

**Open Aircraft Performance Modeling
Based on an Analysis of Aircraft Surveillance Data**

Sun, Junzi

DOI

[10.4233/uuid:af94d535-1853-4a6c-8b3f-77c98a52346a](https://doi.org/10.4233/uuid:af94d535-1853-4a6c-8b3f-77c98a52346a)

Publication date

2019

Document Version

Final published version

Citation (APA)

Sun, J. (2019). *Open Aircraft Performance Modeling: Based on an Analysis of Aircraft Surveillance Data*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:af94d535-1853-4a6c-8b3f-77c98a52346a>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Open Aircraft Performance Modeling

Based on an Analysis of Aircraft Surveillance Data

Open Aircraft Performance Modeling

Based on an Analysis of Aircraft Surveillance Data

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen;
Chair of the Board for Doctorates
to be defended publicly on
Friday, 14 June 2019 at 15:00 o'clock

by

Junzi SUN

Master of Science in Aerospace Science and Technology,
Polytechnic University of Catalonia, Spain
born in Chaohu, Anhui Province, China

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof.dr.ir. J.M. Hoekstra	Delft University of Technology, promotor
Dr.ir. J. Ellerbroek	Delft University of Technology, copromotor

Independent members:

Prof.dr. V. Grewe	Delft University of Technology
Prof.dr. V.N. Duong	Nanyang Technological University, Singapore
Prof.dr. D. Delahaye	Ecole Nationale de l'Aviation Civile, France
Dr. X. Prats i Menendez	Universitat Politècnica de Catalunya, Spain
Dr. X. Olive	Office National d'Etudes et de Recherches Aérospatiales, France

Reserve member:

Prof.dr.ir. M. Mulder	Delft University of Technology
-----------------------	--------------------------------



Keywords: Aircraft Performance, Air Traffic Management, ADS-B, Drag Polar, Dynamic Model, Engine Fuel Flow, Kinematic Model, Meteo-Particle, Mode-S, Open Data, State Estimation, Thrust

Printed by: Offpage.nl

Copyright © 2019 by J. Sun

ISBN: 978-94-6384-030-9

An electronic version of this dissertation is available at
<http://repository.tudelft.nl/>

This book is dedicated to my parents, wife, and son
谨以此书献给我的父母、妻子和儿子

Preface

In the autumn of 2014, I was searching for a Ph.D. topic that would combine my background in aerospace and computer science. The keywords of *open*, *data mining*, and *aircraft* caught my eye in a proposal from the TU Delft, and I immediately decided to submit an application. On a chilly spring morning half a year later, I joined the TU Delft's Aerospace Engineering faculty. I was warmly welcomed by Prof. Hoekstra and Dr. Ellerbroek and introduced to the department, colleagues, researches, and the BlueSky project. My final stop of the first day - the faculty's *De Atmosfeer* bar - certainly confirmed I had made the right choice.

This dissertation records my progress and findings over the past four years. It essentially answers one question: *how can we use open data to model and estimate aircraft performance?* Most of the chapters are based on my journal articles and conference proceedings published since 2016. The primary goal of my Ph.D. research has been to make an open aircraft performance model. As such, the models and tools produced in this dissertation are shared, and I am proud that some of these open-source tools have already been adopted by other researchers.

This four-year journey seems long at the start but certainly feels much shorter now that it is approaching the finishing line. I would like to thank my promoters, Prof. Hoekstra and Dr. Ellerbroek, who have been extremely supportive and given valuable guidance. I would like to thank Prof. Blom and Ir. Vû, who provided great ideas and co-authored some of the chapters of this dissertation, as well as my Ph.D. committee members for their helpful comments and suggestions on the dissertation. I would also like to extend my thanks to all my colleagues from the Department of Control and Simulation, especially for all the inspiring philosophical conversations we shared at the coffee corner. A final thanks to my wife, Marie, who has spent many hours proofreading and improving the stylistics of my papers and this dissertation. In the end, I feel that she may have secretly mastered all this ADS-B stuff.

