390/rs10040641.
- [4] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, “UAVs for smart cities: Opportunities and challenges,” 2014, pp. 267–273, doi: 10.1109/ICUAS.2014.6842265.
- [5] D. R. Vieira, D. Silva, and A. Bravo, “Electric VTOL aircraft: the future of urban air mobility (background, advantages and challenges),” *International Journal of Sustainable Aviation*, vol. 5, no. 2, p. 101, 2019, doi: 10.1504/ijsa.2019.10023187.
- [6] “Air taxis,” www.rolls-royce.com. <https://www.rolls-royce.com/innovation/propulsion/air-taxi.aspx> (accessed Apr. 20, 2021).
- [7] “Joby Aviation | Joby,” www.jobyaviation.com. <https://www.jobyaviation.com/> (accessed Mar. 20, 2021).
- [8] “Home - Lilium,” Lilium.com, 2019. <https://lilium.com/> (accessed Mar. 27, 2021).
- [9] J.-P. Aurambout, K. Gkoumas, and B. Ciuffo, “Last mile delivery by drones: an estimation of viable market potential and access to citizens across European cities,” *European Transport Research Review*, vol. 11, no. 1, Jun. 2019, doi: 10.1186/s12544-019-0368-2.
- [10] PricewaterhouseCoopers, “Drones could add £42bn to UK GDP by 2030 - PwC research,” PwC, May 19, 2018. <https://www.pwc.co.uk/press-room/press-releases/pwc-uk-drones-report.html> (accessed Apr. 08, 2021).
- [11] “Flying in the open category | UK Civil Aviation Authority,” www.caa.co.uk. <https://www.caa.co.uk/Consumers/Unmanned-aircraft/Recreational-drones/Flying-in-the-open-category/> (accessed Mar. 27, 2021).
- [12] L. Purton, H. Abbass, and S. Alam, “Identification of ADS-B system vulnerabilities and threats,” Jan. 2010.
- [13] “CAP 722: Unmanned Aircraft System Operations in UK Airspace - Guidance & Policy,” publicapps.caa.co.uk. <https://publicapps.caa.co.uk/modalapplication.aspx?appid=11&mode=detail&id=415>.
- [14] “The Air Navigation (Amendment) Order 2020 Article 129,” Legislation.gov.uk, 2020. <https://www.legislation.gov.uk/uksi/2020/1555/made> (accessed Mar. 23, 2021).
- [15] “Regulatory challenges for innovation in aviation | UK Civil Aviation Authority,” www.caa.co.uk. <https://www.caa.co.uk/Our-work/Innovation/Regulatory-challenges-for-innovation-in-aviation/> (accessed Sep. 2, 2021).
- [16] U.S. Department of Transportation, “Integration of Civil Unmanned Aircraft Systems (UAS) in The National Airspace System (NAS) Roadmap,” tech. rep., U.S. Department of Transportation, FAA, Washington, DC, 2013.
- [17] L. Wong, W. Chen, J. Kay, and V. Raska, “Autonomous Sense and Avoid (SAA) for Unmanned Air System (UAS),” in RTO-MP-SCI-202 Symposium on Intelligent uninhabited Vehicle Guidance Systems, (Munich, Germany), 2009.
- [18] “CAP1861: Beyond Visual Line of Sight in Non-Segregated Airspace,” publicapps.caa.co.uk.

<https://publicapps.caa.co.uk/modalapplication.aspx?appid=11&mode=detail&id=9294> (accessed Sep. 2, 2021).

[19] “Air Traffic Management (ATM),” *Icao.int*, 2019.
<https://www.icao.int/safety/airnavigation/Pages/atm.aspx> (accessed Mar. 27, 2021).

[20] “GitHub adsbxchange/dump1090-fa,” *GitHub*.
<https://github.com/adsbxchange/dump1090-fa> (accessed July. 1, 2021).

