ABSTRACT

Title of Thesis:	EXTRACTING CAPACITY METRICS FOR GENERAL AVIATION AIRPORTS FROM ADS-B DATA
	Danae Zoe Mitkas, Master of Science, 2021
Thesis Directed By:	Professor David J. Lovell, Department of Civil and Environmental Engineering, and Institute for Systems Research

General Aviation airports play a pivotal role in the aviation system of the US, with over 5000 small airports existing and operating across the country. Serving almost exclusively small aircraft, these airports have a unique activity profile, compared to larger commercial airports. Like their larger cousins, they occasionally see the need to apply for federal funding for capacity enhancement projects, a process that requires rigorous documentation of the demand and capacity situation at the airport. Existing models for capacity estimation have been calibrated to reflect the much larger scale features that dominate large airports. The main challenge is to develop a method to provide precise data for small airports that operate mainly with small single or multi-engine aircraft. These airports are typically not towered and, hence, do not benefit from traditional automated data collection technologies. This research addresses the issues of a) collecting aircraft data at local airport environments from aircraft equipped with Automated Dependent Surveillance – Broadcast (ADS-B) technology, b) processing the data to determine and classify flights, and c) assessing elements of the operational performance of these aircraft. The thesis proposes a method to extract aircraft approach speeds and runway occupancy times, which are important contributors to capacity estimation. We applied and validated our method in three small airports.

EXTRACTING CAPACITY METRICS FOR GENERAL AVIATION AIRPORTS FROM ADS-B DATA

by

Danae Zoe Mitkas

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2021

Advisory Committee:
Professor David J. Lovell, Chair
Professor Paul Schonfeld
Professor Deb Niemeier

Preface

Almost two years ago, when I joined the Department of Civil and Environmental Engineering at the University of Maryland, College Park, my journey towards a Master's in Transportation Engineering started with a promise of unobstructed flying to the skies only to end in solitary confinement in my apartment monitoring flights on my computer screen. Before I even attended my first class at UMD, I had the good fortune to meet Prof. David J. Lovell and learn about his latest research project. I was excited with the prospect to study airfields and aircraft, and I decided to join his research team.

During the past two years, I had the opportunity, through classwork and research, to learn more about Aviation and Data Analytics, two areas that I had barely touched during my undergraduate studies at Aristotle University in Greece.

My passage through UMD was not as eventful as I had hoped, due to the COVID-19 restrictions. However, my classes were stimulating, my research project was challenging and allowed me to be creative, and I was able to participate in conferences and present early results of my research.

I would like to express my gratitude to Prof. Lovell, my Thesis advisor, for his constant support and guidance, and for providing the opportunity to get involved with the Aviation field. Our frequent discussions would always advance my research and help me move forward and find the ways to overcome any difficulties.

Many thanks go to Yeming Hao, my (virtual) teammate, for generously providing her knowledge, assistance and emotional support.

I would also like to thank Dr. Seth Young, Professor at The Ohio State University and partner in this project, for his constant insight on anything related to General Aviation, both as a Professor and researcher as well as an aviation instructor.

Lastly, I would like to thank Hui Jeong Ha and Sandeep Venkatesh, graduate students at The Ohio State University, and partners along this journey of exploring aircraft performance at General Aviation airports.

Dedication

To my brother Alex, and

My parents Pericles and Sophia

Acknowledgements

This work has been sponsored by the Federal Aviation Administration's Office of Airports via the Administration's university consortium for Aviation Operations Research (NEXTOR III). This work is part of the FAA's larger effort to enhance operational capacity models for airports serving primarily smaller aircraft. Specifically for sponsoring the research on "Small Aircraft Capacity Modeling Factors" and providing the motivation for this study. The author would like to acknowledge Mr. Kent Duffy, National Resource Expert for Airport / Airspace Capacity, FAA for his support of this work.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

As the demand for air transportation increases worldwide, so does the demand for General Aviation (GA) airports. GA airports constitute an important part of the US air transportation system, with thousands of them operating across the US and contributing highly to the daily air traffic within the National Airspace. The small aircraft, which mainly operate at these airports, will not often perform long journeys, but it is common for them to operate at a local level, flying in circles around their base airport or completing short and frequent trips between neighboring airports. Even though most GA airports do not have Air Traffic Control (ATC) towers to monitor the operations, it is still important to have a process that can collect and provide information about their daily activity.

An airport's major goal is to control the flow of traffic in the most strategic and efficient way, to maximize its capacity. Optimizing capacity requires comprehensive knowledge of the existing conditions and demand levels of the airport, and therefore making the most appropriate decisions that will match both the geometry of the airport and the type of activity it accommodates. For the case of GA airports, this procedure has been proven more challenging because small aircraft activity varies widely in terms of aircraft performance and operations, and in many cases relies on the decisions of the pilots. Therefore, aircraft activity cannot be easily predicted or estimated, but rather must be captured at the moment it is occurring [1][2].

Section 1.1: Background

As in the case of larger airports, small airports will eventually reach the need to request federal funding to either expand and maintain existing facilities or construct additional facilities. In order to be eligible for such funding, airports need to submit proof that supports the need for expansion, and most importantly to establish that the airport is operating at capacity levels. For this to be possible, an airport needs the appropriate tools that will ensure both reporting accurate numbers of operations and calculating its actual capacity. Providing self-reported data collected from manual counts can be questionable and generally would not constitute sufficient support for funding. Additionally, as further explained in chapter 2, current capacity estimation models are not suitable for GA airports that accommodate mainly small aircraft.

Small airports often host flight schools, resulting in training aircraft performing multiple takeoffs and landing in short time intervals, creating a unique and dense flight pattern. Thus, when studying the capacity of a small airport it is important to consider the existence of an active flight school and examine its activity closely. However, a high proportion of touch-and-go activity can accommodate a lot more takeoffs and landings, resulting in an increased runway throughput. When an aircraft is obliged to do a full-stop landing, it will require a significant reduction of its speed to be able to turn and exit the runway, and therefore will lead to an increased runway occupancy time.

Section 1.2: Aim of the Research

The aim of this research is twofold: a) to develop a data collection scheme that will provide the necessary inputs for capacity estimation, and b) to utilize the data collected to extract capacity metrics. As thoroughly explained in chapter 3, the proposed method includes collecting Automatic Dependent Surveillance (ADS-B) data from aircraft, a technology that is becoming increasingly popular over the past years. Collecting data that are communicated directly by the aircraft provides a comprehensive and fairly accurate description of each operation because it minimizes human error. For the case of GA airports, it is important to collect data that are as detailed as possible, because small alterations in the performance can affect significantly the operations of the airport. The ability to utilize the collected ADS-B data to calculate the necessary metrics for capacity estimation, such as Average Approach Speed and Runway Occupancy Time, can set the basis for creating an accurate capacity estimation model for small GA airports.

Section 1.3: Methodology of the Research

The methodology followed for this research includes the following steps:

a) Literature Review:
   Existing information relevant to the topics discussed throughout the thesis was reviewed and revealed that neither the methods for data collection in small airports were standardized, nor the existing models for airport capacity estimation were suitable for GA airports. At this point, the objectives of this research were delineated.
b) Data Collection:
   After investigating existing databases, it was decided to proceed with collecting our own data, directly from the study locations. The technology chosen as most appropriate for this study was ADS-B. After understanding the inner workings of ADS-B, receivers were installed at three participating airports, to collect aircraft data. Data collection proceeded for several months.
c) Data Analysis and Preparation:
   Preliminary analysis of the collected data provided useful insight and led to the detection of several anomalies. Further analysis and processing were necessary to recognize all the discrepancies in the data and to add "flags" or correct the data before moving to the next steps. Detecting erroneous values and correcting any problems before utilizing the data increases the reliability of the output.
d) Extracting Capacity Metrics:
   After the data are cleaned, they can be utilized to extract important metrics for capacity estimation. In this research, we chose to analyze two of the factors that determine runway capacity: the Average Approach Speed of aircraft classes and the Average Runway Occupancy Times. The values for these two parameters were calculated for each airport.

In our analysis, the following software was used:

i. **R**: The collected data volume is high and therefore it was decided to R for both the preparation steps and the extraction of metrics. R allows the processing of large amounts of data and the visualization of the results. Graphs and plots throughout the Thesis have been produced using R.
ii. **Postgres SQL**: All data collected are stored in the online Postgres SQL database created for this project. The two most important tables of the database, the "adsb_messages" table which includes all the messages collected and the "flights" table which includes information per flight and the "flags" that have been generated by the preparation steps, are discussed in detail in section 3.
iii. **QGIS**: Various tools were created and used on the QGIS geographic information system software platform to visualize and analyze the relevant performance characteristics of individual flights. QGIS was also used for mapping the data and understanding the different operations.

Chapter 2: Literature Review

Detecting and counting aircraft can be challenging, especially in the case of small aircraft. Visual detection or radar sensors may not always be effective with small size and low velocity aircraft. Capacity studies require data to be collected over long periods of time; hence, manual observations are not feasible. Small airports typically do not have any kind of surveillance radar or other automated data collection mechanisms already installed. Therefore, other techniques must be investigated for data collection in small airports. The aim of this study is to recognize and stratify data collected at small General Aviation (GA) airports. General Aviation refers to all air traffic that is not commercial or military. The GA airports play a pivotal role in the economy and the aviation system of the US, with over 5000 small airports existing and operating across the country. As these airports tend to have different characteristics and unique activity, compared to regular airports, collecting and stratifying data becomes even more complex.

Section 2.1: Airport capacity

Capacity estimation is a very important procedure for any airport and requires careful steps when it comes to small airports, with higher sensitivity and multiple limitations. Small airports are affected more by minor changes in aircraft activity, weather conditions or sudden events. Existing capacity models were calibrated to reflect the much larger scale features that dominate large airports; they do not provide meaningful results when it comes to small airports, since small aircraft and the discriminating features of small airports have low impact on the final result. This can be an important problem for airports that operate mainly with small aircraft. Therefore, the main challenge is to create a method that will provide precise data for small airports that operate mainly with small single or twin-engine aircraft.

Before describing the ways to estimate airport capacity, we first have to define it. Airfield capacity is the maximum number of aircraft that can be accommodated by an airport in a given period of time, and it can be measured either as airside capacity or runway capacity. However, airport capacity does not provide enough information on its own, unless it is compared to a measure of demand. A demand-capacity comparison would provide enough information to understand the performance of an airport and its ability to accommodate aircraft. There are several computer simulation models that can be used for capacity estimation, which require a variety of inputs. However, these models usually provide the most accurate results for larger, commercial airports.

The need to update airfield capacity models is even greater for small airports. The existing capacity estimation methods proposed by the "ACRP Report 79: Evaluating Airfield Capacity" are inadequate for small airports or small aircraft [3]. These airports may encounter capacity issues only during the peak hour, which may not be reflected in the available analysis techniques. The "ACRP Report 79" and the "Airfield Capacity Spreadsheet" are mostly effective for large airports. Additionally, the Advisory Circular (AC) 150/5060-5 contains only one short section (4-5) which refers to capacity estimation for single runway airports or airports used by small aircraft (class A and B) [4]. This technique takes into account only:

- Runway configuration and
- Percent of touch-and-go activity

and provides results of hourly capacity for Visual Flight Rules (VFR) and Instrument Flight Rules (IFR) conditions. These two characteristics might be important for an airport's performance but are not definitive for capacity estimation. Since technology provides the means to collect and analyze more data, these methods ought to be revised and updated, especially for small airports [3][4][5].

Subsection 2.1.1: Airport capacity characteristics and metrics

Capacity estimation models must consider both static and dynamic characteristics of an airport. Some of the most important factors include:

<u>Static Characteristics</u>

1. Runway Configuration:
   It defines the layout of the runway or runways of an airport. Both the number and the position of the runways affect the airport's capacity.
2. Control Tower Availability:
   It describes the presence or not of an Air Traffic Control (ATC) tower. Although one may assume that all airports have ATC towers, it is often that small airports do not have ATC or share towers with another airport.
3. Runway Exits and Parallel Taxiway Availability:
   Runway exits are used by the aircraft to move from a runway to a taxiway or the opposite. The number of exits on a runway is related to its length. A short runway usually has exits only at the two ends of the runway and therefore aircraft have to cross the whole runway to exit. Longer runways may have multiple exits along their length.

