



Combination of ADS-B and QAR Data for Mid-Air Collision Analysis

—

Kombination von ADS-B und QAR Daten für die Analyse von Annäherungen und Kollisionen von Luftfahrzeugen

Bachelor thesis

Author: Franziska Feigl

Matriculation Number: 03669438

Examiner: Prof. Dr.-Ing. Florian Holzapfel

Supervisors: Lukas Höhndorf, Phillip Koppitz

September 2018

Statutory Declaration

I, Franziska Feigl, declare on oath towards the Institute of Flight System Dynamics of Technische Universität München, that I have prepared the present Bachelor thesis independently and with the aid of nothing but the resources listed in the bibliography.

This thesis has neither as-is nor similarly been submitted to any other university.

Garching, 05 September 2018

Franziska Feigl

Kurzfassung

Ein Ziel der Flugdatenanalyse (Flight Data Monitoring) ist es potentielle Unfallrisiken zu erkennen. Dabei werden Quick Access Recorder (QAR) Daten analysiert, die an Bord eines Flugzeugs aufgezeichnet wurden. Allerdings enthalten QAR Daten keine Informationen über andere Flugzeuge in der Umgebung, was es schwierig macht, das Risiko für Flugzeugkollisionen zu analysieren. Diese Arbeit beschreibt Methoden um QAR mit Automatic Dependent Surveillance Broadcast (ADS-B) Daten zu kombinieren und so die QAR Daten mit Risikoindikatoren für eine Flugzeugkollision zu erweitern. Ein MATLAB-Algorithmus wird entwickelt, der ADS-B Daten von der Internetseite adsbexchange.com verwendet, wofür ein Datenmanagementsystem implementiert wird. Die QAR Positionsaufzeichnungen bekommen neue Zeitstempel basierend auf einer Referenzzeit, was es möglich macht sie mit den ADS-B Positionsaufzeichnungen anderer Flugzeuge zu vergleichen. Mit den neuen Zeitstempeln können Risikoindikatoren während eines Fluges berechnet werden, darunter die Distanz zum nächsten Flugzeug, die Luftraumdichte, eine Überprüfung für mögliche TCAS Warnungen und die Distanz zu anderen Flugzeugen im Landeanflug. Außerdem zeigt die Auswertung von ADS-B Daten von Flugzeugen verschiedener Airlines mögliche Anwendungen der Analyse der Risikoindikatoren um Risiken zwischen unterschiedlichen Fluggesellschaften und Flughäfen zu vergleichen.

Abstract

One goal of Flight Data Monitoring is to detect potential accident risks when analyzing Quick Access Recorder (QAR) data recorded on-board an aircraft. However, QAR data does not contain information about surrounding aircraft, which makes it difficult to analyze the accident type *Mid-Air Collision*. This thesis is concerned with finding means to merge QAR and Automatic Dependent Surveillance Broadcast (ADS-B) data to add risk indicators for a Mid-Air Collision to the QAR datasets. An algorithm is developed in MATLAB that uses ADS-B data from the website adsbexchange.com and a data management system is implemented. During merging, the QAR position recordings get new timestamps based on a reference time that make them comparable to the ADS-B position recordings of other aircraft. Using the new timestamps, risk indicators along a flight can be calculated, including the distance to the closest aircraft, the airspace density, a check for potential TCAS alerts and the distance to other aircraft during final approach. Furthermore, the evaluation of ADS-B data of aircraft from different airlines shows possible applications of analyzing the risk indicators for comparing risks between airlines and airports.

Table of Contents

| | |
|--|------|
| List of Figures..... | ix |
| List of Tables..... | xi |
| Table of Acronyms | xiii |
| Table of Symbols..... | xv |
| 1 Introduction | 1 |
| 2 Background information..... | 3 |
| 2.1 ADS-B data | 3 |
| 2.1.1 Message contents | 4 |
| 2.1.2 Selection of data source..... | 4 |
| 2.2 QAR data | 6 |
| 2.3 Mid-Air Collisions | 6 |
| 2.4 Distance calculation | 7 |
| 3 Data handling..... | 8 |
| 3.1 Data reading..... | 8 |
| 3.2 Data preprocessing | 10 |
| 3.2.1 Structuring..... | 10 |
| 3.2.2 Sorting data in sectors..... | 11 |
| 3.3 Merging ADS-B and QAR data..... | 13 |
| 3.3.1 Locating a specific flight in ADS-B data..... | 13 |
| 3.3.2 Merging timestamps | 14 |
| 3.4 Detecting ADS-B data errors | 18 |
| 3.5 Interpolating data of surrounding aircraft | 21 |
| 4 Analysis..... | 23 |
| 4.1 Comparison of ADS-B to QAR data quality | 23 |
| 4.2 Calculation of risk indicators..... | 27 |
| 4.2.1 Minimum distance to surrounding aircraft..... | 27 |
| 4.2.2 Airspace density..... | 29 |
| 4.2.3 Check for TCAS Resolution Advisories | 30 |
| 4.2.4 Minimum distance during final approach | 34 |
| 4.3 Analysis of example data..... | 37 |
| 4.3.1 Comparing risk indicators for different airlines | 37 |
| 4.3.2 Comparing distances in final approach for different airports | 40 |
| 5 Conclusions and perspective..... | 41 |
| References..... | i |

Appendixv

List of Figures

| | |
|--|---------|
| Figure 1-1: World annual traffic growth according to the Airbus Global Market Forecast [1] .. | 1 |
| Figure 1-2: Annual accident rates in commercial air traffic [2]..... | 1 |
| Figure 2-1: ADS-B system architecture and protocol hierarchy [14]..... | 3 |
| Figure 2-2: Location of recorded ADS-B messages from <i>ADSBexchange</i> on 18.04.2018, 12:00 am to 1:00 pm, shown in red, source of the map: MATLAB Mapping Toolbox | 5 |
| Figure 2-3: Azimuth for great circle distance..... | 7 |
| Figure 3-1: Data structure obtained from json file | 8 |
| Figure 3-2: Data structure before unpacking..... | 10 |
| Figure 3-3: Data structure after unpacking..... | 11 |
| Figure 3-4: Map of the world with sectors, source of the map: MATLAB Mapping Toolbox...12 | |
| Figure 3-5: Map of the world with sectors where ADS-B data exists marked green, source of the map: MATLAB Mapping Toolbox | 13 |
| Figure 3-6: Illustration of merging timestamps with a reference point | 15 |
| Figure 3-7: Distance between QAR and ADS-B position with same timestamps after merging, Flight A..... | 16 |
| Figure 3-8: Distance between QAR and ADS-B positions with same timestamps, Flight B, before correction | 16 |
| Figure 3-9: Illustration of timestamp correction | 17 |
| Figure 3-10: Distance between QAR and ADS-B position with same timestamps, Flight B, after correction | 18 |
| Figure 3-11: Altitude profile of Lufthansa aircraft D-AINB landing in London Heathrow with data coming from a faulty receiver..... | 18 |
| Figure 3-12: Altitude profile with altitudes from faulty receiver in red and altitudes from all other receivers in blue | 19 |
| Figure 3-13: Altitude profile with detected outliers marked red..... | 20 |
| Figure 3-14: Altitude profile after faulty receivers and outliers were removed | 20 |
| Figure 3-15: Great circle distance (blue) and direct connection (red)..... | 21 |
| Figure 3-16: Comparison of altitude interpolation between using exact data only and using all data during ascend of an aircraft | 22 |
| Figure 4-1: Distance from QAR to closest ADS-B positions..... | 23 |
| Figure 4-2: Example of lack of available position recordings leading to big distance between ADS-B and QAR data..... | 24 |
| Figure 4-3: Altitude profile from ADS-B and QAR data | 24 |
| Figure 4-4: Change of Latitude over Longitude for one part of the flight compared between ADS-B and QAR data..... | 26 |
| Figure 4-5: Change of Latitude over Time for one part of the flight compared between ADS-B and QAR data | 26 |
| <hr/> | |
| Combination of ADS-B and QAR data for Mid-Air Collision Analysis | |
| Franziska Feigl | Page ix |

List of Figures

| | |
|--|----|
| Figure 4-6: Minimum horizontal distance during flight..... | 28 |
| Figure 4-7: Altitude distance at minimum horizontal distance during flight | 28 |
| Figure 4-8: Number of surrounding aircraft during flight..... | 29 |
| Figure 4-9: Flight path with location of peaks in number of surrounding aircraft, source of the map: Google Earth | 29 |
| Figure 4-10: TCAS protection volume [4]..... | 30 |
| Figure 4-11: Example for the course of τ for a head-on collision (A) and two aircraft passing each other (B) | 32 |
| Figure 4-12: Comparison between τ and modified τ for example B, region where d is below $DMOD$ marked red | 33 |
| Figure 4-13: Distribution of vertical and horizontal τ values for the QAR flights | 34 |
| Figure 4-14: Distribution of vertical and horizontal τ values for the QAR flights - closeup with TA (green) and RA (red) threshold values | 34 |
| Figure 4-15: Distance to next aircraft in front | 35 |
| Figure 4-16: Number of aircraft detected landing in front | 36 |
| Figure 4-17: Distance to next aircraft landing behind..... | 36 |
| Figure 4-18: Comparison of mean number of surrounding aircraft during flight for different airlines..... | 38 |
| Figure 4-19: Comparison of mean horizontal distance to closest surrounding aircraft during flight between airlines..... | 39 |
| Figure 4-20: Comparison of mean vertical distance to closest surrounding aircraft during flight between airlines | 39 |

List of Tables

| | |
|---|----|
| Table 3-1: Stored fields | 9 |
| Table 3-2: Definition of variables in the file name for one sector | 12 |
| Table 4-1: Mean horizontal distance between QAR and ADS-B data for all flights..... | 23 |
| Table 4-2: Sensitivity level definition and alarm thresholds [4] | 31 |
| Table 4-3: Comparison of distances to next aircraft in front in approach for different airports | 40 |
| Table 5-1: Summary of all variables added to a QAR dataset..... | 41 |

Table of Acronyms

| Acronym | Description |
|----------------|--|
| ACAS | Airborne Collision Avoidance System |
| ADS-B | Automatic Dependent Surveillance Broadcast |
| ATC | Air Traffic Control |
| EOFDM | European Operators Flight Data Monitoring |
| FDM | Flight Data Monitoring |
| GNSS | Global Navigation Satellite Systems |
| ICAO | International Civil Aviation Organization |
| MAC | Mid-Air Collision |
| NaN | Not a Number |
| QAR | Quick Access Recorder |
| SMS | Safety Management System |
| SSR | Secondary Surveillance Radar |
| TCAS | Traffic Collision Avoidance System |
| UTC | Coordinated Universal Time |

Table of Symbols

Latin Letters

| Symbol | Unit | Description |
|-----------|---------------|--|
| t | s | Time |
| p | m | Distance of the direct connection between two points |
| d | m | Distance of the great circle connection between two points |
| r | m | Radius |
| \dot{d} | $\frac{m}{s}$ | Closure rate |
| τ | s | Time to closest point of approach |
| $DMOD$ | m | Horizontal distance threshold |
| $ZTHR$ | ft | Altitude distance threshold |
| TAU | s | Tau threshold |
| lat | $^{\circ}$ | Latitude |
| lon | $^{\circ}$ | Longitude |

Indices

| Symbol | Description |
|--------|--|
| R | Reference time |
| k | Timestep |
| $vert$ | Variable concerning vertical dimension |
| hor | Variable concerning horizontal dimension |
| mod | Modified tau |
| SL | Sensitivity Level |
| RA | Boundaries for ACAS Resolution Advisory |
| QAR | From QAR data |
| $ADSB$ | From ADS-B data |

1 Introduction

According to the Airbus Global Market Forecast, air traffic from now on will more than double in the next 20 years as visible in Figure 1-1 [1]. However, annual accident rates per million departures have been almost consistent over the last few years, as Figure 1-2 shows [2]. Hence, if no further actions are taken to improve aviation safety, the number of aircraft accidents are likely to increase with growing air traffic in the future.

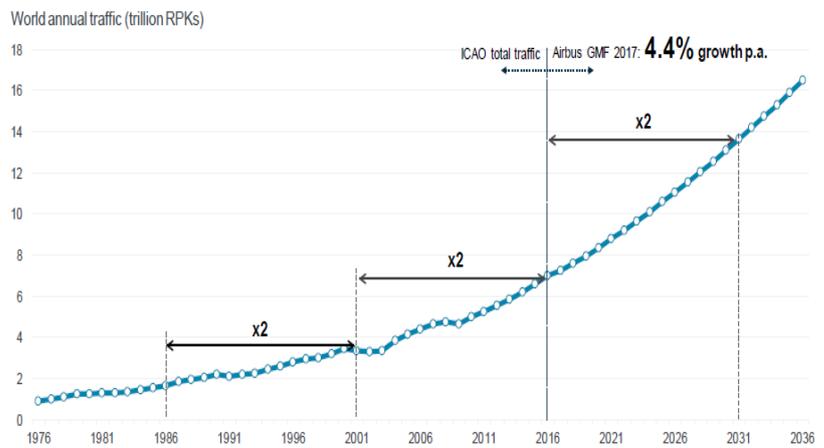


Figure 1-1: World annual traffic growth according to the Airbus Global Market Forecast [1]

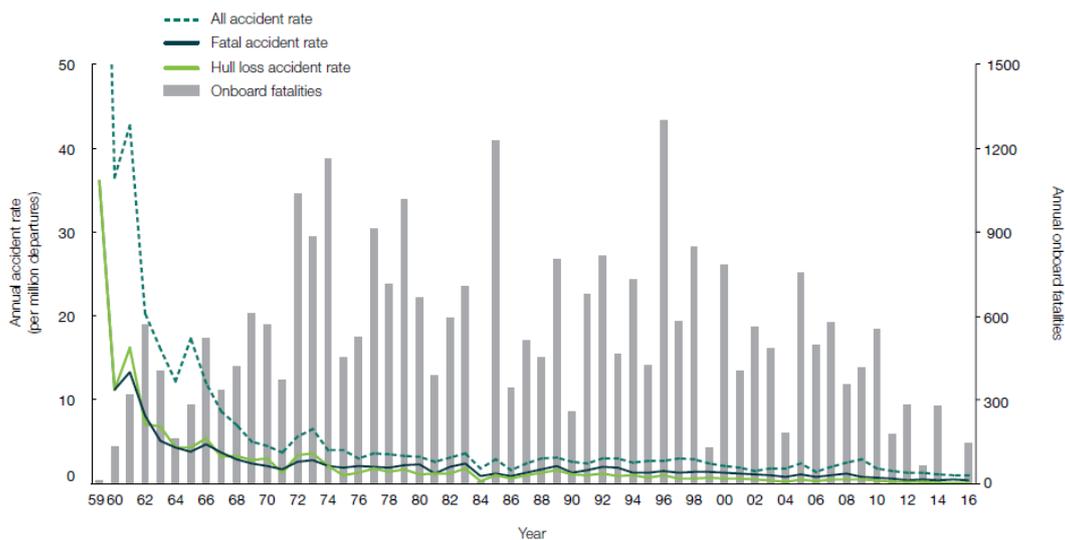


Figure 1-2: Annual accident rates in commercial air traffic [2]

One of the worst accidents in recent history was the collision of a Boeing 757 with a Tupolev-154 over Überlingen on the night of July 1, 2002, resulting in the death of 71 people. The accident got blamed partly on the air traffic controller, who realized the upcoming conflict between the two aircraft very late. The two aircraft were on crossing flightpaths on the same altitude. After the controller had finally noticed the danger, he instructed the Tupolev to descend to avoid a collision with the Boeing on the same altitude. At that time however, the Traffic Collision Avoidance System (TCAS) installed on both aircraft had already alerted the pilots telling the Boeing to descend and the Tupolev to climb. While the Boeing followed the TCAS order to descend, the Tupolev followed the instructions from the controller and therefore descended as well. This led to the collision of the aircraft [3]. After this collision actions were taken to standardize that all pilots have to follow TCAS instructions over those from air traffic control [4].

Another one of the measures that has been taken to identify and mitigate accident risks, is that airlines have to implement a Safety Management System (SMS), which is a systematic approach to control safety risks. The SMS is composed of four components: safety policy and objectives, safety risk management, safety assurance and safety promotion. One important part of this system to identify hazards and risks and to assure safe performance is Flight Data Monitoring (FDM) [5].

The European Operators Flight Data Monitoring (EOFDM) forum wants to support operators to improve their flight data monitoring for maximum safety benefits. The main data source for data recorded on-board an aircraft is the Quick Access Recorder (QAR). One of the current objections of the EOFDM forum is to find means to monitor the risk for the incident type Mid-Air Collision (MAC), which is a collision of aircraft while both are in flight. This incident type includes also all other separation related occurrences [6].

As a first step, the EOFDM forum working group A defined precursors for a MAC that can be monitored through FDM programs. Among others, they named high airspace density as a contextual factor that might increase the risk for a MAC. However, for contextual information about the air traffic it is not enough to have only the data recorded on-board one aircraft [7]. This is where Automatic Dependent Surveillance Broadcast (ADS-B) data comes into play, as it can be used to get information about other aircraft in the airspace. Many aircraft are broadcasting ADS-B data that includes information about their current position. This data can be received with antennas on the ground. It is used by flight tracking websites such as *ADSBexchange* and *The Opensky Network*, that collect ADS-B data from receivers all around the world. From these websites it is possible to obtain the data for research.

The main goal of this thesis is to find risk indicators for a MAC while combining ADS-B and QAR data. To achieve this first a data management system has to be implemented for the ADS-B data. Afterwards, means have to be found to merge QAR and ADS-B data. Finally, MAC risk indicators are defined, and the ADS-B data is used to enrich the QAR data with the information about MAC risk indicators.

In the following thesis, background information about the utilized data and the definition of MAC is given in chapter 2. In chapter 3, the algorithm to handle the ADS-B data and how it is merged with a QAR dataset, as well as the detection of data errors and interpolation of the ADS-B data is described. Afterwards, the risk indicators for a MAC that can be calculated using ADS-B data are depicted in section 4.2. In section 4.3, some examples are given of what could be done in analyzing flights of a certain fleet, before a conclusion and perspective of further work is provided in chapter 5.

2 Background information

Before starting with the description of the developed algorithm, this chapter provides background information about the characteristics of ADS-B and QAR data, as well as the incident type MAC.