[22] NextGen, the Next Generation Air Transportation System: Transforming Air Traffic Control from Ground-Based and Human-Centric to Satellite-Based and Airplane-Centric

[23] S.-H. Lee, Y.-K. Kim, J.-W. Han, and D.-G. Lee, “Protection Method for Data Communication between ADS-B Sensor and Next-Generation Air Traffic Control Systems,” *Information*, vol. 5, no. 4, pp. 622–633, Dec. 2014.

[24] D. Jeon, Y. Eun, and H. Kim, “Estimation fusion with radar and ADS-B for air traffic surveillance,” *International Journal of Control, Automation and Systems*, vol. 13, no. 2, pp. 336–345, Feb. 2015, doi: 10.1007/s12555-014-0060-1.

[25] “TCAS Version 7.1 requirement in EU airspace,” *EASA*, Jul. 22, 2015. 25
<https://www.easa.europa.eu/newsroom-and-events/news/tcas-version-71-requirement-eu-airspace> (accessed Mar. 15, 2021).

[25] M. Kochenderfer, J. Holland, and J. Chryssanthacopoulos, “Next generation airborne collision avoidance system,” *Lincoln Laboratory Journal*, vol. 19, pp. 17–33, Jan. 2012.

[26] “Navigating Airspace Classifications - When & Where to Fly Your Drone,” *dronegenuity*, Nov. 17, 2016. <https://www.dronegenuity.com/navigating-airspace-classifications/> (accessed Mar. 14, 2021).

[27] S. Drinkwater, “ADS-B OUT Implementation Delayed,” *COPA*, Oct. 24, 2019.
<https://copanational.org/en/2019/10/24/ads-b-out-implementation-delayed/> (accessed Mar. 13, 2021).

[28] K.-Y. Baek and H.-C. Bang, “ADS-B based Trajectory Prediction and Conflict Detection for Air Traffic Management,” *International Journal of Aeronautical and Space Sciences*, vol. 13, no. 3, pp. 377–385, Sep. 2012, doi: 10.5139/ijass.2012.13.3.377.

[29] M. Varga, Z. A. Polgar, and H. Hedesiu, “ADS-B based real-time air traffic monitoring system,” *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2015, doi: 10.1109/tsp.2015.7296255.

[30] “ACAS X - SKYbrary Aviation Safety,” *www.skybrary.aero*, Mar. 02, 2021.
https://www.skybrary.aero/index.php/ACAS_X#:~:text=ACAS%20XU%20standards%20we re (accessed Mar. 28, 2021).

