TFG TITLE: Taxi time analysis and prediction with ADS-B data. A case study in Barcelona-El Prat Airport

DEGREE: Grau en Enginyeria d'Aeronavegació i Grau en Enginyeria d'Aeroports

AUTHOR: Marc Aicart i Font

ADVISORS: Xavier Prats i Menéndez
          Oscar Maldonado Diaz

DATE: September 12, 2017
**Titol:** Anàlisi i predicció de temps de rodatge amb dades ADS-b. Estudi a l’Aeroport de Barcelona-El Prat

**Autor:** Marc Aicart i Font

**Directors:** Xavier Prats i Menéndez
Oscar Maldonado Díaz

**Data:** 12 de setembre de 2017

---

**Resum**

Aquest document conté un estudi dels temps de rodatge de l’aeroport de Barcelona-El Prat utilitzant dades ADS-B. En la Secció 1 es mostra com descodificar les dades ADS-B i quina informació útil se’n pot extreure de cara a fer l’anàlisi de temps de rodatge. La Secció 2 mostra com modelar l’aeroport, incloent pistes, carrers de rodatge i llocs d’estacionament a partir de dades de l’AIP, imatges de satèl·lit i plànols oficials. La Secció 3 mostra com es pot relacionar les posicions obtingudes a partir d’ADS-B i l’aeroport modelat per tal de poder definir inequívocament la trajectòria que ha seguit un avió dins d’aquest aeroport modelat. La Secció 4 mostra com a partir de tota la informació recopilada en les anteriors seccions es poden determinar els factors que afecten al temps de rodatge i com crear un model que permeti estimar aquests temps.

En anteriors estudis s’havia restringit el càlcul de temps de rodatge a unes situacions o localitzacions molt concretes, però en aquest document s’intenta relativitzar tots els paràmetres (treballant amb velocitats, cues relatives i diferenciació per operacions) per tal de poder estendre aquest càlcul a totes les operacions d’un aeroport. Els resultats demostren que aquest objectiu s’ha complert amb una precisió en 2 minuts de marge (requeriment A-CDM) del 73% en sortides i del 97% en arribades.

El model que es proposa no només es caracteritza per una bona precisió en condicions estàtiques sinó que demostra una bona adaptació a les condicions canviants. Si bé és cert que el model no funciona bé quan els sets de dades d’entrenament i validació tenen condicions diferents, el model ha demostrat ser vàlid en noves condicions amb sets de dades d’entrenament molt petits en les noves condicions. Els models tradicionals es basen en el càlcul d’històrics punt a punt i necessiten un període llarg de dades per poder extreure conclusions, cosa que fa inviable el càlcul de temps de rodatge just després de canviar les condicions. En el model proposat, gràcies als càlculs en velocitats i relativitzacions, és capaç d’extreure informació de totes les dades disponibles i crear prediccions amb bona precisió encara que les condicions hagin canviat recentment.

En el present document es presenta, a més, un cas real de càlcul en condicions extraordinàries on es pot veure la capacitat d’adaptació del model a l’aeroport de Barcelona-El Prat al tancar una pista per manteniment durant un mes.
This document contains a study about taxi time analysis in Barcelona-El Prat airport using ADS-B data. Section 1 shows how to decode ADS-B data and what useful information can be recovered to perform the taxi time analysis. Section 2 shows how to model the airport, including runways, taxiways and stands based on AIP data, satellite images and official maps. Section 3 shows how the positions obtained from ADS-B and the modeled airport can be related to be able to unequivocally define the trajectory that an airplane has followed through this modeled airport. Section 4 shows from all the information compiled in the previous sections, how can be determined the factors that affect taxi time and how to create a model that allows estimating them.

In previous studies taxi time calculation had been restricted to very specific situations or locations, but in this document, will be tried to relative all parameters (working with speeds, relative queues and differentiation for operations) to be able to extend this calculation to all airport operations. The results show that this goal has been achieved with an accuracy of 2 minutes (A-CDM requirement) of 73% in departures and 97% in arrivals.

The proposed model is not only characterized by high accuracy in static conditions but also shows a good adaptation to changing conditions. Although it’s true that the model doesn’t work well when the training and evaluating data sets have different conditions, the model has proven to be valid under new conditions with a very small set of training data in the new conditions. Traditional models are based on calculation of point-to-point histories, which need a very large period of data to extract conclusions. With this methods it’s difficult to calculate taxi time just after a condition change. With the proposed model, thanks to calculations in velocities and relativities, model is able to extract information from all the available data and create predictions with good accuracy even if the conditions have recently changed.

This document also presents a real case of calculation in extraordinary conditions where adaptation capacity of the model can be seen in Barcelona-El Prat airport during a one-month runway closure due to maintenance.
A la meva família;
sense els quals no hauria arribat fins aquí.

A tots els companys becaris;
per totes les estones que hem compartit junts.
CONTENTS

Acronyms ................................................................. xv

Acknowledgements .................................................... 1

INTRODUCTION .......................................................... 3

CHAPTER 1. ADS-B data collection and message decoding ........ 5
  1.1. ADS-B fundamentals .............................................. 5
  1.2. ADS-B message layout ........................................... 6
  1.3. ADS-B messages reception and decoding ....................... 7
      1.3.1. Timestamp decoding from Radarcape .................. 8
      1.3.2. Callsign decoding ........................................... 8
      1.3.3. Ground position decoding ................................. 9

CHAPTER 2. Airport modelling ........................................ 13
  2.1. Airport Identification, ARP and Runway Modelling ............ 13
  2.2. Taxiway Modelling ............................................... 14
  2.3. Stand Modelling .................................................. 15
  2.4. Ground Vehicles and Calibration aids ........................... 16
  2.5. Aircraft with non-accurate location broadcast .................. 17
  2.6. Aircraft size modelling: MTOW .................................. 17
  2.7. Runway Configuration modelling ................................ 18

CHAPTER 3. Path Recognition ........................................... 19
  3.1. Mode Recognition ................................................ 19
  3.2. Runway recognition .............................................. 20
  3.3. Stand Recognition ............................................... 21
  3.4. Taxiway recognition ............................................. 24
CHAPTER 4. Taxitime calculation  .................................................. 29
  4.1. Selection of Explanatory Variables ........................................ 29
  4.2. Variables Modeling ............................................................... 31
  4.3. Function Fitting ................................................................. 33
  4.4. Results evaluation ............................................................... 34
  4.5. Testing under extraordinary conditions ................................... 35
  4.6. Future development ............................................................. 36

CONCLUSIONS ............................................................................. 37

BIBLIOGRAPHY ........................................................................... 39
LIST OF FIGURES

1.1 ADS-B principals: Each aircraft calculates its own position and broadcasts it through Mode S ................................................................. 5
1.2 Barcelona El-Prat Airport low ADS-B coverage zones from EETAC’s station 8

2.1 LEBL runway modelling ............................................................. 14
2.2 LEBL taxiway modelling ........................................................... 15
2.3 LEBL stand modelling ............................................................. 16
2.4 D-ALED path on May 26th 2017 in LEBL .................................... 18

3.1 Runway recognition using first method ........................................ 20
3.2 Runway recognition using second method (angle not to scale) ......... 21
3.3 Stand recognition using a radius of tolerance .............................. 21
3.4 Stand recognition using a trigonometric oval .................................. 22
3.5 Overlapped stands with different entry angle .............................. 23
3.6 Overlapped stands with the same entry angle and parallel center lines 24
3.7 Standard taxi route in LEBL from runway 25R vacated via R6 to Gate ES 25

4.1 Taxi distance vs Taxitime in KDFW (real data and SFFS model)[6] ....... 30
4.2 Comparison between a 3rd and a 9th order approximation on a taxi speed vs #flights plot ................................................................. 32
4.3 Taxi Observed speed vs Taxitime model error on LEBL ................. 35
4.4 AENA e-CDM system screen shot ............................................. 36
# LIST OF TABLES

1.1 ADS-B message structure ................................................. 6  
1.2 ADS-B Type Code references .......................................... 7  
1.3 Aircraft identification and category message structure .................. 8  
1.4 Surface position message structure ...................................... 9  
1.5 Ground Speed decoding table ........................................... 10  

2.1 runways.dat file structure ............................................... 13  
2.2 taxiways.dat file structure ............................................. 14  
2.3 stands.dat file structure ................................................ 15  
2.4 gndvehicles.dat file structure ......................................... 16  
2.5 toberemoved.dat file structure ........................................ 17  
2.6 mtow.dat file structure .................................................. 18  
2.7 runwayconfig.dat file structure ....................................... 18  

3.1 3-taxiway segments guess table from R6 rapid exit ...................... 27  

4.1 Function constants found on March 2017 training data set ................ 34
ACRONYMS

A-CDM  Airport Collaborative Decision Making.

ACARS  Aircraft Communications Addressing and Reporting System.

ADS-B  Automatic Dependant Surveillance Broadcast.

ADS-C  Automatic Dependant Surveillance Contract.

AENA  Aeropuertos Españoles y Navegación Aérea.

AFT  Aircraft.

AIP  Aeronautical Information Publication.

AIRAC  Aeronautical Information Regulation and Control.

ANSP  Air Navigation Service Provider.

AOBT  Actual Off Block Time.

ARP  Airport Reference Point.

ATC  Air Traffic Control.

ATFM  Air Traffic Flow Management.

CRC  Cyclic Redundancy Check.

CTOT  Calculated Take Off Time.

DME  Distance Measuring Equipment.

ED50  European Datum 1950.


EXIT  Estimated Taxi In Time.

EXOT  Estimated Taxi Out Time.

FAA  Federal Aviation Administration.

GNSS  Global Navigation Satellite System.

GPS  Global Positioning System.

GS  Ground Speed.

HDG  Heading.

ICAO  International Civil Aviation Organization.
INS  Inertial Navigation System.

LDT  Landing Time.

MLAT Multilateration.

MTOW Maximum Take Off Weight.

REA  Ready to Depart Message.


RTO Rejected Take Off.

RWY Runway.

SATR Standard Airport Taxi Routes.

SIBT Stand In-Block Time.

SMR Surface Movement Radar.


SUP Supplemento.

TCAS Traffic Collision and Avoidance System.

THR Threshold.

TIS-B Traffic Information Service Broadcast.