2.1 ADS-B data

With Automatic Dependent Surveillance Broadcast (ADS-B), aircraft transmit information about their identity, current position, speed and track. It is broadcasted **automatically** periodically without necessary interrogation, it is **dependent** on information from the aircraft's on-board equipment coming from global navigation satellite systems (GNSS) or on-board sensors, and it is used for **surveillance** by other aircraft and ground stations [8]. This is different to the signals used for Secondary Surveillance Radar (SSR), called Mode-S, which are only sent as a response to interrogation from a ground station or another aircraft [9]. There is a distinction between *ADS-B Out*, which is the ability to broadcast ADS-B messages, and *ADS-B In*, which is the ability to receive and process these messages [10]. The ADS-B system architecture is depicted in Figure 2-1. The equipment with an *ADS-B Out* transmitter will become mandatory in the EU for all aircraft with a maximum takeoff weight greater than 5700 kg or a maximum cruising true airspeed capability greater than 250 knots in 2020 [11], the same applies to all aircraft flying in dedicated airspace in the US [12]. In 2016, around 70 % of Mode-S equipped aircraft in Europe and the US were already capable to broadcast ADS-B [9]. The use of ADS-B in Air Traffic Control (ATC) improves accuracy of position information, and thus allows a higher density of aircraft in the airspace while still maintaining safety. Another advantage is that it can be used to receive information from aircraft on-ground, which is not possible with conventional radar. As the installation of an ADS-B receiver is far cheaper than a conventional radar station, it is used to improve airspace surveillance coverage in big, sparsely populated areas. Finally, the information is also received by other *ADS-B In* equipped aircraft, so that pilots have more awareness of air traffic around them and can avoid potential collisions [13]. The technology used for *ADS-B Out* is mostly an extension, called Extended Squitter, to the 1090MHz Mode-S system already used for SSR. The signal is transmitted line-of-sight to the receiver, which means that it works best if no obstacles exist [8]. Depending on the specific conditions, such as environment, altitude and terrain, ADS-B can have a range of more than 120nm [14].

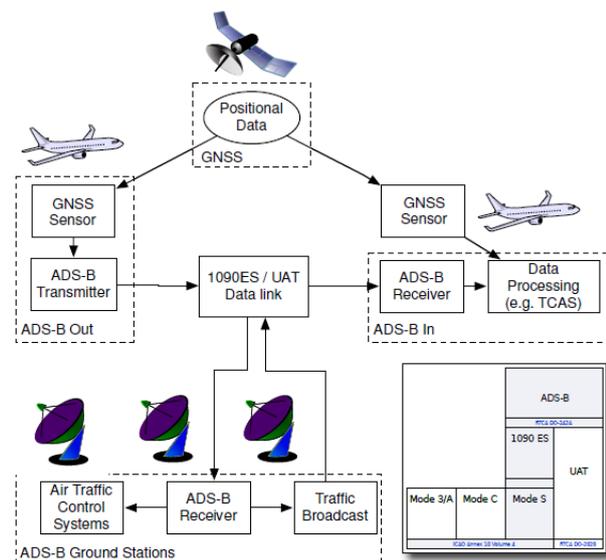


Figure 2-1: ADS-B system architecture and protocol hierarchy [14]

2.1.1 Message contents

An ADS-B message includes the following information [14]:

Aircraft identification

Using the 24-bit ICAO address. This is the six-digit hexadecimal identifier broadcasted by the aircraft over the air in order to identify itself. Blocks of these codes are assigned to countries by the International Civil Aviation Organization (ICAO). Each country then assigns individual codes to aircraft registered in that country [15]. For example, the Lufthansa aircraft with the registration *D-AINB* has the ICAO address 3C65C2.

Airborne position

Aircraft position is broadcasted twice per second and includes aircraft horizontal position (latitude and longitude based on the WGS84) and aircraft barometric altitude in a compressed format which allows a vertical accuracy of +/- 25 ft

Airborne velocity

Including speed and heading information, broadcasted in North-South and East-West components of the velocity with an accuracy of 4 knots. If available, which is mostly the case, this velocity is the ground speed. In addition, the vertical rate is included in 64 ft/min steps.

Status messages

Indicating the accuracy or uncertainty of the broadcasted position data, information on emergencies and other information

Not all aircraft that are equipped with ADS-B report all of the information, but sometimes only parts of it. Strohmeier et al. [13] found out that although ADS-B messages are transmitted in a high frequency they did not receive them all due to message collisions. This was caused by too much traffic on the 1090 MHz channel, as it is also used for Mode-S SSR transmissions. Another issue with ADS-B that has been addressed is the security risk caused by hackers who might insert false ADS-B messages or alter existing ones [13].

2.1.2 Selection of data source

The ADS-B data can be received not just by other aircraft or ATC ground stations, but also by everyone else who sets up an inexpensive antenna. There are several public ADS-B data sources available on the internet. They all collect the data from their own antennas and many private receivers around the world. However, they differ in their intentions to collect the data, in terms of how much of the historical data they provide for free and how they filter and alter the data. This chapter gives a short comparison of available websites offering ADS-B data and shows why *ADSBexchange* was chosen as data source for this thesis.

Flightradar24.com is a flight tracking website with the biggest network of around 17,000 ADS-B data receivers. They provide live tracking of aircraft, which is interesting for private users who want to track a specific flight. However, to analyze past flights with recorded QAR data in this thesis, historical ADS-B data is needed. On *Flightradar24.com* historical data is only available for specific flights with a subscription that one has to pay for. Even then, one can only download data for a limited amount of flights every month [16].

Flightaware.com is the biggest commercial flight tracking website that obtains its data from air traffic control systems, its own community of around 15,000 data receivers and the satellite

supported Aireon ADS-B. As with *Flightradar24.com*, live tracking of specific aircraft is publicly available. Furthermore, one can see the tracking protocol of the last three months for every aircraft for free, so it is possible to pull this information from the website. However, information about all aircraft at a certain location and time was needed for this thesis. To gain this, it is necessary to download the tracking protocol separately for every aircraft. In doing so, it would be almost impossible to find all relevant aircraft, as one would have to know all of their flight numbers. Another problem is that quite a lot of information is lost because the protocol only contains data in steps of around 30 seconds. Again, the full historical data is only available for purchase [17].

Opensky-network.org is a non-profit association with a smaller network of around 1,000 receivers which aims to make the ADS-B data accessible for research. As the complete historical database is available for research purposes, it would have been an option to use this website as data source [18].

ADSExchange.com is a cooperative of over 1,000 data receivers from around the world and at the beginning of this thesis provided the complete historical database for non-commercial use. An advantage comparing to the other websites mentioned above is, that *ADSExchange* does not filter any of the aircraft and for example also provides ADS-B data broadcasted by military and state aircraft. The other websites do not show these for privacy reasons, but *ADSExchange* claims, that it will not remove any information unless they are legally obliged to do so. It used to be possible to download the complete worldwide dataset for one day contained in several json (JavaScript Object Notation) files directly from the website. *ADSExchange* started storing historical data on 09.06.2016, so data is only available starting from that date. The data coverage is good in Europe and North America, but quite sparse in other regions of the world as shown in Figure 2-2[15].

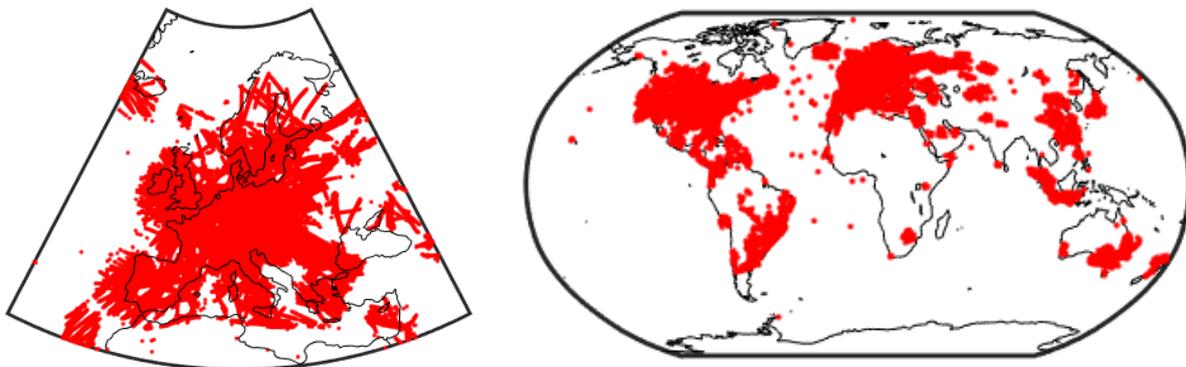


Figure 2-2: Location of recorded ADS-B messages from *ADSExchange* on 18.04.2018, 12:00 am to 1:00 pm, shown in red, source of the map: MATLAB Mapping Toolbox

As with *ADSExchange* the raw data was the easiest to access and the json files are easy to decode in MATLAB, this website was chosen as the data source for this thesis. However, during the time this thesis was written, they changed the access regulations for the historical data. Now access is only provided if one purchases the data transfer and becomes a feeder oneself.

2.2 QAR data

A Quick Access Recorder (QAR) is a device onboard an aircraft recording raw flight data to make it quickly accessible for analysis. The QAR receives its inputs from the flight data acquisition unit, which can record over 2000 flight parameters. Unlike the compulsory flight data recorder that is used to reconstruct the circumstances leading to an accident, the QAR data is used by airlines in their FDM to improve their safety and operating procedures [19]. The parameters utilized within this thesis (which are a subset of all available parameters) are: time span since beginning of recording, latitude, longitude, barometric height, vertical speed, ground speed, heading ψ and the weight-on-wheels information. UTC time was not available in that particular dataset but would have been useful to locate the flights in the ADS-B data. The QAR data has a sampling rate of up to 16 Hz for some parameters, but the position, the most important information for this thesis, was only recorded every two seconds in the available dataset.

2.3 Mid-Air Collisions

The aviation occurrence type Mid-Air Collision (MAC) is defined by the ICAO as “Airprox, TCAS alerts, loss of separation as well as near collisions or collisions between aircraft in flight” [6]. This definition includes all separation related issues of aircraft in flight. A collision between two aircraft is one of the most dangerous accidents, because it usually results in a loss of both aircraft [20].

An *Airprox* (Aircraft proximity) is a situation where either pilot or ATC assumes that two aircraft were so close to each other that their safety was not assured [21]. This definition is very subjective and not bound to exact measures. This is different for a *loss of separation* which occurs if set separation minima in controlled airspace are breached [22]. Generally, the minimum separation standards for an aircraft in cruise are 2,000 ft vertical separation and 5 NM longitudinal separation, but this can be reduced to 1,000 ft vertical separation if the aircraft is below 29,000 ft and to 2.5 NM longitudinal separation during final approach. However, the applied separation minima depend a lot on the relative trajectories of the aircraft involved, therefore these numbers are only a point of reference [23]. If a collision can still be avoided, but the two aircraft were less than 500 ft apart or an actual risk of a collision happening was seen, this is called a *Near Mid-Air Collision* and has to be reported to the authorities [24].

There are several possible precursors and contextual factors leading to a MAC event named by the EOFDM working group including among others level bust or lateral deviation [7]. As there are several barriers preventing a loss of separation, including ATC and onboard systems, usually multiple failures have to occur before a MAC happens. This is also one of the reasons why it is a very rare event: Since 2007 only two fatal MAC accidents have occurred with commercial jet airplanes [2].

If two aircraft already have come too close, the most important barrier preventing them from a collision is the Airborne Collision Avoidance System (ACAS), issuing Resolution Advisories telling the pilot what to do to avoid a collision. The Traffic Collision Avoidance System TCASII, which is a commonly used version of ACAS, uses the Mode-S data link to interrogate other aircraft about their position and processes the replied information to detect any threats. Surrounding aircraft are then also depicted on a traffic display. To reduce the number of

surveillance interrogations, hybrid surveillance was introduced. This uses ADS-B information, which does not need interrogation for aircraft still far enough away not to be a threat [4].

2.4 Distance calculation

All horizontal distances in this thesis are calculated with the MATLAB *distance* function, which computes the distances on a sphere or ellipsoid. This function uses the WGS84 coordinates *lat* and *lon* of two aircraft (Index 1 and 2) and the earth radius *r*, which is set to the average of 6371km. The output is the great circle distance *d* between the two aircraft and the azimuth of the second aircraft with respect to the first, which is the angle at which the arc crosses the meridian containing the first aircraft [25], as depicted in Figure 2-3. The great circle distance *d* is calculated in the MATLAB function with the Haversine formula (2-1), taken from the MATLAB source code. In this formula *atan2* is the four-quadrant inverse tangent.

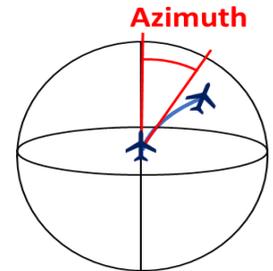


Figure 2-3: Azimuth for great circle distance

$$d = 2 \cdot r \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$a = \sin^2\left(\frac{\text{lat}_2 - \text{lat}_1}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\text{lon}_2 - \text{lon}_1}{2}\right) \quad (2-1)$$

The *distance* function allows furthermore to compute the distances on the WGS84 ellipsoid instead of a sphere. This however takes more time to compute, so the approximation with the mean earth radius was chosen for this thesis to allow a more efficient calculation of distances.

3 Data handling

This chapter explains how the ADS-B data is obtained and preprocessed, so that it can be used for the analysis described in chapter 4. All of the data handling was done in MATLAB. The reading and contents of the ADS-B data from *ADSExchange* is described in section 3.1. The algorithm to process, sort and store the raw data is depicted in section 3.2. Subsequently, section 3.3 explains how ADS-B and QAR data are merged so that the ADS-B data can be used to enrich the QAR data. Section 3.4 describes how ADS-B data errors are detected and deleted, before the ADS-B data is interpolated to fill missing entries in the dataset, as described in section 3.5.

To process this large amount of data in MATLAB in a reasonable time, a considerable effort was used to make the storage process as efficient as possible. Parallel to this thesis, another semester thesis was conducted by Camilla Zimmermann, who also used the ADS-B data for MAC analysis, but with a location-based approach. When deciding about the final data structure, it was important to consider that the solution had to work for both usages of the ADS-B data. Therefore, this part of the algorithm was a team effort.

3.1 Data reading

ADS-B data for a specific date *yyyy-mm-dd* is downloaded using the link <http://history.adsbexchange.com/Aircraftlist.json/yyyy-mm-dd.zip>. When the download is unzipped, it results in several json files each containing the ADS-B data for one minute. This means 1,440 files for one day, named *yyyy-mm-dd-hhmmZ.json* containing all data received around the world during this minute, with *hhmm* being the UTC time and *Z* (short for Zulu) a suffix included in all filenames. These files are then decoded in MATLAB using the function *jsondecode*, which gives back a MATLAB struct with the data structured as shown in Figure 3-1.

| acList | | | | | | | | | | | | | | | | | | | | | |
|----------|------|-----|------|-----|-----|-----|----------|------|-----|------|-----|-----|-----|----------|------|-----|------|-----|-----|-----|-----|
| ACStruct | | | | | | | ACStruct | | | | | | | ACStruct | | | | | | | ... |
| Icao | Rcvr | Lat | Long | Alt | Cos | ... | Icao | Rcvr | Lat | Long | Alt | Cos | ... | Icao | Rcvr | Lat | Long | Alt | Cos | ... | ... |
| | | | | | | | | | | | | | | | | | | | | | |

Figure 3-1: Data structure obtained from json file

To begin with, the contents of this struct will be described. Every struct element, called *ACStruct* from now on, contains the information for one aircraft recorded by one receiver. There can be several *ACStructs* for one aircraft in one file if there were several receivers that recorded data from this aircraft during the respective minute. The fields in the *ACStructs* contain only one scalar value each with the last received position, except for the *Cos* field. The *Cos* field contains an array with sets of time, latitude, longitude and altitude position reports that were picked up from the receiver during the minute of the file.

The unique identifier included in the data to connect it to a certain aircraft is the ICAO identifier code. The aircraft registration is also available in the dataset from *ADSExchange*. However, it is not originally broadcasted in the ADS-B message but added by *ADSExchange* via an external database.

A full list of variables available from *ADSBexchange* is included in the appendix. Table 3-1 shows the variables that were identified to be reliable and useful for later analysis with a short description of their content, unit and the data type in which they are stored. The variables listed as mandatory were recognized to be necessary for a useful position recording, the others are beneficial, but do not have to be included. For the variables containing numeric values, also the resolution of the data is given. The number in this column is the smallest possible interval between values for this variable. It is worth to mention that the altitude information is only precisely rounded to 25 feet intervals outside of the *Cos* element, which is as precise as the altitude is transmitted in the ADS-B message [13]. This means that there is only one precise altitude report per minute, while the altitude information in the *Cos* arrays is rounded to 500 feet intervals. This leads to a stepped altitude profile if all altitude recordings are used. The frequency of the position recordings in the *Cos* arrays can be up to several recordings per second but is very irregular.

The time is given in epoch milliseconds, which is the number of milliseconds since January 1, 1970 midnight in UTC [26]. As this is not easily interpretable, it is rounded by the algorithm to full seconds and converted to the number of seconds since 00:00 UTC the date the data is from.

| Name | Mandatory | Content | Unit | Accuracy | Data Type |
|----------------|------------------|--|--------------------|---|------------------|
| Icao | Yes | 6-digit hexadecimal ICAO aircraft identifier | - | | String |
| Reg | Yes | Aircraft Registration number added via a database | - | | String |
| Alt | Yes | Standard pressure altitude | Feet | 25 | Double |
| Lat | Yes | Latitude | Degree | 0.0001 | Double |
| Long | Yes | Longitude | Degree | 0.0001 | Double |
| PosTime | Yes | Time when position was reported | Epoch milliseconds | 1 | Double |
| Spd | No | Ground speed | Knots | 0.1 | Double |
| Trak | No | Track angle across the ground | Degree | 0.1 | Double |
| Vsi | No | Vertical Speed | Feet/minute | 64 | Double |
| Cos | Yes | All received Lat, Long, PosTime, Alt (n/4 positions) | See above | Same as for the variables above, except for Alt it is 500 here | Nx1double array |
| Rcvr | No | Receiver identifier, available from April 27, 2017 | - | | Double |

Table 3-1: Stored fields

3.2 Data preprocessing

3.2.1 Structuring

The algorithm to preprocess the data is described next. The first step is to decide which of the available ACStruct data packs for one aircraft to store. Therefore, the data of one hour with the variables listed in Table 3-1 are sorted into a struct containing one field for every aircraft. The goal is to reduce the data size and thus reduce runtime by storing only one ACStruct data pack per aircraft per minute.

To keep relevant information only, the variables from one ACStruct are only transferred into this struct if it contains all the variables listed as mandatory in Table 3-1. This way uncomplete data is filtered out, where for example only latitude and longitude are given, but no altitude. A position report is of no use if it does not contain all of the mandatory variables, but the optional variables like speed might sometimes be missing while the position report is still good. This does not mean that an aircraft does not send out the mandatory information at all if some of it is missing in one ACStruct, as the information might be available from other receivers.

If there is more than one ACStruct available that includes all necessary information for one aircraft, only the one with the most position recordings is kept, which means it has the longest Cos array. This sorting is then repeated for all files of one hour. The result is a struct with one field for every recorded aircraft during this hour, with a separate line for the data of one minute, but with all other position recordings received during that minute together in the Cos array. This data structure is depicted in Figure 3-2. During the process, the program keeps track of how many position reports in total (from ACStruct top level and Cos arrays) are stored, as this is important to pre-allocate the arrays where the data is stored in the next step. The pre-allocation and the reduction to only necessary data proved to reduce the runtime significantly, which is why this intermediate step is required.

| | Icao | Reg | Alt | Lat | Long | PosTime | Spd | Trak | Vsi | Cos | Rcvr |
|-----------|--------|--------|------------|------------|-------------|---------|---------|---------|------------------|--|---------------|
| ICAOABCE1 | ABCDE1 | XY-XYZ | Altitude x | Latitude x | Longitude x | Time x | Speed x | Track x | Vertical Speed x | Lat x1 Lonx1 Timex1 Altx1 ... Latxn Lonxn Timexn Altxn | Receiver ID x |
| | ABCDE1 | XY-XYZ | Altitude y | Latitude y | Longitude y | Time y | Speed y | Track y | Vertical Speed y | Laty1 Lony1 Timey1 Alty1 ... | Receiver ID y |
| ICAOABCE2 | Icao | Reg | Alt | Lat | Long | PosTime | Spd | Trak | Vsi | Cos | Rcvr |
| | ABCDE2 | XY-LMN | Altitude z | Latitude z | Longitude z | Time z | Speed z | Track z | Vertical Speed z | Latz1 Lonz1 Timez1 Altz1 | Receiver ID z |

Figure 3-2: Data structure before unpacking

Once all relevant ADS-B data of one hour is stored in the struct, the Cos array has to be unpacked to separate the different variables time, latitude, longitude and altitude and to bring these position recordings to the same level as the position recordings directly from ACStruct.