<u>Dynamic Characteristics</u>

1. Average Approach Speed of Aircraft Classes:
   It is the speed of the aircraft while approaching the runway for landing. This speed will vary for different segments of an approach as well as by aircraft weight and configuration.
2. Average Arrival Runway Occupancy Time (AROT) of Aircraft Classes:
   It is the average time an aircraft (or certain type of aircraft) occupies a runway after its landing and is measured from the time the aircraft crosses the runway threshold, until the time it fully exits the runway. An aircraft is able to exit a runway either at the end of the runway, or (if available) using an exit at some other point of the runway, leading to the taxiway.
3. Aircraft Separation:
   It describes the spacing (either longitudinal or time) between consecutive aircraft approaching for landing or aircraft getting ready to depart. This spacing can be prescribed either by Air Traffic Control (ATC) or by the pilots.
4. Touch-and-Go Operations:
   Touch-and-Go describes the type of activity where an aircraft arriving to the runway makes a touchdown and immediately (without slowing down or stopping) takes off again. This operational pair is counted as one arrival and one departure and therefore two operations. This type of activity is common in the case of small airports with associated flight schools, where training aircraft perform multiple touch-and-go's daily for practice purposes. This type of activity is not regular for larger airports and in those locations may only be performed in case of an emergency.

The first three factors can be easily identified, whereas the remaining four constitute important metrics related to the airport's activity and require aircraft movement data to be properly measured and identified [3].

Section 2.2: Previous attempts

While the importance of collecting and processing accurate and real-time aircraft data is omnipresent, the challenge of completing this task successfully and effortlessly is yet to be solved. Various attempts have been made for automated aircraft data collection and some of the most relevant will be listed below. Multiple methods have been researched and patents have been filed related to automated aircraft counting with acoustic technology. The first related patent was filed in Dec. 2005; it refers to an automated acoustic data collection system using an Unmanned Aerial Vehicle (UAV) equipped with an antenna array. The UAV is able to collect data while in flight. However, the major challenges encountered include wind noise and the UAV's engine noise. Also, this system can only detect the appearance of something (e.g., aircraft) in the environment, with no additional data. Therefore, such a method would only be useful to provide a rough number of operations counts or only the detection of activity during non-busy hours (e.g., late night) [6].

More recently, another patent was filed (Nov. 2011), which claims to achieve low-cost aircraft detection in areas where ground surveillance radars do not exist or are limited. Referring to both a method and the apparatus needed to detect aircraft in an airport environment, the system takes advantage of the acoustic emissions of the aircraft and translates them to "positional" and "aircraft type" information. Aircraft can be detected and, in some cases, identified by their acoustic emissions. A strong advantage of the system is that it does not require any additional equipment to be carried by the aircraft, since it relies solely on the acoustic emissions. At the same time, this remains one of the main disadvantages, since acoustic emissions do not provide any additional information regarding precise positional information, and the type of aircraft is only estimated by the emission detected by the acoustic sensors, and not communicated by the aircraft [7].

These methods would be considered effective in detecting small aircraft, since they do not rely on the size or type of the aircraft, but only on their acoustic emissions. Also, another significant advantage is that both methods do not require additional equipment to be installed on the aircraft. However, they can only provide gross information regarding the position and in some cases the type of the aircraft. Airport capacity estimation requires detailed data for all the aircraft approaching, taking off or taxiing around the airfield. Such data would require either additional equipment for wider coverage and more detailed collection, or more elaborate procedures for data extraction, processing, and interpretation. In most cases, these data types simply cannot be gathered by these methods.

Subsection 2.2.1: Existing data sources

The next step was to investigate existing data sources that might be able to provide the data needed. These data sources could be either private or open-source databases utilized for research purposes or commercial use. However, the common disadvantage of such databases is that, even though they may contain all the necessary data types that would theoretically be helpful for capacity estimation, they tend to focus on larger airports, of major concern, and prove to be inadequate for small airports. As mentioned, small airports tend to have different behavior and need to be observed closely. For example, the number of based aircraft at a small airport is generally known, but quite often a significant percentage of the based aircraft do not participate in many flying operations per year, so they do not represent the actual activity of the airport. Moreover, flight schools are often located in small airports, and these tend to have aircraft that are utilized more often and contribute highly to the airport's traffic.

The Aviation System Performance Metrics (ASPM) database was investigated, and it was found that it contains information only for 77 ASPM airports and for the ASPM carriers. This led to the conclusion that it does not contain complete records for small airports [8]. Next, the System Wide Information Management System (SWIM) was examined. It can be considered a useful database since it provides real-time, relevant aeronautical, flight and weather information. However, it also proved to be insufficient for small airports and small aircraft [9]. Finally, the Traffic Flow Management System (TFMS) was considered, which contributes data to both ASPM and SWIM. TFMS provides Aircraft Situation Display (ASDI) data, which include aircraft scheduling, routing, and positional information. As in the previous cases, TFMS also lacks data on small aircraft [10].

Aviation related open-source databases collect data from individuals who are willing to set up devices to collect and feed data from any part of the world to a central database. In this way large amounts of data can be collected instantly, with no regional restrictions. Therefore, it was deemed more plausible to find data for small airports in such databases. The first database considered was "FlightAware", which collects and provides aircraft data, mostly for commercial flight tracking purposes. It also provides access to the collected data to researchers or individuals willing to place tracking equipment (receivers) in their own space. The data provided are mostly accurate and useful for the purposes of this study and include information for General Aviation airports as well. However, FlightAware focuses on applications such as flight progress tracking, and visual displays. These do not need high resolution data, and the specific behavior of the aircraft at the endpoints of its journey are not important. As a result, these data tend to be heavily filtered. Unfortunately, it is the data around the airports that are most important for the present purposes, and the en route data are the ones that are irrelevant. Finally, because FlightAware and other such services are essentially crowd sourced, they are restricted to locations where volunteers have installed equipment, which does not include many small airports. The resulting data are only as good as the implementation (e.g., software quality, hardware quality, antenna placement, robustness to communications dropouts and power outages, etc.), which we have no control over when using crowd sourced data.

Another open-source database for aviation related data is "Opensky", which is a community-based receiver network collecting air traffic surveillance data. The main advantage of the Opensky network database is that it keeps all the raw unfiltered data as collected by the receivers. The reason this database was rejected is its lower coverage, especially across the US. It is still a developing network and therefore receives data mostly from areas closer to major airports, lacking data for small airports. The heat map in Fig. 1 indicates the reception coverage of the network, with the darker spots (higher reception) concentrated around major airports and big cities. Again, it is subject to the vagaries in the volunteer installations.

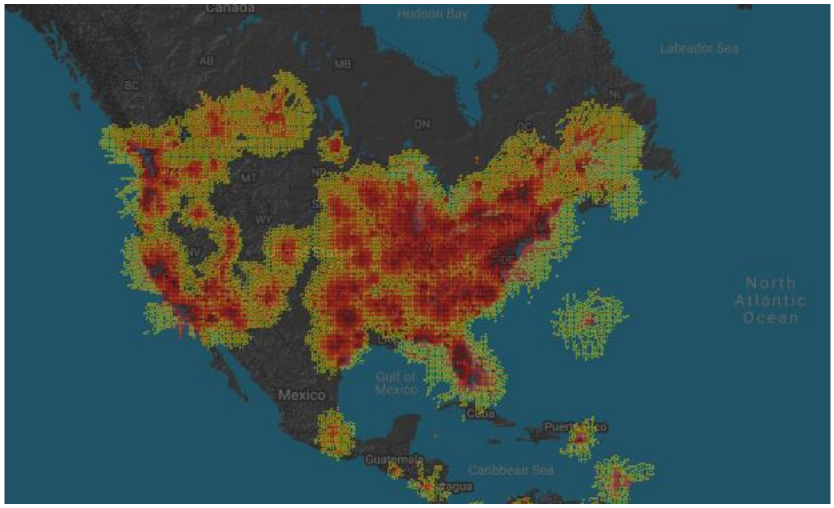


Figure 1: Opensky network coverage across the US (opensky-network.org)

Therefore, all mentioned databases were decided to be inadequate for the purposes of this study.

Finally, after identifying the data needed for capacity estimation, and investigating the available databases, it was decided that the most preferable option is to develop a novel system to collect the necessary data from General Aviation airports. Since the mentioned approaches of aircraft detection do not seem to provide the information needed, the use of ADS-B (Automatic Dependent Surveillance – Broadcast) technology appeared to be the most appropriate method. A more detailed description of ADS-B technology and the data it yields, will be given in the following chapter.

Chapter 3: Data Collection and Preparation

For this study, data were collected from specific General Aviation (GA) airports in various locations across the US. The data collection was performed using the ADS-B technology, which is an integral part of air transportation. Automatic Dependent Surveillance – Broadcast (ADS-B) is a technology intended to supplement ground-based radars by enabling participating aircraft to broadcast their own kinematic data (position, altitude, and speed), as well as other relevant data at regular intervals. According to the FAA, all aircraft that want to fly in controlled airspace must be equipped with ADS-B out, making ADS-B data widely available. Aircraft can be equipped with 1090ES or UAT transponders and transmit messages at the 1090 or 978 MHz frequency, respectively. ADS-B 1090ES is required for aircraft flying above 18,000 ft., or for locations outside the USA. UAT transponders are limited to use within the United States and for aircraft flying at lower altitudes. Ground stations receive and repeat messages both in 978 MHz (UAT) and 1090 MHz (1090ES) and an aircraft transmits a message twice every second. ADS-B receivers collect aircraft data from any equipped aircraft that is detected within range. Thus, ADS-B can help in automated data collection and accurate operation counts.

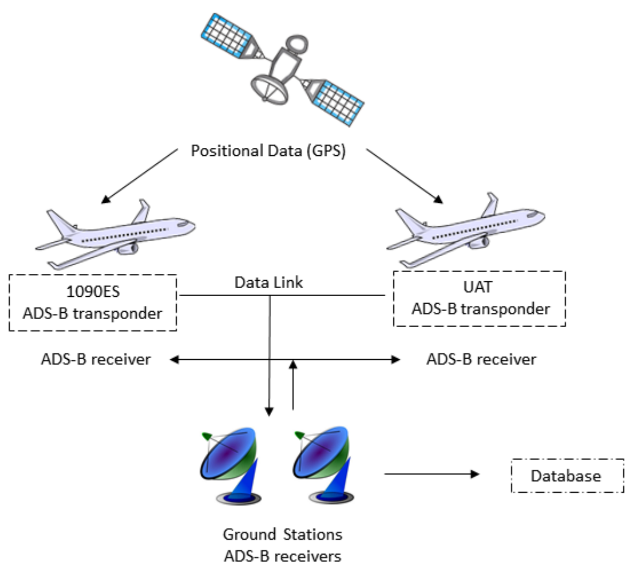


Figure 2:ADS-B system architecture

The ADS-B technology is shown schematically in Figure 2. An aircraft receives its position information from a constellation of GPS (Global Positioning System) satellites. Using its ADS-B transponder, it feeds the position information along with other data to ADS-B receivers. Receivers feed their collected raw data to larger databases and retransmit the information to other aircraft [11][12][13][14].

Section 3.1: Data Collection

For the data collection, ADS-B receivers have been placed at each location, in this case at each airport participating in the study (presented in subsection 3.3.1). Two receivers are set at each airport, one for 1090 MHz and one for 978 MHz frequency messages. Technically, it is possible to receive both types of messages with one receiver; however, this division was selected for convenience and to avoid possible corruption of messages in the cases where a lot of messages arrive at the same time. An ADS-B receiver is composed of the antenna, bandpass filter, and RF (radio frequency) amplifier (the analog section), a software-defined radio (SDR, the digital section), and some processing unit. To ensure high quality ADS-B reception, the antenna must have good line-of-sight. In this study it was ensured that in all cases the antenna had direct line-of-sight to the runway(s) and was close to the runway level. As mentioned, two receivers were placed in each location, meaning two separate antennae and two separate processors to handle the incoming data, all of which then feed the data to a common server [14].

Subsection 3.1.1: Explaining the data

After the receivers have been set, they automatically start collecting any data from aircraft within reception range. At this point, it is important to mention that the gain of the RF amplifier has to be adjusted accordingly based on the location and the amount of activity in the area. If the airport at which the receivers are set is located close to other airports and experiences a lot of overflight activity from larger commercial aircraft, then a lot of messages will be detected at the same time. This will cause issues, especially in the case of the 1090 receiver, since this is the frequency mostly used by large aircraft. As a result, the receiver will end up collecting corrupted messages and missing the ones that are important for the airport. Lowering the gain might help reduce the number of messages received from overflights and will increase the reception of activity near and on the runway. Examples of the effect of different gain will also be presented later in this chapter.

The messages collected must be filtered and decoded to provide meaningful information. Once it is demodulated in the SDR, each message appears as a string of 112 binary bits, which are mapped into 16 hexadecimal characters. Of those 112 bits, 24 bits are the unique ICAO (International Civil Aviation Organization) aircraft identification number (actually the ID of the transponder) and 56 bits are the ADS-B data. The remaining bits are used for parity checking and other communications details. The following table describes the structure of the ADS-B message.