On a personal note, I am deeply grateful for my parents' love, support, and encouragement of my pursuit of science since I was a young boy. I have also been extremely blessed to have fallen in love with and married Marie, as well as welcomed my son William to the world, during my doctoral studies.

Junzi Sun
Delft, May 2019

Contents

Preface	vii
Summary	xvii
Samenvatting (Summary in Dutch)	xix
1 Introduction	1
1.1 Background	1
1.2 Data	3
1.3 Models	5
1.3.1 Kinematic model	5
1.3.2 Dynamic model	6
1.3.3 Weather	6
1.4 Estimations	7
1.5 Research questions	8
1.6 Structure	8
1.6.1 Part One: Data	8
1.6.2 Part Two: Models	9
1.6.3 Part Three: Estimations	10
1.6.4 Conclusion	10
I Data	11
2 Mode-S data decoding	13
2.1 Introduction	14
2.2 Background	16
2.2.1 Fundamentals of Mode-S, ADS-B, and Comm-B	16
2.2.2 Regulation and availability	17
2.2.3 Data structure	17
2.3 Basic decoding of ADS-B data	19
2.4 Interference of the Comm-B downlink parameters	20
2.4.1 Source aircraft identification and error detection	21
2.4.2 BDS inference using the Heuristic-Probabilistic method	24

2.5	Experiment and validation	27
2.5.1	Experiment I: Examining the heuristic logic	28
2.5.2	Experiment II: Examining the identification accuracy	28
2.5.3	Experiment III: Analyzing the global data	29
2.6	Discussion	31
2.7	Conclusion	32
2.8	Acknowledgment	33
3	Flight trajectory processing	35
3.1	Introduction	36
3.2	Flight extraction from large ADS-B datasets	36
3.2.1	Data fields	36
3.2.2	Simple flight extraction	37
3.2.3	Machine learning flight extraction	37
3.3	Flight phase identification	39
3.3.1	Flight phase identification process	39
3.3.2	Membership functions	40
3.3.3	Identification logic	42
3.4	Verification of the algorithms	44
3.4.1	Benchmarking of the clustering methods	44
3.4.2	Examination of flight phases	44
3.5	Discussion	46
3.5.1	Data storage	46
3.5.2	Tuning of clustering algorithms	46
3.5.3	Flight phase identification limitations	46
3.6	Conclusion	46
II	Models	49
4	Meteorological model	51
4.1	Introduction	52
4.2	Meteorological observations from Mode-S data	54
4.2.1	Accurate models of temperature and wind speed	55
4.3	The Meteo-Particle model	57
4.3.1	Assumptions	57
4.3.2	Measurements and probabilistic rejection	58
4.3.3	Particles	59
4.3.4	Particle propagation	59
4.3.5	Probabilistic re-sampling	61
4.3.6	Information reconstruction	61
4.3.7	Representation of confidence level	63
4.4	Short-term prediction with the Meteo-Particle model	65

4.5	Experiments and validations	67
4.5.1	Construction of sample wind and temperature fields	67
4.5.2	Validation of MP model with NWP data	69
4.5.3	MP model estimation accuracy	72
4.5.4	Short-term prediction accuracy	74
4.6	Uncertainty analysis	75
4.6.1	Model uncertainty	75
4.6.2	Data uncertainty	76
4.6.3	Error tolerance	77
4.7	Discussion	79
4.8	Conclusion	80
5	Kinematic performance model	83
5.1	Introduction	84
5.2	The WRAP model	85
5.2.1	Model parameters	85
5.2.2	Parametric statistical models and selection	86
5.2.3	Interpretation of the model	89
5.3	The data	90
5.3.1	Data source	90
5.3.2	Trajectory processing	90
5.3.3	Atmospheric conditions and speed conversions	90
5.4	Construction of WRAP parameters	92
5.4.1	Takeoff	92
5.4.2	Initial climb	93
5.4.3	Climb	93
5.4.4	Cruise	96
5.4.5	Descent	96
5.4.6	Final Approach	97
5.4.7	Landing	98
5.5	Results	98
5.5.1	An Airbus A320 example	99
5.5.2	Correlations among parameters	102
5.5.3	Summary based on common aircraft types	105
5.6	Comparison with existing models	105
5.7	Discussion	107
5.7.1	Geo-correlation	107
5.7.2	Point of lift-off and touch-down	107
5.7.3	Speed restrictions	108
5.7.4	Cruise altitude	109
5.7.5	Continuous descents	109
5.7.6	Data and model use cases	110