Figure 3-3 shows the data structure after unpacking. Every position record corresponds to one line in the new data structure. During unpacking, the data is stored in separate string and double arrays for every variable, that are pre-allocated with the complete number of datapoints counted in the previous step. This turned out to be the fastest way, as MATLAB is fast in accessing arrays. The datapoints for separate aircraft are not separated anymore, but all stored in the same arrays, as every position record can be matched to an aircraft by the ICAO code. For the position recordings coming from a Cos array speed, vertical speed and track are not known, so these positions in the respective arrays are filled with NaN (not a number). ICAO code, Registration and Receiver ID are copied from the ACStruct top level. After all data points are transferred to the arrays, all separate arrays are put together into one table. In the first versions of the data pre-processing algorithm, the data was written directly into a struct or table during the unpacking, but both versions were significantly slower.

| Icao | Reg | Alt | Lat | Long | PosTime | Spd | Trak | Vsi | Rcvr |
|--------|--------|-------------------|-------------------|-------------------|--------------------|---------|---------|------------------|---------------|
| ABCDE1 | XY-XYZ | Altitude x | Latitude x | Longitude x | Time x | Speed x | Track x | Vertical Speed x | Receiver ID x |
| ABCDE1 | XY-XYZ | Alt _{x1} | Lat _{x1} | Lon _{x1} | Time _{x1} | NaN | NaN | NaN | Receiver ID x |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ABCDE1 | XY-XYZ | Alt _{xn} | Lat _{xn} | Lon _{xn} | Time _{xn} | NaN | NaN | NaN | Receiver ID x |
| ABCDE1 | XY-XYZ | Altitude y | Latitude y | Longitude y | Time y | Speed y | Track y | Vertical Speed y | Receiver ID y |
| ABCDE1 | XY-XYZ | Alt _{y1} | Lat _{y1} | Lon _{y1} | Time _{y1} | NaN | NaN | NaN | Receiver ID y |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ABCDE2 | XY-LMN | Altitude z | Latitude z | Longitude z | Time z | Speed z | Track z | Vertical Speed z | Receiver ID z |
| ABCDE2 | XY-LMN | Alt _{z1} | Lat _{z1} | Lon _{z1} | Time _{z1} | NaN | NaN | NaN | Receiver ID z |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Figure 3-3: Data structure after unpacking

3.2.2 Sorting data in sectors

The next processing step is to divert the big table into smaller sections. This is necessary as always loading the complete table, even if only a small portion of the data is needed, is very inefficient. Sorting by location was seen as the most useful option, as it allows loading data specifically for certain regions. The partitioning is conducted by using latitude and longitude information and sorting the data into sectors of 10x10 degrees. This separates the earth into 648 sectors as shown in Figure 3-4.

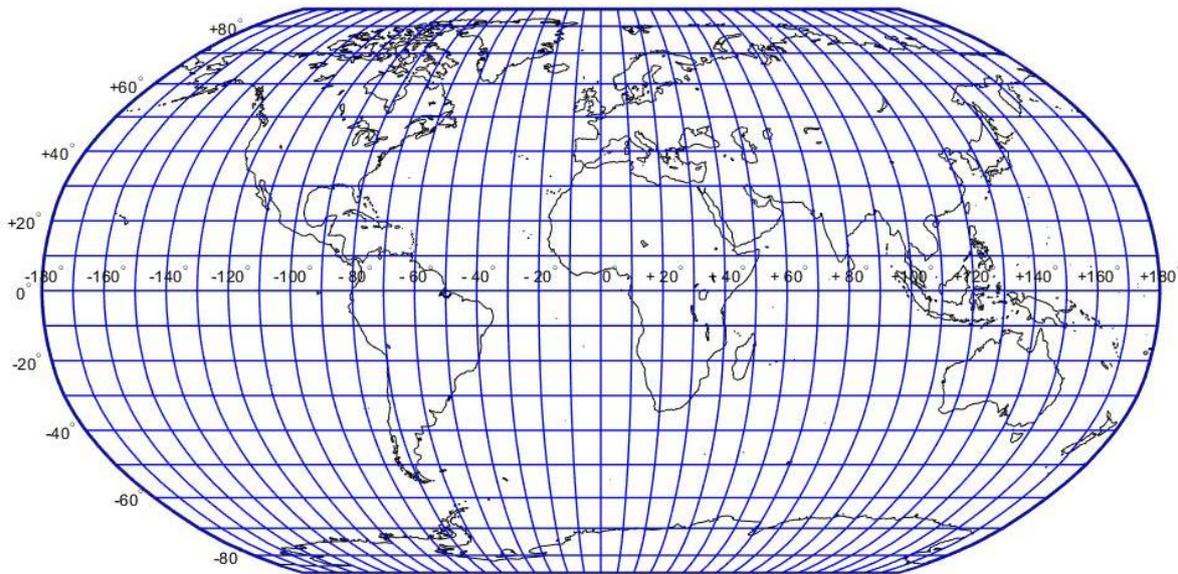


Figure 3-4: Map of the world with sectors, source of the map: MATLAB Mapping Toolbox

This partitioning allows a simple query on latitude and longitude to find the associated sectors efficiently. All datapoints that are found for a sector are saved in a *.mat* file named "sector"+*sign1*+*lat*+*sign2*+*long*+*on*+*date*+*hour*+*hour*, with the variables as defined in Table 3-2. For example, the sector reaching from 40-50°N and 20-30°E with data from 11:00 am UTC to 12:00 am UTC on 05.08.2018 is referred to as "sectorn40e20on20180805hour11".

| Variable | Definition |
|----------|--|
| sign1 | "n" or "s" depending on the hemisphere of the position |
| lat | latitude of the south edge of the sector |
| sign2 | "e" or "w" depending on the hemisphere of the position |
| long | longitude of the west edge of the sector |
| date | date which the data is from |
| hour | stored hour of data in UTC time |

Table 3-2: Definition of variables in the file name for one sector

Not for all sectors ADS-B position recordings can be found. Especially over oceans and sparsely populated areas there are no ADS-B receivers transmitting data to *ADSBexchange*. Figure 3-5 shows a map of the world where only sectors that contained data from 12:00 am UTC to 1:00 pm UTC on 18.04.2018 are marked green. There are some outlier sectors over the ocean very far north and south. When checking the data in these sectors it revealed that they originated from wrong position recordings.

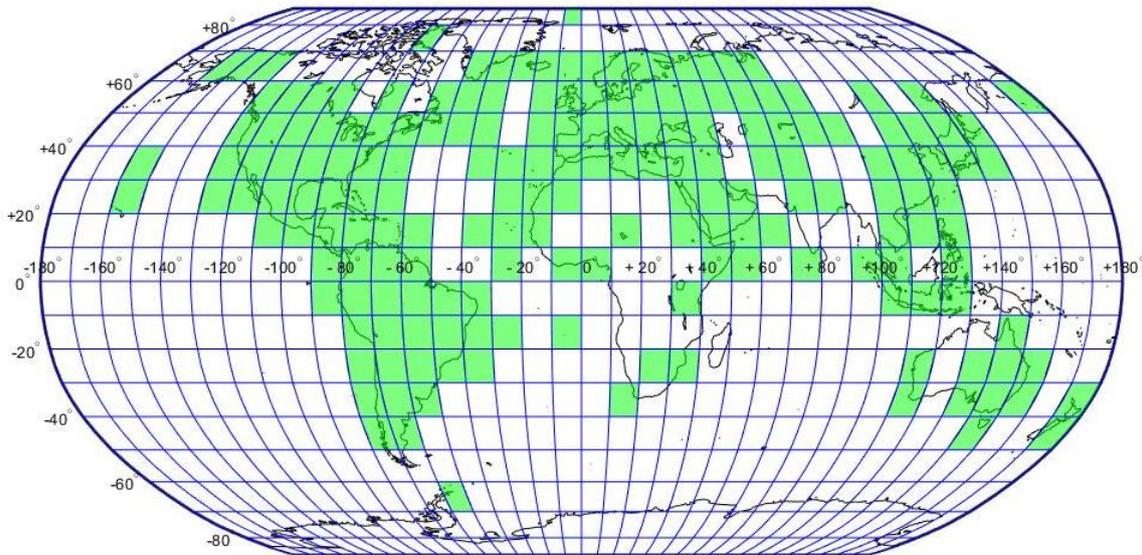


Figure 3-5: Map of the world with sectors where ADS-B data exists marked green, source of the map: MATLAB Mapping Toolbox

3.3 Merging ADS-B and QAR data

To use ADS-B data for evaluating the air traffic surrounding a flight for which QAR data is available, the timestamps of ADS-B and QAR data have to be merged. The aim when merging the datasets is not to expand the QAR position recordings of one aircraft with the ADS-B position recordings of that aircraft, because the QAR data is assumed to be sufficiently accurate concerning the position information about the own aircraft. The aim is rather to find the ADS-B position recordings of other aircraft surrounding the considered aircraft. This means that the ADS-B data of the considered aircraft itself is only used to find the correct timestamps for the QAR data. These timestamps are used in further calculations with the ADS-B data of surrounding aircraft.

The first step before being able to compare QAR and ADS-B data is to convert the QAR data to the same units as the ones that are used in the ADS-B data. For all of the following calculations, only QAR data of airborne aircraft is used, because this is where a MAC can happen. Therefore, only QAR data from when there is no weight-on-wheels signal is used.

3.3.1 Locating a specific flight in ADS-B data

First, the ADS-B data for the same flight as given in the QAR data has to be found. Flight number or destination airport however are not included in the ADS-B data. Either registration number or ICAO code should be known of the aircraft that the QAR data is from, as well as the date on which the QAR data was recorded. With this information, all ADS-B data of the aircraft on the respective date can be obtained.

Usually the UTC time is included in QAR datasets, therefore it can be used to find the ADS-B data of the aircraft from the time when the QAR data was recorded. As the UTC time was not available in the QAR datasets utilized within this thesis, another way to find a specific flight in the ADS-B data had to be found.

Therefore, an algorithm was developed that checks the ADS-B data of one day for breaks of more than 1,000 seconds, which corresponds to 16.7 minutes. Thereby, it partitions the data into sections that are assumed to be separate flights. The next step is to find out which of these separate flights is the desired flight for the QAR data. As origin and destination airport are known based on the QAR data, the separate flights in the ADS-B data are searched for those. To find the origin and destination airport for the ADS-B flights, the distances from their start and endpoints to the desired origin and destination airports are calculated, and the closest airport within 100 km is added as departure/destination airport to the ADS-B flights. The matching flight for the QAR data is the one that connects the desired origin and destination airport. This worked well for the available QAR flights, but it would not work if the ADS-B data recording stopped too far from the airport or if the same aircraft did the same trip twice on the same day. Then, the algorithm could not detect which of the two flights belongs to the QAR dataset. The algorithm could be further improved to work more reliable, but as it was only constructed to work with the available QAR data despite the missing UTC time, this was not necessary for this thesis. It is assumed that the algorithm will not be necessary for future work, because then the UTC time should be available in the recorded QAR data.

3.3.2 Merging timestamps

After the flight has been detected in the ADS-B data like described in section 3.3.1, there are now two datasets for the same flight: one from ADS-B data and one from QAR data. To be able to add information to the QAR data derived from the ADS-B data, for example the minimum distance to another aircraft, it is necessary to know which QAR position recording is from the same time as an ADS-B position recording. Thus, the QAR dataset needs timestamps that are comparable to the timestamps used in the ADS-B data. The algorithm used to achieve this is described in this section.

A reference point is used, from which then the other timestamps of the QAR data are calculated. This is necessary because the QAR dataset only contains the time elapsed since the beginning of the recording and not the UTC time. This step is illustrated in Figure 3-6. First, the distances between all points of the flight's ADS-B data and the QAR data are calculated. Then, the pair of ADS-B and QAR position recordings with the smallest distance to each other are taken as reference points. For the following calculation it is assumed that these two points are from exactly the same time t_R , which is the time stamp of the ADS-B reference point in seconds since 00:00 UTC that date. It can be assumed that these position recordings were recorded at the same time, as on all tested flights the distance between the reference points was only somewhere around or even below 2 m.

The other timestamps for the QAR data are then computed from the reference time t_R and the time difference from the datapoint to the reference point. For example, if there is a QAR position recording recorded 2 seconds before the reference point, this gets the timestamp $t_R - 2s$. After calculating this for every QAR position recording there is a timestamp for every QAR datapoint that can be used to find matching ADS-B datapoints from the same point in time.

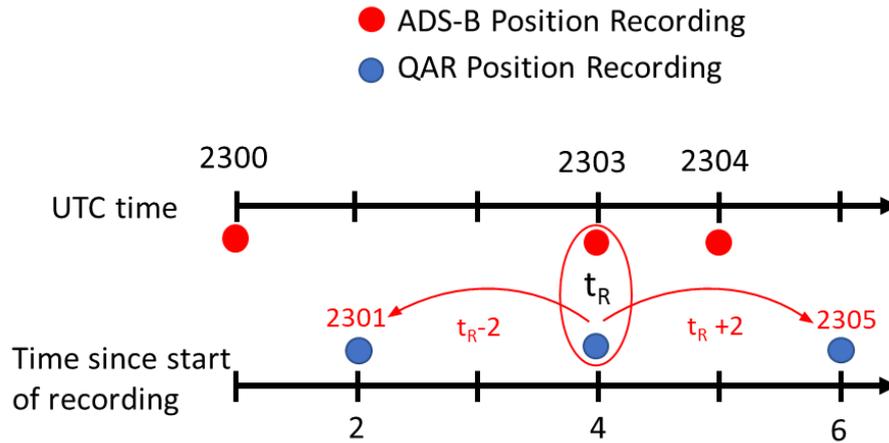


Figure 3-6: Illustration of merging timestamps with a reference point

To get the ideal set of QAR time steps for which the distances to the ADS-B dataset are minimal, the new QAR timestamps in vector \vec{t}_{QAR} are then moved ± 10 s in 1s steps from the ones calculated from t_R . For every step of the flight, that is from takeoff to touch down, the horizontal distances between QAR and ADS-B points with the same timestamp are calculated. After the calculation, the timeline with the smallest median of the distances is used. This means that from the vectors with the QAR timestamps $\vec{t}_{QAR,n} = \vec{t}_{QAR} + n$ for $n = -10, -9 \dots 9, 10$ the one for which the expression (3-1) is minimal is chosen. \vec{pos}_{QAR} is a vector of all QAR position recordings, \vec{pos}_{ADSB} contains all ADS-B position recordings and \vec{t}_{ADSB} the matching ADS-B timestamps.

$$median \left(distance \left(\vec{pos}_{QAR}(k), \vec{pos}_{ADSB} \left(\vec{t}_{ADSB} == \vec{t}_{QAR,n}(k) \right) \right) \right) \tag{3-1}$$

$k = 1 \dots \text{number of QAR positions}$

Figure 3-7 shows the distances between QAR and ADS-B positions with the same timestamps after merging along one of the flights, Flight A, which shows that the position recordings are mostly closer than 500m.

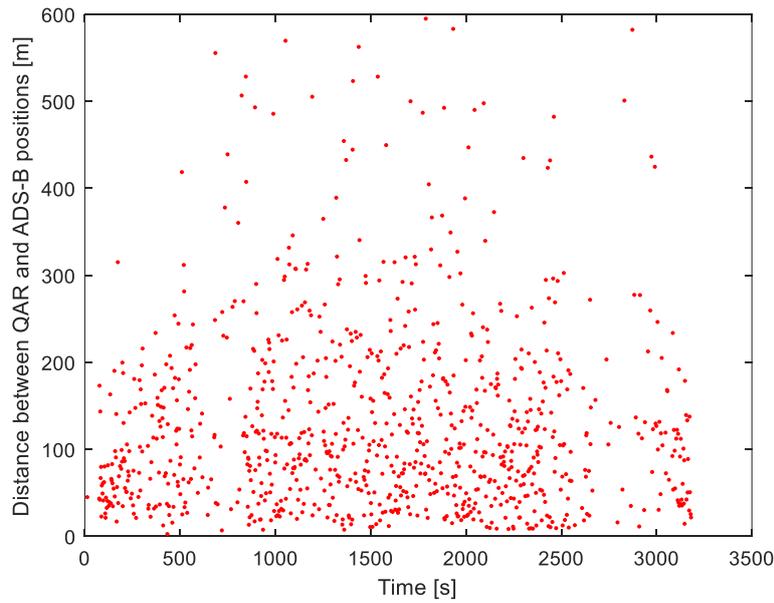


Figure 3-7: Distance between QAR and ADS-B position with same timestamps after merging, Flight A

However, this was not equally close for all flights. Figure 3-8 shows the distances for another flight, Flight B. There a section lasting around 300 s is visible where the distance between the ADS-B and QAR positions is constantly between 800 m and 1500 m. As explained further in section 4.1, the trajectories are not actually that far apart, but the ADS-B data was 8 s behind the QAR data in this section, probably caused by an inaccurate receiver. Unfortunately, as *ADS-Bexchange* started adding the receiver ID to the dataset only in 2017 and QAR data was only available from 2016 and older, the faulty receivers could not be identified, and their data could not be corrected. With more recent QAR data this might be possible.

