# Innovative and Low-cost Technique to Identify Airport Taxi Congestion Points

Post-operational data analysis at Malaga airport

Alan R. Groskreutz Senior Researcher CRIDA A.I.E. Madrid, Spain argroskreutz@crida.es Fernando Domingo Merino Air Navigation Engineer Universidad Politecnica de Madrid Madrid, Spain f.dmerino@alumnos.upm.es Iván Díaz Pérez Air Traffic Controller ATFCM software developer ENAIRE OPS Directorate Madrid, Spain idiaz@enaire.es

*Abstract*— One factor contributing to the uncertainty of a flight's takeoff time is related to the variation between the planned taxi out time vs the actual time due to congestion delay between the stand and the runway. By determining which taxiway intersections create the most instances of delay, procedures and alternate routes could later be developed to try to alleviate these delays and create a more predictable operation. However, medium and small airports generally do not have surface surveillance systems in place necessary to record this type of data.

This paper will propose and investigate a method to record and analyze aircraft movement on the airport surface through the use of inexpensive ADS-B antennas and open source programs that can be quickly installed at airports that want to perform postoperational analysis regarding their surface movements.

Results show that the main congestion points can be determined through the use of this system, that the amount of data necessary to perform the analysis can be reduced from what is initially recorded by 99,9%, and that the reduction does not remove necessary data quality.

### Keywords-data analysis; delay; airport; taxi; congestion

#### I. INTRODUCTION

One of the main performance concerns at airports is delay. A prime source of this delay can be inefficient taxi times caused by one aircraft having to slow down or stop their taxi to let another aircraft taxi by. Two of the metrics in the Airport Performance Framework are tied to the Taxi-out time; Departure Predictability and Ground Movement Efficiency[1]. To be able to find means to reduce this type of delay, airports first need to find out where the delay is happening and why by analyzing many months of recorded taxi movements. However, the airports that have the necessary surveillance systems in place to be able to obtain this data are mainly large airports with multilateration systems. Many midsize and smaller airports are not currently equipped to do this kind of analysis.

In SESAR 2020, PJ04.01 performed and validated research regarding the Post-Operational analysis that a medium sized airport could accomplish without having an Airport Operational Plan or the associated systems to record that information[2]. In

line with the results of that research, the use of ADS-B antennas coupled with open source data analysis tools provide an inexpensive alternative that can be quickly installed at airports that want to perform post-operational analysis regarding their surface movements.

This paper presents the implementation methodology, and analytical results from the use of ADS-B data at Málaga-Costa del Sol airport to determine where the areas of taxi congestion are located. Thus, from the study of this output, information can be obtained regarding the total movements that enter these areas, the areas of greater or lesser influx, and the hours of greater activity in each area. This information is important for carrying out congestion analyses during taxi.. Results from congestion analysis can lead to changes in procedures and taxi routes which can bring benefits not only in reducing delays, but also Fuel/CO2 emissions, saving costs, and improving punctuality and predictability

The paper is split into three parts; how the data was gathered and stored, how the data was reduced and filtered to a manageable level, and the final analytical results.

### II. METHODOLOGY

### A. Aircraft movement data gathering

The foundational element of the data capture system, shown in Figure 1, is a Rasberry Pi Model 3 with a 1.2 Ghz BCM2837 processor. Connected to this is a 256G memory card, A USB Digital Terrestrial Television receiver, an ADS-B antenna, a monitor and keyboard for local configuration, and a mini 4G card for transmitting the data and for remote monitoring of the system.













Figure 1. ADS-B data capture and storage diagram

The approximate cost of the installation of a system with one antenna was 200 EUR. If direct line of sight of aircraft is hindered, a multi-antenna setup could be configured as well.