Bit	No. of bits	Abbreviation	Information
1-5	5	DF	Downlink Format
6-8	3	CA	Transponder capability
9-32	24	ICAO	ICAO aircraft address
33-88	56	ME	Message, extended squitter
(33-37)	(5)	(TC)	(Type code)
89-112	24	PI	Parity/Interrogator ID

Table 1: Structure of ADS-B frame

The most useful amount of information is contained in the Type Code. More specifically, the data frames include the aircraft identification, the surface position, the airborne position, the airborne velocities, and aircraft status messages. In our system, each hexadecimal message is stored along with its timestamp, the

Downlink Format, and the ICAO address. The timestamp must be added by the downstream data collection process because raw ADS-B messages do not contain any inherent timing information. Once it is stored, the next step is to decode the message to make it comprehensible. The messages get decoded using the *pyModeS* library, which is a Python library designed to decode Mode-S messages, including ADS-B messages. This large amount of data collected and decoded is then loaded into a PostGreS SQL database and populates the tables. Figure 3 depicts the process of collecting and storing the data on the online database [16][17][18][19].

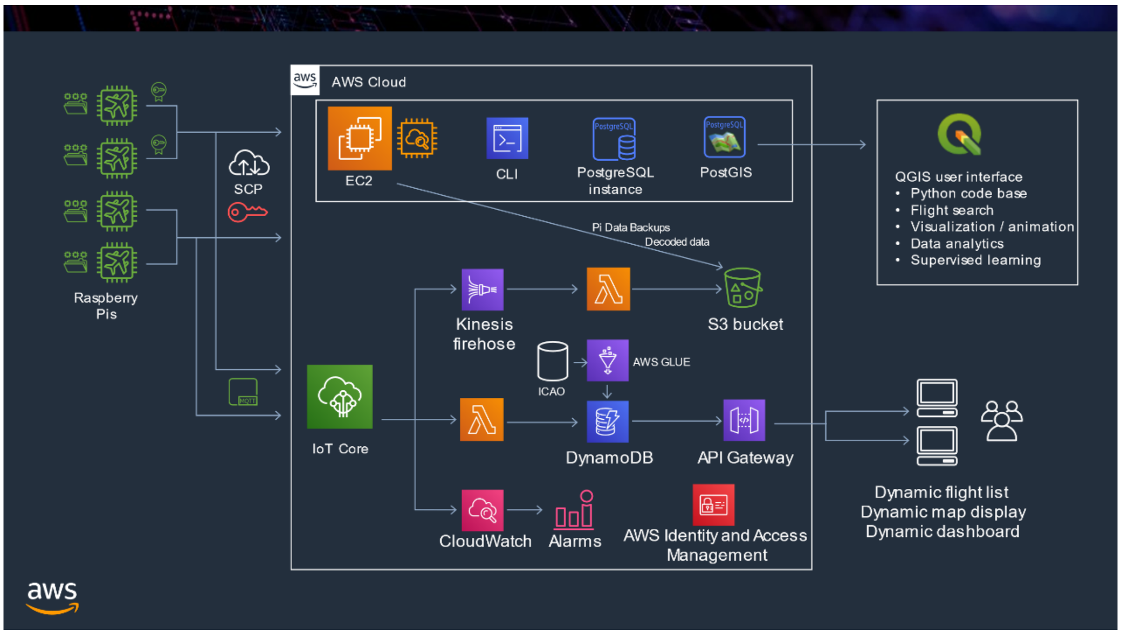


Figure 3: Storing the collected data (AWS architecture)

The processing system of the receivers (Raspberry Pis) is able to run autonomously. Thus once installed no further intervention is required. This system includes the following important features:

a) The configuration script that contains necessary information related to the location of the receiver (such as airport lat/long coordinates and altitude) as well as the receiver's frequency.
b) Data are decoded and stored on the receiver and then transmitted to an Amazon Web Services (AWS) portal in real time over the MQTT protocol, to support real-time mapping and flight display applications (Figure 4).
c) The receiver holds complete logs of system events, message transactions, etc.
d) At the beginning of each hour, any recent data and log files are uploaded via SCP to an AWS EC2 computing instance, and subsequently archived on the local computer, where they are retained for a month. In the event of a communications malfunction, this operation is re-attempted at every hourly upload event until it is successful. Files older than a month are deleted from local storage, because by this time they have been uploaded to AWS and stored in several places.
e) The system monitors real-time communications coming from AWS through the MQTT protocol, which allows the users to ping the receivers, remotely reboot them, tunnel into them to provide terminal window access, and change the gain levels on the Radio Frequency amplifier (Section 3.4).

f) Available upgrades on the AWS database are downloaded and installed automatically and the system reboots. This check is performed every hour after the collected files have been uploaded to the EC2.

The AWS Cloud software (the main part in Figure 3) consists of the following components:

a) *The EC2 computing instance*, which initially receives the data collected and the log files that are uploaded each hour. Then the data are decoded, filtered, and loaded into the PostGreSQL database, as shown in Figure 5. The filtering referred here removes data transmitted by aircraft that are too far from the studied airport and therefore not related to this study. Further processing and organizing of the data is described in Section 3.2.
b) *Lambda functions*, of which one processes the MQTT data submissions and populates a Dynamo DB with the last 5 minutes of real time data for the mapping and flight list web pages and the other processes PULL requests from the mapping, flight list, and dashboard web pages, and invokes the AWS API to send responses to the associated HTTP requests.
c) *The S3 bucket* is used as the final archive for all decoded message strings. It is also the project web server, hosting the mapping, flight list, and dashboard web pages, as well as a project information web page meant for research dissemination.
d) *Cloudwatch* watches various system functions and issues alerts. This helps optimize the storage and processing levels and warns the users of possible remote unit failures.

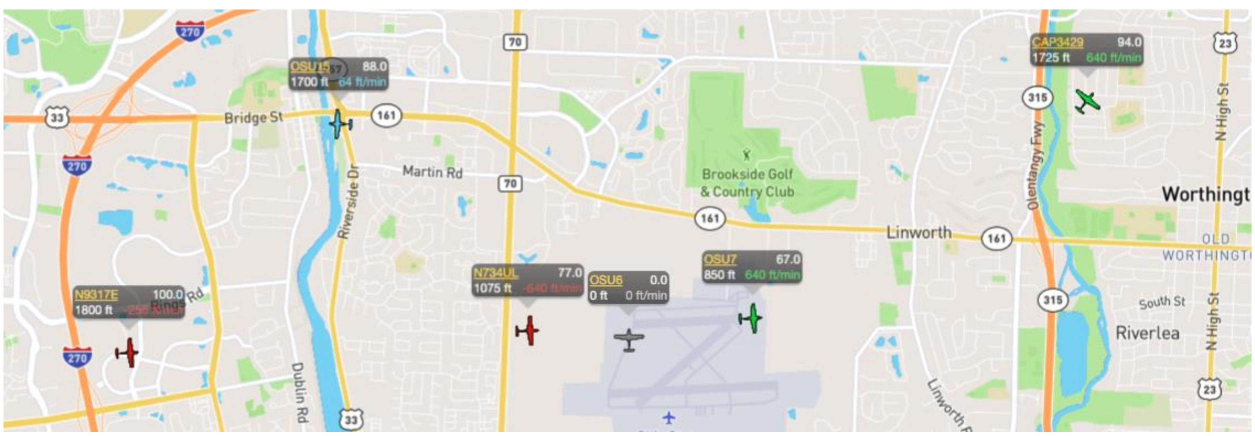


Figure 4: Instance of the real-time map at KOSU

The real-time map, apart from being an illustration of the data collection and processing, is also a useful tool for instantly identifying discrepancies in the data, or malfunction of the receivers. A map that fails to update the position of the aircraft indicates that the receivers have stopped collecting or feeding data to the AWS system. If something unreasonable appears on the map, it is first checked through the recorded data before proceeding to any alterations. In some cases, the data might have been decoded and stored properly, but failed to be properly handled by the mapping process. Additionally, for the case of the Ohio State University (KOSU), the map provides access to the ATC tower communication. Therefore, listening to the aircraft communication with the tower, and at the same time seeing the respective aircraft on the map, provides extra validation for the performance of the method. Moreover, the real-time map is provided to the airport managers of the participating locations, as an additional monitoring tool. [1]

The result of this first process is stored in a table of data ("adsb_messages" table) containing the following columns:

- Message id in an increasing numbering,
- Timestamp in seconds since January 1st, 1970,
- ICAO address,
- Latitude (lat),
- Longitude (long),
- Altitude (alt) in ft,
- Groundspeed (gs) in knots,
- Track (trk),
- Rate of climb (roc) in ft per second, and
- Callsign.

msg_id [PK] integer	datetime numeric (12,2)	icao character (10)	lat real	long real	alt real	gs real	trk real	roc real	callsign character (8)
57672233	1612328488.31	A62778	40.7166	-73.4823	7925	301	353.53	2432	
57672234	1612328489.24	A62778	40.7179	-73.4825	7975	301	352.57	2496	
57672235	1612328490.17	A62778	40.7192	-73.4827	8025	302	351.63	2496	
57672236	1612328491.20	A62778	40.7206	-73.483	8050	302	350.88	2560	
57672237	1612328492.24	A62778	40.7221	-73.4834	8100	303	349.36	2496	
57672238	1612328492.67	A62778	40.7227	-73.4836	8125	302	348.76	2496	GTI2678_
57672239	1612328493.16	A62778	40.7233	-73.4837	8150	302	348.76	2496	GTI2678_
57672240	1612328494.26	A62778	40.7248	-73.4841	8200	303	348.39	2496	
57672241	1612328494.69	A62778	40.7254	-73.4843	8200	303	348.39	2496	
57672242	1612328495.13	A62778	40.726	-73.4844	8225	303	348.39	2496	

Figure 5: Example of the data after been decoded

Subsection 3.1.2: Initial Processing

An aircraft transmits a message twice every second and each message received is stored individually. When received, a message does not provide immediate information on its own. The messages must be organized into groups, which from now on will be called *flights*, which are created by clustering the messages based on their ICAO address and their timestamps. If an aircraft fails to transmit a message in 600 secs (a number selected after tests on collected and decoded data) or more, then a new flight is created in the database. Each flight has a unique flight_id. Messages received continuously and from the same aircraft, receive the same flight_id. This new column is added to the table of messages, along with the location and the frequency (freq) column, which are provided by the configuration files created for each airport. The final structure of the messages table called "adsb_messages" is shown in Figure 6.

[1] The AWS storing and processing system as well at the real-time map and flight list were implemented by Dr. Lovell. Each step was validated and monitored by the student's analysis.

msg_id [PK] integer	datetime numeric (12,2)	icao character (10)	lat real	long real	alt real	gs real	trk real	roc real	callsign character (8)	flight_id integer	location text	freq character varying
57672233	1612328488.31	A62778	40.7166	-73.4823	7925	301	353.53	2432		351431	KFRG	1090
57672234	1612328489.24	A62778	40.7179	-73.4825	7975	301	352.57	2496		351431	KFRG	1090
57672235	1612328490.17	A62778	40.7192	-73.4827	8025	302	351.63	2496		351431	KFRG	1090
57672236	1612328491.20	A62778	40.7206	-73.483	8050	302	350.88	2560		351431	KFRG	1090
57672237	1612328492.24	A62778	40.7221	-73.4834	8100	303	349.36	2496		351431	KFRG	1090
57672238	1612328492.67	A62778	40.7227	-73.4836	8125	302	348.76	2496	GTI2678_	351431	KFRG	1090
57672239	1612328493.16	A62778	40.7233	-73.4837	8150	302	348.76	2496	GTI2678_	351431	KFRG	1090
57672240	1612328494.26	A62778	40.7248	-73.4841	8200	303	348.39	2496		351431	KFRG	1090
57672241	1612328494.69	A62778	40.7254	-73.4843	8200	303	348.39	2496		351431	KFRG	1090
57672242	1612328495.13	A62778	40.726	-73.4844	8225	303	348.39	2496		351431	KFRG	1090

Figure 6: Final structure of adsb_messages table

Apart from the "adsb_messages" table, a "flights" table was also created on the Postgres database. Both tables get populated once every hour, with the new data that have been collected by the receivers over the previous hour. The "flights" table holds information about every flight detected by the receivers, and each row corresponds to a different flight with a unique flight_id. Each row includes the following fields: the flight_id, the icao address of the aircraft, the callsign of the flight, the first and last timestamp of the series of messages collected for the flight, and the location at which it was detected. The structure of the "flights" table can be seen at Figure 7.

flight_id [PK] integer	icao character (10)	callsign character (8)	datetime_first numeric (12,2)	datetime_last numeric (12,2)	location_first text
359124	A3CD99	EJA344__	1613141967.30	1613142044.03	KOSU
358964	A8CED2	ENY3649_	1613132946.04	1613133036.26	KOSU
358970	A0CBD2	OSU15___	1613134617.20	1613135469.88	KOSU
358971	A2CF9E	OSU28___	1613134690.67	1613135597.46	KOSU
359188	A96577	DAL325__	1613142370.70	1613142464.30	KOSU
359043	A56FF8	N45CC___	1613137801.00	1613139309.48	KOSU
359201	A13ADF	OSU9____	1613142464.67	1613144971.94	KOSU
359242	A9E55E	N737DG__	1613145634.94	1613146345.16	KOSU
359251	A87F51	AAL2635_	1613146870.43	1613146941.59	KOSU
359250	A59D52	N461A___	1613146741.10	1613147081.43	KOSU
359246	A5F6EF	N4836G__	1613146842.30	1613147262.33	KOSU

Figure 7: Example of data in "flights" table

This table was created to provide a quick view of the data collected and overall information for each flight. It will later be populated with more useful metrics for each flight [20][22].