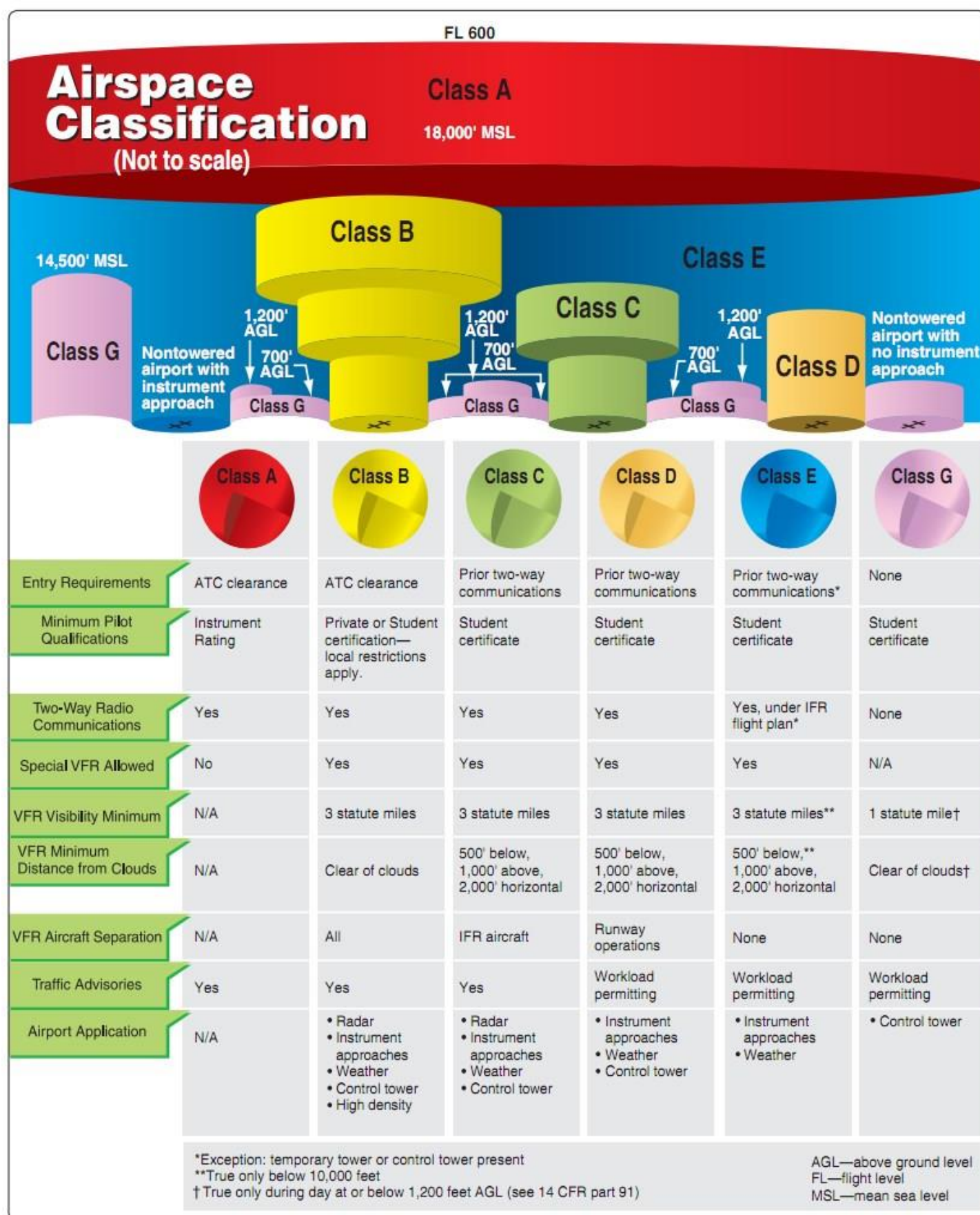
[31] S. Ayhan and H. Samet, “Aircraft trajectory prediction made easy with predictive analytics,” *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 21–30, Aug. 13–17, 2016, Accessed on: July 10, 2018.