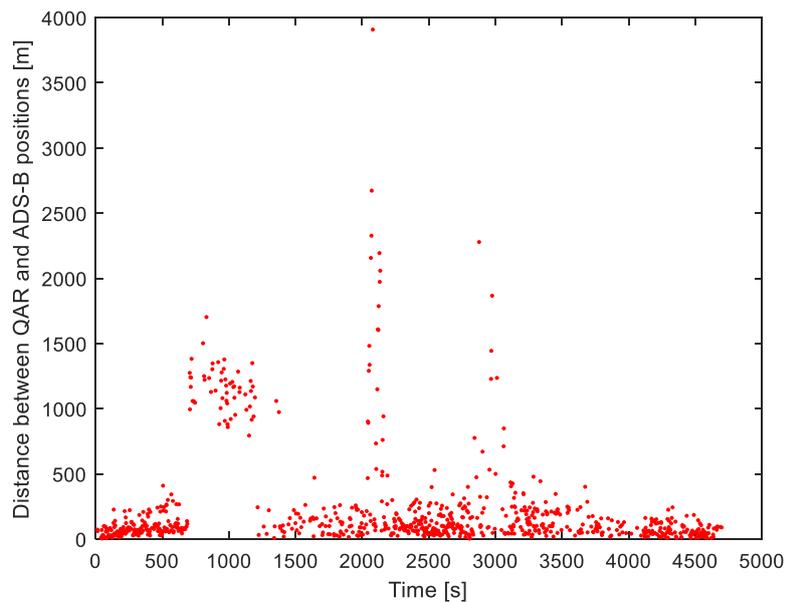


Figure 3-8: Distance between QAR and ADS-B positions with same timestamps, Flight B, before correction

To be able to work with the older datasets, it was assumed that all ADS-B data from other aircraft around the time with the offset comes from the same receiver, so that it must have the same offset. To be able to compare the QAR data with the ADS-B data of the aircraft surrounding it, it needs the same timestamps as the surrounding aircraft. This is why it was decided to shift the QAR timestamps to match the ADS-B data, even though this means that it is shifted to the wrong time and the time distances between the corrected and the not corrected sections are not constant anymore. Shifting only one section also means that there is the same timestamp for two different QAR position recordings at the transition between two sections. The correction is illustrated in Figure 3-9, depicting ADS-B position recordings in red and QAR position recordings in blue. In this example, the ADS-B position with the timestamp 2309 is actually closest to the QAR position recording with the original timestamps 2307, which leads to a bigger distance between the position recordings with the same timestamps. This is corrected by adding the timestamp of the closest ADS-B point to the QAR position recording, instead of the one that is calculated from the timespan to the reference point. This however leads to an error in the time differences, as now the QAR position recordings with the original timestamps 6 and 8, which is two seconds apart, have the timestamps 2305 and 2309, which is four seconds apart.

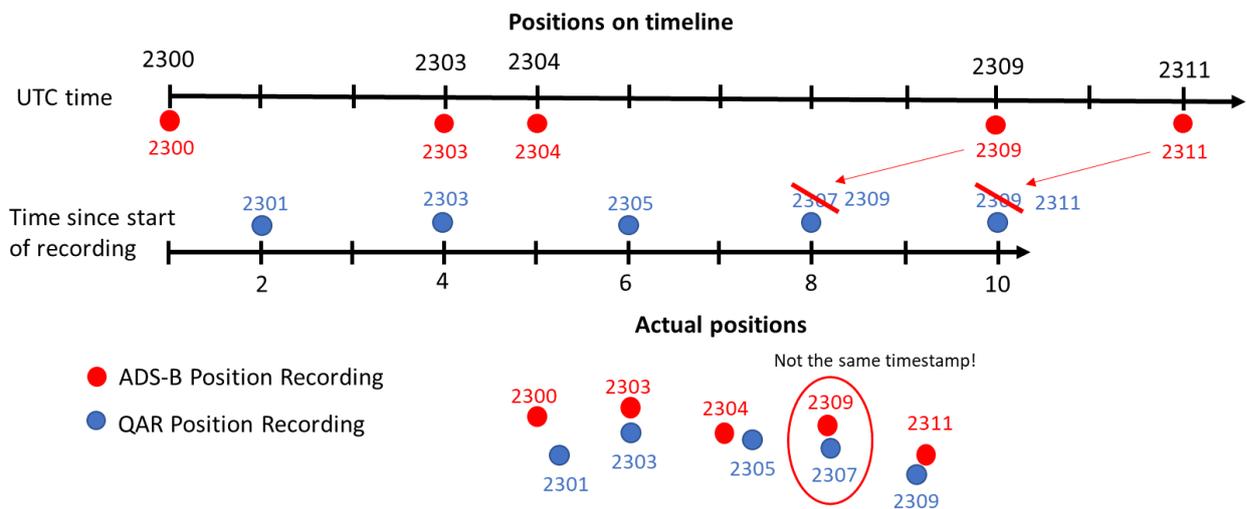


Figure 3-9: Illustration of timestamp correction

The algorithm corrects whole sections of timestamps if the distance between QAR and ADS-B data with the same timestamps is bigger than 250 m for more than five QAR position recordings in a row. The timestamps of these sections are moved again +/- 10 s, and for these sections again the timestamps for which expression (3-1) is minimal are chosen. Figure 3-10 shows the distances from the same flight as in Figure 3-8, but after the correction was conducted. There are still some outliers with large distances, however, these are data errors in the ADS-B data that will be dealt with in Section 3.4.

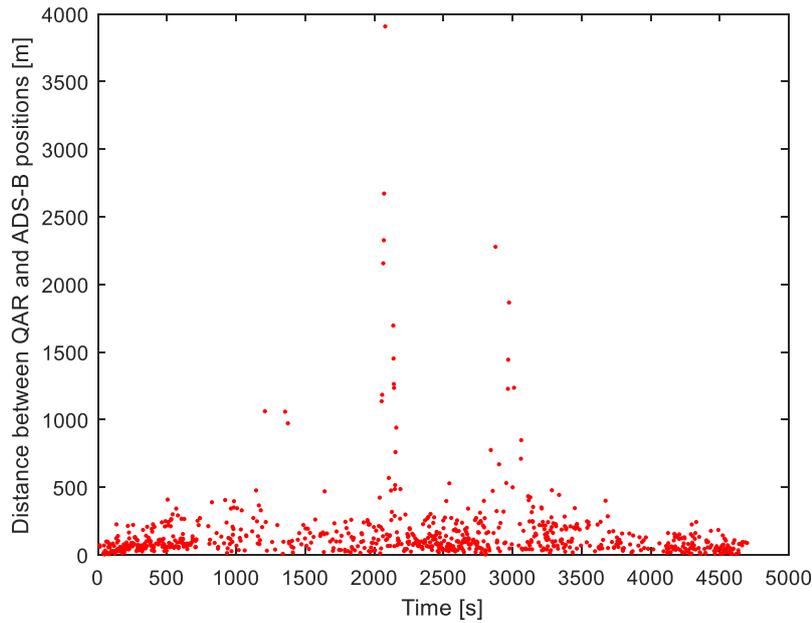


Figure 3-10: Distance between QAR and ADS-B position with same timestamps, Flight B, after correction

3.4 Detecting ADS-B data errors

As described in the previous section, parts of the ADS-B data can be erroneous. Data errors are most common if the aircraft are very low, likely because the reception of the ADS-B receivers is not that good if obstacles are present. Figure 3-11 shows the altitude profile of Lufthansa aircraft *D-AINB* landing in London Heathrow on 18.04.2018 as recorded in the ADS-B data. It looks as if the aircraft was jumping between altitudes and landing twice.

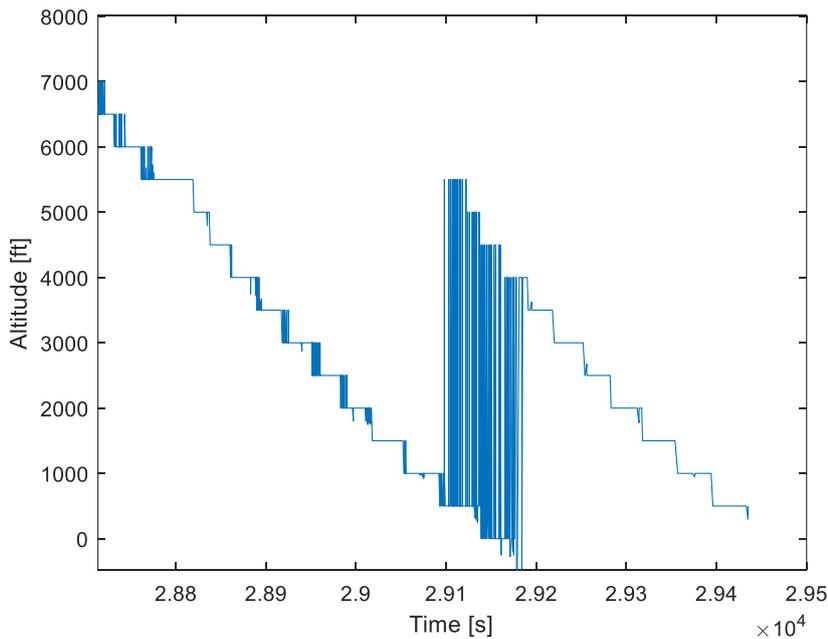


Figure 3-11: Altitude profile of Lufthansa aircraft *D-AINB* landing in London Heathrow with data coming from a faulty receiver

As the ADS-B data is from 2018, the receiver ID is included and therefore further investigation was possible. The data that resembles a second landing came from a single receiver while the data from the first landing came from several different receivers. This is visible in Figure 3-12, where the altitudes from the receiver with the ID 73071 are plotted in red and the altitudes from all other receivers are plotted in blue. Therefore, it is very likely that the first landing is the actual correct time of the landing. The single receiver for the second landing had such a big error in its timestamps that it appears as if the aircraft was landing again five minutes later.

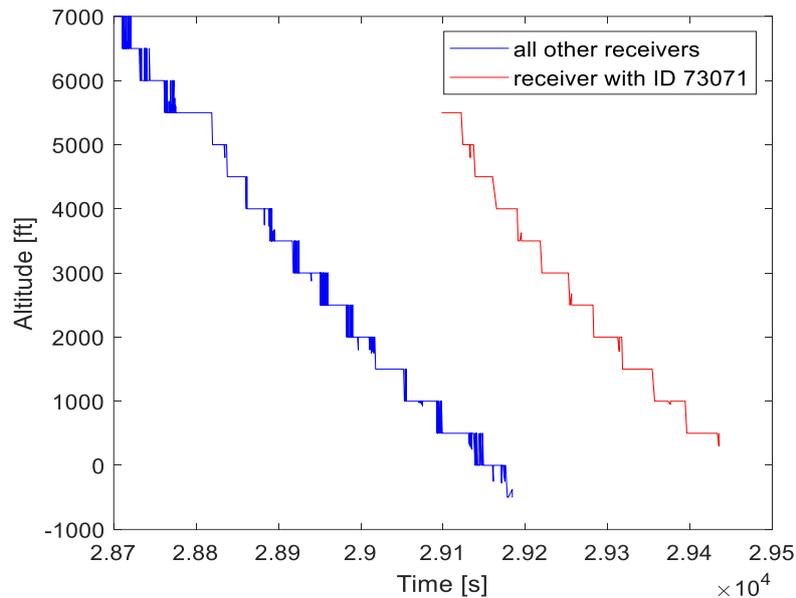


Figure 3-12: Altitude profile with altitudes from faulty receiver in red and altitudes from all other receivers in blue

To be able to provide consistent information about distances to other aircraft, these false position recordings have to be filtered out. To find these outliers in the data the MATLAB function *isoutlier* is used. This function provides different methods to find outliers. For finding local outliers there are the *movmedian* and the *movmean* method. Here, the *movmedian* method is used, because according to the MATLAB documentation this method is more robust than *movmean*. The *movmedian* method flags a value as an outlier if it is more than three locally scaled Median Absolute Deviations (MAD) from the local median over a specified window length. The scaled MAD for a vector *A* is defined as in equation (3-2) [27]. The function *erfcinv* is the inverse complimentary error function.

$$MAD = c \cdot median(|A - median(A)|) \tag{3-2}$$

$$c = -\frac{1}{\sqrt{2} \cdot erfcinv\left(\frac{3}{2}\right)} = 1.48$$

Of several tested window length, 20 proved to be the best compromise between finding all outliers and not flagging too much good data.

The *isoutlier* function is applied on altitude, latitude and longitude of the ADS-B data, separately for every aircraft included, and all flagged data is deleted before further calculations. Furthermore, if the receiver ID is included in the data, receivers that transmit wrong data are

detected. In Figure 3-13 outliers that were detected at the same time in altitude, latitude and longitude in the previous example are marked red.

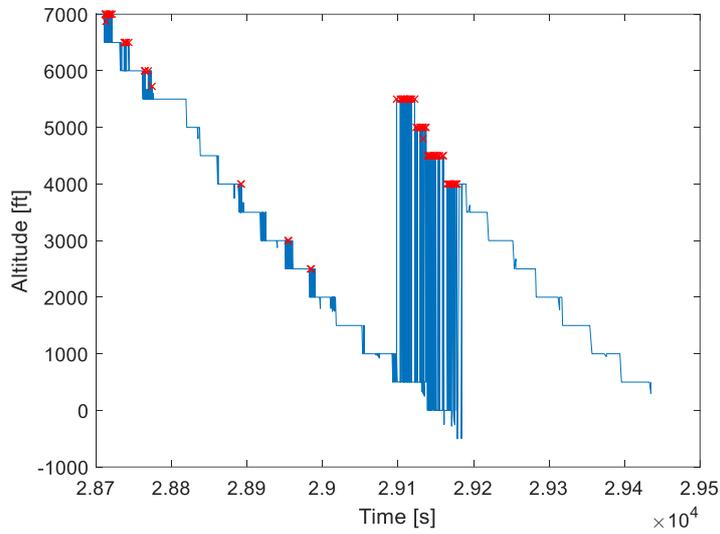


Figure 3-13: Altitude profile with detected outliers marked red

In areas where several receivers record data from the same aircraft and one is behind the others with the timestamps, like in the example in Figure 3-11, its data will be flagged by the *isoutlier* function. To prevent data from these faulty receivers providing false results also in areas where it is the only data source, thus the data will not be flagged as an outlier there, it is necessary to delete the data coming from these receivers completely. To prevent too much data getting deleted, the algorithm only deletes all data from a receiver if all three values, altitude, latitude and longitude, have been detected to be outliers in more than one position recording coming from this receiver. Figure 3-14 shows the altitude profile from the example flight after all the outliers and the data from faulty receivers have been removed.

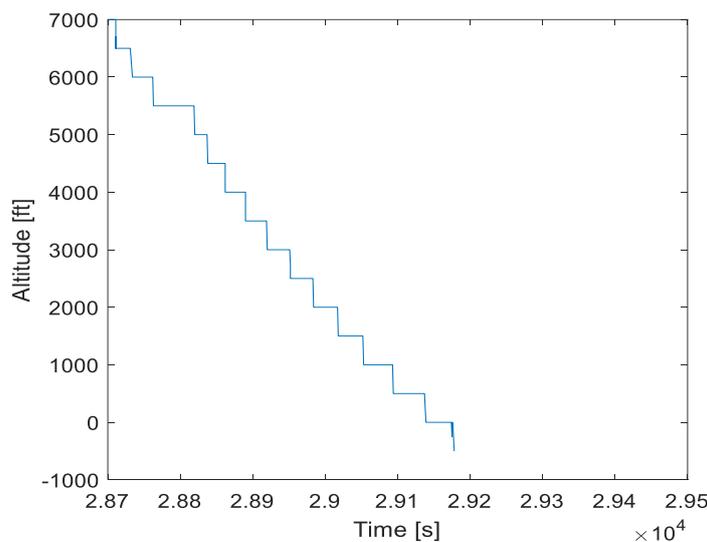


Figure 3-14: Altitude profile after faulty receivers and outliers were removed

3.5 Interpolating data of surrounding aircraft

For the following calculation, all ADS-B data of aircraft within 50 km horizontal distance of the QAR flight path is used. To obtain this, data of all sectors that the flight passes, as well as the neighboring sectors, if the flight is close to the border, are loaded. Afterwards, the distance of every QAR position recording to every ADS-B position recording with the same timestamp is calculated and only the ADS-B data with a distance below 50 km is kept. The time steps of the ADS-B data are very irregular, sometimes there are several messages per second, sometimes there are only two per minute. Therefore, to have the position of every surrounding aircraft for every QAR position recording, the ADS-B data is interpolated.

The positions in the ADS-B data are given in WGS84 coordinates. As interpolating the latitude and longitude coordinates directly would not work very well, especially around the poles, the data is transformed separately for every flight to the Earth-Centered-Earth-Fixed (ECEF) coordinate system. The data is then interpolated linearly and transformed back to the WGS84 coordinates. The interpolation in the ECEF system is done along the shortest distance between two points. In reality, an aircraft flies close to the great circle distance around the earth and keeps the altitude constant, so the distance that it covers between the two points is actually longer. The distance of the direct connection p between two points with a great circle distance d at a radius r can be calculated by formula (3-3).

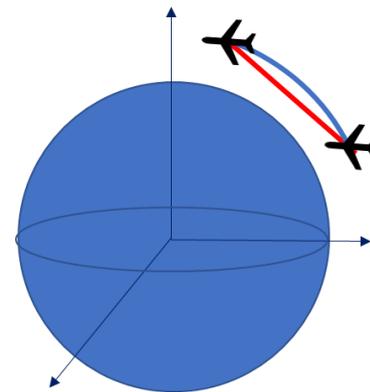


Figure 3-15: Great circle distance (blue) and direct connection (red)

$$p = 2 \cdot r \cdot \sin\left(\frac{d \cdot 180^\circ}{2 \cdot r \cdot \pi}\right) \tag{3-3}$$

As the difference between great circle distance and direct connection at a great circle distance of 1,000 m is only 10^{-6} m, and usually the ADS-B positions are even closer together than that, this difference is negligible.

However, to get a smoother dataset with constant altitudes, the altitude is interpolated separately and not taken from the ECEF interpolation. For a better result, only the exact altitude data is used. Exact altitude data has a precision of 25 ft and did not come from the Cos arrays because in the Cos arrays the altitude is rounded to 500 ft. Figure 3-16 shows the difference of the altitude profile when using only the exact data or when using all data. When using the inexact data also, the steps are visible that occur due to the rounded altitude values. This is why the inexact data of the Cos arrays is only used if no exact data is available for this aircraft and as the first and last value to extend the interpolated range.

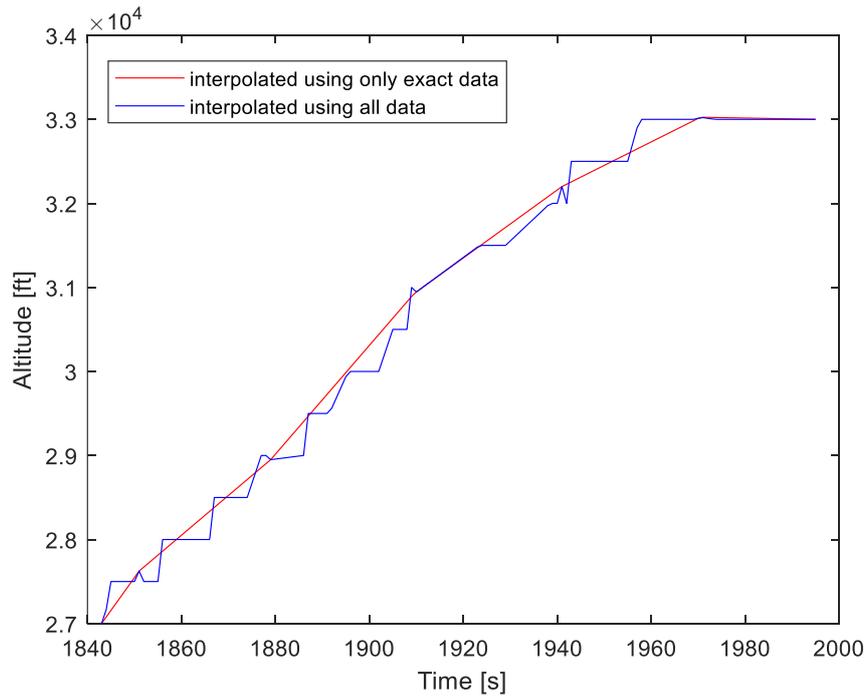


Figure 3-16: Comparison of altitude interpolation between using exact data only and using all data during ascend of an aircraft

The other variables speed, track and vertical speed exist only in the lines that do not come from the Cos arrays, which is in the same lines where the altitude is exact. They are also interpolated linearly, but only if more than one datapoint exists. Otherwise these variables are set to NaN. Furthermore, if an aircraft only has one datapoint at all in the area of interest it is considered to be non-relevant and deleted completely.