TOT Take Off Time.

TRK Track.

TSAT Target Start-up Approval Time.

TWY Taxiway.

VOR VHF Omnidirectional Range.

VTT Variable Taxi Time.

I would like to express my very great appreciation to all ICARUS research group members for all the support and help they gave me. In particular I wanted to thank Mr. Marc Pérez-Batlle for all his work on ADS-B decoding software.

Also, I wish to acknowledge the help provided by members of the AENA Operations and Aeronautical Information divisions in Barcelona-El Prat Airport.
Taxi time analysis and prediction with ADS-B data. A case study in Barcelona-El Prat Airport
INTRODUCTION

The taxitime is a parameter that indicates the time that takes an aircraft to move in an airport. The taxitime is defined for departures as the time between off-block and take-off times and for arrivals as the time between landing and in-block times. This parameter had a limited utility and, therefore, was roughly approximated. Its utility was only considered by ATC (Air Traffic Control) when sending a Ready to Depart Message (REA) to the ATFM unit (Air Traffic Flow Management), that must include a taxitime indication[1]. This taxitime information did not have to be very accurate because it was only for slot allocation and errors on taxitime approximation could be recovered using the slot tolerance.

In in A-CDM (Airport Collaborative Decision Making) airports[2], however, the situation is different and a good taxitime estimation is needed to make some predictions. For instance, taxitime is needed for all aircraft to calculate its TSAT (Target Start-up Approval Time) and SIBT (Stand In-Blocks Time) and accuracy is required to be higher. For departing aircraft from a A-CDM airport TSATs are assigned by taking into account the departure time (TOT/CTOT) and the taxi-out time (EXOT) while for arrivals SIBT is calculated using the landing time (LDT) and the taxi-in time (EXIT). In both cases, a high-accuracy taxitime calculation is required to guarantee better planning and not to generate and propagate delays in case that actual taxitime differs from estimated one.

Motivated by the implementation of A-CDM, ANSPs (Air Navigation Service Provider) and Airport Authorities tried to improve their taxitime calculations. In Barcelona-El Prat Airport (LEBL) an improved point-to point taxitime table was implemented. To do this, AENA (Aeropuertos Españoles y Navegación Aérea) processed a big amount of historical data and established a set of tables defining the standard taxitime. For departing flights taxitime was defined from each stand to each runway holding point and for arrival flights from each runway exit to each stand using SATRs (Standard Airport Taxi Routes)[3].

Tables solution improved accuracy but have a major disadvantage, which is they are static. Even if airport conditions change, taxitime estimations using a table remain constant because are only dependant on starting and ending point. This was a problem during the A-CDM test phase in LEBL. To overcome this, A-CDM procedures were shut-down when real taxitime highly differed from the predicted ones due some airport disturbance (bad weather, rejected take off...).

Aiming to improve the static tables methodology, some dynamic solutions started to be developed and a first solution was implemented for taxi-out time calculation in Boston-Logan Airport (KBOS)[4]. These calculations relied on a mathematical method that had to be trained using historical data. Additional methods like [5] and [6] were developed to improve accuracy against the mentioned one and added new features like taxi-in time calculation.

Since these methods try to determine taxitime only from airport surrounding traffic and queues, they only work for a small group of stands with the same aircraft mix and with a small group of modelled airlines.

To overcome the taxitime calculation limitations, some papers like [7] and [8] focus on determining taxi speeds to avoid effects on distance and being able to compare data from more locations.
In this project, taxi speeds will be calculated for Barcelona-El Prat Airport using ADS-B (Automatic Dependant Surveillance Broadcast) data. ADS-B data provides accurate location and time stamps on most aircraft, but may have coverage losses from a specific station. In this study a modelled airport will be used to relate taxi time and taxi speed using the taxi distance. An airport model should include runways, taxiways and stands and its recognition methods. In this project a solution is proposed where runways, taxiways and stands are defined with coordinates and an orientation. Also a set of recognition methods that have been proven effective on LEBL even with low-coverage zones around the airfield are shown here. This method includes a tool which allows to recover the most probable taxi route in case some ADS-B messages are lost.

With this treated data, taxi speed will be calculated by defining and modelling several variables that affect it. A logarithmic sum of polynomial fittings of this variables will be used to determine the taxi speed. The values of the constants of this formula will be determined using an iterative method. This method will help finding an optimal solution for the set of constants that will ensure accuracy on taxi speed calculations.
CHAPTER 1. ADS-B DATA COLLECTION AND MESSAGE DECODING

1.1. ADS-B fundamentals

ADS-B or Automatic Dependent Surveillance Broadcast is a function on an aircraft or a surface vehicle that periodically broadcasts its identification, position, altitude, horizontal and vertical velocity, heading, TCAS status and other information. This information is previously calculated on board using a navigation system like GNSS, INS, VOR-DME, etc. ADS-B supports an improved use of airspace, reduced ceiling/visibility restrictions, improved surface surveillance, and enhanced safety such as conflict management.

Figure 1.1: ADS-B principals: Each aircraft calculates its own position and broadcasts it through Mode S.

Regarding the reason of its name, it is Automatic in the sense that no pilot or controller action is needed, and Dependent because the information that is sent depends on external navigation systems in the aircraft. The main distinction of ADS-B when compared with other ADS systems, such as ADS-C (Automatic Dependent Surveillance Contract), is that the information is broadcast without being asked.


ADS-B and SSR systems behave in a similar way, but there are some differences between them. Main difference is that an SSR system is bidirectional because an interrogation is needed to generate a response. On the other hand, ADS-B data is periodically broadcast regardless the interrogations. To complement this system ADS-C technology was created and allows bidirectional contact between two stations. Currently ADS-C is mandatory in some oceanic regions like the North Atlantic airspace[12].

Nowadays ADS-B is mandatory in some regions like Australia (some exceptions apply[13]) and is used for ATC in en-route phases. Since in Europe and US is not yet mandatory for non-new aircraft[14], approximately only a 70% of aircraft are equipped with this technol-
ogy. To provide the aircraft that are already in service with the needed equipment a retrofit has to be done.

To increase ADS-B coverage Multilateration (MLAT) technology was developed. This system allows positioning of any Mode A/C equipped aircraft or ground vehicle. MLAT uses the Mode A/C messages triangulation to position an aircraft (Mode S and ADS-B messages could be used also). Unlike other triangulation systems like GPS, in MLAT messages are sent from the moving target and computation is made on ground stations which are interconnected. As any triangulation positioning system, MLAT needs to be in contact with at least 4 receivers to determine a position. This requirement is achieved easily over 12000ft, but needs several additional stations to work below this altitude due to obstacles. This restriction becomes critical on-ground and, for example, Barcelona El-Prat Airport (LEBL) needs 36 MLAT receivers to guarantee coverage in all the movements area[15].

1.2. ADS-B message layout

The message structure in mode-S and ADS-B is defined by an RTCA standard[16] and FAA[17], EUROCONTROL[18] and ICAO[19][20] have published implementation guides that make decoding easier to understand.

ADS-B messages consist on a 112bits sequence defining 5 parts[21] as it’s shown on Table 1.1.

<table>
<thead>
<tr>
<th>Item</th>
<th>DF</th>
<th>CA</th>
<th>ICAO</th>
<th>DATA</th>
<th>PI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 bits</td>
<td>3 bits</td>
<td>24 bits</td>
<td>4+52=56 bits</td>
<td>24 bits</td>
</tr>
</tbody>
</table>

Table 1.1: ADS-B message structure

Item **DF** stands for Downlink Format and indicates message type. ADS-B messages have a DF equal to 17 (1001 in binary), whereas Mode-S squitter for civil aircraft equals 11 and for military aircraft equals 19. Downlink Format can be read directly on binary on the first 5 bits of the message.

Item **CA** stands for Mode-S squitter Capability and indicates characteristics of the provided information from the aircraft. Capability can be read in bits [6:8].

Item **ICAO** stands for ICAO-24 aircraft address and indicates aircraft identification. This field contains 24 bits in binary, which makes 3 bytes each of them containing 2 hexadecimal characters. Thus makes a 6-character hexadecimal address which fully identifies an aircraft. Unlike SSR-squawk codes, ICAO-24 addresses are unique for each aircraft and cannot be repeated. ICAO-24 address can be read directly once converted to hexadecimal bits [9:32].

Item **DATA** contains the transmitted data on the message. This field contains a Type Code (TC) in bits [33:37] in binary which indicates message type as shown in table 1.2 and the message (MSG) which can be decoded using [17].

---

\(^1\) Computation based on received messages in EETAC ADS-B station during March 2017
<table>
<thead>
<tr>
<th>Type Code</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No position information</td>
</tr>
<tr>
<td>1-4</td>
<td>Aircraft identification and category</td>
</tr>
<tr>
<td>5-8</td>
<td>Surface position and velocity</td>
</tr>
<tr>
<td>9-18</td>
<td>Airborne position (Barometric altitude)</td>
</tr>
<tr>
<td>19</td>
<td>Airborne velocity</td>
</tr>
<tr>
<td>20-22</td>
<td>Airborne position (GNSS height)</td>
</tr>
<tr>
<td>23-31</td>
<td>Reserved for other applications</td>
</tr>
</tbody>
</table>

Table 1.2: ADS-B Type Code references

Item PI stands for Parity/Interrogator ID and is used to check for errors on the message. With this part of the message and using CRC (Cyclic Redundancy Check) a bit error can be found and even corrected[22].

1.3. ADS-B messages reception and decoding

In December 2016 a Radarcape ADS-B receiver was installed on EETAC in Castelldefels, Spain to study surrounding air traffic. Statistics show that system can detect aircraft up to 325nm but mean distance happened to be less than 175nm. The receiver is located less than 5 miles westbound of Barcelona El-Prat Airport and is capable of detecting ground traffic in it.

EETAC’s Radarcape was programmed to time-stamp all ADS-B and Mode-A/C messages and send them to a server where were stored. As an average, the receiver produces 900Mb of binary messages from more than 1500 aircraft per day.

After months of data, was found that some airport zones had a bad ADS-B coverage due to electromagnetic shadows produced from the airport terminal building. In Figure 1.2 low coverage zones from EETAC’s station are highlighted. Terminal 1 building is assumed to cause this low coverage zones, because as it can be seen on the figure, it is located between the problematic zone and the receiver location.