A preliminary version of this analysis was demonstrated in Mitkas & Lovell, 2020 [25] and presented at the ICRAT 2020 virtual conference.

Subsection 3.1.3: Collected Data Volume

Given that an aircraft transmits a message twice every second and that the aircraft operating at GA airports will often fly close or around the airport and consequently remain within the radius of the receivers, the amount of data collected is significant. An overview of the data values collected until April 24, 2021 is provided in Table 2.

Airport	No. of ADS-B messages	No. of Detected Flights	Date of receiver installation
KCGS	4,992,640	80,353	July 2020
KOSU	36,839,529	105,868	June 2020
KFRG	21,175,178	53,145	October 2020

Table 2: Collected Data Volume

Section 3.2: Data Preparation

To ensure the accuracy of the results, the data must go through some preparation steps. Feeding data with major discrepancies to a model would produce unreliable outputs. The initial steps take place at the message level, while the messages are received, decoded, and stored. The first stage includes checking if the message received is more than 5nm (nautical miles) away from the airport. Each message's latitude and longitude are used to calculate the distance of the aircraft from the airport; if the distance is more than 5nm from the center of the airfield, the message is considered not significant for the analysis of the particular airport and, therefore, is discarded. In addition, it must be mentioned that if a message is received with missing positional data, it has already been rejected by the system. The 5nm radius was selected because, for a GA airport, anything at a greater distance will most probably not affect its activity. Since subsequent analysis will focus on the operations of a specific airport and not the overall activity in the National Airspace (NAS), there is no need to burden the database with unnecessary data.

The second step relates to Altitude, one of the message parts that has been observed to be the most unstable. At the same time, altitude is one of the most important metrics when dealing with aircraft and classifying activity types at an airport. Before explaining the different cases, it is important to understand why some of these anomalies occur in the altitude data. Aircraft measure their pressure altitude while in flight and report it through their ADS-B transponder. However, deviations of environmental temperature and pressure from standard conditions cause the altitude estimate to be slightly erroneous. Also, most transponders have a "standby" operation, to which the aircraft can switch either when they do not want to transmit altitude anymore or when instructed to do so to reduce clutter in a high traffic area. When in "standby" mode, the aircraft will continue to transmit positional information and will not be lost from the receiver. It is common for aircraft to switch the transponder to "standby" mode after landing, in which case the transponder is still able to report an altitude equal to zero if the speed is low enough for it to be impossible for an aircraft to be flying. This is a clear example in which the aircraft do not transmit the actual pressure (or barometric) altitude, at any airport that is not at sea level.

In this phase, the altitude of each message is compared to zero; if it is negative, then the "negative_alt" field of the corresponding flight is set to true, meaning that at least one of the messages of the flight has negative altitude. Once set to true, this field will not change again. The case of negative altitude can be due to corrupted messages, or wrongful installation of the aircraft's altimeter, or because in some cases the aircraft might transmit their barometric altitude (baroaltitude) instead of their actual Above Ground Level (AGL) altitude. This phenomenon is mostly observed at airports with low airfield elevation altitude (lower than 100 ft). At this point, the negative altitude is not modified or neglected, it remains as is in the database.

The next stage could be characterized as a premature clustering of the flights and populates the "on_ground" field based on the messages of each flight. Every incoming message's altitude and groundspeed is checked. If the altitude is equal to zero (alt=0), meaning that the aircraft was at some point on ground level then the "on_ground" field is set to true. If not, then the groundspeed is checked; if the groundspeed is less than 20 knots, then the aircraft is considered to be on ground level. An aircraft cannot be flying if its groundspeed is lower than 20 knots unless it is a helicopter. In the case of helicopters, the latter check is not performed. To determine if an aircraft is a helicopter, its ICAO address is used to extract information about the aircraft type, model, engines, etc. from the FAA registry. If none of these checks are positive, the "on_ground" field remains false. This labeling gives a rough estimation of operation counts in each airports, since all the "on_ground"=true flights are possible airport operations. This estimation is not definitive for operation counts but could be indicative of the level of airport activity.

The flowchart for the steps performed during the pre-processing of a message is depicted in Fig. 8.

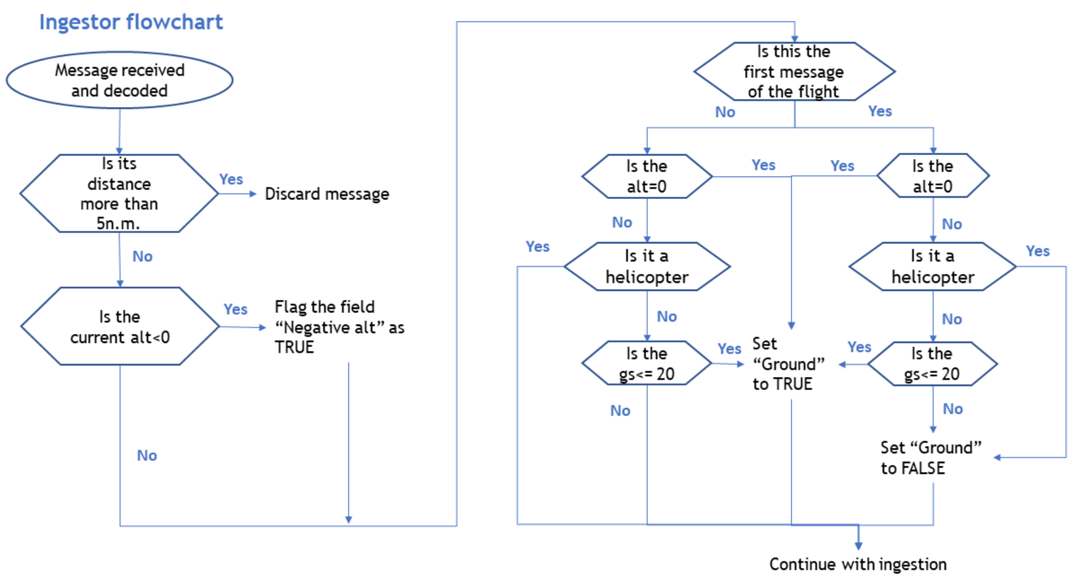


Figure 8: Pre-processing at message level

The following steps are performed at the flight level, meaning that the messages must be already grouped into a flight and the flight must be completed before starting them. Altitude is again the main concern of the following checks and procedures. As a first step, the minimum positive altitude (min_alt) of the flight is observed. Aircraft tend to transmit their sea level altitude while on flight and switch to AGL altitude when on the runway or taxiway. This might cause problems, especially at the airports with high field elevation. For example, in an airport with field elevation of 800ft an landing aircraft will transmit a decreasing altitude until 800ft, until it touches the runway. Once the aircraft is on ground and has decreased its groundspeed, the transponder will be switched to AGL altitude, communicating a message of zero altitude. This will create "jumps" in the altitude profile of the flight, from 800 to 0ft, which initially would seem wrong, but the data are accurate. This applies to takeoffs as well. Collecting the minimum positive altitude of each flight provides information about the field elevation of each location (as it was understood in that time and place according to atmospheric parameters) and rationalizes this anomaly in the altitude data. However, this is not a correction that can be applied permanently to the data, because it differs from airport to airport based on its elevation, and even for the same airport, it will differ from day to day based on the weather conditions (temperature and pressure).

To address this "jump" an AGL normalization process is performed. Specifically, if the minimum positive altitude collected is far from zero (more than 10 ft), and the flight has the "On_ground" field set to TRUE, then the "min_alt" is subtracted from the entire altitude profile and the result indicates the actual Above Ground Level altitude. This process also improves the cases were the transponder does not remain steady at alt=0 after the aircraft has landed and switches between 0 and some higher altitude (e.g. 800 ft). Before applying the AGL normalization, the messages are checked to ensure that the aircraft's speed is low enough that it would not be possible for it to be flying.

Next, the maximum time gap (max_dt) between consecutive messages is identified. Large time gaps (Δt) between messages can affect the data consistency and cause misinterpretation. A time gap is created when the aircraft leaves from the line-of-sight of the transponder or travels far enough at such a distance that it is not within the radius of the receiver or the transponder fails to transmit messages. This Δt cannot be greater than 600secs, which has been chosen as the threshold to separate flights of the same aircraft. It is important to know this gap, to be used to explain some of the anomalies that might occur.

The third metric collected is the maximum altitude (max_alt) of a flight. The maximum altitude can also provide a rough estimation of the accuracy of the data and the relativeness to the airport operations. If max_alt is found to be high (e.g. 30,000 ft) then the flight is most probably a commercial large aircraft flight, not related to the traffic of the GA airport. However, this cannot be a solid conclusion because this increased altitude can be just one or more corrupted altitude messages while the rest of the data remain accurate. The max_alt is also used in the next and final step of the data preparation process.

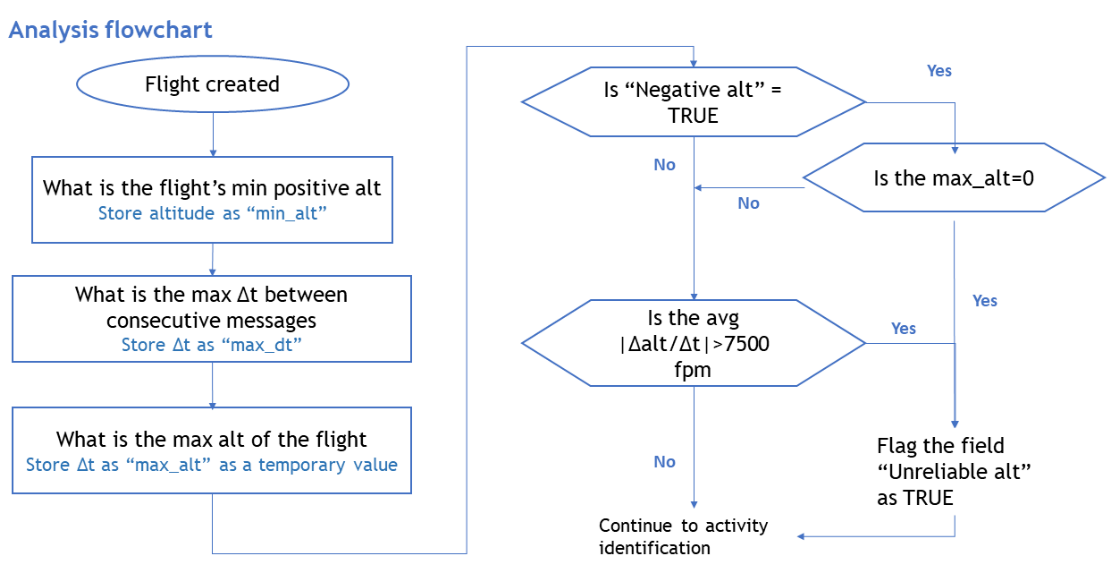


Figure 9: Pre-processing at flight level

The flow chart in Figure 9 illustrates the processing steps described above. The blue text in the boxes indicates the values that are collected for each flight either for correction purposes (AGL normalization) or for system monitoring purposes (major gaps in messages, max_dt). The right part of the graph illustrates the steps followed for identifying if a flight has unreliable altitude, as described below.