4 Analysis

4.1 Comparison of ADS-B to QAR data quality

Merging QAR and ADS-B data allows to compare both data sources with each other and to evaluate how precise the ADS-B data is in comparison to the QAR data. As there were some problems with the timestamps, see section 3.3.2., it is necessary to compare the position reports independently from these. To achieve this, the ADS-B position recordings were interpolated as described in section 3.5, but with a smaller timestep of 0.05 s. Then, the horizontal distance of every QAR position report to all ADS-B data points was calculated, to find the closest ADS-B position to every QAR position. Figure 4-1 shows the distances to the closest ADS-B position during the example Flight B. The distance between both flight paths is never above 70 m and only 23.9 m on average. This was similar for all flights analyzed for this thesis, as depicted in Table 4-1. The average distance in total for all flights was 37.8 m. Some higher distances between the flight paths occurred only in case not enough ADS-B data points were available to get a good interpolation, as visible in the example illustrated in Figure 4-2.

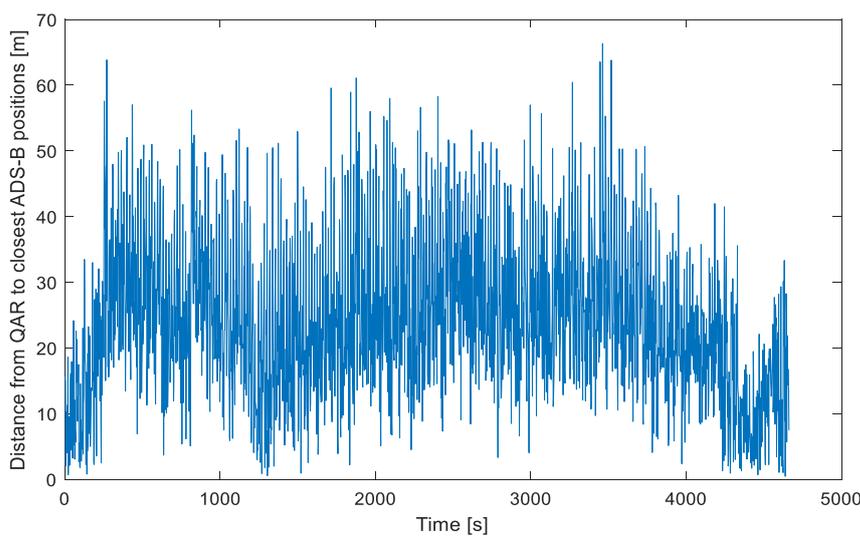


Figure 4-1: Distance from QAR to closest ADS-B positions

| Mean horizontal distance between QAR and ADS-B data for all flights |
|---|
| 24.4 |
| 19.6 |
| 14.4 |
| 23.9 |
| 18.5 |
| 141.8 |
| 55.1 |
| 29.1 |
| 13.3 |

Table 4-1: Mean horizontal distance between QAR and ADS-B data for all flights

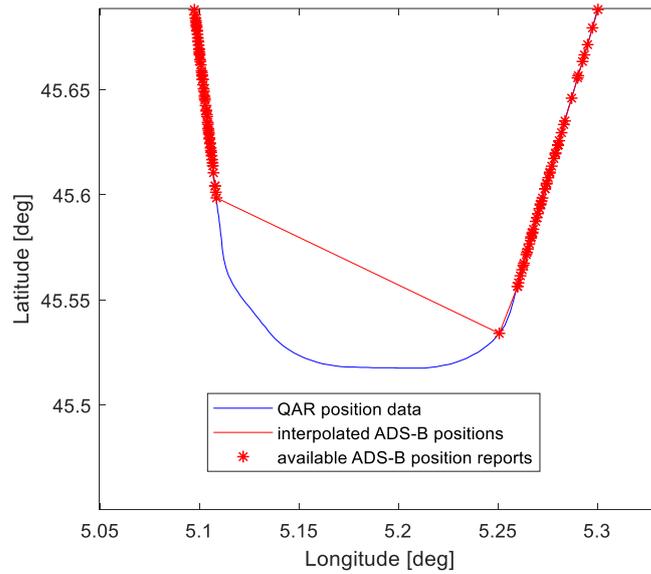


Figure 4-2: Example of lack of available position recordings leading to big distance between ADS-B and QAR data

The precision of the exact altitude (the altitude that did not come from the Cos arrays) in the ADS-B data is also quite good, as Figure 4-3 shows. There, the altitude profile of the example flight with the QAR data (blue) and the ADS-B altitudes (red) that are quite close to the QAR altitude profile can be seen. The biggest distance between the two datasets is 250 feet at one point, but mostly smaller. The problem with the timestamps did not influence the altitude difference because the wrong timestamps are only in parts of the flights with constant altitude. The only difficulty with the altitude from the ADS-B data is that a precise altitude is only available once every minute, therefore it cannot indicate all small shifts in altitude.

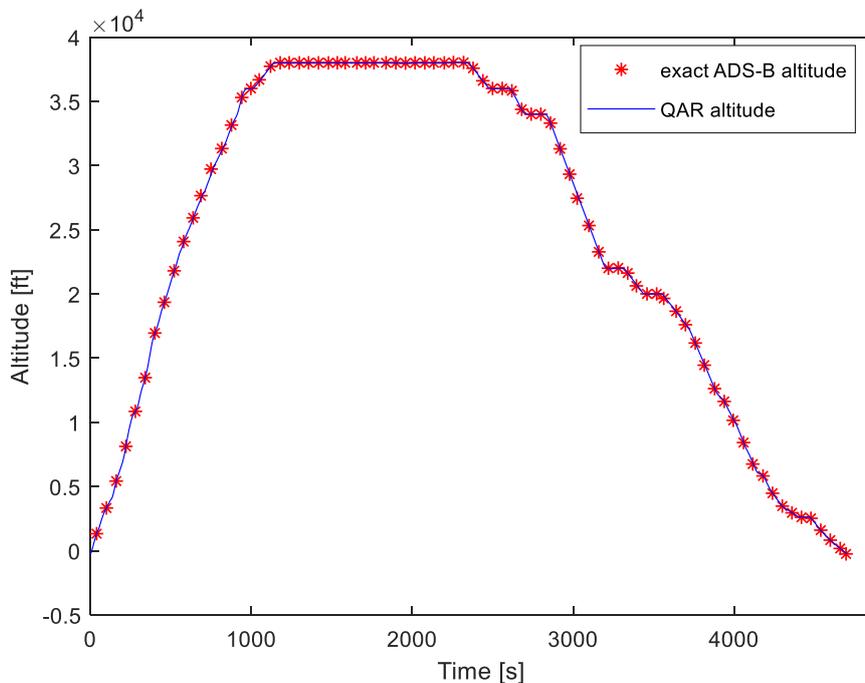


Figure 4-3: Altitude profile from ADS-B and QAR data

Generally, it can be stated that the position recordings available from the ADS-B data are accurate enough to evaluate how close aircraft come to each other, and to assess the risk of a MAC. However, a problem becomes evident when comparing Figure 3-10 to Figure 4-1. Both show the distance between QAR and ADS-B position recordings for the same flight, however Figure 4-1 shows the distance from each QAR position to the closest ADS-B position, where the distance for the k^{th} QAR position recording $pos_{QAR,k}$ with a vector of all ADS-B position recordings \overline{pos}_{ADS-B} is calculated as in equation (4-1).

$$d_k = \min(\text{distance}(pos_{QAR,k}, \overline{pos}_{ADS-B})) \quad (4-1)$$

Figure 3-10 in contrast shows the distance from each QAR position to the ADS-B position report with the same timestamp $t_{QAR,k}$ as described in equation (4-2).

$$d_k = \text{distance}(pos_{QAR,k}, pos_{ADS-B}(t_{ADS-B} == t_{QAR,k})). \quad (4-2)$$

The distances in Figure 3-10 are a lot bigger than in Figure 4-1. Additionally, there are some outliers with distances of almost 4000 m in Figure 3-10, which are not visible in Figure 4-1. To detect the reason for the differences, the section of the flight with the outliers has to be considered closer. Figure 4-4 shows the positions in ADS-B and QAR data during this part of the flight with just latitude and longitude, but no time information. Here, the position recordings are so close together that no difference is visible. However, Figure 4-5 shows the change of latitude over time for ADS-B and QAR data for the same part of the flight where bigger differences are visible. The edges in the line of the QAR data are where the timestamps of the QAR data got changed by the merging algorithm to better fit the ADS-B data. This shows that the big distances of positions with identical timestamps occur, because the aircraft is just reported to be at a position at the wrong time. This happens because the time of the report is not added by the aircraft itself but added by the receiver when it gets the message. Therefore, if the receiver does not work with the correct synchronized time, the ADS-B data includes wrong timestamps.

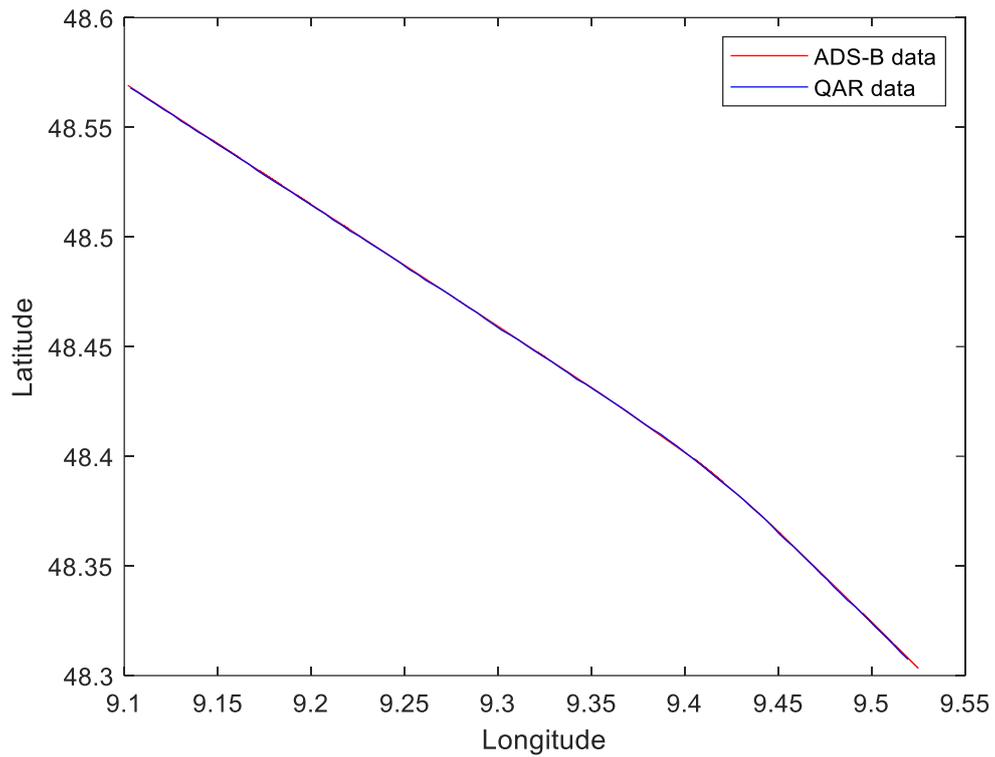


Figure 4-4: Change of Latitude over Longitude for one part of the flight compared between ADS-B and QAR data

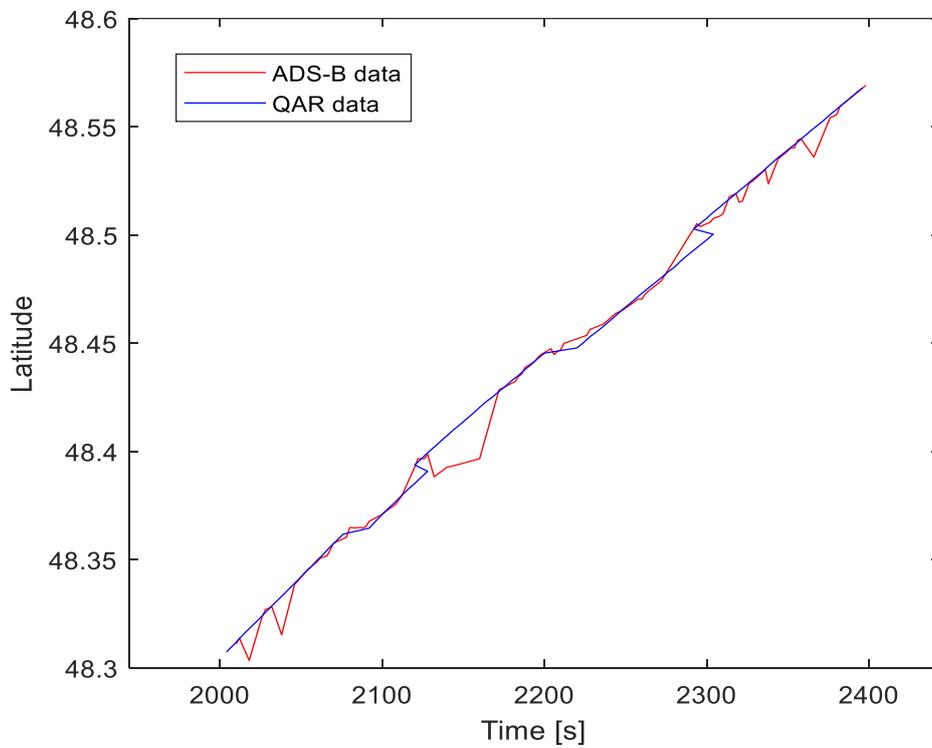


Figure 4-5: Change of Latitude over Time for one part of the flight compared between ADS-B and QAR data

4.2 Calculation of risk indicators

With the data prepared in the previous steps, several parameters are defined and calculated that help to evaluate the concentration of air traffic and along that the risk of a MAC. All of these parameters can be calculated for every QAR position recording of the flight and are added to the existing QAR dataset so that the information can be used for further evaluations part of FDM.

First, the distances of the analyzed aircraft to the surrounding aircraft at every timestep are calculated. This is later used to calculate all risk indicators. In a second step, also the altitude distance between the two aircraft is calculated by subtracting the altitude of the analyzed aircraft from the one of the surrounding aircrafts. Therefore, the altitude distance is negative if the surrounding aircraft is below the analyzed aircraft and positive if it is above.

4.2.1 Minimum distance to surrounding aircraft

To find out how close aircraft come to each other during the flight, the minimum distance to the next aircraft at any given point is stated. By doing so, it can be verified whether separation minima were met during the flight. To assess which other aircraft is the closest with respect to horizontal and altitude distance, the first thought was to use the Euclidian distance calculated by $\sqrt{\text{Horizontal distance}^2 + \text{Altitude distance}^2}$. This however would not make sense in the context of two aircraft passing each other, because the horizontal distance to each other has to be much bigger than the altitude distance to assure the safety of the two aircraft. If the separation minima described in section 2.3 are compared, the horizontal distance has to be about ten times bigger than the vertical distance. Using the Euclidian distance might lead to an aircraft 1 km below the analyzed aircraft being chosen as the closest at that point, even though maybe another aircraft is only at 2 km horizontal distance at the same altitude, which would be far more critical. Subsequently, only the smallest horizontal distance is used to decide which aircraft is the closest. The horizontal distance is available for all aircraft within 50 km from the previous calculation. It is set to NaN for an aircraft if the vertical distance to this aircraft is less than 4,000 ft in any direction. This leaves only the horizontal distances to aircraft that are in the same altitude section. The closest aircraft at one point in time is the one with the smallest of these distances at that time. Figure 4-6 shows the horizontal distance to the closest aircraft during Flight B. Figure 4-7 shows the altitude distance to the same horizontally closest aircraft for the same flight. If there is no minimum distance given for a specific time, this means that no other aircraft was at the same altitude section of +/- 4,000 ft closer than 50 km horizontal distance at that time. To offer the possibility to reconstruct where exactly the other aircraft was, also the information about the direction to the other aircraft in respect to the heading of the analyzed aircraft as well as the flight track of the other aircraft are added to the QAR dataset.

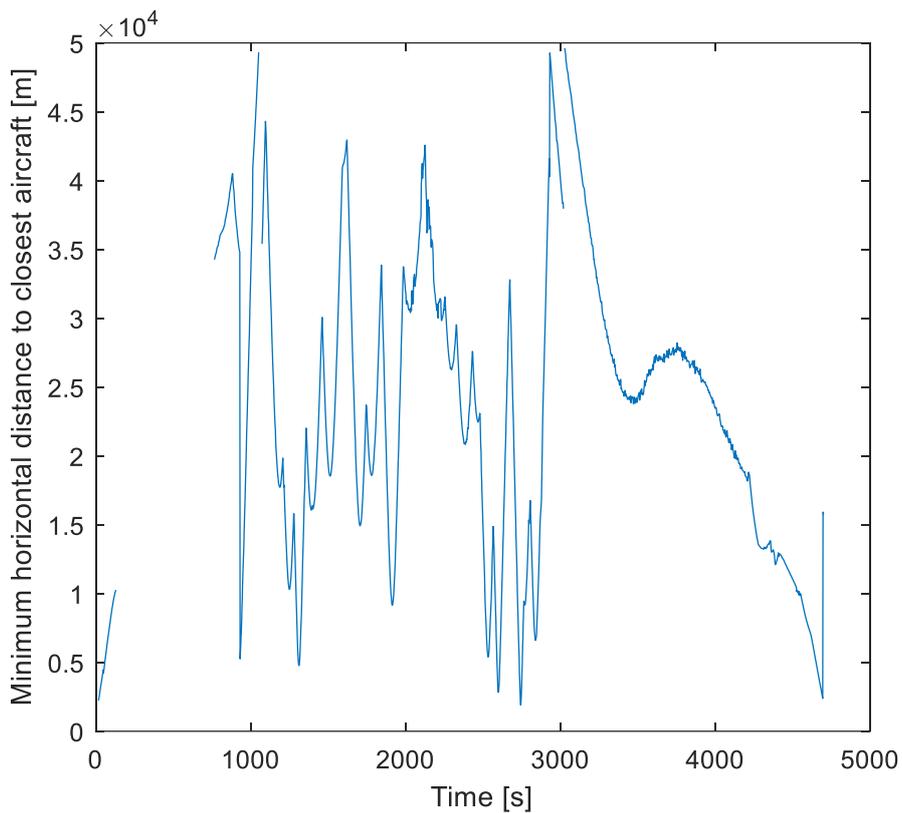


Figure 4-6: Minimum horizontal distance during flight

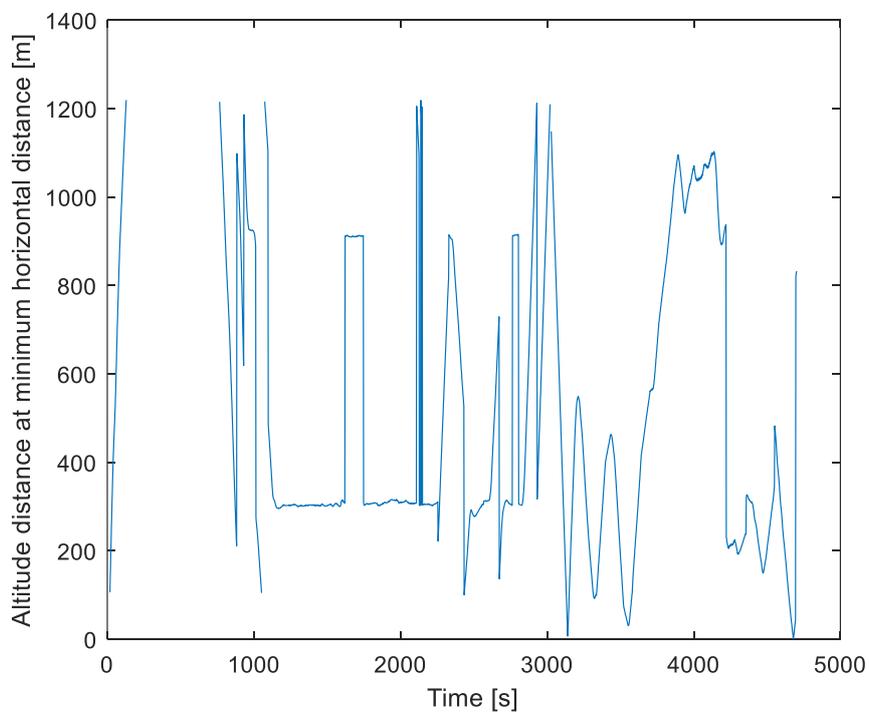


Figure 4-7: Altitude distance at minimum horizontal distance during flight

4.2.2 Airspace density

Next, for an evaluation of how busy the airspace was during a flight, other aircraft in a specified space around the analyzed aircraft are counted. For this thesis, a cylinder with a radius of 20 km and a height of 4000 ft above and below the aircraft was chosen. Figure 4-8 shows the number of aircraft in this cylinder for Flight A. The location of the peaks in this plot along the flight path are displayed on a map in Figure 4-9.