After further analysis was found that near all terminal buildings some messages were also lost. Thus made determining parking positions difficult in some cases, especially where aircraft was parked next to terminal building and facing it.

In order to perform the taxitime study only callsign and ground position ADS-B messages and their timestamp will be considered. In the following sections a detailed explanation about decoding those messages will be given.

ADS-B messages will be decoded using pyModeS python package developed by Mr. Junzi Sun. This package is one of the most complete tools for ADS-B and Mode-S mes-
sage decoding. At the time this project was being developed, the package did not include some surface decoding methods. After some contributions and implementation suggest-ions made by Mr. Marc Pérez-Batlle and the author to the package developer, those were implemented.

### 1.3.1. Timestamp decoding from Radarcape

Since ADS-B and Mode-A/C/S messages does not include a timestamp, for non real-time applications, it has to be added by the receiver. Thus makes the accuracy of the timestamp related to the receiver internal clock. In the studied case, Radarcape includes a GPS antenna, which makes timestamp accurate up to nanoseconds. This accuracy is enough to use this device as an MLAT receiver.

As stated by Radarcape Support[23], a 48 bits timestamp is added before each message. 18 upper bits are used to code the seconds of the day, while 30 lower bits contain nanoseconds. Since day information is not provided, server has been configured to store messages in a file per day basis. That way, day information can be easily found (from server clock) in the file name.

### 1.3.2. Callsign decoding

Callsign information is one of the simplest ADS-B message to decode. As it can be seen on Table 1.2 this messages have a Type Code from 1 to 4 depending on aircraft category. The 56bits sequence is divided as shown in Table 1.3.

<table>
<thead>
<tr>
<th>TC</th>
<th>EC</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1:5]</td>
<td>[6:8]</td>
<td>[9:14]</td>
<td>[15:20]</td>
<td>[21:26]</td>
<td>[27:32]</td>
<td>[33:38]</td>
<td>[39:44]</td>
<td>[45:50]</td>
<td>[51:56]</td>
</tr>
<tr>
<td>5bits</td>
<td>3bits</td>
<td>6bits</td>
<td>6bits</td>
<td>6bits</td>
<td>6bits</td>
<td>6bits</td>
<td>6bits</td>
<td>6bits</td>
<td>6bits</td>
</tr>
</tbody>
</table>

Table 1.3: Aircraft identification and category message structure
Items **TC** and **EC** refer to Type Code and Emitter Category respectively and indicate type of aircraft or ground vehicle that is emitting the message.

Items **C1-C8** refer to the callsign characters. Each one uses 6 bits which makes 64 possible characters. To relate this field to a specific character a mapping needs to be done as follows from 0 to 63:

```
#ABCDEFGHIJKLMNOPQRSTUVWXYZ####0123456789####
```

With this mapping characters from A to Z use values [1:26], numbers form 0 to 9 use [48:57] and blank space use [32].

### 1.3.3. Ground position decoding

ADS-B uses CPR (Compact Position Reporting) format to encode aircraft position[24]. The main idea is to encode accurate locations in less bits using two messages. The information provided in one message leads to multiple solutions around the world, but using two messages a single solution is found. In CPR the globe is divided into several zones. Position inside the zone is defined in each message, but zone can only be determined combining two messages called *odd* and *even*.

In CPR the number of bits to encode position differs from airborne encoding ($Nb = 17$) and surface encoding ($Nb = 19$). This parameter determines the accuracy, which is approximately 5m for airborne and 1.25m for surface.

The Earth is divided into 15 geographic latitude zones between equator and the poles for Mode-S ($NZ = 15$). The $NZ$ parameter determines the unambiguous range, which corresponds to the distance between to consecutive latitude zones, and is 360NM for airborne and 90NM for surface (because high-order 2 bits are omitted).

The number of geographic longitude zones is depends on latitude using equation 1.1 which leads to a result between 1 and 59.

$$NL(lat) = \text{floor}\left(2\pi \left(\arccos\left(1 - \frac{1 - \cos\left(\frac{\pi}{2NZ}\right)}{\cos^2\left(\frac{\pi}{180}\left|lat\right|\right)}\right)^{-1}\right)\right) \quad (1.1)$$

Surface position messages have are structured as table 1.4 shows. Message includes speed, heading and CPR position.

<table>
<thead>
<tr>
<th><strong>TC</strong></th>
<th><strong>MV</strong></th>
<th><strong>S</strong></th>
<th><strong>GT</strong></th>
<th><strong>IMF</strong></th>
<th><strong>CPR F</strong></th>
<th><strong>CPR lat</strong></th>
<th><strong>CPR lon</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1:5]</td>
<td>[6:12]</td>
<td>[13]</td>
<td>[14:20]</td>
<td>[21]</td>
<td>[22]</td>
<td>[23:39]</td>
<td>[40:56]</td>
</tr>
<tr>
<td>5bits</td>
<td>7bits</td>
<td>1bit</td>
<td>7bits</td>
<td>1bit</td>
<td>17bits</td>
<td>17bits</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.4**: Surface position message structure

Item **TC** refers to Type Code. For surface position messages this field is compressed between [5:8].

Item **MV** stands for Movement and indicates ground speed. A detailed decoding explanation is shown in section 1.3.3.1..
Item \( \mathbf{S} \) stands for ground track status and indicates the validity of the ground track field. This field admits two values: \( 0= \text{invalid} \) and \( 1= \text{valid} \).

Item \( \mathbf{GT} \) stands for ground track and heading. A detailed decoding explanation is shown in section 1.3.3.2.

Item \( \mathbf{IMF} \) stands for ICAO Mode-A Flag and indicates the type of identity associated with the aircraft data reported in the TIS-B message. IMF equals to \( 0 \) if TIS-B data is identified by ICAO-24 address. Otherwise if IMF equals to \( 1 \) if TIS-B is identified by Mode-A code.

Item \( \mathbf{CPR F} \) stands for CPR Format and indicates if message is coded with \textit{odd} (CPR F=1) or \textit{even} (CPR F=0) format.

Items \( \mathbf{CPR lat} \) and \( \mathbf{CPR lon} \) indicate latitude and longitude coded using CPR. Decoding can be done in two ways: using two messages (section 1.3.3.3.) or using one message (section 1.3.3.4.).

### 1.3.3.1. Ground Speed decoding

Ground Speed is coded using 7 bits which leads to 128 values. A non-linear scale is used as it is shown on Table 1.5. Thus makes Ground Speed accuracy be greater for lower values of GS, where quantization steps are smaller.

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Meaning</th>
<th>Quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No information available</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>( GS &lt; 0.125kt )</td>
<td></td>
</tr>
<tr>
<td>2-8</td>
<td>( 0.125kt \leq GS &lt; 1kt )</td>
<td>0.125kt steps</td>
</tr>
<tr>
<td>9-12</td>
<td>( 1kt \leq GS &lt; 2kt )</td>
<td>0.25kt steps</td>
</tr>
<tr>
<td>13-38</td>
<td>( 2kt \leq GS &lt; 15kt )</td>
<td>0.5kt steps</td>
</tr>
<tr>
<td>39-93</td>
<td>( 15kt \leq GS &lt; 70kt )</td>
<td>1kt steps</td>
</tr>
<tr>
<td>94-108</td>
<td>( 70kt \leq GS &lt; 100kt )</td>
<td>2kt steps</td>
</tr>
<tr>
<td>109-123</td>
<td>( 100kt \leq GS &lt; 175kt )</td>
<td>5kt steps</td>
</tr>
<tr>
<td>124</td>
<td>( 175kt \leq GS )</td>
<td></td>
</tr>
<tr>
<td>125-127</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.5: Ground Speed decoding table

### 1.3.3.2. Heading decoding

Heading is coded using 7 bits which leads to 128 values. A linear scale is used with \( \frac{360}{128} = 2.8125^\circ \) steps. This way, a 0000000 means 0\(^\circ\) and 1111111 means 357.1875\(^\circ\).

### 1.3.3.3. CPR decoding using two messages

CPR can be decoded using two messages from the same aircraft and opposite format (\textit{odd} and \textit{even}) sent in a short interval. This method is also known as \textit{Globally Unambiguous Position} because no reference is needed. Algorithm to decode this messages works as follows[17][21]:
1. Compute latitudes and longitudes from messages for CPR 17-bits:

\[
CPR(\text{lat}/\text{lon})_{\text{odd/even}} = \frac{(\text{lat}/\text{lon})_{\text{odd/even}}}{2^{17}}
\]  
(1.2)

2. Compute latitude zone sizes \(dLat_0\) and \(dLat_1\):

\[
dLat_i = \frac{90}{4NZ - i}
\]
(1.3)

3. Compute latitude index \(j\):

\[
j = \text{floor} \left( 59 \cdot CPRlat_{\text{even}} - 60 \cdot CPRlat_{\text{odd}} + \frac{1}{2} \right)
\]
(1.4)

4. Compute relative latitudes:

\[
Lat_i = dLat_i \cdot (\text{mod}(j, 60) + CPRlat_i)
\]
(1.5)

5. Correct latitudes for southern hemispheres, so are in the range [-90,90]. Final latitude used will be the newest. If latitude from both messages are in different latitude zones, computation is not possible.

6. Calculate longitude parameters:

\[
n_i = \max(NL(\text{lat}) - i, 1)
\]
(1.6)

\[
dLon = \frac{360}{n_i}
\]
(1.7)

\[
m = \text{floor} \left( CPRlon_0 \cdot [NL(Lat) - 1] - CPRlon_1 \cdot NL(\text{lat}) + \frac{1}{2} \right)
\]
(1.8)

\[
Lon = dLon \cdot (\text{mod}(m, n_i) + CPRlon_i)
\]
(1.9)

7. Correct latitudes, so are in the range [-180,180].

This method is used when receiver first detects an aircraft and position is unknown. When position is already determined, another method can be used instead. Method presented in next section works with one message providing that a recent location is known.

1.3.3.4. CPR decoding using one message and a reference

This method presents some differences with the previous one because only one message and a reference are needed. The method is useful when an aircraft position has been previously calculated or aircraft is sure to be located inside a radius (for example an airport).