- [32] M. G. Hamed, D. Gianazza, M. Serrurier and N. Durand, "Statistical prediction of aircraft trajectory: regression methods vs point-mass model", *ATM 2013 10th USA/Europe Air Traffic Management Research and Development Seminar Chicago IL*, pp. hal-00911709, June 2013.
- [33] Zhao, Ziyu & Zeng, Weili & Zhibin, Quan & Chen, Mengfei & Yang, Zhao. (2019). Aircraft Trajectory Prediction Using Deep Long Short-Term Memory Networks. 124-135. 10.1061/9780784482292.012.
- [34] Nivison, S.A.; Khargonekar, P.P. Development of a robust deep recurrent neural network controller for flight applications. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; IEEE: New York, NY, USA, 2017; pp. 5336–5342. [CrossRef]
- [35] X. Tang, L. Zhou,, Z. Shen,, & M.Tang, (2015). 4D Trajectory Prediction of Aircraft Taxiing Based on Fitting Velocity Profile. CICTP 2015. doi:10.1061/9780784479292.001
- [36] Y. Bar-Shalom, X.-R. Li and T Kirubarajan, Estimation with Applications to Tracking and Navigation, New York, NY:John Wiley & Sons Ltd., 2001.
- [37] T. I. Baumgartner and U. Maeder, "Trajectory prediction for low-cost collision avoidance systems," *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 1.C.5-1-1.C.5-8, doi: 10.1109/DASC.2009.5347567.
- [38] Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software; John Wiley & Sons: Hoboken, NJ, USA, 2004.
- [39] Q. Li, R. Li, K. Ji and W. Dai, "Kalman Filter and Its Application," *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74-77, doi: 10.1109/ICINIS.2015.35.
- [40] D. Peng and Y. Gu, "IMM Algorithm for a 3D High Maneuvering Target Tracking," *Lecture Notes in Computer Science*, pp. 529–536, 2011, doi: 10.1007/978-3-642-21524-7_65.
- [41] S. Gamse, T. A. Wunderlich, P. Wasmeier and D. Kogoj, "The use of Kalman filtering in combination with an electronic tacheometer," *2010 International Conference on Indoor Positioning and Indoor Navigation*, 2010, pp. 1-9, doi: 10.1109/IPIN.2010.5647500.
- [42] "Airborne Collision Avoidance System (ACAS) - SKYbrary Aviation Safety," *Skybrary.aero*, 2017. [https://www.skybrary.aero/index.php/Airborne_Collision_Avoidance_System_\(ACAS\)](https://www.skybrary.aero/index.php/Airborne_Collision_Avoidance_System_(ACAS)).
- [44] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, Art. no. 1, 1998, doi: 10.1109/7.640267
- [45] J. Hegde, E. Henriksen, I. Utne, and I. Schjølberg, "Development of safety envelopes and subsea traffic rules for autonomous remotely operated vehicles," *Journal of Loss Prevention in the Process Industries*, vol. 60, Mar. 2019, doi: 10.1016/j.jlp.2019.03.006.