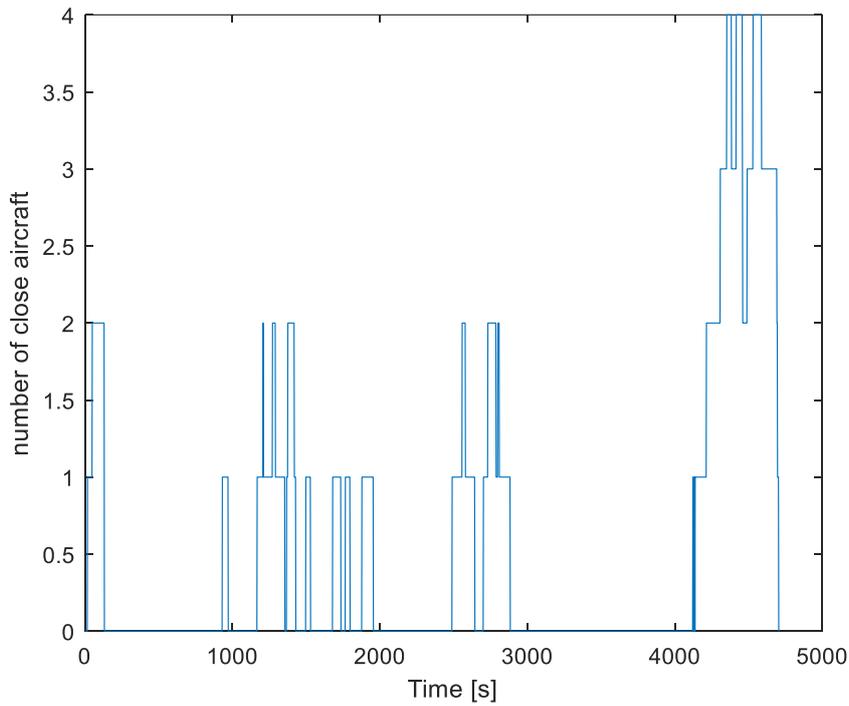


Figure 4-8: Number of surrounding aircraft during flight

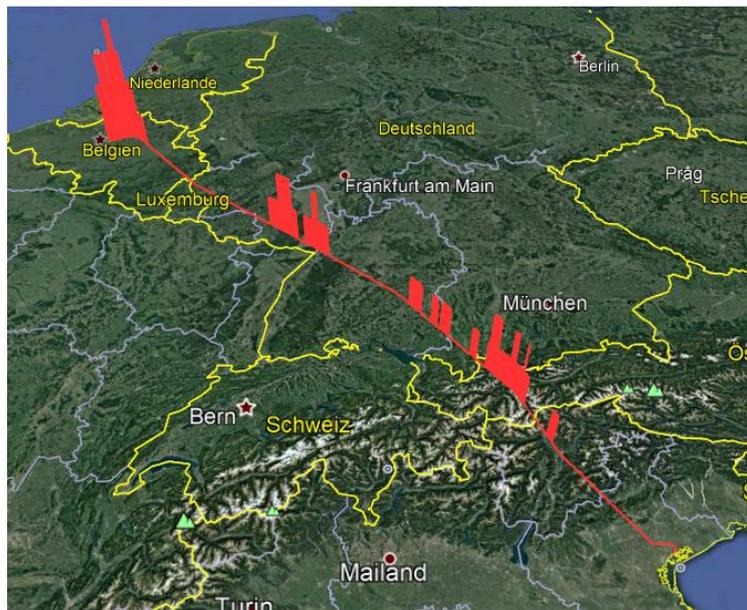


Figure 4-9: Flight path with location of peaks in number of surrounding aircraft, source of the map: Google Earth

The calculation of the airspace density relates to the thesis of Camilla Zimmermann mentioned before. With the algorithm she developed, this variable is calculated as well, however not during the course of a flight, but for a specified region. This region might for example be the airspace around an airport. This way, certain areas with high airspace density can be detected not just for certain flights, but in general.

4.2.3 Check for TCAS Resolution Advisories

The Traffic Alert and Collision Avoidance System (TCAS) is a family of airborne devices that provide collision avoidance protection by a traffic display and collision threat alerts. If another aircraft intrudes the protected airspace surrounding an aircraft a traffic advisory (TA) is issued to alert the pilot of the intruder. After a TA the pilot is supposed to look for the other aircraft, but not divert from any clearances by ATC. If the intruder aircraft comes even closer, a resolution advisory (RA) is issued for both aircraft which provides a recommended escape maneuver in the vertical dimension to either increase or maintain the existing vertical separation between aircraft. Pilots have to follow a RA immediately, even if this contradicts instructions from ATC [4]. Figure 4-10 shows the TCAS protected volume around each aircraft.

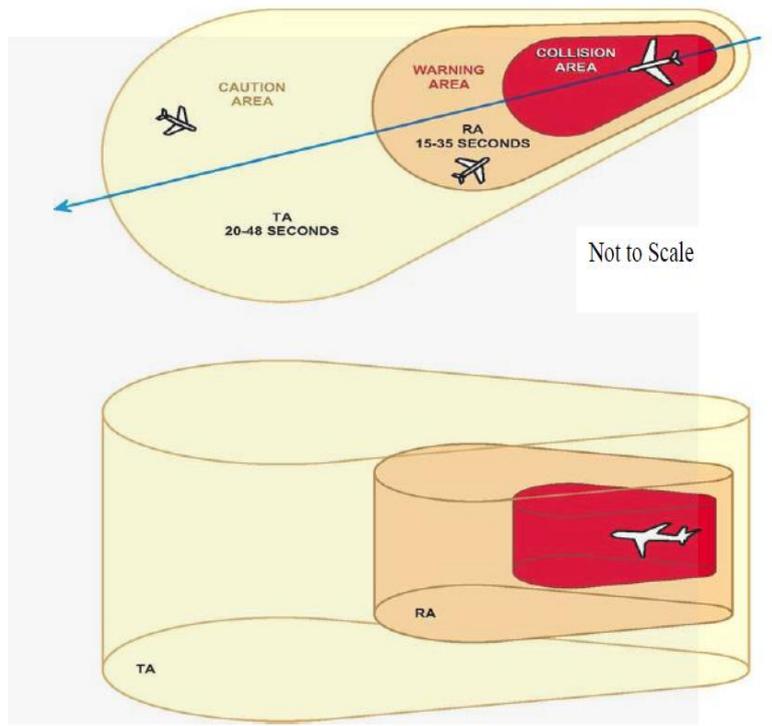


Figure 4-10: TCAS protection volume [4]

Using the TCAS logic with the ADS-B data gives an idea of when an TA or RA would have been issued, and thus marks critical situations. TCAS uses different sensitivity levels depending on the altitude of the aircraft that define the threshold levels for the time to closest point of approach TAU , the horizontal distance threshold $DMOD$ and the vertical distance threshold $ZTHR$, which control the size of the protected volume. Table 4-2 [4] contains the definition of the sensitivity levels (SL) and the alarm thresholds for a TA or RA that are used in the TCAS logic. This definition was also implemented for the analysis of the ADS-B data, with $DMOD$ converted to meters. Sensitivity level 2 and 3 are actually defined using the Above Ground Level (AGL). But as only barometric altitude was available for the implementation with

the ADS-B data, this was used instead. This might lead to a higher SL if the barometric altitude is bigger than the above ground level.

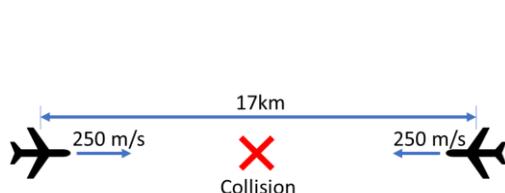
| Own Altitude (feet) | SL | TAU (seconds) | | DMOD (NM) | | ZTHR (feet) | |
|---------------------|----|---------------|-----|-----------|------|-------------|-----|
| | | TA | RA | TA | RA | TA | RA |
| < 1000 (AGL) | 2 | 20 | N/A | 0.30 | N/A | 850 | N/A |
| 1000 - 2350 (AGL) | 3 | 25 | 15 | 0.33 | 0.20 | 850 | 600 |
| 2350 – 5000 | 4 | 30 | 20 | 0.48 | 0.35 | 850 | 600 |
| 5000 – 10000 | 5 | 40 | 25 | 0.75 | 0.55 | 850 | 600 |
| 10000 – 20000 | 6 | 45 | 30 | 1.00 | 0.80 | 850 | 600 |
| 20000 – 42000 | 7 | 48 | 35 | 1.30 | 1.10 | 850 | 700 |
| > 42000 | 7 | 48 | 35 | 1.30 | 1.10 | 1200 | 800 |

Table 4-2: Sensitivity level definition and alarm thresholds [4]

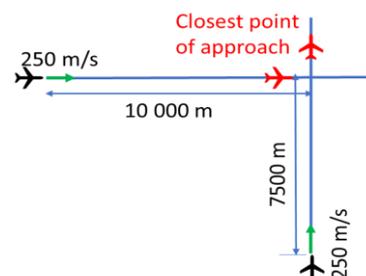
The primarily used measurement to decide if a TA or RA is issued, is the time τ . The horizontal τ is the approximated time to closest point of approach and vertical τ is the time until both aircraft are on the same altitude. Both horizontal and vertical τ are calculated as in equation (4-3) by the horizontal or vertical distance d divided by the corresponding closure rate \dot{d} .

$$\tau = \frac{d}{\dot{d}} \tag{4-3}$$

The closure rate is positive when the aircraft are approaching each other and negative when they are getting further away from each other. This is the same for τ as the distance is always positive. τ and the actual time to closest point of approach are only exactly the same if the two aircraft are going to collide with each other, otherwise τ is just an approximation. If the aircraft are passing each other with some distance, τ will decrease until shortly before the closest point of approach, and then increase sharply until the closest point of approach is reached, after which it becomes negative [28]. This is illustrated with the two examples in Figure 4-11. Example A shows a head-on collision with constant speed. As the closure rate \dot{d} is constant and the distance d is decreasing linearly in this example, τ is also decreasing linearly and shows the actual time to the collision. In example B the flight paths of the two aircraft cross each other in a right angle. As the two aircraft are at different distances from the point of intersection with the same constant speeds, they are not at the point of intersection at the same time and do not collide. As the course of distance d and closure rate \dot{d} is not linear in example B and \dot{d} approaches zero at the closest point of approach, τ behaves as described earlier and has a peak shortly before the closest point of approach. The plot B3 in Figure 4-10 shows also the actual closest point of approach compared to τ . τ is very close to the time to closest point of approach at the beginning but differs later.



A1: Illustration of example A



B1: Illustration of example B

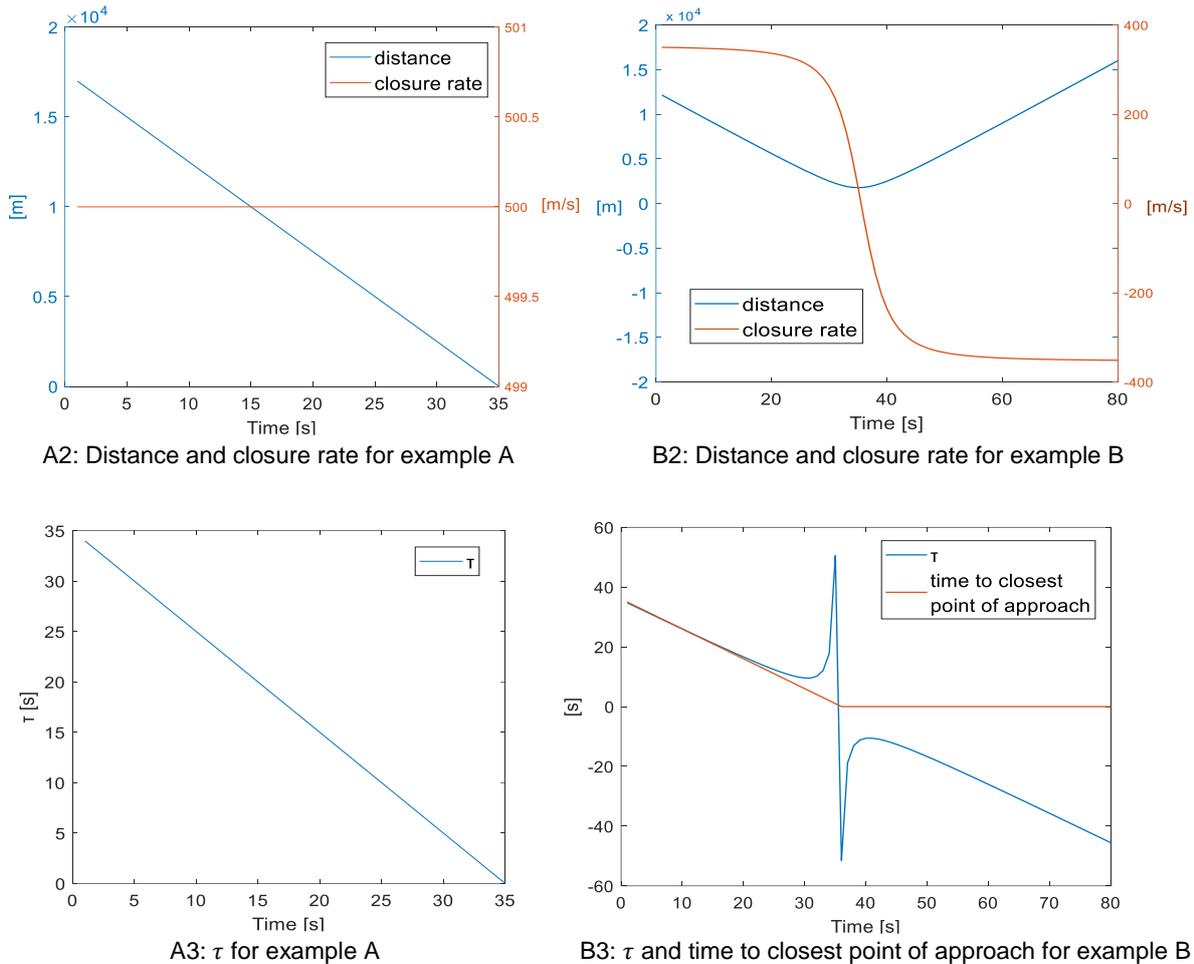


Figure 4-11: Example for the course of τ for a head-on collision (A) and two aircraft passing each other (B)

Using the discrete ADS-B data the closure rate \dot{d} is approximated by the difference between the current distance between the two aircraft and the distance at the previous timestep divided by the time difference. The vertical or horizontal closure rate at timestep k are both calculated by equation (4-4), where d is the vertical or horizontal distance between two aircraft as described in the introduction to section 4.2.

$$\dot{d}_k = \frac{d_{k-1} - d_k}{t_k - t_{k-1}} \tag{4-4}$$

To avoid that aircraft come very close but with such a low closure rate that the TAU thresholds are not breached, a modified definition of τ is used in TCAS. This definition uses the horizontal distance threshold $DMOD$ as in equation (4-5) [28]. This modified version is also used for the horizontal τ in the implementation for the ADS-B data.

$$\tau_{mod} = \frac{d^2 - DMOD^2}{d\dot{d}} \tag{4-5}$$

The modified τ is almost the same for high closure rates and distances, but smaller for small distances and closure rates. To use formula (4-5), the distance has to be bigger than $DMOD$ and the closure rate cannot be zero. Figure 4-11 compares τ and τ_{mod} for the example B from before. The area where the distance is smaller than $DMOD$ is marked red. In this area the values of τ_{mod} are not useful for the evaluation of the time to closest point of approach, because $d^2 - DMOD^2$ is negative and this reverses the signs of τ_{mod} . As τ_{mod} is smaller when d approaches $DMOD$, the TAU threshold levels will be breached earlier.

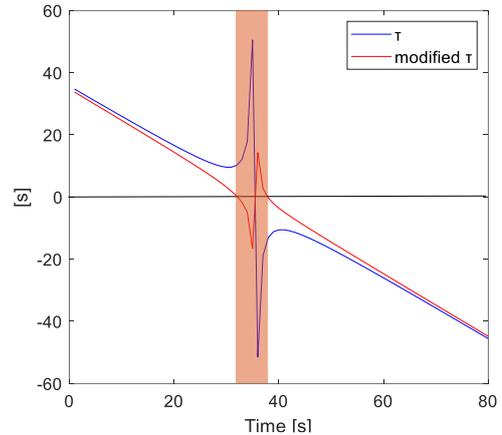


Figure 4-12: Comparison between τ and modified τ for example B, region where d is below $DMOD$ marked red

After horizontal and vertical τ are calculated, it can be decided if a TA or RA would have been issued. This is the case if either τ_{hor} is below the threshold at the applied sensitivity level, or the horizontal distance is below $DMOD$, and also τ_{vert} is below the threshold, or the vertical distance is below $ZTHR$. So, an RA is issued if the statement (4-6) is true.

$$(\tau_{hor} < TAU_{SL,RA} \parallel d_{hor} < DMOD_{RA}) \& (\tau_{vert} < TAU_{SL,RA} \parallel d_{vert} < ZTHR_{RA}) \tag{4-6}$$

With the ADS-B data, τ is calculated and evaluated for every aircraft within 10 km horizontal distance and 2000 ft altitude distance, and the information if a TA or RA was detected at a point of the flight is added to the QAR dataset. For the available QAR datasets, no TAs and RA were detected with this analysis after false detections resulting from ADS-B data errors had been removed. Figure 4-13 shows the distribution of vertical and horizontal τ values for the QAR flights. To cause a TA or RA both τ values have to be below the corresponding threshold TAU , but above zero. These areas are marked in the closeup of the distribution in Figure 4-14. The threshold for a TA at the highest SL is marked green, the threshold for a RA is marked red. It is visible that none of the τ values are in the critical area.