Provided reference must be within 180NM range for airborne messages and 45NM for surface messages (which is half unambiguous range). Algorithm to decode this messages works as follows [17][21]:
1. Compute latitude and longitude from message for CPR 17-bits:

\[
CPR(\text{lat}/\text{lon}) = \frac{(\text{lat}/\text{lon})}{2^{17}}
\]  \hspace{1cm} (1.10)

2. Compute latitude zone sizes \(d\text{Lat}_0\) and \(d\text{Lat}_1\):

\[
d\text{Lat}_i = \frac{90}{4\text{NZ} - i}
\]  \hspace{1cm} (1.11)

3. Compute latitude index \(j\):

\[
j = \text{floor}\left(\frac{\text{Lat}_\text{ref}}{d\text{Lat}}\right) + \text{floor}\left(\frac{\text{mod}(\text{Lat}_\text{ref}, d\text{Lat})}{d\text{Lat}} - \text{CPRlat} + \frac{1}{2}\right)
\]  \hspace{1cm} (1.12)

4. Compute latitude:

\[
\text{Lat} = d\text{Lat} \cdot (j + \text{CPRlat})
\]  \hspace{1cm} (1.13)

5. Compute dLon:

\[
d\text{Lon} = \begin{cases} 
360 & \text{if } \text{NL(lat)} > 0 \\
\frac{360}{\text{NL(lat)}} & \text{if } \text{NL(lat)} = 0
\end{cases}
\]  \hspace{1cm} (1.14)

6. Compute longitude index \(m\):

\[
m = \text{floor}\left(\frac{\text{Lon}_\text{ref}}{d\text{Lon}}\right) + \text{floor}\left(\frac{\text{mod}(\text{Lon}_\text{ref}, d\text{Lon})}{d\text{Lon}} - \text{CPRlon} + \frac{1}{2}\right)
\]  \hspace{1cm} (1.15)

7. Compute longitude:

\[
\text{Lon} = d\text{Lon} \cdot (m + \text{CPRlon})
\]  \hspace{1cm} (1.16)

This method will be used due to its simplicity and also because a position is known. Airport ARP position will be used as a reference to decode all surface messages.
CHAPTER 2. AIRPORT MODELLING

In this chapter airport modelling is shown. The aim of the modelling is to define some airport characteristics that will allow to determine, for a given ADS-B ground track, the runway, taxi-route and stand used by the aircraft and their timestamp. WSG-84 coordinate system is used to model the airport because is the same system used for ADS-B messages. Besides the airport, some aircraft-characteristics will be also modelled to use them during taxitime analysis and calculation.

2.1. Airport Identification, ARP and Runway Modelling

Airport identification modelling consists on ICAO 4-letter code of the airport. Currently this is used only for labeling tasks but would be needed when integrating VTT (Variable TaxiTime) and A-CDM systems.

ARP (Airport Reference Point) modelling is needed for decoding ADS-B positions using only one message as section 1.3.3.4. shows. Since this position is usually located halfway the main runway it can be considered to be the central point of the airport.

ARP is also used to only consider reported positions closer than a given range to the airport. With this filter, system makes sure that all ground positions received proceed from aircraft in the studied airport.

Runway modeling is based on threshold position and runway track. To guarantee high accuracy on the data, AIP coordinates and tracks were considered.

File containing this data has the following structure (where RWY stands for Runway and THR stands for Threshold):

<table>
<thead>
<tr>
<th>File Name: runways.dat</th>
<th>Separator: Tabulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport Identification</td>
<td>ARP Latitude</td>
</tr>
<tr>
<td>RWY Name</td>
<td>THR1 Name</td>
</tr>
<tr>
<td>RWY2 Name</td>
<td>…</td>
</tr>
<tr>
<td>ARP Longitude</td>
<td>THR1 Lon</td>
</tr>
<tr>
<td>…</td>
<td>THR1 HP</td>
</tr>
</tbody>
</table>

Table 2.1: runways.dat file structure

Items THR HP indicate the holding points for a given threshold. This information is used to improve taxiway recognition. Here it will be included all the aircraft possible entries to the specified runway, including some crossing taxiways. Rapid exits will not be included because are specified to be one way only.

Figure 2.1 shows the 3 modeled runway in LEBL airport overlaying an airport diagram where threshold position can be observed.

1 In EETAC’s ADS-B station several surface positions were received from LEPA, LEGE and LERS while studying LEBL ground movement
2 Refer to AIP for details
2.2. Taxiway Modelling

The modelling of taxiways is based on dividing each taxiway in several segments. Usually taxiways are divided on sub-taxiways segments between each taxiway crossing, stub\(^3\), by-pass\(^4\) or rapid exit. This segments are allowed be named using a letter and a digit as [25] and [26] state.

Here sub-taxiways are defined with the central point coordinates and its track. Segment distance and the linking taxiways on each end are also noted on the file defining them. This file has the following structure (where TWY stands for taxiway and TRK stands for track):

<table>
<thead>
<tr>
<th>File Name: taxiways.dat</th>
<th>Separator: Tabulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport Identification</td>
<td></td>
</tr>
<tr>
<td>TWY Name</td>
<td>TWY Lat</td>
</tr>
<tr>
<td>TWY2 Name</td>
<td>. . . . . . . . . . .</td>
</tr>
</tbody>
</table>

Table 2.2: taxiways.dat file structure

Items LINK and LINK Reverse indicate the taxiways, gates and runways that can be reached from a specific sub-taxiway. Parameters on the LINK item can be reached using the same direction as the parameter TCK indicates. Otherwise parameters on LINK Reverse can be reached using the opposite direction as parameter TCK indicates.

Parameter LINK and LINK Reverse also indicate if the mention taxiway is a gate to enter the apron. From stand modelling (section 2.3.), available stands from a specific gate can be found.

Parameters on both fields should be ordered by ATC preference taking into account local taxi procedures. Usually, straight routes are preferred, unless SATRs indicate a turn. This preferences will be taken into account when determining the taxi route recognition in section 3.4.1..

\(^3\)Taxiway that connects a runway to a parallel taxiway or a taxiway to an adjacent apron area

\(^4\)Non-procedural taxi route authorized by ATC
In order to avoid miss-recognition between two crossing taxiways it is important to apply some considerations. Sub-taxiway coordinates should be placed over the center-line and halfway the segment, this point should also avoid crossing and turning areas where track cannot be determined accurately. Figure 2.2 shows modeled taxiways in LEBL airport overlaying an airport diagram where this considerations have been taken into account.

Figure 2.2: LEBL taxiway modelling

2.3. Stand Modelling

The airport stand modeling consists on stand coordinates and entry line track. In stand modeling data must be accurate so coordinates in AIP and entry line tracks on official maps will be used\(^5\).

File containing this data has the following structure (where S stands for stand and AFT stands for aircraft):

<table>
<thead>
<tr>
<th>File Name: stands.dat</th>
<th>Separator: Tabulator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S Group</strong></td>
<td><strong>S Name</strong></td>
</tr>
<tr>
<td>S2 Group</td>
<td>S2 Name</td>
</tr>
</tbody>
</table>

Table 2.3: stands.dat file structure

Stand position published in Spanish AIP is considered to be the latest stop bar in the stand center line by default\(^6\). In general aviation and cargo apron some stands can be used in both directions. In this cases, stand coordinates are located on the preferred orientation.

\(^5\)During LEBL stand modelling some AIP stands coordinates were detected wrong by the author. After contacting the airport authorities (AENA) a correction was made on AIRAC 1707. Further investigation showed that the errors were produced when AIS department miss-converted coordinates from ED50 to ETRS89(WSG84)

\(^6\)As AENA Operational Management Department in LEBL confirmed to the author
Stand entry line track is necessary for stand recognition. Most "Taxi-in, Push-out" stands are oriented perpendicular to the terminal building using a straight entry line which corresponds to the stand track. This stands may have a simple or displaced entry line, but this makes no difference on defining stand track, where the last entry line track will prevail[27].

Autonomous stands or "Taxi-in, Taxi-out" work on a different way. The stand shape is usually different and includes a large turn. Stands may have a perpendicular or tilted entry (like 45° or 55° entry)[28]. In this cases stand track will be defined as the aircraft orientation at the stop bar regardless of the fore orientation.

Items Max AFT and Incompatibilities are currently not in use, but will be necessary when integrating VTT with stand allocation system in A-CDM.

Figure 2.3 shows modeled stands in LEBL airport.

Figure 2.3: LEBL stand modelling

2.4. **Ground Vehicles and Calibration aids**

In some airports, ground vehicles are being equipped with a Mode-S transponder and ADS-B[29] to monitor their position and interventions. In order to detect this vehicles and differentiate them from aircraft some solutions were considered.

First solution was to identify the vehicle by using identification ADS-B message. This solution was disregarded when finding out that not all ground vehicles and transponders were broadcasting this information correctly.

Second solution was to identify all the ground vehicles in an airport by its ICAO-24 code in order to be omitted. This solution was applied by using a file containing this data with the following structure:

<table>
<thead>
<tr>
<th>File Name: gndvehicles.dat</th>
<th>Separator: Tabulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Vehicle ID</td>
<td>ICAO-24 code</td>
</tr>
<tr>
<td>Ground Vehicle2 ID</td>
<td>. . .</td>
</tr>
</tbody>
</table>

Table 2.4: gndvehicles.dat file structure
In some airports, where tests are being held, it is probable to find static transponders in airport buildings for calibration and testing purposes. To remove them from the system and prevent errors, its ICAO-24 codes have been included to the previous file like a ground vehicle.

### 2.5. Aircraft with non-accurate location broadcast

During ADS-B tests in EETAC station it has been discovered that some aircraft report their position in a non-accurate way. This means that they have quick sudden turns on heading and offset from its true position. It has been found that this happens on older aircraft like B737-400 or A300 where broadcast position is calculated from INS (Inertial Navigation System) and not from GPS.

Aircraft that suffer this issue should be removed from the system and ignored because its data is not valid for taxi analysis. Determining which aircraft suffer from this issue can be a difficult task. Some aircraft may have been retrofitted, and detecting them only with aircraft specifications broadcast on ADS-B messages is not feasible.

To remove the specified aircraft from the system a file containing its ICAO24 code will be used. Including all aircraft that visit the airport and are found not to be valid may seem a big task. But this traffic represents less than 0.1% of the average aircraft mix\(^7\).