As stated previously, altitude can be really sensitive to data corruption and therefore, it is important to know from the start whether a flight has incorrect altitude data. For this reason, the average $|\Delta alt/\Delta t|$ of each flight is computed and compared to a threshold, which is currently set at 7500fpm (ft/min). This number was selected based on the type of aircraft and their climb power, that operate mainly at GA airports. A small aircraft cannot reasonably change its altitude by more than 7500ft in one minute. For the calculation of the average $|\Delta alt/\Delta t|$, this finite difference of rate of climb is computed for each pair of consecutive messages. In case the altitude is missing, the row is ignored. If a result exceeds the threshold, the unreliable_alt flag is set to True, and it indicates that some messages have extreme changes in altitude, or an altitude value that would not be expected in that phase of the flight. There is one additional case where the flight's altitude is deemed unreliable, which is when the altitude is constantly negative and the aircraft communicates the actual altitude only while on ground (alt=0). It is obvious that a constant negative altitude cannot be rational and therefore those flights also have unreliable_alt set to true. Some examples of these cases are shown in the flight profiles in Figure 10, in which the altitude data (top graph) seem completely incomprehensible, while groundspeed and rate of climb have reasonable values for fixed wing single-engine aircraft taking off.

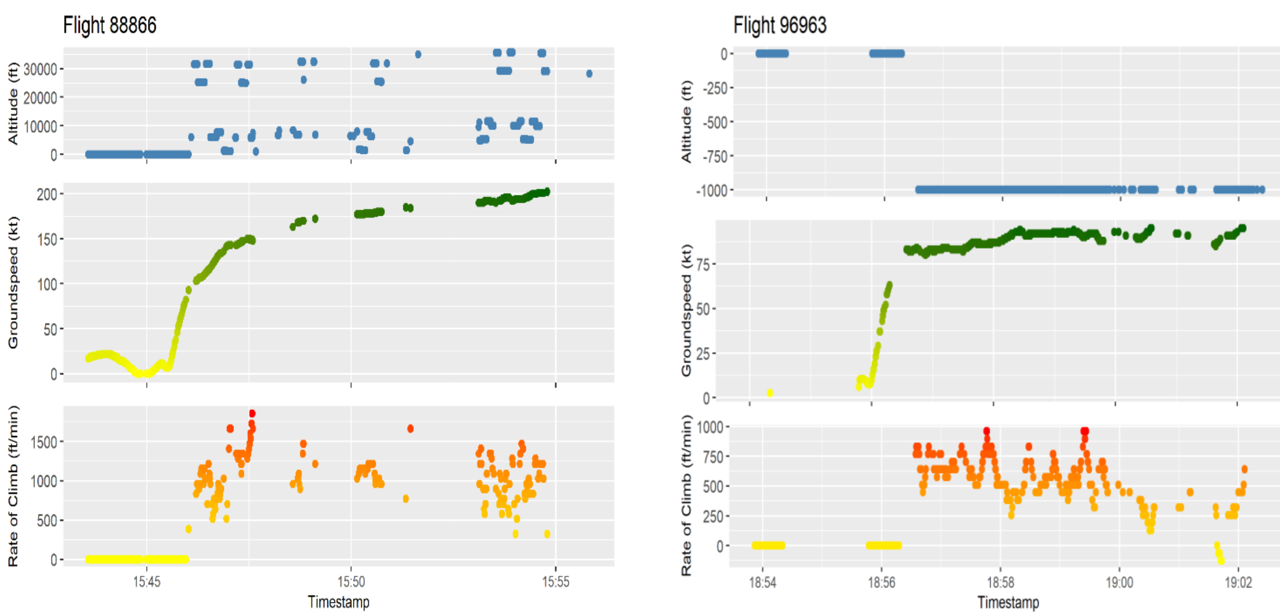


Figure 10: Example of flights with "unreliable_alt"=TRUE

The altitude data appear to be the most unstable, but also comprise one of the most important metrics; the rest of the data contained in each message tend to be generally stable and trustworthy. Thus, when the altitude is unreliable, it can be estimated by integrating the rate of climb, which provides the increase or decrease of the altitude in ft per minute. This process has been validated using flights with good altitude and when the data are adequate the results between the actual altitude and the estimated are almost identical.

The denser the messages, the better the results. Some examples of flight data that were used to validate this method can be seen in Figure 11, where blue color indicates the altitude transmitted from the receiver, and the light green color indicates the altitude estimated from the rate of climb. Deviations might occur in the parts where the messages are not as dense, such as the beginning or the end of the flight, when the signal is not as strong or stable.

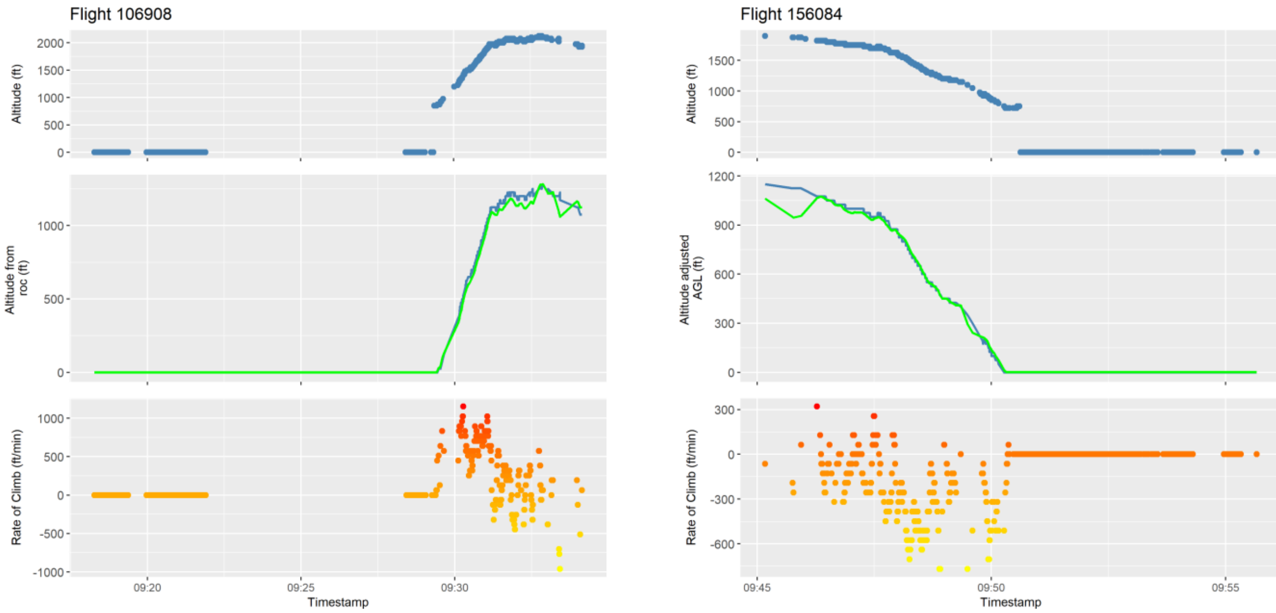


Figure 11: Estimating altitude from rate of climb (method validation)

Therefore, by estimating the altitude from rate of climb, the examples from Figure 10 will be transformed as follows.

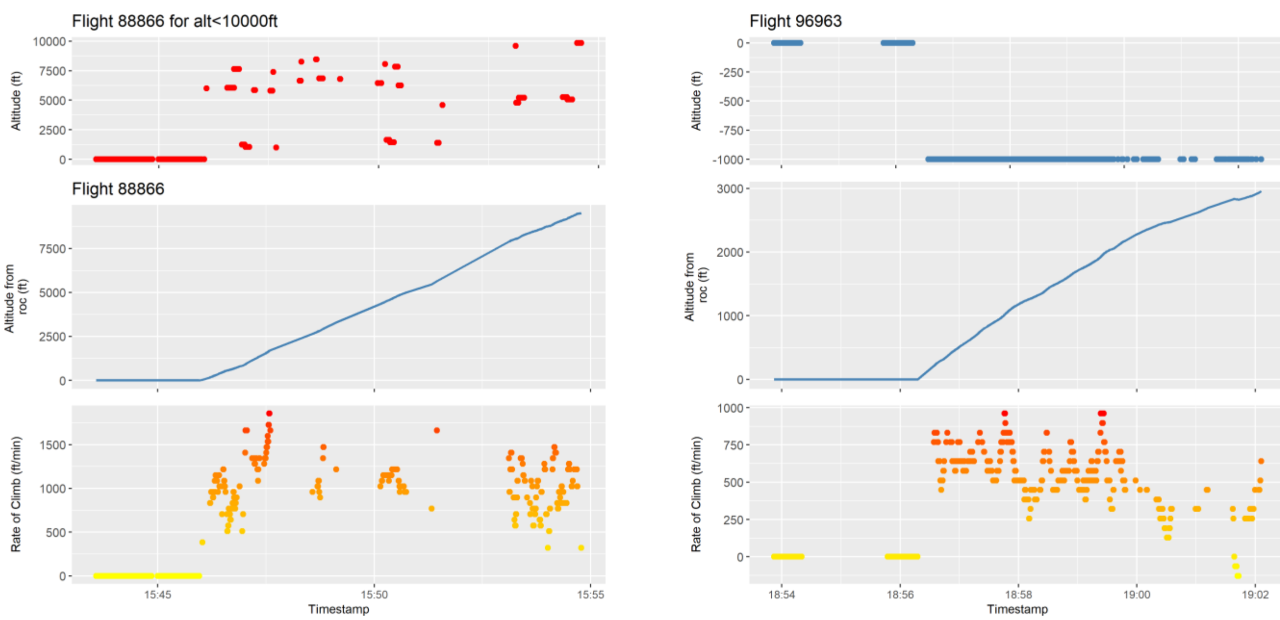


Figure 12: Examples of estimating altitude from rate of climb (unreliable_alt=TRUE cases)

At this point the preparation process of the data is concluded and useful information has been collected for each flight at this preliminary stage. The final structure of the "flights" table, including the additional columns, can be seen in Figure 13. The last column (ac_type) indicates the type of the aircraft, which as mentioned before is extracted from the FAA registry, using the ICAO address of each aircraft.

flight_id [PK] integer	icao character (10)	callsign character (8)	datetime_first numeric (12,2)	datetime_last numeric (12,2)	location_first text	flight_path json	processed boolean	neg_alt boolean	on_ground boolean	op_type integer	unreliable_alt boolean	ac_type text
359124	A3CD99	EJA344__	1613141967.30	1613142044.03	KOSU	[{"datetime": "16...	true	false	false	[null]	false	5
358964	A8CED2	ENY3649_	1613132946.04	1613133036.26	KOSU	[{"datetime": "16...	true	false	false	[null]	false	5
358970	A0CBD2	OSU15___	1613134617.20	1613135469.88	KOSU	[{"datetime": "16...	true	false	true	[null]	false	4
358971	A2CF9E	OSU28___	1613134690.67	1613135597.46	KOSU	[{"datetime": "16...	true	false	true	[null]	false	4
359188	A96577	DAL325__	1613142370.70	1613142464.30	KOSU	[{"datetime": "16...	true	false	false	[null]	false	5
359043	A56FF8	N45CC___	1613137801.00	1613139309.48	KOSU	[{"datetime": "16...	true	false	true	[null]	false	4
359201	A13ADF	OSU9____	1613142464.67	1613144971.94	KOSU	[{"datetime": "16...	true	false	true	[null]	false	U
359242	A9E55E	N737DG__	1613145634.94	1613146345.16	KOSU	[{"datetime": "16...	true	false	true	[null]	false	4
359251	A87F51	AAL2635_	1613146870.43	1613146941.59	KOSU	[{"datetime": "16...	true	false	false	[null]	false	5
359250	A59D52	N461A___	1613146741.10	1613147081.43	KOSU	[{"datetime": "16...	true	false	true	[null]	true	4
359246	A5F6EF	N4836G__	1613146842.30	1613147262.33	KOSU	[{"datetime": "16...	true	false	true	[null]	true	4
359252	A0AD24		1613147124.44	1613147188.72	KOSU	[{"datetime": "16...	true	false	false	[null]	false	5
359245	A997BF		1613145997.43	1613145997.43	KOSU	[{"datetime": "16...	true	false	true	[null]	false	4
359244	A2CF9E	OSU28___	1613145969.58	1613146215.61	KOSU	[{"datetime": "16...	true	false	true	[null]	false	4
359247	A07C0D	N130JV__	1613146203.00	1613146646.36	KOSU	[{"datetime": "16...	true	false	false	[null]	false	6

Figure 13: Final structure of the "flights" table

The last column (ac_type) indicates the type of the aircraft and each number corresponds to a different aircraft type as shown in Table 2.

Ac_type	Description
1	Glider
2	Balloon
3	Blimp/Dirigible
4	Fixed wing single engine
5	Fixed wing multi engine
6	Rotorcraft
7	Weight-shift-control
8	Powered Parachute
9	Gyroplane
H	Hybrid Lift
O	Other
U	Unknown

Table 3: Aircraft type list of codes

Results of this process were presented in the Opensky Symposium 2020, poster session [26].