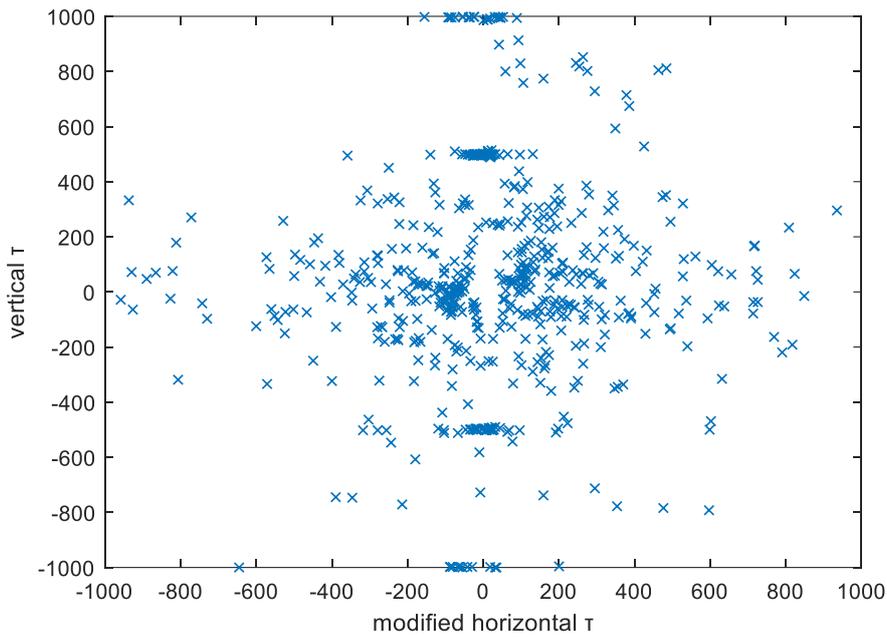


Figure 4-13: Distribution of vertical and horizontal τ values for the QAR flights

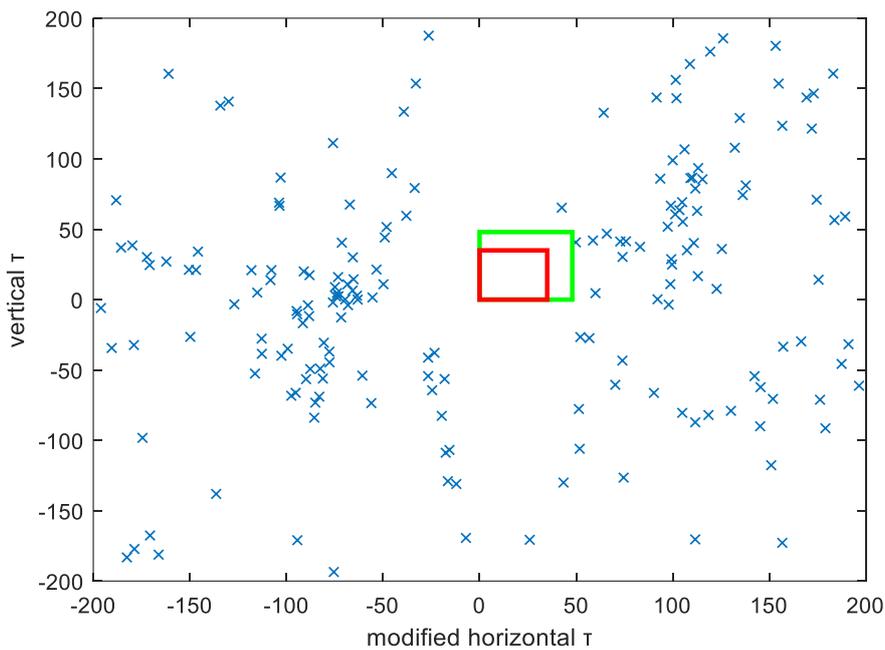


Figure 4-14: Distribution of vertical and horizontal τ values for the QAR flights - closeup with TA (green) and RA (red) threshold values

4.2.4 Minimum distance during final approach

Next, all distances of the aircraft to other aircraft landing in front or behind on the same arrival flight path are evaluated, as the airspace is usually quite busy around airports. To find out how close other aircraft get on the arrival flight paths, first a logic has to be found to distinguish which other aircraft are on the same final approach path. This is the case if an aircraft meets all the following three conditions:

- It is in final approach, this means here that it is below 5,000 ft and its vertical speed is negative.
- It is on the same track, which means that the track does not differ more than 20 degrees from the track of the QAR flight
- It is either behind or in front of the QAR flight, which means that the direction from the QAR flight to the ADS-B aircraft, counted clockwise with 0° directly in front of the QAR aircraft, is either between 170° and 190°, bigger than 350° or smaller than 10°

Using this definition, other aircraft on the same arrival flight path are detected and the horizontal distance to the next aircraft in front and the next one behind can be indicated. Another parameter shows the number of aircraft that are still to land in front of the own aircraft. An issue with the definition of final approach being below 5,000 ft is that only barometric altitude is available. Therefore, the final approach could not be detected for airports with an elevation above 5,000 ft.

Figure 4-15 shows the distance to the next aircraft in front during final approach for the example flight B. Figure 4-16 shows the number of aircraft in front and Figure 4-17 the distance to the next aircraft behind. The plot in Figure 4-14 does not start until 2500 s because no other aircraft is detected in approach behind on the same flight path before. The big jump at the end occurs because the aircraft detected closest behind is not detected anymore. For some seconds the closest aircraft is one far further behind. After that, no aircraft are detected behind anymore because the track information for these ADS-B flights was missing from there on. Therefore, it could not be compared to the track of the QAR flight.

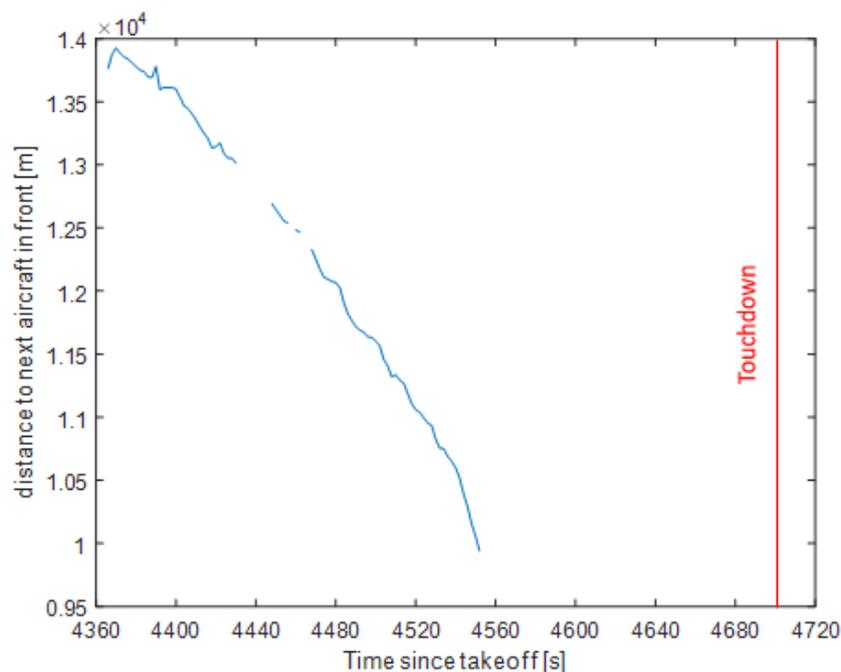


Figure 4-15: Distance to next aircraft in front

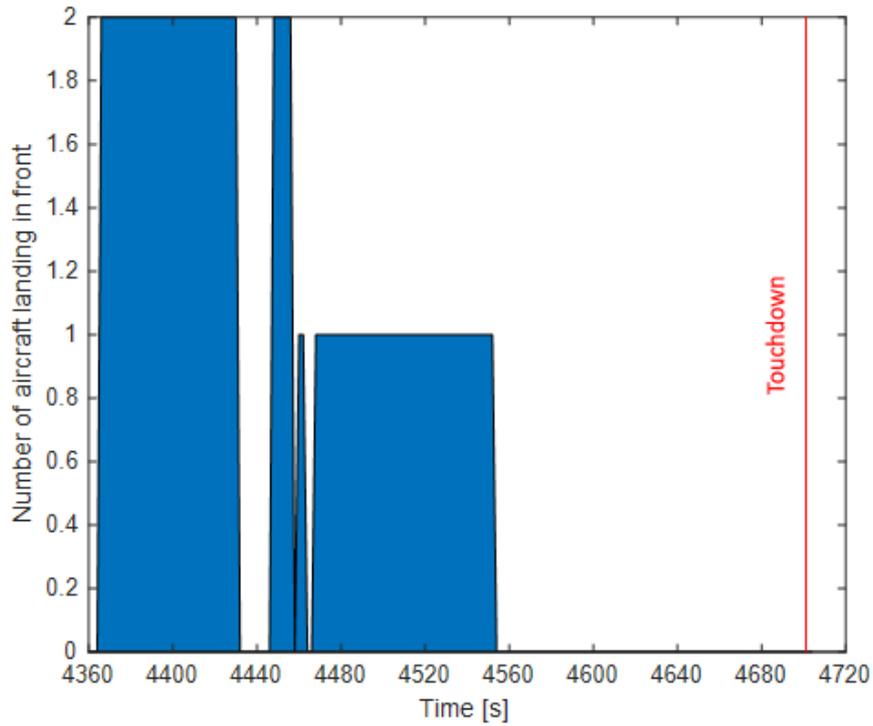


Figure 4-16: Number of aircraft detected landing in front

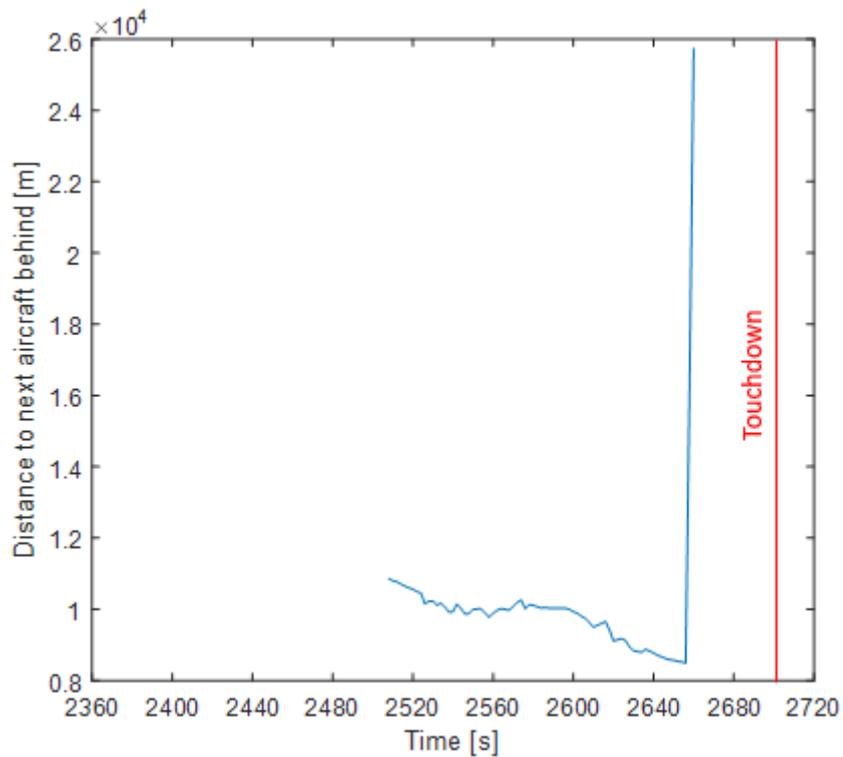


Figure 4-17: Distance to next aircraft landing behind

4.3 Analysis of example data

In this section, some example data is analyzed to show the application of the risk indicators from section 4.2 and to show typical values during a flight. As not enough QAR data was available to make any statistical assessments, only ADS-B data was used for this chapter. The ADS-B data of certain aircraft was prepared to act as if QAR data from these aircraft was available. First, the ADS-B data for the desired aircraft was retrieved by looking for the corresponding registrations in the complete available ADS-B dataset for a certain time span. This resulted into all ADS-B data of the desired aircraft during that time span, possibly from several different flights for each aircraft. To partition the ADS-B data from one aircraft into separate flights, the same method as described in section 3.3.1 was used, which looks for breaks of more than 1000 s in the ADS-B data. Then, the ADS-B data was interpolated as described in section 3.5. Finally, the same methods for calculating the risk indicators as described in the previous sections were applied.

4.3.1 Comparing risk indicators for different airlines

A list of the registrations of their A320 aircraft was retrieved for three different European Airlines (Brussels Airlines, Lufthansa and Vueling Airlines) from *airfleets.net*. The A320 was chosen, because all of the airlines had a big fleet of this aircraft type and it is mainly used for short range flights. This means that there is better coverage of ADS-B data along the entire flight. The ADS-B data of these aircraft was then analyzed using data from 6am to 1pm on 18.04.2018. To be able to compare the risk indicators during different flights that in general have a different length, only 1,000 datapoints from equally distributed positions were used for every flight. The time is thus only given in percentage of the flight elapsed. Afterwards, the mean value for every airline for every normalized point in time was calculated.

Figure 4-18 shows the average over all flights for the values of the airspace density around the aircraft during the course of the flight, calculated as described in section 4.2.2. The values for all airlines are similar during cruise flight but differ more closer to takeoff and landing. Around departure and destination airports, Lufthansa has the highest number of aircraft in close vicinity, Brussels Airlines is on the second position and Vueling Airlines has the least. This is reflected in Figure 4-19, which shows the mean horizontal distance to the closest surrounding aircraft, calculated according to section 4.2.1. The distance to other aircraft is smallest for all airlines close to takeoff and landing, while Lufthansa has the smallest distances of all three airlines.

This might be explained by the type of airport these airlines fly to commonly. Lufthansa has many flights going to busy airports like Frankfurt or London Heathrow, where there is immense traffic. Brussels Airlines is also going to large airports, while Vueling Airlines is a low-cost carrier mainly going to cheaper and smaller airports. This theory could not be confirmed completely by the size of their home airports though. Lufthansa has its home airport in Frankfurt, which is one of the biggest European airports with 462,885 aircraft movements in 2016 [29]. 30 % of the Lufthansa flights, where the destination airport could be detected, were landing there. Vueling is based in Barcelona, which explains why 25 % of its flights landed there. This airport had 302,458 aircraft movements in 2016 [30], which is less than Frankfurt, thus it fits the trend in the airspace density. Brussels airport however had only 223,688 movements in 2016 [31], while the mean airspace density for Brussels Airlines flights is higher

than for Vueling flights. This means that probably more factors than just the number of aircraft movements contribute to the height of the airspace density.

Figure 4-20 shows the vertical distance to the closest aircraft. The distance is mostly around 2,000 ft or larger for all three airlines, which makes sense as this is generally the separation minimum as described in section 2.3. This decreases to 1,000 ft close to landing, as is allowed in controlled airspace, again with the trend of Lufthansa having the smallest distances.

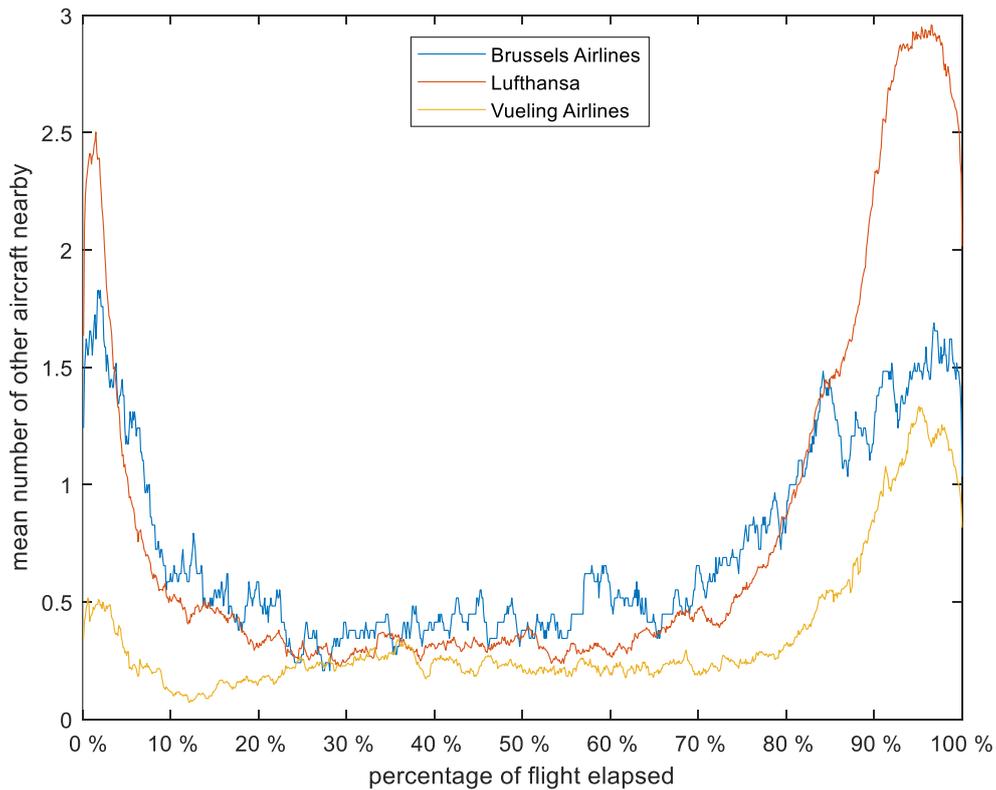


Figure 4-18: Comparison of mean number of surrounding aircraft during flight for different airlines