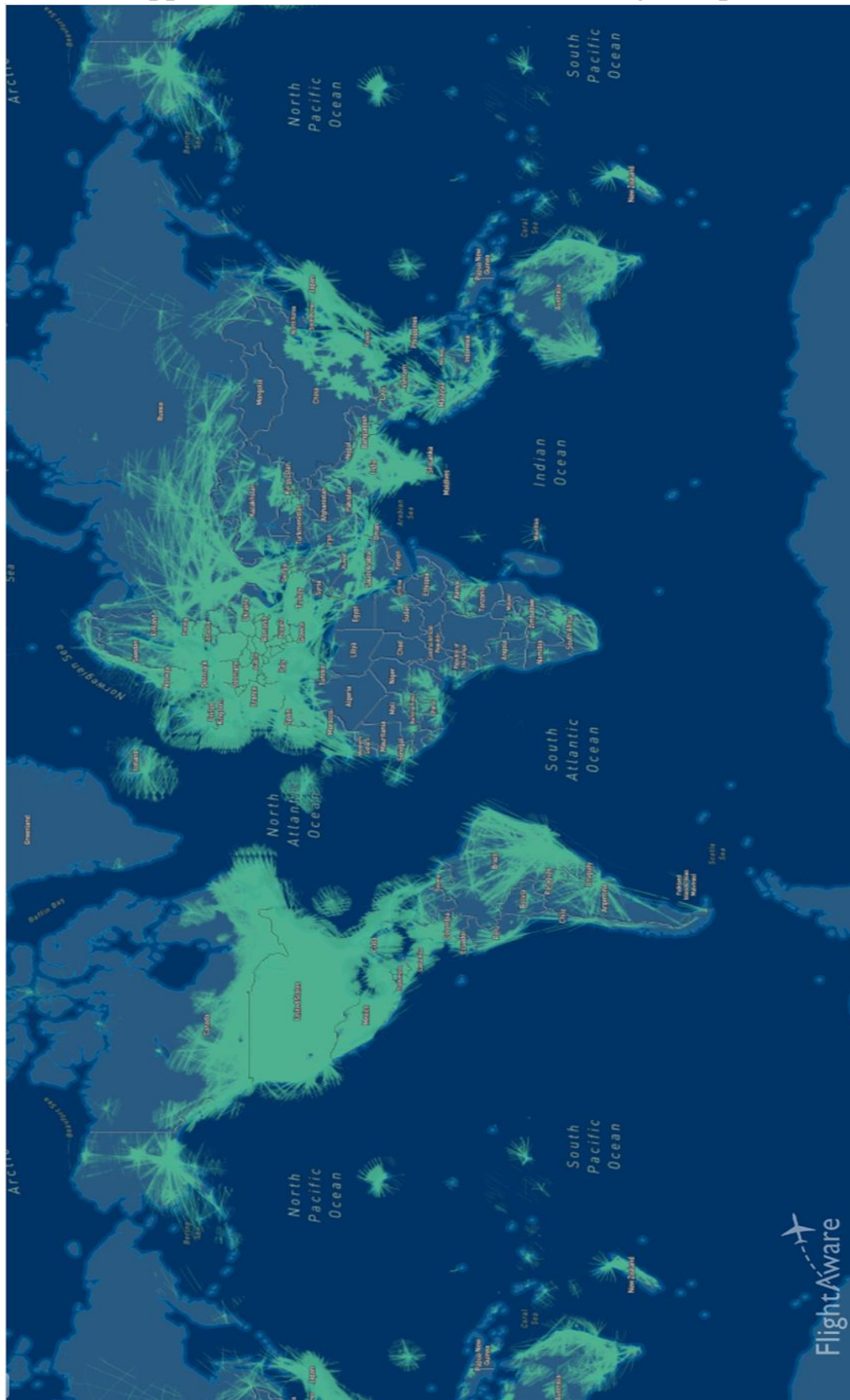
- [46] A. F. Genovese, “The interacting multiple model algorithm for accurate state estimation of maneuvering targets,” *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 22, pp. 614–623, Oct. 2001.
- [50] “FlightAware ADS-B Coverage Map,” *FlightAware*. <https://uk.flightaware.com/adsb/coverage/#data-coverage> (accessed Sep. 01, 2021).
- [51] V. A. Pereboom, “A Data-Driven comparison of Trajectory Prediction methodologies using ADS-B data,” Thesis, TU Delft, 2019.
- [52] C. Muñoz, A. A. Dutle, A. Narkawicz, and J. Upchurch, “Unmanned aircraft systems in the national airspace system: A formal methods perspective,” *ACM SIGLOG News*, vol. 3, Art. no. 3, Aug. 2016, doi: 10.1145/2984450.2984459
- [53] A. Abdulaziz, Y. Shehu, A. Ahmad, and S. Namadi, “Surveillance radar system limitations and the advent of the automatic dependent surveillance broadcast system for aircraft monitoring,” vol. 7, pp. 9–15, Apr. 2019.
- [54] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*, 2nd ed. TU Delft OPEN Publishing, 2021.
- [55] A. Abdulaziz, Y. Shehu, A. Ahmad, M. Kabir, and Bello Salau, Habeeb, “Optimum receiver for decoding automatic dependent surveillance broadcast (ADS-B) signals,” *American Journal of Signal Processing*, vol. 5, pp. 23–31, Jan. 2015, doi: 10.5923/j.ajsp.20150502.01.
- [56] B. Erton. "RF Coverage analysis methodology as applied to ADS-B design. Aerospace conference, 2009 IEEE. IEEE, 2009.
- [57] H. Zha, Q. Tian, and Y. Lin, “Real-world ADS-B signal recognition based on radio frequency fingerprinting,” 2020, pp. 1–6, doi: 10.1109/ICNP49622.2020.9259404.
- [58] Amrhar A, Kisomi A A, Zhang E, et al. “Multi-Mode reconfigurable Software Defined Radio architecture for avionic radios” 2017 Integrated Communications, Navigation and Surveillance Conference (ICNS). IEEE, 2017: 2D1-1-2D1-10
- [59] L. Sahawneh, M. O. Duffield, R. Beard, and T. McLain, “Detect and avoid for small, unmanned aircraft systems using ADS-B,” *Air traffic control quarterly*, vol. 23, pp. 203–240, 2015.
- [60] W. Blythe, Anderson, H., and King, N. 2011. *Ads-b implementation and operations guidance document*. International Civil Aviation Organization.
- [61] S. Drinkwater, “ADS-B OUT Implementation Delayed,” *COPA*, Oct. 24, 2019. <https://copanational.org/en/2019/10/24/ads-b-out-implementation-delayed/> (accessed Mar. 13, 2021).
- [62] *Automatic Dependent Surveillance Broadcast (ADSB) Out Performance Requirements To Support Air Traffic Control (ATC) Service Final Rules Federal Aviation Administration 14 CFR Part 91*, 2010.
- [63] “Equip ADS-B,” *www.faa.gov*, Mar. 15, 2021. <https://www.faa.gov/nextgen/equipadsb/> (accessed Mar. 20, 2021).