Section 3.3: Study locations

For this research, specific study locations were selected, at which the ADS-B receivers have been placed. These locations are small General Aviation airports with significant activity. The participating airports are:

1. College Park Airport (KCGS), College Park, Maryland
2. The Ohio State University Airport (KOSU), Columbus, Ohio
3. Republic Airport (KFRG), Farmingdale, New York

The aim was to select airports with different characteristics that would provide various inputs to this research. Eventually, our options were limited by the COVID-19 travel restrictions and our final choice was determined by our ability to drive to these airports to install the equipment. There are plans to add more airports for the duration of the project. The most imminent addition will be the airports at Grand Forks, ND (KGFK) and Daytona Beach, FL (KDAB). More detailed information for each current airport are given in the following section.

Subsection 3.3.1: Airport characteristics

College Park Airport (KCGS)

KCGS is a public airport located in the City of College Park, Maryland. It has a single runway, runway 15/33, which accommodates on average 62 operations per week, of which 70% are local GA aircraft, 26% are transient GA aircraft, and the remaining are air taxi or military operations. The airport has a total of 31 based aircraft, of which 26 are single engine, 4 are helicopters, and the last one is a glider. The airport's average field elevation is 45ft above sea level. The airport might not have as much activity as other GA airports; however, it was selected for its proximity to the University of Maryland campus, providing the opportunity for in-person observations of the operations [21].

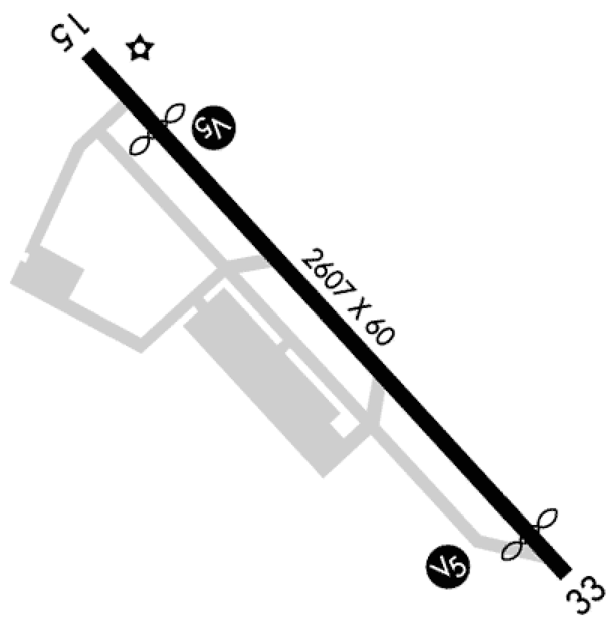


Figure 14: KCGS airport diagram

The Ohio State University airport (KOSU)

KOSU is a public airport located 6 miles northwest of Columbus, Ohio and is owned and operated by The Ohio State University. It is a three-runway airport with two parallel runways, one intersecting runway, and one helipad (H1). Runways 9R/27L and 9L/27R are used the most, while 5/23 usually serves as a taxiway, except in unusual crosswind situations. The airport has on average 246 operations per day, of which 45% are local GA, 34% are transient GA, 20% are air taxi and the rest are military and commercial flights.

KOSU has 148 based aircraft, which are divided into 121 single engine, 11 multi engine, 12 jet airplanes and 4 helicopters. The airport's average field elevation is 905ft. The important aspect of this airport, apart from its different configuration, is the based OSU flight school. Training flights contribute highly to the airport's daily operations and to the percentage of touch-and-go activity [21].

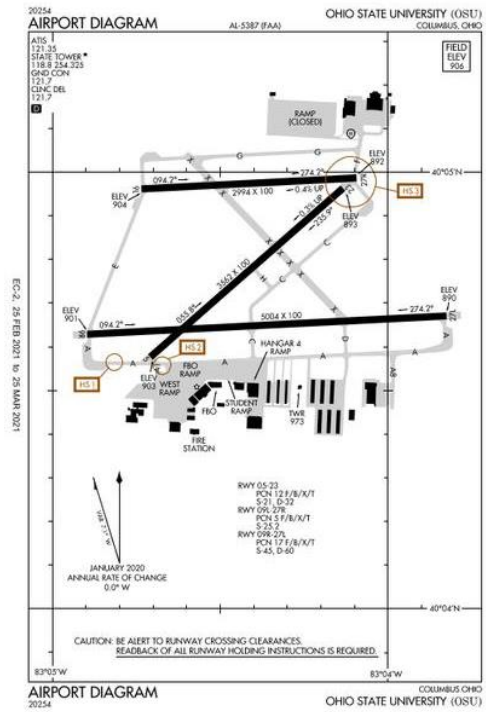


Figure 15: KOSU airport diagram

Republic airport (KFRG)

KFRG is a regional airport located in East Farmingdale, New York. It has two intersecting runways and two helipads (H1 and H2). Runways 14/32 and 1/19 accommodate a significant number of aircraft each day, with the first handling the most traffic. The airport has an average of 543 operations per day, with 49% being local GA, 45% transient GA and the remaining are air taxi, military, and commercial operations. KFRG has 350 based aircraft, of which 238 are single engine, 47 are multi engine, 54 are jet airplanes and 11 helicopters. The airport's average field elevation is 80ft. KFRG hosts multiple flight schools and jet aviation services creating a complex and demanding airport environment. Moreover, its proximity to one of the largest commercial airports (JFK) affects the behavior of aircraft using KFRG [21].

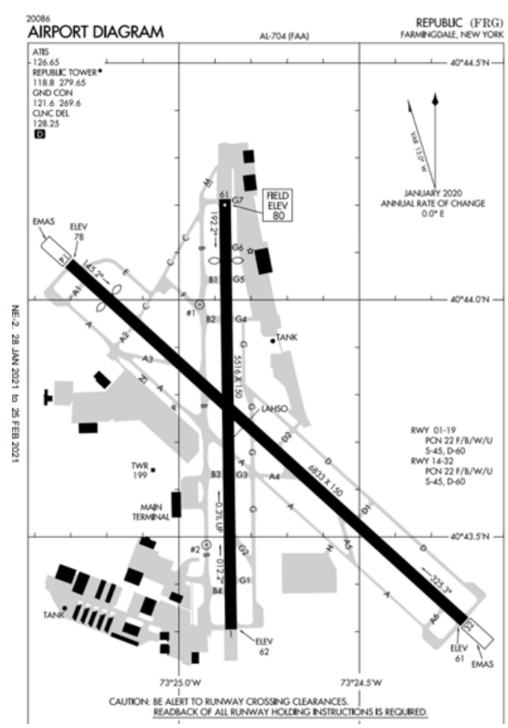


Figure 16: KFRG airport diagram

Section 3.4: Adjusting the RF gain

As mentioned before, it is important to select the gain settings of the RF amplifier accordingly. The gain is selected based in the location and the amount of aircraft activity in the area. Specifically, for the case of KFRG airport, adjustments needed to be made to the gain of the receivers. KFRG is located close to other airports and really close to one of the largest commercial airports in the US (JFK) and therefore experiences a lot of overflight activity from aircraft approaching to land. This causes a lot of message load for the receivers to be able to process correctly and eventually causes message corruption. This is especially true in the case of the 1090 receiver, which is the frequency that most large commercial aircraft use. Because of this situation, it was observed that the receiver would miss aircraft approaching to land at KFRG, or taxiing on the runways, but would still receive message from far way. Therefore, it was decided to lower the gain (originally set to 40 dB), of the KFRG receivers, to 10 dB. This change enhanced the reception of aircraft moving close to the airport; however, it also created some blind spots, mostly at the northwest side of the airport. Later, the gain was increased slightly and set to 20 dB. The result of the gain adjustments can be seen in the density maps of Figure 17. From left to right, the maps show gain set to 40 dB, 10 dB, and 20 dB, respectively. A useful feature of the software architecture is that the gain on any of the units can be set remotely by issuing commands from the AWS dashboard over the MQTT messaging protocol.

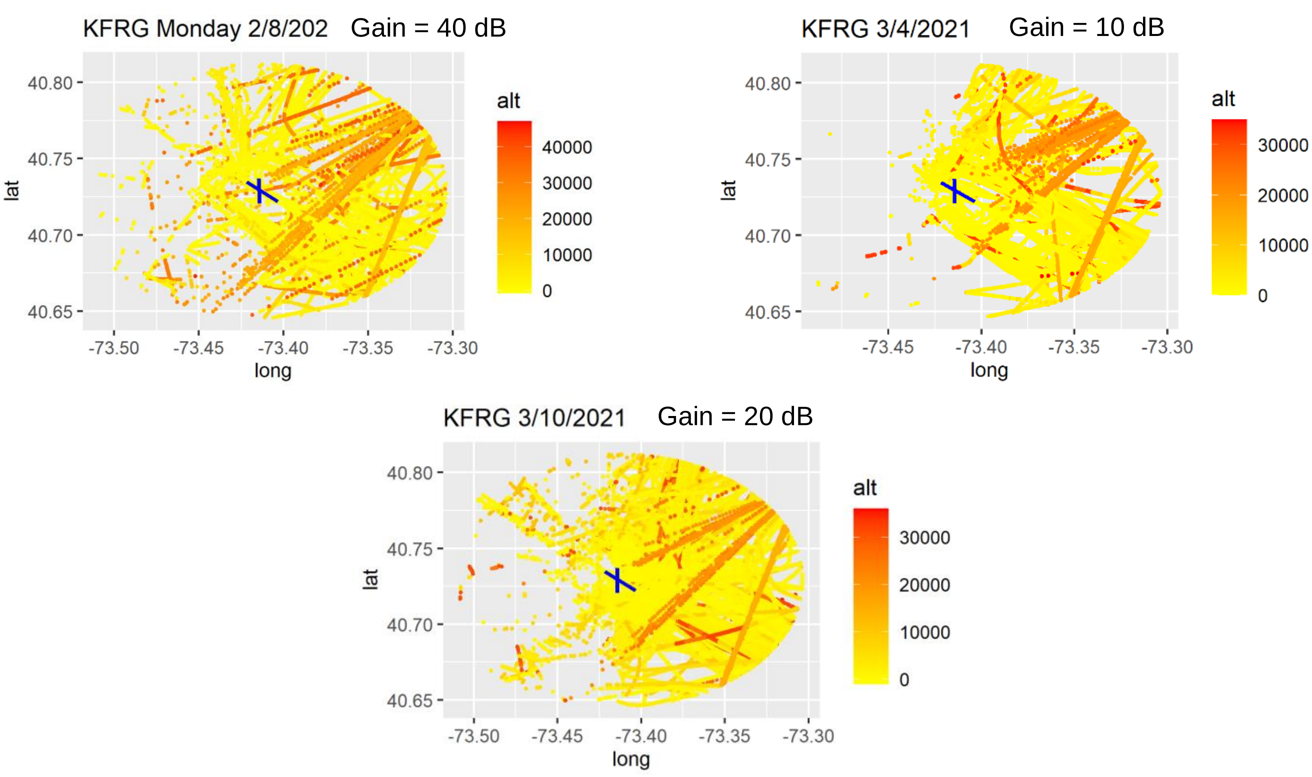


Figure 17: Adjustment of gain at KFRG airport

This adjustment not only brought favorable results in the reception of messages on or near the runway, but it also improved the quality of the data. The reduction of the gain reduced also the amount of messages received at the same time and therefore, lowered the probability of message corruption. This improvement was measured by checking the amount of "unreliable_alt" flights in our database, with every gain setting. The results are shown in Table 3, where the reduction in flights reporting "unreliable_alt" can be clearly observed.

	No. of flights	No. of flights unreliable_alt=TRUE	Percentage
Gain = 40 dB (2/3-7/2021)[2]	1466	128	8.9%
Gain = 10 dB (3/3-7/2021)	2440	259	10%
Gain = 20 dB (3/10-14/2021)	2655	66	2.5%

Table 4: Results of gain change in the quality of data

[2] Indicates the period of measurements for each gain change. The number of days is the same (5) in all three cases.