The file containing the aircraft that need to be removed from the system because provide unreliable data has the following structure:

<table>
<thead>
<tr>
<th>File Name: toberemoved.dat</th>
<th>Separator: Tabulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft Reg</td>
<td>ICAO-24 code</td>
</tr>
<tr>
<td>Aircraft2 Reg</td>
<td>. . .</td>
</tr>
</tbody>
</table>

Table 2.5: toberemoved.dat file structure

Figure 2.4 provides a path from D-ALED (Boeing 757-236SF), which is 33 years old, while landing on runway 02 in LEBL and taxing towards the cargo apron on May 2017. As it can be seen, pattern suffers from offsets and deviations that make data unusable for taxi analysis.

### 2.6. Aircraft size modelling: MTOW

Aircraft size modelling is needed for taxi analysis. MTOW (Maximum Take-Off Weight) variable is the one chosen among all aircraft parameters to determine aircraft size. Aircraft have been separated using the ICAO Aircraft Type Designator as DOC8643 shows\(^8\). MTOW units have no effect on modelling as long as are consistent on the whole file.

File containing this data has the following structure:

\(^7\)Computation based on received messages in EETAC ADS-B station during March 2017

\(^8\)https://www.icao.int/publications/DOC8643/Pages/Search.aspx
18 Taxi time analysis and prediction with ADS-B data. A case study in Barcelona-El Prat Airport

Figure 2.4: D-ALED path on May 26th 2017 in LEBL

File Name: mtow.dat   Separator: Tabulator

<table>
<thead>
<tr>
<th>ICAO TYPE</th>
<th>MTOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICAO TYPE 2</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 2.6: mtow.dat file structure

2.7. Runway Configuration modelling

For traffic analysis, runway configurations have been taken into account. Runway Configuration information can be taken from ANSP AIP but only those that can be unambiguously detected must be added to the file in order not to be miss-recognized. Each runway configuration should include all available runways for each operation, even not being the preferred one or require a special authorization.

File containing runway configurations has the following structure:

File Name: runwayconfig.dat   Separator: Tabulator

<table>
<thead>
<tr>
<th>Airport Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration ID</td>
</tr>
<tr>
<td>Configuration2 ID</td>
</tr>
</tbody>
</table>

Table 2.7: runwayconfig.dat file structure
CHAPTER 3. PATH RECOGNITION

In this chapter the path recognition will be analyzed. From ADS-B positions and a modelled airport, a full aircraft taxi route and its milestones will be deduced. In order to do that, several steps have been taken into account. First, a mode recognition has to be performed in order to detect if the analyzed aircraft is an arrival or a departure.

After that, the software has to follow the aircraft using the LINK parameter mentioned in previous chapter to detect the runway, taxiways and stand used. Due to the bad coverage from EETAC’s ADS-B station, it is not possible to detect the stand for some aircraft, and sometimes the entire taxi route. Following sections will delve into this issues and try present some alternatives to guess not detected data to complete the aircraft taxi route.

3.1. Mode Recognition

When a position message is received from a new flight, mode (arrival/departure) must be detected. The software has to know if the flight is an arrival, a departure or a towed taxi inside the airport (treated as a departure with no runway used). To detect the mode a first point has to be recognized.

At first, four alternatives are considered when a flight is started on the system: The flight can be either on a runway, a taxiway, a stand or inside the apron. To determine the first position, following steps will be performed:

After receiving the first messages only two options are considered: the aircraft is either on a runway or on a stand. These are the most probable scenarios because the aircraft may just have landed, and switched from ADS-B airborne position messages to ADS-B surface position messages, or has tuned-on the transponder and started broadcasting. Sections 3.2. and 3.3. show in deep detail process to recognize runways and stands, respectively. If an aircraft is on a runway, the flight is considered to be an arrival and if it is on a stand it is considered to be a departure.

If the system is unable to recognize either a runway or a stand, possible locations increase. The aircraft may be also inside the apron taxing out. This is a probable solution in LEBL when a stand is located in a low coverage area from EETAC’s ADS-B station. When this situation occurs, the system will try to identify the taxiways only concerning gates. If a gate is found, the flight is considered to be a departure.

If a gate is not recognized, the system will try to recognize the aircraft in any taxiway. This situation means that the aircraft first position was detected once the flight was started and the flight mode cannot be recognized at this moment. If a flight has ended on a runway, the flight is considered to be a departure, but if it is finished on a stand or a gate, it is considered to be an arrival.
3.2. **Runway recognition**

The aim of this section is to determine if a position and a heading (or a set of positions) from ADS-B can be placed inside a modelled runway. In order to do that, several solutions were tested:

First solution was to calculate the runway’s corner coordinates and assess if the analyzed position was located inside the rectangle defined by the four runway corners. This method was disregarded because the runway direction could only be determined using the aircraft heading and the runway track parameters comparison. Figure 3.1 shows the detected zone using this method.

![Figure 3.1: Runway recognition using first method](image)

The method applied on the system is more sophisticated than the previous one and combines two analysis. The first analysis is to determine if an aircraft is located below a certain distance from the threshold coordinates\(^1\). In full-length departures this method allows to determine the runway and the runway-end used.

The second analysis on this method is based on the runway center line. The method will analyze if a position is contained on the runway center line by using the threshold position and the runway track. A tolerance angle (around 0.5\(^\circ\)) will be applied to detect aircraft slightly deviated from the runway center line. A restriction that the position must be between the two runway thresholds will be applied. This restriction protects the system from detecting aircraft taxiing aligned with the runway.

In case a runway is detected by the second analysis, the aircraft heading will be compared with the runway track to determine the used runway direction. This comparison allows to detect any aircraft that is crossing a runway but it is not operating in it.

Aircraft heading can be extracted from ADS-B messages as section 1.3.3.2. show. But experimentally it has been found that some aircraft does not broadcast this information\(^2\). When this situation occurs, heading is calculated as the mean track angle between 5 last reported positions.

This runway recognition method is useful on intersection departures and landings, where the aircraft does not reach any of the thresholds. Figure 3.2 shows detected zone using this method (angle no to scale).

In case a backtrack\(^3\) is needed, this method will recognize runways correctly. When back-

---

\(^1\)To avoid errors this distance should be equal or below than half runway width  
\(^2\)ADS-B surface position message bit 13=0  
\(^3\)Operation where a portion of runway is used for taxi
tracking before departure, runway will be detected by first analysis when turning back on the turning area. In case of a backtrack after landing, runway will be detected by second method before backtracking.

3.3. **Stand Recognition**

This section shows the method applied to determine if a position and a heading reported from an aircraft belongs to any of the modelled stands.

First solution applied to determine a stand was to compare the aircraft and the stand position, assuming a radius of tolerance. Figure 3.3 schematize this solution assuming a big radius over an AENA standard contact “Taxi-in, Push-out” perpendicular stand[30].

This solution, however, was disregarded because did not provide the desired reliability. The distance radius is a variable that can be adjusted to control the tolerance. A low radius offers no errors on recognition on consecutive stands but may lead to miss-recognition. Aircraft that stop at an earlier stop bar[27] or taxiing in low ADS-B coverage zones may
suffer from this issue. A high radius value could resolve this problem, but may result in wrong-recognition. Some positions could be fitted into the circumferences from several consecutive stands if the radius is big enough.

In order to correctly detect stands, a system with three checks was developed.

1. **Distance check:** Distance between the stand coordinates and the aircraft position is a key parameter to evaluate this situation. But, as it has been developed previously, an uncertainty circumference is not an efficient way to handle this issue. Here, a more complex tolerance region is developed based on a trigonometric oval. Equation 3.1 shows maximum tolerance distance allowed depending on the angle from the stand coordinates (0° corresponds to entry line):

\[
\text{Dist}_{\text{MAX}}(\text{ang}) = \begin{cases} 
R & \text{for } -90 < \text{ang} < 90 \\
R + |K \cdot R \cdot \cos(\text{ang})| & \text{for } 90 < \text{ang} < 270
\end{cases}
\] (3.1)

This equation leads to a maximum tolerance distance dependant on the angle around the stand coordinates. On the front part of the stand, distance follows a circumference, but on the back, distance depends on the angle, being the greatest distance over the center line at 180°. Parameters \(R\) and \(K\) allow determining the size of the trigonometric oval. Figure 3.4 shows this distance over the same AENA standard contact "Taxi-in, Push-out" stand.

![Figure 3.4: Stand recognition using a trigonometric oval](image)

With this check, system is able to determine not-overlapped stands positions. When some stands are overlapped, several stands may fulfill the distance check. When this occurs, system is not able to determine the correct stand. To discard wrong solutions additional checks involving stand position and angles can be performed as it is shown here.

2. **Entry angle check:** This check is based on comparing aircraft heading and the stand center line track.
With this this check, system is determining if the aircraft is following the same track as the stand center line. This check is useful to determine overlapped stands with different entry angles like Figure 3.5. This figure shows a Boeing 747-400 and a Bombardier CRJ-1000 with a close stop bar position but oblique stand entry lines.

![Overlapped stands with different entry angle](image)

Figure 3.5: Overlapped stands with different entry angle

On the figure it can be seen that, using only distance check, stand could not be determined because both stands fill the distance conditions. Using entry angle check, correct stand can be confirmed because aircraft heading will match one of the two possible stands track. A tolerance angle is accepted to this check (10° has been tested successfully on LEBL).

This check, however, becomes unusable when the two overlapped stands have the same entry angle.

3. **Center line angle check**: This check is based on comparing stand track with bearing from the aircraft position to the stand.

Relative bearing can be found using aircraft position and stand coordinates. If aircraft is aligned with the stand center line, this bearing will be equal than stand track. With this this check system is determining if the aircraft is aligned with the stand center line. This check is useful to determine overlapped stands with the same entry angles but parallel center lines like Figure 3.6. This figure shows a Boeing747-400 and a Bombardier CRJ-1000 with a close final position and same entry angle.

On the figure it can be seen that using only distance and entry angle check, the stand could not be determined. The two stands may be close enough not to be detected using distance check, and entry angle check would not resolve the uncertainty because the two entry lines are parallel. Using center line angle check, correct
stand can be confirmed because aircraft will be over just one of the two possible center lines. A smaller tolerance angle is accepted to this check (1° has been tested successfully on LEBL).