The software installed on said raspberry is a Raspbian operating system, installed using the noobs tool prior to the installation of the station at the airport – The following packages from Github were used (other options might be used as well):

sudo apt-get update sudo apt-get install git git clone <u>https://github.com/jprochazka/adsb-receiver.git</u> cd ~ / adsb -receiver chmod + x install.sh ./install.sh

It is also feasible to directly install preconfigured images, such as those available in the following Github :https://github.com/jprochazka/adsb-

receiver/releases/tag/v2.6.3 During installation, both the antenna location and elevation must be configured. Once the previous packages have been installed and configured, it must be verified that the station is functioning correctly. To do this, in addition to being able to view traffic logs directly, traffic can be viewed directly in a browser running on the raspberry itself at the address http://localhost/dump1090.php which will look similar to Figure 2.



Figure 2. Local view of ADS-B data

In order to connect to this local server via the internet, the server was configured through NGrok [3]. The use of this tool, or a similar one, is necessary since the ownership of the external IP is of the own Mifi, not of the raspberry itself. Through NGrok, ports 22 (SSH and SSFS) as well as port 5900 (VNC standard for remote desktop connection) were left open. Once configured, the system offers connectivity to the outside, which permits the long term storage of the data in a database not located at the airport.

The system is based entirely on software-based-radio decoding of the signals emitted by aircraft equipped in the 1090MHz band. These radio signals, once decoded (in our case, through the SDR dump1090 mutability), are obtained through hexadecimal format strings, each corresponding to a single message issued by an ADS – B emitter.

The decoding is documented in multiple sites on the Internet, although two of the most complete are [4] and [5].

Although these messages offer great complexity for direct exploitation, the installation carried out in the previous steps offers a data flow on port 30003 in the known SBS format[6], which looks similar to those in Figure 3.

STA,,5,179,400AE7,1	0103,2008/11/28,14:58:51.153,2008/11/28,14:58:51.153,RM
M5G,4,5,211,4CA2D6,	10057,2008/11/28,14:53:49.986,2008/11/28,14:58:51.153,,,408.3,146.4,,,64,,,,
MSG,8,5,211,4CA2D6,	10057,2008/11/28,14:53:50.391,2008/11/28,14:58:51.153,,,,,,,,,,,,,,,0
MSG,4,5,211,4CA2D6,	10057,2008/11/28,14:53:50.391,2008/11/28,14:58:51.153,,,408.3,146.4,,,64,,,,
M5G,3,5,211,4CA2D6,	10057,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,37000,,,51.45735,-1.02826,,,0,0,0,0
MSG,8,5,812,ABBEE3,	10095,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,,,,,,,,,,,0
MSG,3,5,276,4010E9,	10088,2008/11/28,14:53:49.986,2008/11/28,14:58:51.153,,28000,,,53.02551,-2.91389,,,0,0,0,0
MSG,4,5,276,4010E9,	10088,2008/11/28,14:53:50.188,2008/11/28,14:58:51.153,,,459.4,20.2,,,64,,,,,
MSG,8,5,276,4010E9,	10088,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,,,,,,,,,,0
MSG,3,5,276,4010E9,	10088,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,28000,,,53.02677,-2.91310,,,0,0,0,0
MSG,4,5,769,4CA2CB,	10061,2008/11/28,14:53:50.188,2008/11/28,14:58:51.153,,,367.7,138.6,,,-2432,,,,,
MSG,8,5,769,4CA2CB,	10061,2008/11/28,14:53:50.391,2008/11/28,14:58:51.153,,,,,,,,,,,,0

Figure 3. SBS formatted ADS-B messages

#### B. Data reduction and formatting

Málaga-Costa del Sol airport has an average of around 250 aircraft going in to and out of it daily during its high season. This translates to ADS-B data files of around 1,5 GB per day. In order to analyze months, or even weeks of data, certain data reduction measures need to be taken to complete the analysis in a timely manner. An overview of the algorithm is shown in Figure 4 where the boxes represent the processes and the text between the boxes represent the logical structure that store the information (dataframes) during the data reduction process.