Chapter 4: Extracting Capacity Metrics

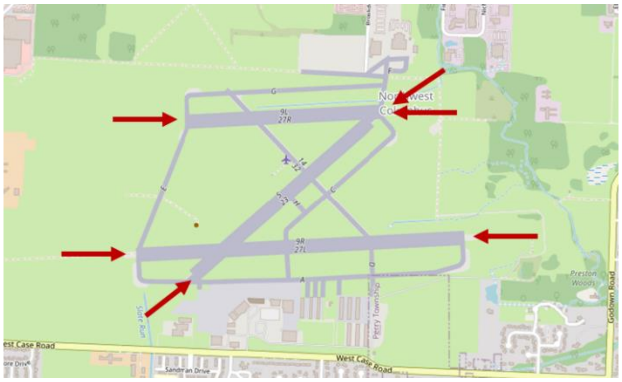
Having collected, organized, and processed such a large amount of data, from three different locations, the next step is to utilize the data for the initial purpose of this project, to extract metrics necessary for capacity estimation. The results in the following section are mostly for KOSU and KFRG airports, since KCGS has a much lower activity and did not present as much interest. Table 4 includes the aircraft approach speeds per aircraft category given by the FAA Advisory Circular AC 150/530-13.

Aircraft Category	Approach Speed (knots)	Example
A	<91	Cessna 172
B	91 to <121	King Air 200
C	121 to <141	B-737
D	141 to <166	B-767
E	166 or more	SR-71

Table 5: Aircraft approach speed categories (FAA)

Section 4.1: Approach Speed

The first metric computed and analyzed was approach speed. As explained earlier, approach speed is the speed that aircraft have before approaching the runway to land. To measure the approach speed, it is first necessary to identify all the possible runway options that an aircraft has to land at each airport. Usually all runways, from all ends, are used for landing, depending on the day and the winds. However, it is often that some runways are not used as much and mostly serve as taxiways. In the case of KOSU, runway 5/23 is used only in extreme crosswind situations, and in the case of KFRG, runway 1/19 is rarely used.



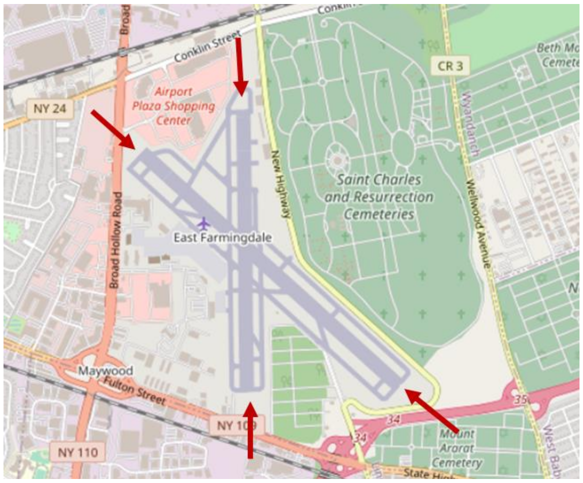


Figure 18: Possible options for landing at KOSU and KFRG airports

The steps for calculating approach speed for either runway at either airport are as follows. Boundary boxes were created at either end of each runway studied. Each box started at the end of the runway and spanned a couple of miles beyond it. The length of the box depends on the length of the runway. The longer runways can be used by heavier aircraft, which will start their landing procedure a lot earlier. The width of the boxes is slightly larger than the width of the runway, so as not to miss any points that might be slightly off. Once the latitude and longitude dimensions of each box are set, the last metric used is track. Track is involved in the procedure to ensure that all points captured within the box are from aircraft heading to land at the correct direction. For example, for aircraft approaching to land on runway 9L at KOSU, the track should be between 87 and 92 [22].

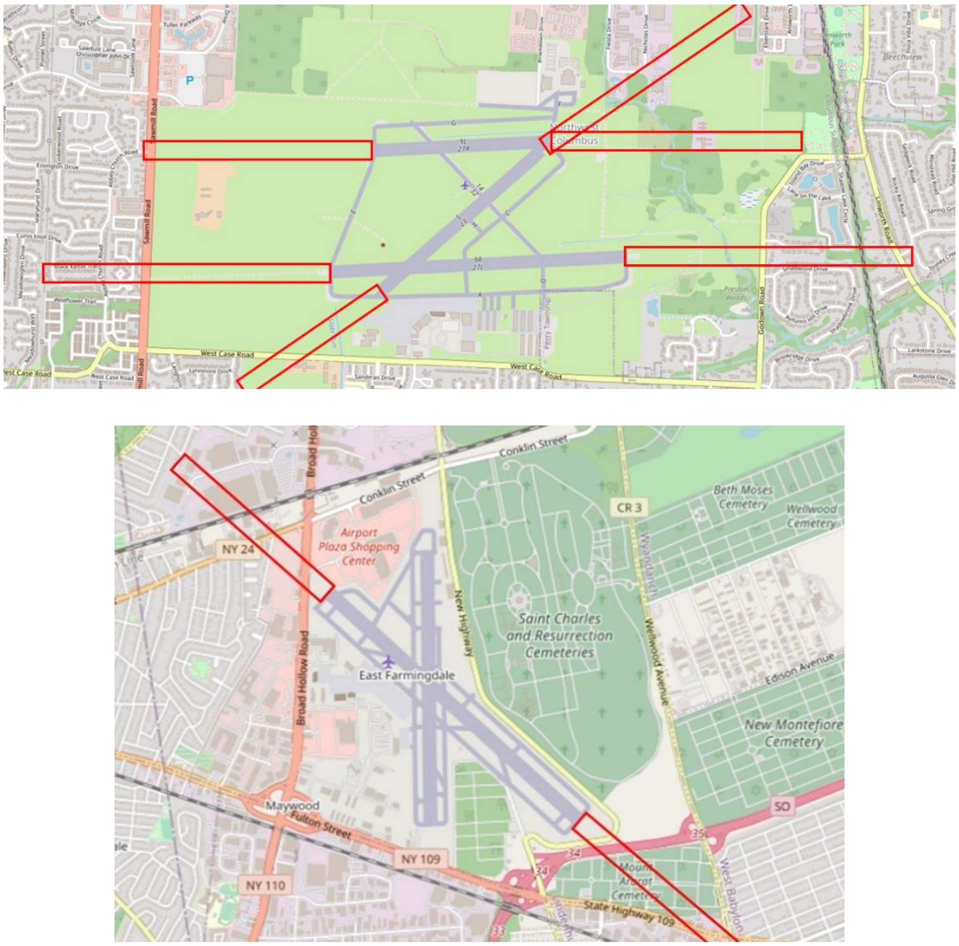


Figure 19: Boundary boxes used to collect approach speed

The approach speed was collected for samples of data and was categorized based on aircraft type. An example of the results of aircraft landing on runway 9R at KOSU airport is depicted in the graphs of Figure 20.

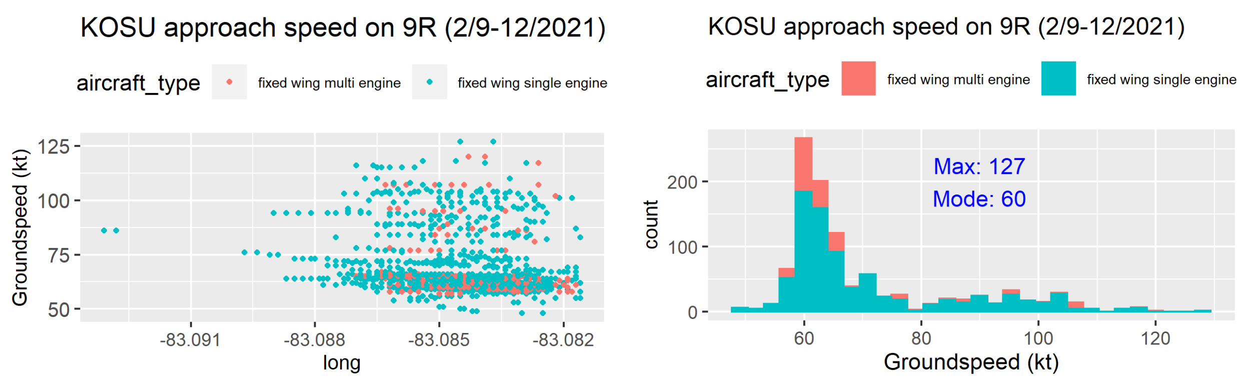


Figure 20: Approach speed results for runway 9R at KOSU

As seen, the approach speed ranges from 50 to 120 knots, with most points gathered between 55 and 70 knots and the predominant value being 60 knots. This runway is expected to experience a variable approach speed, since it is the longer runway of the airport and is utilized both by small, fixed wing single engine aircraft and larger fixed wing multi engine aircraft. Bigger aircraft arrive at a higher speed. The overall results for all runways and both airports can be seen in Table 5.

Airport	**Runway used for landing**	**Groundspeed range (knots)**	**Average Approach Speed per aircraft type (knots)[3]**	
			Fixed wing single engine	Fixed wing multi engine
KOSU	9L	60-75	66	62.5
	9R	60-110	71	68
	27R	60-80	70	69.5
	27L	60-110	71	73
	5	55-65	58	63
	23	55-65	59	60
KFRG	14	55-85	62	82
	32	50-85	68	83
	1	Not used		
	19	Not used		

Table 6: Approach speed results

At KOSU, runway 9L/27R is mostly used by small training aircraft and therefore the approach speed is expected to be lower compared to runway 9R/27L, which is used by both small and larger aircraft. This also explains the bigger range of values for runway 9R/27L. Lower approach speeds are found on runways accommodating solely the smaller single engine aircraft (KOSU 9L) and aircraft operating on crosswind runways when headwinds tend to be greater (KOSU 5/23).

At KFRG the runway mainly used is 14/32, and usually aircraft take off at 14 and land on 32, unless weather conditions require a change in the pattern. The range of approach speed is particularly steady between 50 and 85 knots. This runway is also used by both single and multi-engine aircraft.

[3] Results computed for the same number of days in each case (4 days).

Section 4.2: Runway Occupancy Time

Runway occupancy time indicates the time an aircraft spends on the runway, from the moment it passes the runway threshold until the moment it turns and exits the runway and poses an important limitation to the runway's capacity. Runway occupancy time (ROT) is not only related to the type and speed of the aircraft landing, but also to the geometric characteristics of the runway (i.e., runway length, number of exits, position of exits etc.). The steps for calculating ROT at either runway are as follows. The first point a landing aircraft is seen on the runway or right before the runway is recorded as the Touch Time (t_t). The point at which the same aircraft exits the runway is recorded as Exit Time (t_e). To identify the Touch Time, a small boundary box is used at either end of the runway. To identify the Exit Time, the position of the aircraft is compared to the position of the exits. If the track of the aircraft changes near an exit, then the aircraft is exiting the runway and that moment is distinguished as the Exit time [23][24].

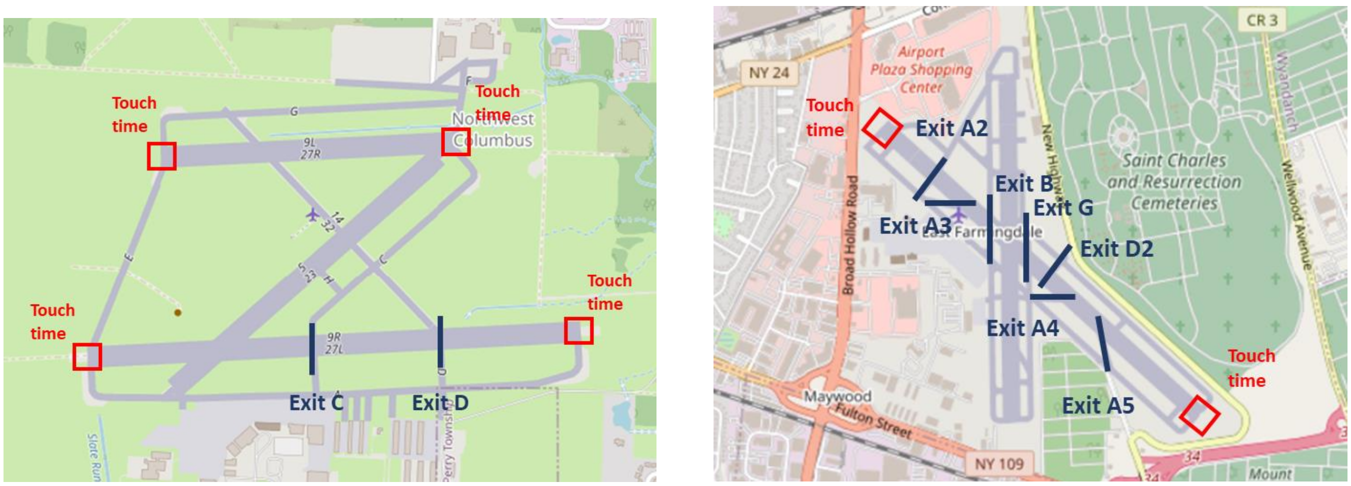


Figure 21: Runway thresholds and Exit positions

The exit used at each landing depends on how quickly the aircraft manages to decelerate after its landing. In order for an aircraft to turn and exit the runway it must have adequately lowered its speed. If the speed is still high, then the aircraft will have to taxi all the way to the end of the runway. In the case of runway 9L/27R, the runway is short and does not have exits that can be used, therefore, all aircraft landing on this runway must exit at the far end of the runway. The results for the Average Runway Occupancy Time (AROT) for each runway and for the different exits used, are shown in Table 6.