Even with this check there is still an ambiguous situation. When two stands are face-to-face, close enough and their center lines are aligned, this method would not be able to find the correct solution. This is a rare scenario in contact stands, but this situation can be found in remote aprons, specially those designed specifically for general aviation. An easy solution for that is forcing center line check to only work if aircraft is behind the aircraft stop bar. With this correction, an airport stand can be determined unambiguously from a set of aircraft positions and headings.

### 3.4. Taxiway recognition

This section shows the method applied to determine if a position and a heading reported from an aircraft belongs to any of the modelled taxiways.

The applied solution differs from those applied on previous sections. In previous section aim was to determine the position of the aircraft at any point of the runway or the stand, respectively. On this section, position will only be determined in a small portion of the sub-taxiway, were errors cannot be committed. This position is at halfway each taxiway segment (as it has been modelled on section 2.2.).

In this recognition task, it is preferable to skip a taxiway rather than recognize-it wrong. If a taxiway is not recognized and skipped it can be guessed later, but recognizing it wrong
would compromise the whole taxi route recognition as section 3.4.1. explains.

To recognize a taxiway unequivocally, position will be compared to the central point of the taxiway as it is modelled. Radius of uncertainty will be smaller than half taxiway width to avoid errors. Direction used on the taxiway will be determined when comparing taxiway track to aircraft heading. If this comparison leads to a non-valid result (more than 30° difference), taxiway will be omitted assuming that evaluated taxiway was a crossing one and not the taxied one.

3.4.1. Taxi Route

This section gets into detail on taxi route recognition. A taxi route is a set of ordered taxiways and their directions that create a coherent path. Being coherent mean that two consecutive recognized taxiways must be connected between them.

In order to create the coherence needed for the taxi route, only linking taxiways, gates, stands or runways will be evaluated. When a taxiway is recognized and heading evaluated, system comes with a list of links and headings that aircraft can take from its position. This links are evaluated when a new ADS-B surface position messages arrives.

As it has been mentioned on previous sections, ADS-B coverage from EETAC’s station has its limitations and some taxiways could be skipped. If this happen, system will need to be able to guess possible further taxi routes options and evaluate them.

Applied solution is based on an up to 5-segments guess. If system detects that none of the linking paths from the last taxiway can be considered correct, starts a 1-segment guess: system lists all the links form each of the last links (concerning heading) and evaluates them. If none of the solutions is valid, system starts a 2-segment guess using same schema. The method it is programmed to guess a maximum of 5 times to save resources and not to create redundant solutions. With this method, correct guessed solution will be the recognized one with less guessed segments. An example is shown in Figure 3.7 and Table 3.1 where R6 is last recognized taxiway.

![Figure 3.7: Standard taxi route in LEBL from runway 25R vacated via R6 to Gate ES](image)

This example is tabulated so it is easier to understand. First column shows the last recognized taxiway, which is R6. From there, three taxiway segments are available, and those appear on the LINK column (N9, N10 and ES).
Next column shows a 1-segment guess, which lead to the available taxiways in case one has been skipped. In this example it will be assumed that taxiway N10 was skipped, because plane taxied slightly on the right of the center line. If taxiway ES was recognized using the south direction, N10 taxiway could be recovered using this technique. In the table ES only appears once using 1-segment guess and matches the taxiway direction, so solution is unequivocal and taxi route would be R6 N10 ES.

As it can be seen on the table, ES appears in the 2-segments guess field using the route R6 FS M10 ES. But this solution is considered wrong for two reasons. First, the taxiway ES would be used in the opposite direction than the found one. Also, if there is a solution with less steps, this is the one that is considered due to simplicity.

In the case that N10 and ES were missed and GATE-ES was detected using south direction, two possibilities appear. Using the 2-segments guess, two routes can be found: R6 N10 ES GATE-ES and R6 FS M10 GATE-ES. Without knowing any ATC taxi preferences or airport restrictions, both solutions can be considered correct. To decide one, system will use the preferred taxi link as it is modelled on the taxiways file.

### 3.4.2. Taxi Time

To determine taxi speed, two elements are needed: taxi route distance and taxitime. EUROCONTROL defines taxitime as the time between off-block and take-off for departures and between landing and in-block for arrivals\(^2\). Since stand is difficult to determine in all occasions using the data from EETAC’s ADS-B station, in-block and off-block times are not always determined. In this studio taxitime will be considered as follows.

For departures taxitime will be considered from trespassing a gate (no push-back or pull-forward\(^4\) to consider since it is apron’s border) to a runway holding point.

For arrivals taxitime will be considered from runway vacation to a gate.

In an A-CDM environment delays are expected to be done on stand rather than on runway holding point. Pre-departure sequence and taxitime calculations help ATC to provide TSAT accurately so no delays occur on runway holding point. Sometimes this is not achieved and a queue is formed for departure in some taxiways (due to weather, RTO\(^5\), bad planing, LVP\(^6\)...). In this case taxitime will be calculated between the same points but the queue time will be subtracted.

As [8] states, in order to calculate a taxi velocity affected by some disturbances, data with no disturbances should be considered and then apply them. Since this is not possible, because aircraft interactions are too often in big airports, data with disturbances can be considered if those are known. In next chapter variables that affect taxi speed will be discussed and modelled, in order to be considered when analyzing and predicting taxi speed and taxitime.

---

\(^4\)As AIP notes, stands 95, 96, 184, 185 and 217 must use this procedure to avoid jet-blast on buildings

\(^5\)A Rejected Take-Off can force an aircraft to block a runway and have to be towed

\(^6\)In Low Visibility Procedures some restrictions apply. Most of them include a runway capacity reduction
<table>
<thead>
<tr>
<th>Last Position</th>
<th>LINK</th>
<th>1-segment guess</th>
<th>2-segments guess</th>
<th>3-segments guess</th>
</tr>
</thead>
<tbody>
<tr>
<td>R6 N9 N8</td>
<td>R6 N9 N8</td>
<td>R6 N9 N8 N7</td>
<td>R6 N9 N8 N7 N6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 N9 N8 N7 E5</td>
<td>R6 N9 N8 N7 E5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 N9 N8 N7</td>
<td>R6 N9 N8 N7 R6</td>
<td></td>
</tr>
<tr>
<td>R6 N9 N8 HS</td>
<td>R6 N9 N8 HS</td>
<td>R6 N9 N8 HS M7</td>
<td>R6 N9 N8 HS M8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 N9 N8 HS</td>
<td>R6 N9 N8 HS M8</td>
<td></td>
</tr>
<tr>
<td>R6 N9 N8 HS M7</td>
<td>R6 N9 N8 HS M7</td>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS GATE-HS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 N9 N8 HS GATE-HS</td>
<td>R6 N9 N8 HS GATE-HS</td>
<td></td>
</tr>
<tr>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS GATE-HS</td>
<td></td>
</tr>
<tr>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS GATE-HS</td>
<td></td>
</tr>
<tr>
<td>R6 N9 N8 HS M7</td>
<td>R6 N9 N8 HS M7</td>
<td>R6 N9 N8 HS M8</td>
<td>R6 N9 N8 HS GATE-HS</td>
<td></td>
</tr>
<tr>
<td>R6 N9 GS M8</td>
<td>R6 N9 GS M8</td>
<td>R6 N9 GS M8 M7</td>
<td>R6 N9 GS M8 M7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 N9 GS M8</td>
<td>R6 N9 GS M8 M7</td>
<td></td>
</tr>
<tr>
<td>R6 N9 GS M9</td>
<td>R6 N9 GS M9</td>
<td>R6 N9 GS M9 M10</td>
<td>R6 N9 GS M9 FS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 N9 GS M9</td>
<td>R6 N9 GS M9 GATE-FS</td>
<td></td>
</tr>
<tr>
<td>R6 N9 GS GATE-GS</td>
<td>R6 N9 GS GATE-GS</td>
<td>R6 N9 GS GATE-GS Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 N9 GS GATE-GS Stand</td>
<td></td>
<td>R6 N9 GS GATE-GS Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 N10 N11 DS</td>
<td>R6 N10 N11 DS</td>
<td>R6 N10 N11 DS GATE-DS</td>
<td>R6 N10 N11 DS GATE-DS</td>
<td></td>
</tr>
<tr>
<td>R6 N10 ES M10</td>
<td>R6 N10 ES M10</td>
<td>R6 N10 ES M10 M9</td>
<td>R6 N10 ES M10 FS</td>
<td></td>
</tr>
<tr>
<td>R6 N10 ES M10</td>
<td>R6 N10 ES M10</td>
<td>R6 N10 ES M10 GATE-FS</td>
<td>R6 N10 ES M10 GATE-FS</td>
<td></td>
</tr>
<tr>
<td>R6 N10 ES M11</td>
<td>R6 N10 ES M11</td>
<td>R6 N10 ES M11 M12</td>
<td>R6 N10 ES M11 DS</td>
<td></td>
</tr>
<tr>
<td>R6 N10 ES M11</td>
<td>R6 N10 ES M11</td>
<td>R6 N10 ES M11 GATE-DS</td>
<td>R6 N10 ES M11 GATE-DS</td>
<td></td>
</tr>
<tr>
<td>R6 N10 ES GATE-ES</td>
<td>R6 N10 ES GATE-ES</td>
<td>R6 N10 ES GATE-ES Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 N10 ES GATE-ES Stand</td>
<td></td>
<td>R6 N10 ES GATE-ES Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 FS M9</td>
<td>R6 FS M9 M8</td>
<td>R6 FS M9 M8 M7</td>
<td>R6 FS M9 M8 M7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 FS M9 M8</td>
<td>R6 FS M9 M8 M7</td>
<td></td>
</tr>
<tr>
<td>R6 FS M9 GS</td>
<td>R6 FS M9 GS</td>
<td>R6 FS M9 GS N8</td>
<td>R6 FS M9 GS N8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 FS M9 GS</td>
<td>R6 FS M9 GS N9</td>
<td></td>
</tr>
<tr>
<td>R6 FS M9 GATE-GS</td>
<td>R6 FS M9 GATE-GS</td>
<td>R6 FS M9 GATE-GS Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 FS M9 GATE-GS Stand</td>
<td></td>
<td>R6 FS M9 GATE-GS Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 FS M10 M11</td>
<td>R6 FS M10 M11</td>
<td>R6 FS M10 M11 M12</td>
<td>R6 FS M10 M11 M12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 FS M10 M11</td>
<td>R6 FS M10 M11 DS</td>
<td></td>
</tr>
<tr>
<td>R6 FS M10 M11</td>
<td>R6 FS M10 M11</td>
<td>R6 FS M10 M11 GATE-DS</td>
<td>R6 FS M10 M11 GATE-DS</td>
<td></td>
</tr>
<tr>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES N10</td>
<td>R6 FS M10 ES N11</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES N11</td>
<td></td>
</tr>
<tr>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES GATE-ES</td>
<td>R6 FS M10 ES GATE-ES</td>
<td></td>
</tr>
<tr>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES</td>
<td>R6 FS M10 ES GATE-ES Stand</td>
<td>R6 FS M10 ES GATE-ES Stand</td>
<td></td>
</tr>
<tr>
<td>R6 FS GATE-FS</td>
<td>R6 FS GATE-FS</td>
<td>R6 FS GATE-FS Stand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R6 FS GATE-FS Stand</td>
<td></td>
<td>R6 FS GATE-FS Stand</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: 3-taxiway segments guess table from R6 rapid exit
CHAPTER 4. TAXITIME CALCULATION