5.8	Conclusion	111
5.9	Acknowledgment	111
6	Thrust and fuel flow model	113
6.1	Introduction	114
6.2	Model thrust of turbofan engines	116
6.2.1	Takeoff thrust at sea level	116
6.2.2	Takeoff thrust with the influence of altitude	116
6.2.3	Maximum thrust for en-route flights	117
6.2.4	Reference cruise thrust	118
6.2.5	Summary	119
6.3	Model engine fuel flow	119
6.4	Discussion	121
6.4.1	Evaluation of thrust model coefficients	121
6.4.2	Coefficients of engine performance	122
6.4.3	Comparison of OpenAP and BADA	122
6.4.4	Comparison of OpenAP, BADA, and flight data	124
6.5	Conclusion	126
7	Drag polar model	129
7.1	Introduction	130
7.2	Theory of point-mass aerodynamic model	131
7.2.1	Drag polar in point-mass models	131
7.2.2	Analytical model for lift dependent drag coefficient	132
7.2.3	Different aircraft aerodynamic configurations	133
7.2.4	Wave drag at high Mach numbers	135
7.2.5	Summary	136
7.3	Estimation of drag polar under clean configuration	136
7.3.1	The hierarchical stochastic model	137
7.3.2	Bayesian computing - Markov Chain Monte Carlo	140
7.3.3	The posterior estimates	141
7.4	Experiment	141
7.4.1	Estimation of the drag polar using STE model	142
7.4.2	Multiple aircraft types	142
7.4.3	A drag polar using CFD simulation	143
7.4.4	Compressibility effect on drag polar	144
7.5	Summary and discussion	145
7.5.1	Drag polar for different aircraft types	145
7.5.2	Uncertainties	147
7.5.3	Comparison with BADA drag polar	148
7.6	Conclusion	149

III	Estimations	151
8	Turn performance estimation	153
8.1	Introduction	154
8.2	Coordinated turns	154
8.3	Turn trajectory extraction	156
8.4	Turn performance estimation	157
8.5	Experiments and results	161
8.5.1	Experiment and validation data	161
8.5.2	Estimation results	161
8.5.3	Estimation errors	163
8.5.4	Correlation between speed and bank angle	164
8.6	Discussion	164
8.6.1	Mode-S track and turn report	164
8.6.2	Least-square regression	164
8.6.3	Turn under heavy wind conditions	165
8.6.4	Aircraft mass and turn performance	165
8.7	Conclusion	166
9	Mass estimation using Bayesian inference	169
9.1	Introduction	170
9.2	Calculation of mass at different flight phases	171
9.2.1	Computing mass at takeoff	172
9.2.2	Computing mass at liftoff and final approach	173
9.2.3	Computing mass at climb and descent	174
9.2.4	Computing fuel consumption	176
9.3	Bayesian Inference	176
9.4	Experiments and results	177
9.4.1	Single flight demonstrations	178
9.4.2	Distributions for A320 and B737-800	178
9.4.3	Sensitivity analysis	179
9.5	Demonstration with Cessna Citation flight data	183
9.6	Discussion	184
9.6.1	Priors in Bayesian inference	185
9.6.2	The mass observations	185
9.6.3	Uncertainties	185
9.7	Conclusion	186
10	Mass estimation using particle filtering	187
10.1	Introduction	188
10.2	The point-mass aircraft dynamic model	189
10.2.1	Aircraft state	189
10.2.2	State evolution	190