The blank cells of the table indicate that either the runway is not frequently used by that aircraft type (multi engine aircraft do not operate on short runways), or the exit cannot be used by that aircraft type. Specifically, multi engine aircraft will not use the exit closer to the runway threshold and single engine aircraft rarely taxi to the end of the runway if there is a sooner exit. Figure 21 indicates the different runways and different exit locations at the two airports.

Airport	Runway used for landing	Exit used for landing	AROT per aircraft type (secs)[4]	
			Fixed wing single engine	Fixed wing multi engine
KOSU	9L	End of runway	43	--
	9R	Exit C	34	--
	9R	Exit D	48	34
	9R	End of runway	--	64
	27R	End of runway	49	67
	27L	Exit D	28	--
	27L	Exit C	45	30
	27L	End of runway	--	98
	5	End of runway	82	--
	23	End of runway	80	--
KFRG	32	Exit A5	29	30
	32	Exit A4	33	34
	32	Exit B	42	40
	14	Exit B	29	--
	14	Exit G	32	36
	14	Exit A5	--	50

Table 7: Average Runway Occupancy Time

As expected, the further the exit is from the runway threshold, the more time the runway is occupied by the aircraft. However, the expected lower values for multi-engine aircraft, which, as seen previously, approach the runway with a higher speed and therefore should cover the distance in less time, were not confirmed. After examining the case of multi-engine aircraft closer, it was observed that even though the aircraft touch the runway at a higher speed, in many cases they slow down more abruptly and then taxi at a low speed until the exit. As long as an aircraft is still on the runway, no other aircraft can use the runway to either take off or land, until the runway is completely cleared. Therefore, as the runway occupancy time increases, the number of aircraft that can be accommodated by the runway, within a given period of time, decreases.

[4] Results computed for the same number of days in each case (2 days).

Chapter 5: Conclusions and Future Work

The aim of this research was to develop an efficient aircraft data collection process for General Aviation airports, which would yield the necessary information that, in turn, would enable the extraction of capacity metrics. The previous chapters presented the data collection method that was selected and the challenges that emerged by this decision, as well as the extraction of the average approach speed and the average runway occupancy time, two important capacity metrics, from the collected data.

Section 5.1: Conclusions

This section will sum up the conclusions both for the data collection method and the results on the extracted capacity metrics.

Subsection 5.1.1: The use of ADS-B data

Overall, the results of this research have shown that leveraging ADS-B data to understand aircraft performance and enhance the capacity estimation procedure for General Aviation airports, has been effective. ADS-B data can provide adequate information for the detailed tracking of aircraft both while on flight and when taxiing, by receiving direct information communicated by the aircraft and not estimated by the system. This method can become an important and useful tool equally for research purposes as well as for traffic management and monitoring around the airport environment, especially for non-towered airports.

As in most cases, when dealing with raw data some pre-processing steps were required before using them for any computation. However, this procedure did not become an impediment, but assisted the further refinement of the information received. In most cases, ADS-B messages were consistent and dense, creating a comprehensive profile for each flight captured by the receivers. For cases where anomalies are detected, such as the case of unreliable altitude, an effective procedure has been created to identify those discrepancies and normalize the data to approach reality. Also, the three airports that participated in this initial study provided multivariate data and different airfield environments that helped creating a stable and trusted data collection system, which will be able to adapt in any GA airport.

Subsection 5.1.2: Extracted metrics

Initial performance measures of aircraft approach speeds and runway occupancy times extracted from the ADS-B data are a good representation of aircraft behavior at GA airports and can form the basis for a capacity estimation model. Regarding approach speed the results were anticipated, based on the values provided by the FAA, and as expected, were higher for the longer runways that are used both by single and multi-engine aircraft. Longer runways provide additional length for an aircraft to slow down when landing with a higher speed and being heavier. Higher approach speeds will allow higher throughput rate; however major discrepancies between approach speeds on the same runway can lead to various challenges when trying to maintain the separation between aircraft.

Runway Occupancy Time is an important limitation for an airport's capacity, since a higher ROT invariably leads to fewer operations per runway. Apart from the optimal design of runways and exit position, aircraft characteristics will also affect ROT. Carefully collecting and calculating the AROTs for each runway at each airport is key to accurate capacity estimation. The further the exit from the runway threshold, the more time is needed for an aircraft to adequately lower its speed. However, an exit located really close to the runway threshold would not provide a solution to this, as it would be missed by most aircraft. Moreover, even though multi engine aircraft might land at a higher speed, and would be anticipated to cover the length of the runway in less time, this was not confirmed as aircraft are required to reduce their speed enough and are not able to exit earlier. After all, both average approach speeds and average runway occupancy times, along with other characteristics (e.g., minimum separation requirements, average aircraft fleet mix etc.) must be used to estimate the maximum sustainable airport capacity.

Section 5.2: Future Work

The use of ADS-B data to measure aircraft performance is still at an initial stage, especially for GA airports. There are more steps that can be taken to further develop and refine the data collection method proposed or to include more parameters in the performance analysis.

Subsection 5.2.1: Including more airports in the study

One of the next steps of this research will be to include more airport environments, that will provide different and equally significant data. The main reasons behind this step are a) to evaluate the validity and performance of the already designed procedures, b) to determine how well they can adapt to new conditions, and c) to identify different operational characteristics and constraints at airports with higher activity or unique geographical positions. The airports that we have already decided to include are the Grand Forks International Airport, ND (KGFK) and the Daytona Beach International Airport, FL (KDAB). KGFK has four runways (17R/35L, 09L/27R, 17L/35R, 09R/27L), two of which are intersecting, and 12 helipads. The airport's average field elevation is 845 ft. KDAB has three runways (07L/25R, 16/34, 07R/25L), of which two are parallel and one intersecting, and the average field elevation is 34 ft [21]. The airports participating at the moment are not confronted with capacity issues, however both KDAB and KGFK are known to have high aircraft activity, and to often operate at capacity limits. Therefore, it will be interesting to see how both the data collection method and the extraction process will adapt in these environments. The steps that have already been implemented, regarding the data preparation, will help to quickly identify any anomalies and to rectify the data, if needed.

Subsection 5.2.2: Refining and validating the extraction method

Repeating the extraction of metrics for larger datasets, over longer periods of time, will help to identify any anomalies that might not have been observed yet. This procedure will ensure the accuracy of the method and its adaptability to any ADS-B dataset. Having validated the method, it will then be possible to automate the procedures and incorporate them to the system and have the ROT and average Approach Speed of each flight stored at the "flights" table of the online database.

Subsection 5.2.3: Incorporating weather data in the analysis

One more important parameter that needs to be considered in the analysis is the impact of varying weather conditions, and how they affect each airport's operations and therefore capacity levels. Processing data for longer time intervals will cover the different seasons and will also provide input for a comparative analysis of aircraft performance from "good" weather to extreme winds or rainy days with wet surfaces. Especially for locations where inclement weather conditions are frequent and the option to shift operations from one runway to another, to avoid strong crosswind, is not available, this additional analysis will provide significant results.

Section 5.3: Extensions

The data collected using the method presented in this Thesis may give rise to several applications, in addition to the improvement of the operation and the facilities of a GA airport.

- Flight schools may use the data to generate flight patterns of their trainees and help them improve their piloting skills. One can imagine a student being able to observe their approach to the runway or their taxiing maneuvers.
- The FAA may use the data to identify aircraft with malfunctioning ADS-B transponders and/or wrong aircraft ids in their database. These aircraft can be informed about their faulty equipment or other deficiencies.
- Since several small airports lack automated data collection systems and are not staffed 24/7, they cannot record the full extent of their activity. Installing a system like the one developed in this project will greatly improve their flight monitoring (and reporting) capabilities.
- Widespread use of automated data collection systems in the majority of small airports in the country will provide a wealth of data to the FAA and enable the Agency to redesign the rules for General Aviation and to improve the overall capacity of the National Airspace.

References:

[1] Bubalo, B. and Dadun, J. R., 2011, "Airport capacity and demand calculations by simulation—the case of Berlin-Brandenburg International Airport", *Netnomics*, vol.12, pp.161-181.

[2] Gilbo E.P., 1993, "Airport Capacity: Representation, Estimation, Optimization", *IEEE Transactions on Control Systems Technology*, vol.1, issue 3, pp.144-154.

[3] National Academies of Sciences, Engineering, and Medicine, 2012, "ACRP Report 79, Evaluating Airfield Capacity", Washington, DC: The National Academies Press.

[4] Federal Aviation Administration, 1983, "Airport Capacity and Delay", Advisory Circular AC: 150/5060-5.

[5] National Academies of Sciences, Engineering, and Medicine, 2014, "ACRP Report 104, Defining and Measuring Aircraft Delay and Airport Capacity Thresholds", Washington, DC: The National Academies Press.

[6] Zelnio A.M., 2009, "Detection of Small Aircraft Using an Acoustic Array", M.S. Thesis, Wright State University.

[7] Lee J.S., Nielsen R.O., Davis H.H. and Dunholter P.H., 2011, "Acoustic Airport Surveillance System", Patent No. US 8,059,489 B1, United States Patent.

[8] Federal Aviation Administration, "Aviation System Performance Metrics (ASPM)", last update 5.10.19.

[9] Federal Aviation Administration, "System Wide Information System (SWIM)", last update 10.18.19.

[10] Federal Aviation Administration, "Traffic Flow Management System (TFMS)", last update 7.5.18.

[11] "Automatic Dependent Surveillance-Broadcast (ADS-B) Out equipment and use", Title 14 of the Code of Federal Regulations (CFR) § 91.225.

[12] "Automatic Dependent Surveillance-Broadcast (ADS-B) Out equipment performance requirements", Title 14 of the Code of Federal Regulations (CFR) § 91.227.

[13] RTCA, Inc., 2004, "Minimum aviation system performance standards for Automatic Dependent Surveillance – Broadcast (ADS-B)," Report DO242A.

[14] Piracci E.G., Galati G. and Pagnini M., 2014, "ADS-B signals reception: a Software Defined Radio approach", in 2014 IEEE Metrology for Aerospace (MetroAeroSpace), Benevento, Italy.

[15] Strohmeier M., Schäfer M., Lenders V., and Martinovic I. and Wilhelm M., 2014, "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, Germany.

[16] International Civil Aviation Organization, Asia and Pacific Office, 2018, "ADS-B Implementation and Operations Guidance", Edition 11.0.

[17] Strohmeier M., Schäfer M., Lenders V., and Martinovic I., 2014, "Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B", *IEEE Communications Magazine*, Vol: 52, Issue: 5.

[18] Sun J., Ellerbroek, J., Vu, H. & Hoekstra, J., 2019, "pyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research", *IEEE Transactions on Intelligent Transportation Systems*, DOI=10.1109/TITS.2019.2914770, ISSN=1524-9050.

[19] Sun J., 2020, "The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals", TU Delft OPEN Publishing, 2nd edition.

[20] Sun J., Ellerbroek J., & Hoekstra J., 2016, "Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods", In D. Lovell, & H. Fricke (Eds.), 7th International Conference on Research in Air Transportation, Philadelphia, USA.

[21] Aircraft Owners and Pilots Association, aopa.org, (last updated March 2021)

[22] Sun J., Ellerbroek J., & Hoekstra J., "Modeling aircraft performance parameters with open ADS-B data", In 12th Seminar Papers: 12th USA/Europe Air Traffic Management Research and Development Seminar, 2017.

[23] Meijers N.P. & Hansman R.J., 2019, "A Data-Driven Approach to Runway Occupancy Time", AIAA Aviation 2019 Forum, Dallas, Texas, USA.

[24] Kumar V., Sherry L. & Kicinger R., 2009, "Runway Occupancy Time Extraction and Analysis using Surface Track Data", George Mason University, USA

Conference Papers and Posters

[25] Mitkas D.Z. & Lovell D.J., 2020, "A Data Collection Scheme to Support Applications for Capacity Enhancements at Small Airports", ICRAT2020 (International Conference on Research in Air Transportation), Doctoral Symposium, USA

[26] Mitkas D.Z., Lovell D.J., 2020 & Young S., 2020, "Learning to recognize and stratify flight training activities at GA airports using ADS-B data", Opensky Symposium 2020, Poster Session, Europe