Figure 4. Data Reduction Algorithm







Using Python scripts in a Jupyter Notebook[7] to manipulate the messages as individual rows in a dataframe, the data can be easily handled to reduce the content to just what is needed for the study. There are six different types of messages. These messages are described in Table 1. All the information that interests this study is contained in the MSG messages.

TABLE 1 ADS-B MESSAGE DESCRIPTIONS

Message ID	Description
SEL	Generated when the user changes the selected aircraft in
	BaseStation
ID	Generated when an aircraft being tracked sets or changes
	its callsign
AIR	Generated when the SBS picks up a signal for an aircraft
	that it isn't currently tracking
STA	Generated when an aircraft's status changes according to
	the time-out values in the Data Settings menu
CLK	Generated when the user double-clicks (or presses return)
	on an aircraft (i.e. to bring up the aircraft details window)
MSG	Generated by the aircraft. There are eight different MSG
	types.

The first step in the data reduction process is to remove the unnecessary, or unused information. Since some of the datafiles can be on the order of tens of gigabytes in size (if more than one day's worth of flights are stored in a single file), the memory needed to process these files can be a limiting factor. Therefore, this data reduction step will be performed iteratively in dataframes of 1 million rows.

Each of the messages contain 22 data fields, but 16 fields are not required for ground movement studies and are removed. The remaining 6 fields are Hexident (the unique aircraft identifier code), Date message generated, Time message generated, Altitude, Latitude, and Longitude.

The next data reduction step is to remove all the rows that have an altitude above 300ft or do not have latitude information. This will remove any non-position data and data captured from overflights which are not of interest, but still leave the data regarding final approach and takeoff that are interesting for landing point, takeoff point, and runway occupation times. This reduced file is then saved separately and referenced in Figure 4 as SUP. This leaves a dataset that, depending on the amount of overflights, has been reduced nearly 90%, and can be stored for later use in other studies. And additional step combines the columns of Latitude and Longitude into an addition column titled "Point". This is done to make the next steps easier to perform. This output dataframe is called SUP-COMP.

The final data reduction step for this study is to reduce the data to only those datapoints that are related to the position of the aircraft in and near intersections on the taxiway. In order to do this, a map of these areas - or nodes - needs to be created. This map, or overlay, is like a piece of swiss cheese where the holes are the nodes of interest. This overlay is a JSON file made using the QGIS program. QGIS is an open source Geographic Information System (GIS) software package designed to capture, store, manipulate, analyze and display geographically differentiated information in all its forms[8]. With QGIS,

polygons are created that have geographical references as their vertices. Each polygon is a node which represent thresholds, full runways, runway entrances and exits, waiting points, intersections of taxiways and aprons. As the definition of the nodes is stored in an exterior JSON file[9], referred to in Figure 4 as POLI, modification of the node shapes will not affect the general operation of the program, so it is simple to study different applications depending on the objectives pursued.

With these nodes defined, a Boolean logic Python script checks each "Point" value from SUP-COMP to determine if it is within one of the nodes in POLI. If it is, the location message is tagged with the node and saved to the dataframe GEO-FLTR, The process then moves on to the next location message as the same message can't be in two nodes.

Once all the messages from SUP-COMP have been checked against POLI, the resulting dataframe GEO-FLTR is organized by node and saved as GEO-FLTR\*.

Many of the stored position records correspond to the same aircraft within each of the nodes but at times differentiated by just tenths of a second. Defining a minimal time interval and eliminating these redundant messages corresponds to the total number of movements that cross an intersection of paths, or node, during the time period in which the input data is taken. The redundancy filter then outputs the dataframe GEO-LMP.

#### C. Data analysis

Now that we have our reduced list of location points, we need to define what is meant as "congestion". Congestion addresses those situations in which an aircraft is forced to stop because of the proximity of another aircraft in the taxiway, causing a delay in the taxi-time of this first aircraft. Taking from orbital operations, we will refer to these individual moments as "conjunctions".