10.3 Recursive Bayesian estimation	192
10.3.1 Particle filtering	193
10.3.2 Application for the aircraft mass and thrust estimation . . .	195
10.4 Observation and process noise models	196
10.4.1 The observation noise models	196
10.4.2 Process equations for vertical rate, wind, and temperature .	197
10.5 The algorithm	199
10.6 Experiments	200
10.6.1 Experiment I: The simulation of a Boeing B737 climb	200
10.6.2 Experiment II: A real Boeing B737 flight	201
10.6.3 Experiment III: Demonstration with Cessna C550 flights . .	202
10.7 Discussion	204
10.7.1 Estimation uncertainty	204
10.7.2 Choice of observation noise model	207
10.7.3 Influence of variation in mass and thrust setting	207
10.8 Conclusion	210
11 Conclusions and recommendations	213
11.1 Overview	214
11.2 Research questions and conclusions	214
11.3 Limitations and recommendations	218
11.3.1 Mode-S	218
11.3.2 Citation flights	219
11.3.3 Thrust model	220
11.3.4 Non-linear system	220
11.4 Final conclusions	221
Appendix	223
A The OpenAP Python Library	225
A.1 Basics of the OpenAP	226
A.2 Aircraft properties	226
A.3 Kinematic performance	228
A.4 Dynamic performance	229
A.4.1 Aircraft dynamic model	230
A.4.2 Thrust	231
A.4.3 Drag	231
A.4.4 Fuel flow	232
A.5 Other databases and utility libraries	232
A.5.1 Flight phase	232
A.5.2 Aeronautical calculations	233
A.5.3 Navigation database	233

A.6	Discussion	233
A.7	Conclusion	234
B	Drag polar calculation using CFD simulations	235
B.1	Introduction	236
B.2	3D model	236
B.3	Pre-processing of 3D models	237
B.4	Meshing	239
B.5	Navier-Stoke solver	241
B.6	Computation of the drag polar	242
C	Aircraft engine performance	243
	References	247
	Curriculum Vitae	261
	List of research outputs	263

Summary

Given the large number of stakeholders in the modern air traffic management ecosystem, air transportation studies benefit from collaboration and the sharing of knowledge and findings between these different players. However, not all parties have equal access to information. Due to the lack of open-source tools and models, it is not always possible to undertake comparative studies and to repeat experiments. The barriers to accessing proprietary tools and models create major limitations in the field of air traffic management research.

This dissertation investigates the methods necessary to construct an aircraft performance model based on open data, which can be used freely and redistributed without restrictions. The primary data source presented in this dissertation is aircraft surveillance data that can be intercepted openly with little to no restriction in most regions of the world. The aircraft performance model addressed in this dissertation consists of kinematics, thrust, drag polar, fuel flow, and weather components. The eleven chapters in this dissertation follow the sequence of open data, open models, and performance estimations. This order corresponds to the three main parts of the dissertation.

In the first part of the dissertation, open surveillance data is investigated. In Chapter 2, the focus is on how to decode hidden information contained in the secondary surveillance data. The algorithms designed in this chapter can effectively infer the message types and decode the messages. In addition to automatically broadcast data, secondary surveillance messages can greatly increase the amount of information regarding aircraft states. In Chapter 3, the main goal is to use machine learning algorithms to construct and segment flights. Based on previously obtained surveillance data, a clustering algorithm is used to efficiently extract continuous flights from the scattered flight data. Then, a fuzzy logic based identification algorithm is designed to segment a flight into different flight phases. These segmented flights become one of the foundations for later performance analysis and modeling.

The chapters in the second part examine the main components of the open aircraft performance model. First, a novel weather model, the Meteo-Particle model, is proposed in Chapter 4. This model is able to utilize the continuous stream of surveillance data to construct accurate wind and temperature fields. It can be used for both offline weather reconstruction and real-time local weather assimilation. With improved knowledge of wind speed, the accuracy of aircraft performance studies can be further improved. In Chapter 5, the kinematic performance of different aircraft types at each flight phase is modeled. A large quantity of flight data collected through a crowd-sourced receiver network is used to construct models for these parameters. Kinematic parameters, such as distance, speed, vertical rate, and acceleration are modeled for the flight phases of takeoff, initial climb, climb, cruise, descent, final approach, and landing respectively. At the end of this chap-