Several papers have been written regarding taxi time analysis and prediction. Each of them uses a different source of information. ATC system was used in [4], ACARS on [5] and SMR raw data on [6]. In this project, ADS-B is being used. ADS-B is a cheap, accurate and flexible source of information that allows to have information of all the flight phases.

Although the broadcast information in ADS-B it is very accurate in most cases, some aircraft are not equipped with this system (or does not provide the required accuracy) and information about those aircraft is lost. This is a drawback if compared with the SSR data that detects all aircraft equipped with a transponder aircraft, but on the other hand, ADS-B does not need to be calibrated and does not provide false alarms as the radar does.

This raw data (either ADS-B and SMR positions) has to be treated to provide good results. Several solutions were studied, and finally data was related to a modelled airport to identify taxi routes and its milestones. Solutions like an unscented Kalman filter[31] were studied but not implemented. This solution it is focused on recovering the taxi trace and making it continuous and derivable (position, velocity and heading against time) on every point of the route. This project is focused on relating the raw data with a modelled airport, and the filter did not provide any improvement on it.

Solutions based on ACARS are implemented in a slightly different way. ACARS provides timestamp for push-back start and lift-off very accurately, but does-not provide any positions. Only with this information an analysis is difficult to perform. Additional information is needed from ATC/SATR[3] to complete the information provided by ACARS.

Solutions based on ATC data require ANSP/Airport authorities collaboration. The authors requested radar data and A-CDM milestones from Barcelona-El Prat airport but and AENA refused to provide them. A detailed example on VTT and A-CDM integration is on section 4.6. on a real-time scenario.

4.1. Selection of Explanatory Variables

On each of the analyzed papers, different variables are studied and considered when creating a model. Older papers like [4](2001), [5](2011) and [6](2010) try to focus on determining taxitime for each point-to-point possibility connection in the airport. In the last paper, for example, an analysis is performed on Dallas-Fort Worth airport (KDFW) for a small group of stands, with the same aircraft mix and with only 8 airlines modelled. In this studio, two taxi routes are considered and the studio is done separately for each of them.

All of these papers relate taxitime only to the take-off queue, the number of taxing aircraft, the departure sequence changes, stand group and runway configuration. These studies require lots of data for aircraft with the same characteristics, and since distance is not considered, each runway configuration and stand group has to be treated separately.

More modern papers like [7](2013) and [8](2013) calculate taxi speed. In this way, the effect on distance is avoided and the method allows to calculate links between any part of the airport (the effects on some ramps or runways can be modelled also). Although distance may have an effect on taxi speed calculation, this can be modeled and compared
in taxi speed. When trying to relate distance with taxi-time, the resulting function can be linearized as [6] shows. This linearization does not take into account the distance and accuracy is not achieved (black line vs blue circles). Since distance is not considered, predictions using this method result on constant speed (red crosses), which means a straight line in a distance vs time diagram as Figure 4.1 shows.

![Figure 4.1: Taxi distance vs Taxitime in KDFW (real data and SFFS model)[6]](image)

In this study, taxi speeds will be modelled considering distance. Thus makes that distance can be used at the end of the process to get predicted taxi times.

From all the following papers, a list of factors that could effect on taxi speed has been made to check its relevance.

- **Taxi distance**: As it has been showed on the previous figure, distance has an effect on taxi speed. Paper [8] shows this element as the most influencing parameter in taxi speed calculation. In LEBL, taxi-out distance highly differs from Terminal 1 and Terminal 2 and is thought to have an important impact on taxi speed calculation. Taxi distance has been calculated as the sum of the length of the segments included on the taxi route field. Rounded turns have not been taking into account in distance calculation, because those are included in another field.

- **Aircraft size**: Although [4] showed that aircraft size is one of the less important factors in KBOS, MTOW modelling has been included especially to predict taxi speeds on smaller jets (like C650 or GLF5). Taking into account the LEBL aircraft mix, this item is thought to have an important effect.

- **Turn Angle**: Aircraft need to reduce speed when turning especially on those which are equal or greater than 90\(^\circ\) [32]. To model that, differences on modelled taxiway track are added to get the accumulated turn angle.
• **Airline:** The number of aircraft based on the studied airport per airline could have an impact on taxi speed since home pilots feel more conformable when taxiing. Paper [8] shows this parameter as one of the possible to explain the variability of their model. To model this behaviour, number of airline operated flights per day has been taken into account.

• **Airport Operations:** Paper [4] relates taxitime mainly on other aircraft operations and two parameters are discussed in it. Parameter N shows the number of taxing aircraft at the same time as the studied one and parameter Q shows the queue that a departing aircraft experiences. Parameter Q does not indicate the queue on holding point but the number of aircraft on the departure sequence that actually take-off between the aircraft taxi start time and the aircraft takeoff time. Depending on ATC, the aircraft included on Q may reflect a delay on the holding point, but if sequence is well managed Q does not imply a delay.

• **Unknown airport situational factors:** There are other factors that affect all aircraft that can be applied on the model. Situations like bad-weather or airport capacity reduction are time dependant. This situations are difficult to model, so all this factors will be modelled by considering the model taxi speed calculation error previous to the studied aircraft.

### 4.2. Variables Modeling

To train the model, a set of data will be analyzed and modelled using the previous variables. All variables will be modelled separating them between arrivals and departures, as [33] suggests. But unlike this paper does, here N parameter will not differentiate between the arrival and departure surrounding traffic. Due to LEBL layout, operational configurations and taxi preferred procedures, most of the aircraft will follow standard taxi routes and will not be interfered between them. In LEBL, this routes are composed mainly with one-way taxiways and no opposite route conflicts will be held\(^1\) \[^2\]. Some intersection conflicts may appear in crossings, but this have a lower impact on taxi speed and most times are caused by ATC to help defining the departure sequence.

On previous studies like the ones mentioned, variables are modelled using a linear regression. In this study a polynomial fitting will be used instead to improve accuracy. As some studies suggest[8], a base 10 logarithm approach could also help to fit a taxi speed diagram.

In this study each of the variables will be plotted and fitted against the taxi speed. Polynomial fitting help on reducing the standard deviation on fitting if data on the graphics are not linear dependent. Higher-order approximation allow more complicated-shapes with more local maxima and minima, which usually leads to a higher accuracy. But in this case, higher-order approximations have been proven unsuitable due to two reasons. The first reason is that a high-order polynomial fitting tends to ±infinity in a very abrupt way outside

---

\(^1\)Unless ATC authorizes a bypass

\(^2\)Opposite route conflicts may occur inside the ramp but will not be considered because ramp is not considered in this project’ taxitime definition
the training data range, and second reason is that approximation may have a very high local maximum/minimum inside the training range if data is separated enough. In both cases this leads to a bad approximation in some feasible points not considered on the data training set.

In figure 4.2 an example is shown. Figure compares a 3rd and a 9th order polynomial approximation on a taxi speed vs number of flights per airline plot. In this figure, red dots represent departing aircraft and blue dots arrival aircraft from the training data set. As it can be seen, if a value of 1400 flights per airline is entered on the method, two different results can be obtained: 9th order approximation will result over 1e7 m/s whereas 3rd order fit will result approximately to 10 m/s. This example reinforces the idea that a very high order approximation may not be the best solution. It is preferred to use a low order fit (between 1 to 4) to guarantee some continuity on the tendency in and out the training data set bounds.

Using this low-order polynomial approximations method, all the mentioned variables have been modelled using 2nd and 3rd order fits separating data between departure and arrival aircraft. With this approximations, the polynomial parameters that fit the expressions have been calculated for each variable. This means that a taxi speed can be found using the related polynomial constants for each parameter. The variables applied to the training data set are the following:

- Taxi Distance
- Aircraft MTOW
- Taxi route accumulated turn angle
- Mean daily flights per airline
- N (number of taxing aircraft at the same time)
- \(Q\) \(^3\) (number of previous aircraft on the departure sequence)
- Cross-product: \(\frac{Q}{\text{TAXI DISTANCE}}\) \(^3\) \(^4\)

\(^3\) Only applicable on departures

\(^4\) This variable is added to play down the effect on Q on long taxi distances. A high value of Q is not critical
4.3. Function Fitting

Using the modelling explained on previous section, 7 taxi speeds can be found for each flight. The idea of this project is to find a formula that is able to combine all the calculated taxi speeds, include some corrections and result on a predicted taxi speed.

Most of the referenced papers does not model the variables and include them directly to a formula. This way, in the formula each variable has a different value range and detecting dominant factors is difficult. Modelling the variables result on a better handling on extraordinary data (very high and very low taxi speed values are light-weighted on the model) and a better comprehension on dominant factors because all speeds have the same value range.

Using the same training data set, and taking into account the squared error between the observed taxi speed and the predicted by the model, several attempts were held to find a good formula approximation.