To locate these conjunctions within the GEO-LMP dataset, a maximum time difference is established between messages from different aircraft. Speaking with airport air traffic controllers, it was decided to use 30 seconds as the maximum time difference with the assumption that anything less than that would not correspond to freely moving traffic. Those aircraft that meet this temporary difference condition will be considered in conjunction, since they occupy the same point on the taxiway almost simultaneously. This time difference takes into account the start and stop times of the aircraft, the taxiways and other variables that affect those start and stop times. [10]

The conjunctions are found by sequentially selecting two location messages within a particular node, determining the time difference between them, and if it is less than the 30 second threshold, the two messages are stored in the final dataframe. As these conflicts are caused by the action of two aircraft, the total number of conjunctions is equal to half the number of records stored in this output.

#### III. APPLICATION AND RESULTS

As stated before, the airport under study was Malaga Costa del Sol airport. The airport layout can be seen in Figure 5. It is











Spain's fourth largest airport and had a total of 141,313 movements in the year 2018, an increase of 14% above 2017. It has two runways (13/31 and 12/30) and three terminals, two of which are currently operational.



Figure 5. Malaga Costa del Sol Airport Layout

The preferential configuration is runway 12 for arrivals and runway 13 for departures. In single runway operations runway 13 is used for both. Ground operations are split into North and South. The Northern runway(12/30) has a lower throughput as 12 is not usable for takeoff and 30 is not useable for landings due to the interference with the southern runway. Without a ground surveillance system like the one described here, in-depth analysis of its ground movements is not feasible. Four ADS-B antennas were placed on top of the control tower as shown in Figure 6 to obtain full coverage of the airport surface.

Using a free software tool named modesmixer2 the 4 SBS format data streams coming from the ADS-B receivers were merged into a single SBS data stream. The data studied was recorded between July 25<sup>th</sup> and July 31<sup>st</sup>, 2018. This represents a full week of traffic from the highest period of traffic at the airport.



Figure 6. GPS Antenna Location

The nodes were defined as shown in Figure 7 by the pink polygons. Six deferent areas of interest are defined as nodes; the intersection points of the runways, access to taxiways from the platform, entrances and exits of the runways, parking spaces and runways.



Figure 7. Definition of Nodes at Malaga Airport

To facilitate the calculation of total movements in the nodes, and to enable an analysis of global movements and conjunctions, these six node types are classified into three categories that serve their function during the rolling process. Figure 8 shows the three categories of the nodes; Road intersection nodes (Green), Runway entry and exit nodes (Blue) and Rapid runway exit nodes (Orange). As this study is not investigation runway movement or occupancy, the runway nodes are not included in these classifications.



Figure 8. Node Classification Types

After applying the data reduction process, a heatmap of the movements through the nodes can be created. Figure 9 shows the distribution of the traffic through the nodes.

Looking at the north side of the airport, it can be seen that the most movements flow through a single node, which is the one connecting the North side with the South side. This is understandable since under segregated operations, more traffic would flow through that node than the other northern ones as this is the connection between the two runways.











On the South side there are eight nodes that have high levels of traffic through them. These correspond to the intersections between the most used rapid runway exit nodes and the road intersection nodes. This follows since these nodes are used by both arrivals and departures.



Figure 9 Total surface movements through the nodes

After calculating which nodes have the most conjunctions, a congestion heatmap can be created. Figure 10 shows the nodes with the most congestion. The node that stands out more than others is the runway entry node corresponding with runway 31



Figure 10. Locations of taxi congestion

However, as stated earlier, the preferred runway for departures is 13, so this data does not seem to correspond to preferred operations. Looking at the results day by day as shown in Figures 11 through 17 it is clear that this congestion occurred on Sunday, July  $29^{\text{th}}$ .