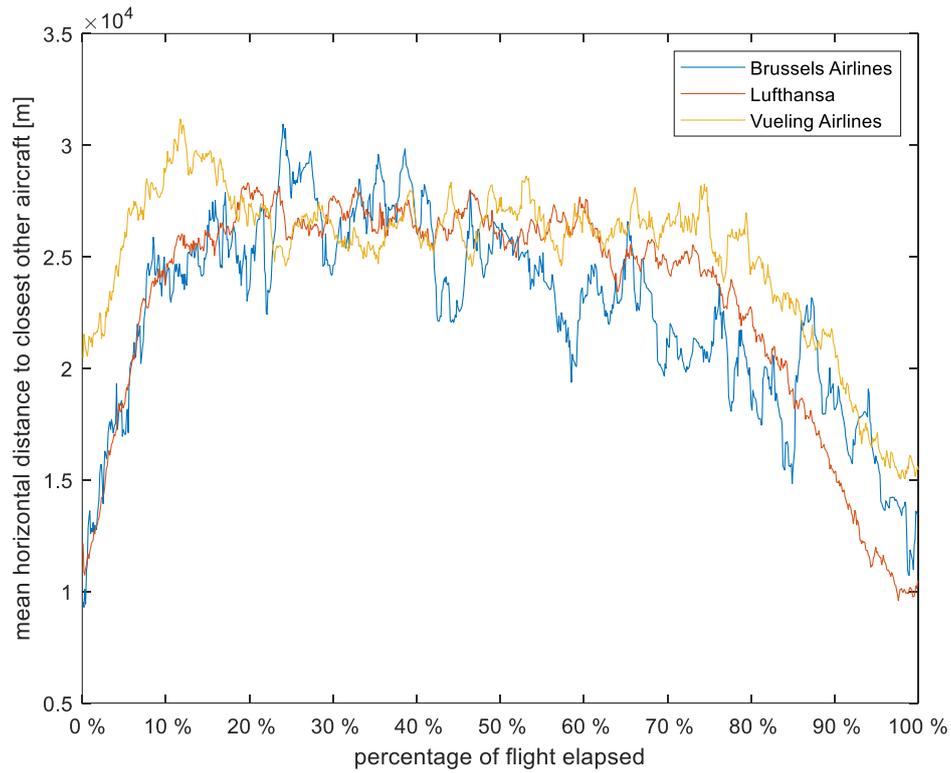


Figure 4-19: Comparison of mean horizontal distance to closest surrounding aircraft during flight between airlines

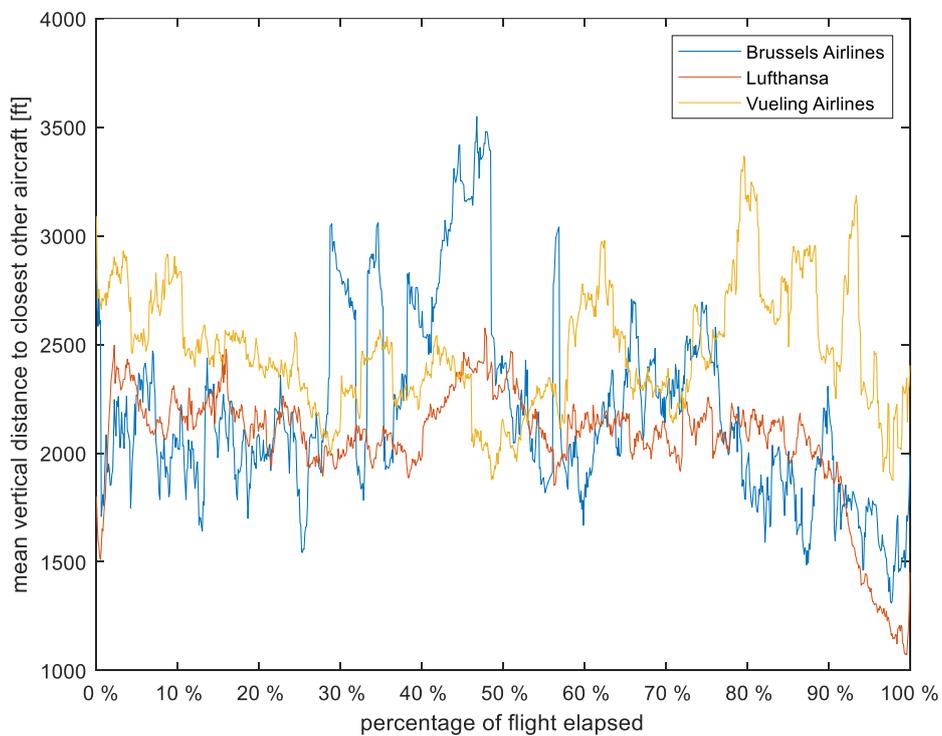


Figure 4-20: Comparison of mean vertical distance to closest surrounding aircraft during flight between airlines

4.3.2 Comparing distances in final approach for different airports

To confirm the differences in separation between airports, the distances in final approach for different airports were evaluated. To assign the destination airport to a flight, the airport database from the Institute of Flight System Dynamics is used. First, a database of all European airports was utilized, but this included also non-commercial and military airports. This led to many of the non-commercial airports being falsely set as destination airports, because they were very close to other commercial airports. To avoid this, the database was filtered to contain only bigger commercial airports. If the endpoint of a flight based on ADS-B data was below 5,000 feet, the distance to all airports was calculated and the closest airport within 30 km was assigned as destination airport. The algorithm works for a lot of the flights, but still no destination airport could be assigned to 40% of the flights, because the ADS-B data stopped too early. Again, if the airport elevation was above 5,000 ft, no landings could be detected because only barometric altitude is known in the ADS-B data.

Using the calculation as described in section 4.2.4, the distance at every position recording to the next aircraft in front during final approach for every flight was calculated. Afterwards, the minimum and the mean value of the distances were stored for every flight. For every flight with a certain destination airport, the average value for both, the mean and the minimum values, was calculated. Table 4-3 shows these values for the flights of the airlines compared in section 4.3.1 evaluated with data from 6:00 am UTC to 9:00 pm UTC on 18.04.2018. Only airports where more than ten flights were detected are shown, to prevent possible outliers from falsifying the result when the sample of flights is too small.

| Airport | Number of detected flights | Mean of minimum distance to next aircraft in front | Mean of mean distance to next aircraft in front |
|----------------------------|----------------------------|--|---|
| Brussels Airlines | | | |
| Brussels Airport | 29 | 16,434 m | 17,639 m |
| Lufthansa | | | |
| London Heathrow Airport | 10 | 6,397 m | 8,383 m |
| Düsseldorf Airport | 12 | 7,777 m | 7,926 m |
| München Airport | 73 | 9,243 m | 10,287 m |
| Frankfurt Airport | 120 | 10,771 m | 13,731 m |
| Vueling Airlines | | | |
| Amsterdam Airport Schiphol | 14 | 7,354 m | 11,331 m |
| Barcelona Airport | 113 | 10,471 m | 11,116 m |
| Madrid Barajas Airport | 13 | 12,335 m | 13,068 m |
| Rome Fiumicino Airport | 26 | 12,843 m | 13,469 m |
| Paris Orly Airport | 12 | 16,391 m | 18,816 m |

Table 4-3: Comparison of distances to next aircraft in front in approach for different airports

5 Conclusions and perspective

In the course of this bachelor thesis, an algorithm was developed to enrich QAR data with MAC risk indicators derived from ADS-B data. First, as source for ADS-B data the website *ADSBexchange* was chosen, mainly because of the free access to the complete historical datasets. A method was developed to process the downloaded ADS-B data. In doing so, runtime and storage problems were solved by optimizing the storage process and keeping only relevant data. Afterwards, methods were developed to merge ADS-B and QAR data using a reference point, which allowed to compare both data sets with each other. This demonstrates that the position recordings in the ADS-B data are similar to the ones from the QAR data but that there are potential issues with incorrect timestamps in the ADS-B data. To compensate this, an algorithm was formulated in order to detect outliers in the ADS-B data and delete them before interpolation. After the ADS-B data had been prepared, MAC risk indicators were defined mainly quantifying the air traffic around a flight with QAR data available. Table 5-1 is a summary of all variables and risk indicators that are added to a QAR dataset by the developed algorithm.

| Name | Description |
|------------------------------|--|
| adsbTime_s | timestamps for QAR position recordings after merging, in seconds since 00:00 UTC that day |
| min_hor_dist_m | horizontal distance to the closest surrounding aircraft in meters |
| min_vert_dist_m | vertical distance to the horizontally closest surrounding aircraft in meters |
| min_direction_deg | direction to the horizontally closest surrounding aircraft counted clockwise in degrees with 0 being directly in front |
| min_track_deg | track of the horizontally closest surrounding aircraft |
| numAC | number of surrounding aircraft |
| TA | 1 if a TA was detected, 0 otherwise |
| RA | 1 if a RA was detected, 0 otherwise |
| min_dist_infront_m | distance to closest aircraft in front during final approach |
| min_dist_behind_m | distance to closest aircraft behind during final approach |
| numAC_landing_infront | number of aircraft on same final approach track still to land in front |

Table 5-1: Summary of all variables added to a QAR dataset

A selection of ADS-B data of A320 aircraft of different airlines was analyzed to compare the MAC risk indicators for their fleet, which proved that based on this selection of three airlines, Lufthansa flights had the highest airspace density during departure and landing. Also, the different airports they flew to were ranked by the distances during final approach for these flights.

Still, there are some possibilities for future work. As *ADSBexchange* meanwhile changed its access policy for historical ADS-B data, either the new requirements have to be met, including providing a receiver and payment, or a different data source has to be found. A possibility for future work might be to use the *Opensky Network*, as the access to their historical database was granted after the work had been started with *ADSBexchange*.

5 Conclusions and perspective

In section 3.4, methods to find errors in the ADS-B data were discussed, but as there might still exist some wrong data that is not detected, these methods can still be improved. When more recent QAR data is available so that the receiver ID of the corresponding ADS-B data is known, this can be used to further identify faulty receivers and to correct or delete data coming from those receivers.

To solve the issue of only barometric altitude being available, the *getTerrainElevation* function from the Institute for Flight Systems Dynamics might be used. This function retrieves the terrain elevation at a certain position. The altitude above ground for the aircraft can be calculated from this, which would increase the accuracy of TCAS SL selection and final approach detection.

In future work, the set of algorithms developed during this thesis can be used to enrich QAR data with information about surrounding air traffic. This can then be used for future risk analysis of the accident type MAC. The use of Subset Simulation as proposed by Mishra et al. [32] is one possibility to estimate the probability of conflict between two aircraft, where the probabilities are usually quite small. With more receivers and more ADS-B equipped aircraft, the data coverage will increase in the future. This will eventually increase the possibilities for analysis even further, especially for long haul flights as at the moment large parts of these flights are without data coverage. Using these methods, it will be possible for airlines to evaluate the MAC risk for their fleet, and eventually reduce the risk where it is particularly high.

References

- [1] Airbus, Growing Horizons Global Market Forecast 2017-2036, Toulouse, 2017.
- [2] Aviation Safety, Boeing Commercial Airplanes, Statistical Summary of Commercial Jet Airplane Accidents–Worldwide 1959-2016, Seattle, 2017.
- [3] A. Nunes and T. Laursen, "Identifying the Factors that Contributed to the Ueberlingen Midair Collision," in *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*, 2004.
- [4] Federal Aviation Administration, *Introduction to TCAS II, Version 7.1*, 2011.
- [5] *Safety Management Manual (SMM), 3rd ed.*, International Civil Aviation Organization, Doc. 9859, Montreal, 2013.
- [6] International Civil Aviation Organization, *Aviation Occurrence Categories, Definitions and Usage Notes, ed. 4.2*, 2011.
- [7] EOFDM Working Group A, *Review of Mid Air Collision (MAC), Precursors from an FDM Perspective*, 2018.
- [8] "ADS-B," SKYbrary, [Online]. Available: <https://www.skybrary.aero/index.php/ADS-B>. [Accessed 22 08 2018].
- [9] M. Schäfer, M. Strohmeier, M. Smith and M. Fuchs, "OpenSky Report 2016: Facts and Figures on SSR Mode S and ADS-B Usage," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, 2016.
- [10] W. R. Richards, K. O'Brien and D. C. Miller, "New Air Traffic Surveillance Technology," *Boeing Aero Quarterly*, pp. 7-13, 2010.
- [11] European Commission, *COMMISSION IMPLEMENTING REGULATION (EU) No 1028/2014*, Brussels, 2014.
- [12] Federal Aviation Administration, *14 CFR Part 91, Automatic Dependent Surveillance-Broadcast (ADS-B) Out Performance Requirements to support Air Traffic Control (ATC) Service, Final Rule*, Washington, 2010.
- [13] M. Strohmeier, M. Schäfer, V. Lenders and I. Martinovic, "Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B," *IEEE Communications Magazine*, pp. 111-118, May 2014.
- [14] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *IPSN-14 Proceedings*

- of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, 2014.
- [15] "ADSBexchange," [Online]. Available: <https://www.adsbexchange.com/>. [Accessed 22 08 2018].
- [16] "Flightradar24," [Online]. Available: <https://www.flightradar24.com/about>. [Accessed 22 08 2018].
- [17] "FlightAware," [Online]. Available: <https://de.flightaware.com/about/>. [Accessed 22 08 2018].
- [18] "The OpenSky Network," [Online]. Available: <https://opensky-network.org>. [Accessed 22 08 2018].
- [19] Federal Aviation Administration, *Advisory Circular 120-82, Flight Operational Quality Assurance*, 2004.
- [20] "Mid-Air Collision," SKYbrary, [Online]. Available: https://www.skybrary.aero/index.php/Mid-Air_Collision. [Accessed 22 08 2018].
- [21] "AIRPROX," SKYbrary, [Online]. Available: <https://www.skybrary.aero/index.php/AIRPROX>. [Accessed 22 08 2018].
- [22] "Loss of Separation," SKYbrary, [Online]. Available: https://www.skybrary.aero/index.php/Loss_of_Separation. [Accessed 31 08 2018].
- [23] "Separation Standards," SKYbrary, [Online]. Available: https://www.skybrary.aero/index.php/Separation_Standards. [Accessed 22 08 2018].
- [24] Federal Aviation Administration, *Order 8020.11D*, 2018.
- [25] Mathworks. [Online]. Available: <https://de.mathworks.com/help/map/ref/distance.html>. [Accessed 22 08 2018].
- [26] M. Kerrisk, "The Linux Programming Interface," San Francisco, No Starch Press, 2010, p. 10.1.
- [27] Mathworks. [Online]. Available: <https://de.mathworks.com/help/matlab/ref/isoutlier.html>. [Accessed 22 08 2018].
- [28] C. Munos, A. Narkawicz and J. Chamberlain, "A TCAS-II Resolution Advisory Detection Algorithm," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, Hampton, 2013.
- [29] Fraport AG, *Frankfurt Airport Air Traffic Statistics*, 2016.
- [30] "Aircraft Movements," Barcelona Airport, [Online]. Available: <https://www.barcelonaairport.net/aircraft-movements.shtml>. [Accessed 22 08 2018].

[31] brussels airport, *BRUtrends2016*.

[32] C. Mishra, S. Maskell, S.-K. Au and J. F. Ralph, *Efficient estimation of probability of conflict between air traffic using Subset Simulation*, Liverpool, 2016.

Appendix

Available variables in datasets from *ADSBexchange*. source:

<http://www.virtualradarserver.co.uk/Documentation/Formats/AircraftList.aspx#response>

| Property | Description |
|----------|--|
| Id | The unique identifier of the aircraft. |
| TSecs | The number of seconds that the aircraft has been tracked for. |
| Rcvr | The ID of the feed that last supplied information about the aircraft. Will be different to <i>srcFeed</i> if the source is a merged feed. |
| Icao | The ICAO of the aircraft. |
| Bad | True if the ICAO is known to be invalid. This information comes from the local <i>BaseStation.sqb</i> database. |
| Reg | The registration. |
| Alt | The altitude in feet at standard pressure. |
| GAlt | The altitude adjusted for local air pressure, should be roughly the height above mean sea level. |
| InHg | The air pressure in inches of mercury that was used to calculate the AMSL altitude from the standard pressure altitude. |
| AltT | The type of altitude transmitted by the aircraft: 0 = standard pressure altitude, 1 = indicated altitude (above mean sea level). Default to standard pressure altitude until told otherwise. |
| TAlt | The target altitude, in feet, set on the autopilot / FMS etc. |
| Call | The callsign. |
| CallSus | True if the callsign may not be correct. |
| Lat | The aircraft's latitude over the ground. |
| Long | The aircraft's longitude over the ground. |
| PosTime | The time (at UTC in JavaScript ticks) that the position was last reported by the aircraft. |
| Mlat | True if the latitude and longitude appear to have been calculated by an MLAT server and were not transmitted by the aircraft. |
| PosStale | True if the last position update is older than the display timeout value - usually only seen on MLAT aircraft in merged feeds. |
| IsTisb | True if the last message received for the aircraft was from a TIS-B source. |
| Spd | The ground speed in knots. |
| SpdTyp | The type of speed that <i>Spd</i> represents. Only used with raw feeds. 0/missing = ground speed, 1 = ground speed reversing, 2 = indicated air speed, 3 = true air speed. |
| Vsi | Vertical speed in feet per minute. |
| VsiT | 0 = vertical speed is barometric, 1 = vertical speed is geometric. Default to barometric until told otherwise. |
| Trak | Aircraft's track angle across the ground clockwise from 0° north. |
| TrkH | True if <i>Trak</i> is the aircraft's heading, false if it's the ground track. Default to ground track until told otherwise. |
| TTrk | The track or heading currently set on the aircraft's autopilot or FMS. |
| Type | The aircraft model's ICAO type code. |

| | |
|--------------|--|
| Mdl | A description of the aircraft's model. Usually also includes the manufacturer's name. |
| Man | The manufacturer's name. |
| CNum | The aircraft's construction or serial number. |
| From | The code and name of the departure airport. |
| To | The code and name of the arrival airport. |
| Stops | An array of strings, each being a stopover on the route. |
| Op | The name of the aircraft's operator. |
| OpCode | The operator's ICAO code. |
| Sqk | The squawk as a decimal number (e.g. a squawk of 7654 is passed as 7654, not 4012). |
| Help | True if the aircraft is transmitting an emergency squawk. |
| Dst | The distance to the aircraft in kilometres. |
| Brng | The bearing from the browser to the aircraft clockwise from 0° north. |
| WTC | The wake turbulence category of the aircraft - see enums.js for values. |
| Engines | The number of engines the aircraft has. Usually '1', '2' etc. but can also be a string - see ICAO documentation. |
| EngType | The type of engine the aircraft uses - see enums.js for values. |
| EngMount | The placement of engines on the aircraft - see enums.js for values. |
| Species | The species of the aircraft (helicopter, jet etc.) - see enums.js for values. |
| Mil | True if the aircraft appears to be operated by the military. |
| Cou | The country that the aircraft is registered to. |
| HasPic | True if the aircraft has a picture associated with it. |
| PicX | The width of the picture in pixels. |
| PicY | The height of the picture in pixels. |
| FlightsCount | The number of Flights records the aircraft has in the database. |
| CMsgs | The count of messages received for the aircraft. |
| Gnd | True if the aircraft is on the ground. |
| Tag | The user tag found for the aircraft in the BaseStation.sqb local database. |
| Interested | True if the aircraft is flagged as interesting in the BaseStation.sqb local database. |
| TT | Trail type - empty for plain trails, 'a' for trails that include altitude, 's' for trails that include speed. |
| Trt | Transponder type - 0 =Unknown, 1 =Mode-S, 2 =ADS-B (unknown version), 3 =ADS-B 0, 4 =ADS-B 1, 5 =ADS-B 2. |
| Year | The year that the aircraft was manufactured. |
| Sat | True if the aircraft has been seen on a SatCom ACARS feed (e.g. a JAERO feed). |
| Cos | Short trails |
| Cot | Long trails |
| ResetTrail | True if the entire trail has been sent and the JavaScript should discard any existing trail history it's built up for the aircraft. |
| HasSig | True if the aircraft has a signal level associated with it. |
| Sig | The signal level for the last message received from the aircraft, as reported by the receiver. Not all receivers pass signal levels. The value's units are receiver-dependent. |