ter, a comprehensive open kinematic database, WRAP, is produced and shared. In Chapter 6, engine thrust and fuel flow are investigated. Based on an open thrust model from the literature, maximum thrust during the flight can be computed for common passenger aircraft types, as long as basic engine performance parameters are known. In this chapter, a fuel consumption model is also proposed. The fuel model is based on the open engine emission data-bank information from the International Civil Aviation Organization. Thanks to the excellent coverage of engine types in this dataset, fuel flow for almost all common turbofan engines can be modeled. In Chapter 7, one of the most important aircraft performance components, the drag polar, is studied. Accurate flight data is combined with the Markov chain Monte Carlo Bayesian method in order to derive drag polar coefficients. The study in this chapter results in a fully open drag polar models that can be applied to common transportation aircraft types.

The third part of the dissertation looks into the possibility of using open surveillance data to estimate aircraft performance parameters of individual flights. In Chapter 8, performance parameters related to aircraft turns are estimated. Using broadcast surveillance data, the radius of turn is first inferred. Then, the bank angle and load factor can be derived. The accuracy of this estimation is validated using the independent reference information from the secondary surveillance communication. In Chapter 9 and 10, two methods are designed to estimate the mass of an aircraft. One of the main differences between these two methods involves the flight stages used for estimation. In Chapter 9, the entire trajectory is required. Aircraft masses are first computed at different stages of a flight. Then, considering the empirical knowledge of the mass as a priori, the initial mass is produced as the maximum a posteriori estimation. Chapter 10 makes use of a short flight segment at the start of the climb and applies a Bayesian estimator to the flight dynamics level. A particle filter is constructed to estimate aircraft states using observations from surveillance data. These states include position, speed, vertical rate, wind, temperature, mass, and thrust setting. The particle filter is able to infer not only the mass but also to determine the uncertainty of estimates. The study indicates the level of data quality needed to produce a meaningful estimation. This represents one of the first studies in the air traffic management domain that quantifies the uncertainty of mass estimation caused by surveillance data accuracy.

In the final chapter of this dissertation, final conclusions and future recommendations are made. With the goal of making future air traffic management studies more transparent, comparable, and reproducible, the models and source code proposed in this dissertation are fully open, including, for example, the py-ModeS, OpenAP, Meteo-Particle, and particle filter libraries. A major part of the open performance model has already been implemented in the BlueSky air traffic simulator. The performance model proposed in this dissertation has proven to be an efficient open-source alternative to current closed-source models. It also has the potential to be used in other studies, for example, those investigating trajectory optimization and prediction. These areas of investigation could be considered as interesting paths to extend this doctoral research.

Samenvatting

Het huidige ecosysteem voor luchtverkeer kent vele belanghebbenden. Onderzoek gerelateerd aan luchtverkeer profiteert van open samenwerking tussen wetenschappers. Er is echter geen sprake van gelijke toegang tot informatie. Door een gebrek aan vrij toegankelijke tools en modellen is het niet altijd mogelijk om vergelijkende studies en herhaalbare experimenten uit te voeren. De beperkte toegang tot commerciële tools en modellen hindert luchtverkeersonderzoek.

Dit proefschrift beschrijft methoden voor het genereren van vliegtuigprestatie modellen op basis van open data, die vrij kunnen worden gebruikt en zonder beperkingen kunnen worden gedistribueerd. De voornaamste bron van data die in dit proefschrift wordt gebruikt, komt voort uit vluchtgegevens die periodiek door moderne vliegtuigen worden verstuurd. Deze informatie kan openlijk worden onderschept met weinig tot geen beperkingen in de meeste delen van de wereld. Het in dit proefschrift gepresenteerde vliegtuigprestatie model beschrijft kinematica, stuwkracht, aerodynamische eigenschappen, brandstofverbruik en weersomstandigheden. De elf hoofdstukken in dit proefschrift zijn gerangschikt in drie onderdelen; open data, open modellen en prestatie model schattingen.