Figure 13. July 27th, 2018 congestion











6



Figure 14. July 28th, 2018 congestion



Figure 15. July 29th, 2018 congestion



Figure 16. July 30th, 2018 congestion



Figure 17. July 31st, 2018 congestion

Looking at the operations on that day show that the airport was in single runway mode and was using runway 31 for departures. It makes more sense that there would be traffic waiting to depart as the arrivals landed on the same runway.

## IV. CONCLUSIONS

Using ADS-B antennas as a means to record and save aircraft taxi data is an inexpensive means to perform airport ground motion analysis. While the raw data initially recorded can be more than 1GB per day for even a medium airport, data reduction based upon altitude, message type, and existence of a latitude position reduced the data over 96,2%. Filtering for position messages that occurred in the nodes defined at Malaga reduced the data again 99,4% for a total data reduction of 99,98%, nearly 4 orders of magnitude. Even with this reduction, by using the method of saving the data within the nodes, it is still possible to show the taxi path of a single flight on the surface as shown in Figure 18.



Figure 18. 28th of July, 2018 taxi path of hexident 344699

The congestion analysis has shown that there are a few intersections in which the majority of the traffic passes and cause the majority of the taxi time delays. More investigation of this











type with a wider database of flights needs to occur before any operational remedies can be suggested to smooth out the congestion. This first step analysis demonstrates that a rapid and inexpensive means to identify the location of these congestion causing points exists, and can be rapidly implemented.

Further investigation to be pursued using this technique include the areas of runway occupancy time analysis, rapid exitway prediction, and queue times at departure runway holdings amongst others. This is already occurring at Tenerife airport, where they are using the same antenna system and data collection method to investigate runway occupancy times and how they relate to weather and visibility conditions. However, results are not yet available.

In the age of Big Data, storing massive amounts of data for future airport post-operational analysis can become cumbersome, expensive and time consuming. Inexpensive data reductions techniques that maintain a high level of information quality are needed for a rapid analysis. While this analysis was carried out at a medium sized airport, there is no reason why the same could not be done at a large, hub airport. The only difference would be the data processing times.

The systems and methodology outlined in this paper individually are not innovative. Each step is either using a COTS product or a known algorithm. What is innovative is the combination of these steps and their application for traffic flow analysis at a medium to small airport. The paper provides an example of how post-operational analysis can be performed at not only large airports, but also at airports that do not have ground surveillance systems installed.

#### REFERENCES

- EUROCONTROL, "Air traffic management airport performance (ATMAP) framework,", December 2009, <u>https://www.eurocontrol.int/sites/default/files/2019-05/atmap-reportdecember-2009.pdf</u>
- [2] SESAR JU (2018), SESAR Solution PJ.04-01:Validation Report (VALR).
- [3] NGROK.(2019) Expose a local web server to the internet: https://ngrok.com/docs
- [4] ICAO (2013). Guide on Technical and Operational Considerations for the Implementation of ADS-B in the SAM region; https://www.icao.int/SAM/eDocuments/ADSB%20Guide%20Vs1.2%20 English.pdf
- [5] Sun, J. (2017). The 1090MHz Riddle. An open-access book about decoding Mode-S and ADS-B data. <u>https://mode-s.org/decode/</u>
- [6] Bones Aviaation Page. Article 4.2 ADS-B Socket Data formating: http://woodair.net/sbs/article/barebones42\_socket\_data.htm
- [7] JUPYTER (2017). The Jupyter Notebook. <u>https://jupyter-notebook.readthedocs.io/en/stable/</u>
- [8] OSGeo Proyect. (2019). Documentation for QGIS 3.4. Obtenido de QGIS Un Sistema de Información Geográfica libre y de Código Abierto: <u>https://www.qgis.org</u>
- [9] Schaub, T. (2016). *The GeoJSON Format.* Internet Engineering Task Force.
- [10] Zhang, T., Ding, M., Zuo, H., Chen, J., Weiszer, M., Qian, X., & Burke, E. (2018). An online speed profile generation approach for efficient airport ground movement.