- [64] RTCA Special Committee 186, "Minimum Operational Performance Standards (MOPS) for Universal Access Transceiver (UAT) Automatic Dependent Surveillance - Broadcast (ADS-B)," RTCA, Inc., 2011.
- [65] A. Henshall, "UAV Collision Avoidance Utilizing ADS-B and TCAS with Independent Ranging," Master's Thesis, Auburn University, 2018.
- [66] M. Kochenderfer, J. Holland, and J. Chryssanthacopoulos, "Next generation airborne collision avoidance system," *Lincoln Laboratory Journal*, vol. 19, pp. 17–33, Jan. 2012.
- [67] N. F. M. Radzi, M. Mubin and N. A. Rahim, "Collision detection algorithm for UAVs based on parametric theorem approach," *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 493-496, doi: 10.1109/ICCSN.2011.6014942.
- [68] EUROCONTROL. *Specification SPEC-0102. EUROCONTROL Specification for the Use of Military Unmanned Aerial Vehicles as Operational Air Traffic Outside Segregated Airspace*; EUROCONTROL: Brussels, Belgium, 2007
- [69] D. Vito and G. Torrano, "RPAS automatic ADS-B based separation assurance and collision avoidance system real-time simulation results," *Drones*, vol. 4, Art. no. 4, 2020, doi: 10.3390/drones4040073.
- [70] S. Ramasamy, R. Sabatini and A. Gardi, "Avionics sensor fusion for small size unmanned aircraft Sense-and-Avoid," *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, 2014, pp. 271-276, doi: 10.1109/MetroAeroSpace.2014.6865933.
- [71] A. Canolla, M. B. Jamoom and B. Pervan, "Interactive multiple model sensor analysis for Unmanned Aircraft Systems (UAS) Detect and Avoid (DAA)," *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018, pp. 757-766, doi: 10.1109/PLANS.2018.8373451.
- [72] Y. Zou, H. Zhang, D. Feng, H. Liu and G. Zhong, "Fast Collision Detection for Small Unmanned Aircraft Systems in Urban Airspace," in *IEEE Access*, vol. 9, pp. 16630-16641, 2021, doi: 10.1109/ACCESS.2021.3053302.
- [73] L. Xi, J. Zhang, Y. Zhu, and W. Liu, "Simulation study of algorithms for aircraft trajectory prediction based on ADS-B technology," *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, Oct. 2008, doi: 10.1109/asc-icsc.2008.4675378.

Appendix A – CAA ATS National Airspace Classification



Appendix B – Global ADS-B Coverage Map



Appendix - C Detailed collision timing results

Flight			IMM CV-CA					IMM CV-CA-CT 2D					IMM CV-CA-CT 3D					
EZY89TP			Lookahead Time Step Ts					Lookahead Time Step					Lookahead Time Step					
collision point	Co-ordinates		Actual collision time	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s
1	x	3971037	170	165	180	175	180	175	165	180	175	180	175	165	180	175	180	175
	y	-147696																
	z	4975721																
2	x	3972050	311	300	300	300	300	315	300	300	300	300	315	300	300	300	300	315
	y	-164867																
	z	4974334																
3	x	3978248	433	420	420	425	420	420	420	420	425	420	420	420	420	425	420	420
	y	-167528																
	z	4968067																
4	x	3985204	584	570	580	575	570	595	570	580	575	570	595	570	580	575	570	595
	y	-172552																
	z	4961663																