The adopted formula is the following:

$$\log_{10}(\text{pred}.\text{vel}) = (1 + K \cdot \text{error}[i - 1]) \cdot (A \cdot \log_{10}(\text{vel}_{\text{dist}}) + B \cdot \log_{10}(\text{vel}_{\text{MTOW}})
+C \cdot \log_{10}(\text{vel}_{\text{angle}}) + D \cdot \log_{10}(\text{vel}_{\#\text{lights}}) + E \cdot \log_{10}(\text{vel}_{\text{N}}) + F \cdot \log_{10}(\text{vel}_{\text{Q}})
+ G \cdot \log_{10}(\text{vel}_{\text{Qdist}}))$$ (4.1)

This formula has two different parts. First part corresponds to the airport situation correction and second part corresponds to taxi speed calculation. In both parts several constants are included, whose value have to be determined. Determining the best value of each of the constants will lead on the most accurate formula using this schema.

First part represents the airport situation correction and tries to deviate the predicted taxi speed taking into account the committed error on the last aircraft. If the model, for some reason, predicted a lower taxi speed than the observed one, may mean that some disturbance has occurred on the airport. On next aircraft system should predict a higher value to correct the deviation and vice-versa. This part has a constant K that relates the deviation that an aircraft has to receive due to the last error. To fulfill this, K should be negative.

Second part of the formula represents the taxi speed calculation. As it can be seen on the equation, the formula represents a weighted average of taxi speeds, each of them calculated taking into account the polynomial fitting parameters found on previous steps. Weight of each parameter is represented by the constants that multiply the tenth logarithm of each taxi speed.

The challenge now is to find the best value of the constants A-G and K, such that best fitting is obtained. Since this formula is non-linear and multi-variable, an optimization algorithm is needed. Broyden-Fletcher-Goldfarb-Shanno \(^5\) \(^6\) algorithm is an iterative method that allows to optimize unconstrained, non-linear, multi-variable problems with an unknown

---

5. https://en.wikipedia.org/wiki/Broyden%E2%80%93Fletcher%E2%80%93Goldfarb%E2%80%93Shanno_algorithm

---

if taxitime is high and aircraft have enough time to depart. This would not create a bottleneck on runway holding point that could create a delay and a taxitime increase
Jacobian. This method will be used to optimize the squared error between the observed and predicted taxi speed on the training data set.

This method is considered quasi-Newtonian and it is useful on this situation because no Jacobian and initial value constraints are known. But like all Newtonian methods, its convergence is not guaranteed on all cases. This is the reason not to expand the optimization function. Several attempts were held to improve accuracy, like adding a first and second derivative discretization on error or adding powers on some of the variables, but all of them failed on converging.

4.4. Results evaluation

On previous sections a training data set has been used to calculate polynomial fitting constants and function constants. Using this calculated values, taxi speeds can be predicted with a new evaluating data set. On this project, training data set corresponds to March 2017 and evaluating data set to the period of April 2017-August 2017.

Applying the proposed method with the found constants from the training data set to the evaluating data set leads to the following results: 73% of the departing aircraft and 97% of the arrival aircraft are predicted with the 2-minutes margin required by EUROCONTROL[2] on an A-CDM airport in short time. With this predictions, mean absolute error for departures is 104s and for arrivals 31s, which in both cases is less than the 120s required by EUROCONTROL.

Using this data set, the found function constants are the following:

<table>
<thead>
<tr>
<th>MODE</th>
<th>K</th>
<th>Dist</th>
<th>MTOW</th>
<th>#flights</th>
<th>N</th>
<th>Q</th>
<th>t.angle</th>
<th>Q/dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dep</td>
<td>-0.045</td>
<td>1.283</td>
<td>0.637</td>
<td>0.532</td>
<td>0.820</td>
<td>-1.561</td>
<td>-0.700</td>
<td>-0.004</td>
</tr>
<tr>
<td>Arr</td>
<td>-0.023</td>
<td>0.596</td>
<td>0.979</td>
<td>0.993</td>
<td>-2.163</td>
<td>0</td>
<td>0.594</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.1: Function constants found on March 2017 training data set

With the found parameters on Table 4.1 some conclusions can be extracted.
First conclusion is that K value is negative in both modes as it is expected. This makes the airport situation correction make its function because it compensates the last error.

Second conclusion that can be extracted are the most influencing parameters, which in this case are distance, MTOW and N. This parameters are the ones with a higher quadratic value (values on Table 4.1 squared).

Figure 4.3 plots the observed speed vs the error committed on the formula where red dots represent departures and blue dots represent arrivals. As it can be seen, maximum error committed by the system is held on departures with low observed speeds.

This error can be assumed by multiple factors: Q parameter does not properly model the holding point queue and un-modelled variables.

A possible explanation for this error is that Q parameter (and Q/dist) does not reflect the airport behaviour that wanted to model and were included for. Although subtracting the

---

[2] Short time is considered from 30 minutes before off-block time, where departure runway and actual stand is already known.
stopped-time from the taxitime when calculating Q, deceleration and acceleration time are not considered and imply a decrease on average taxi speed. This parameter does not model the holding point queue, but the taxing aircraft in front of the studied airport. A holding point aircraft queue was thought unnecessary in an A-CDM airport because delays are expected to be done on stand\cite{2}. As it has been seen, in this airport this is not always accomplished. Especially when some disruption occurs, airport predicted taxitime (based on historical tables) may differ from the observed ones or airport capacity is reduced and queues are formed on holding points\footnote{As AENA Operations Division in LEBL confirmed to the author}.

Errors on the model that cannot be related with the Q parameter or the fitting accuracy loss can be assumed to be variables that are not considered on this project. In this project only some of the variables that could be calculated have been considered, and this is related to the available data source. With the provided data, and contrasting with other author’s work, the mentioned variables are the ones that have been found most suitable to perform this study.

4.5. Testing under extraordinary conditions

During the realization of this project, some extraordinary conditions took place on LEBL. From January 10th 2017 to February 5th 2017 a runway closure took place. Runway 07L/25R and taxiways R1, R5 and R6 were closed to regenerate the paving\footnote{AIP-ESPAÑA SUP99/16 AIRAC27-OCT-16}. During this period, airport was operating with 2 crossed runways or 1 mixed-mode runway depending on the configuration.

Taxi procedures during this period also changed, and mean taxi distance was considerable increased because some stubs and bypasses were not available. Also, since
most procedures changed, mean taxi velocity was reduced from 8m/s (December 2016) to 6.8m/s (runway closure).

To prove the reliability on the model, taxitime on this period have been tried to predict. Using December 2016 as the training data set and the runway closure period as the evaluated data set, the 2-minutes accuracy was 58% for departures and 87% for arrivals. With this sets of data, mean absolute error increase and prediction is not valid for this conditions.

After that another analysis was performed. With only the first 2-days of the closure as training data set, the whole period was tired to predict. With this data, the 2-minutes accuracy was 67% for departures and 94% for arrivals.

With this tests it has been proven the usability of the method: method is not able to predict taxitime in future changing conditions, but is very effective on predicting taxitime from a small set of data with the same conditions. This ability is useful especially on new conditions (like the ones on LEBL during runway closure), where taxitime calculations based on historical point-to-point analysis is not possible due to the lack of data.

### 4.6. Future development

The main future development of the system is the integration with real-time operations. Real time operations allow to use machine learning when new data enters to the system. Recalculation of the constants and polynomial fittings parameters including time-weighted data can be included to detect real-time disturbances.

Although the method in this project is developed to be causal (no future information is used at any time) some aids will be needed to keep this in real-time. Since LEBL is an A-CDM airport, integration with both systems will be essential to get some information needed for the computations.

Figure 4.4 shows a screen shot of the AENA e-CDM system. In this screen shot, the flight departure list can be seen and some useful items appear like stand, departure runway, EXOT, AOBT and TOT. Disadvantages of this system include that no-taxi route is provided and accuracy on the data are up to 1 minute.

![Figure 4.4: AENA e-CDM system screen shot](image)

---

---
CONCLUSIONS

This project shows a method to calculate taxitime in Barcelona-El Prat Airport. To do that, ADS-B data was decoded and analyzed comparing them with a modelled airport. With this information a set of data was used to train the model, and another to evaluate it.

Training data set allowed to determine the main factors that affect taxi speed and a modelling based on a polynomial fitting was implemented to all explanatory variables. These variables included taxi route parameters like distance or turn angle, airport traffic situational parameters like N, Q or Q/dist, aircraft parameters like MTOW or airline parameters like the number of daily flights per airline.

With these modelled variables, a logarithmic sum of polynomial fittings of them and an airport situation corrector has been applied to obtain a predicted taxi speed. To better fit the observed taxi speed on the training data set, function constants have to be optimized. To do so, Broyden-Fletcher-Goldfarb-Shanno algorithm was used.

Variables modelling and formula constants found with the training data set, were applied to the evaluation data set to assess the predictability of the model. The model resulted to have the desired accuracy (120s) on 73% on departures and 97% on arrivals.

But for some applications, the mentioned accuracy could not be enough. The method has proven to consider the main explanatory variables, but there is still a variability that is not considered. To improve this results, more variables could be implemented, and to do so, an additional data source should be used.

A more sophisticated corrector for airport situation in the formula could also help on increasing accuracy. Including discretizations on first and second derivative on last error and including a weighted-mean of last errors could result effective. But all these implementations add additional constants on the formula and optimization method may not converge.

To include more parameters on the formula, an alternative optimization method should be used. But in order to use a non-quasi-newtonian one (to ensure convergence) some assumptions would be needed regarding jacobians and inital values.

This accuracy can be reached also using conventional methods like static tables. But the main advantage of using this model is the capacity to adapt to new conditions. In this project an example has been analyzed where the method was used to predict taxi speeds on a runway closure scenario in LEBL during January and February 2017. This example showed that if an earlier set of data is used for training, method is not able to predict taxi speeds due to the different characteristics that apply. But if a small training data set from the beginning of the disturbance periods is used, accuracy on prediction nearly reaches the one obtained without disturbances.

Conventional methods require large amounts of data with the same characteristics to predict taxi times, and this makes them unusable when conditions suddenly change. Proposed method is able to predict taxitime in new conditions with a small set of training data because it works with speeds, relativizes some variables and parameters are adaptive.


40 Taxi time analysis and prediction with ADS-B data. A case study in Barcelona-El Prat Airport