Flight			IMM CV-CA					IMM CV-CA-CT 2D					IMM CV-CA-CT 3D					
RYR8213			Lookahead Time Step					Lookahead Time Step					Lookahead Time Step					
collision point	Co-ordinates	Actual collision time	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	
1	x	3978810	66	60	60	50	60	70	60	60	50	60	70	60	60	75	60	70
	y	-128312																
	z	4971069																
2	x	3974918	185	180	180	175	180	175	180	180	175	180	175	180	180	175	180	175
	y	-146130																
	z	4972348																
3	x	3983146	344	330	340	330	330	350	330	340	330	330	350	345	340	350	n/d	350
	y	-160182																
	z	4964701																
4	x	3985284	443	420	440	425	420	420	420	425	420	420	435	440	450	450	450	455
	y	-170934																
	z	4961850																

Flight			IMM CV-CA					IMM CV-CA-CT 2D					IMM CV-CA-CT 3D					
EXS8L8			Lookahead Time Step					Lookahead Time Step					Lookahead Time Step					
collision point	Co-ordinates	Actual collision time	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	
1	x	4053596	46	45	40	50	n/d	n/d	45	n/d	50	n/d	n/d	45	n/d	50	n/d	n/d
	y	-72353.8																
	z	4923996																
2	x	4044871	91	90	n/d	90	90	n/d	90	100	90	90	n/d	90	n/d	90	90	n/d
	y	-67537.3																
	z	4930134																
3	x	4033142	166	165	160	n/d	n/d	n/d	165	n/d	n/d	n/d	n/d	165	n/d	n/d	n/d	n/d
	y	-60155.4																
	z	4940888																
4	x	4024435	212	210	220	225	210	210	210	220	225	210	210	210	n/d	n/d	210	210
	y	-58492.8																
	z	4947645																

Flight			IMM CV-CA					IMM CV-CA-CT 2D					IMM CV-CA-CT 3D					
RYR56UE			Lookahead Time Step					Lookahead Time Step					Lookahead Time Step					
collision point	Co-ordinates		Actual collision time	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s	Ts = 15s	Ts = 20s	Ts = 25s	Ts = 30s	Ts = 35s
1	x	3974966	91	90	80	100	90	70	75	80	75	90	70	90	100	100	90	n/d
	y	-186609																
	z	4970988																
2	x	3979818	233	210	200	200	210	210	210	200	200	210	210	225	220	225	n/d	n/d
	y	-168932																
	z	4965807																
3	x	3982314	240	240	240	225	240	245	225	240	225	240	245	240	240	250	240	245
	y	-166775																
	z	4964907																
4	x	3985413	311	300	300	300	300	315	300	300	300	300	315	315	320	325	330	315
	y	-173368																
	z	4961720																

Flight			IMM CV-CA					IMM CV-CA-CT 2D					IMM CV-CA-CT 3D				
VLG8LV			Lookahead Time Step					Lookahead Time Step					Lookahead Time Step				
Intruder		Actual collision time	Ts= 15s	Ts =20s	Ts=25s	Ts=30s	Ts=35s	Ts= 15s	Ts =20s	Ts=25s	Ts=30s	Ts=35s	Ts= 15s	Ts =20s	Ts=25s	Ts=30s	Ts=35s
x (m)	4020820.597	81	70	70	85	n/d	n/d	70	75	80	n/d	n/d	70	75	n/d	n/d	n/d
y (m)	-75090.9785																
z (m)	4939240.11																
Vx (m/s)	-192.429962																
Vy (m/s)	21.58447859																
Vz (m/s)	-14.14272																
heading (deg)	-173.6																