In het eerste deel van het proefschrift worden de toepassingen van open vliegtuiggegevens onderzocht. In hoofdstuk 2 ligt de nadruk op het decoderen van informatie uit antwoorden van vliegtuigen op verzoeken van de secundaire radar. De gepresenteerde algoritmen kunnen de berichttypen effectief afleiden en de berichten decoderen. Met deze gegevensbron kan de hoeveelheid informatie over de vluchtstoestand van vliegtuigen aanzienlijk worden vergroot. In hoofdstuk 3 worden verschillende algoritmen gebruikt om vluchten en vluchtsegmenten te reconstrueren uit ongesorteerde data. Met een clusteralgoritme worden continue vluchten efficiënt gedetecteerd in de verzamelde vluchtgegevens. Met een fuzzy logic algoritme worden de vluchten in verschillende vluchtfasen verdeeld. Deze gegroepede vluchtdata vormt een van de fundamenten voor latere analyse en modellering van de vliegtuigeigenschappen.

De hoofdstukken in het tweede deel onderzoeken de belangrijkste componenten van het open vliegtuigmodel. Allereerst wordt in hoofdstuk 4 een nieuw weermodel voorgesteld. Dit model kan de continue stroom aan vliegtuigdata gebruiken om nauwkeurige wind- en temperatuurvelden te construeren, voor zowel offline weerreconstructie als realtime lokale weerschatting. Met dit weermodel kan de nauwkeurigheid van vliegtuigprestatie modellen verder worden verbeterd. In Hoofdstuk 5 worden de kinematische prestaties van verschillende vliegtuigtypen op elke vluchtfase gemodelleerd. Een grote hoeveelheid vluchtgegevens die zijn verzameld via een openbaar ontvangernetwerk wordt hier gebruikt om modellen voor deze parameters te schatten. Kinematische parameters zoals afstand, snelheid, verticale snelheid en versnelling zijn gemodelleerd voor respectievelijk de vluchtsegmenten start, initiële klim, klim, kruisvlucht, afdaling, eindnadering en landing. Het resultaat is een uitgebreide open kinematische database. In hoofd-

stuk 6 worden de stuwkracht van de motor en het brandstofverbruik bestudeerd. Op basis van een open stuwkrachtmodel uit de literatuur kan de maximale stuwkracht tijdens de vlucht worden berekend voor gewone typen passagiersvliegtuigen, op voorwaarde dat de basisparameters van de motorprestaties bekend zijn. In dit hoofdstuk wordt ook een brandstofverbruikmodel voorgesteld. Het brandstofmodel is gebaseerd op de open-engine-emissie-gegevensbankinformatie van de International Civil Aviation Organization. Dankzij de uitstekende dekking van motortypen in deze dataset kan het brandstofverbruik voor bijna alle gangbare turbofanmotoren worden gemodelleerd. In Hoofdstuk 7 wordt een van de belangrijkste onderdelen van de vliegtuigprestaties, de 'drag polar', bestudeerd. De vluchtgegevens worden hier gecombineerd met een Markov-chain Monte Carlo Bayesiaanse methode om drag-polarcoëfficiënten af te leiden. Het resultaat is de eerste vrij beschikbare set drag-polar-modellen voor gewone typen transportvliegtuigen.

Het derde deel van dit proefschrift onderzoekt de mogelijkheid om open surveillancegegevens te gebruiken om prestatieparameters van individuele vluchten te schatten. In Hoofdstuk 8 worden de prestatieparameters gerelateerd aan bochten geschat. Met behulp van de periodiek verstuurde vluchtgegevens kan eerst de radius van de bocht worden afgeleid. Vervolgens kunnen de rolhoek en de belastingsfactor worden afgeleid. De nauwkeurigheid van deze schatting is gevalideerd met behulp van de secundaire surveillancecommunicatie als onafhankelijke referentie. In hoofdstuk 9 en 10 zijn twee methoden ontworpen om de vliegtuigmassa in individuele vluchten te schatten. Een van de belangrijkste verschillen tussen deze twee methoden betreft de vluchtstadia die worden gebruikt voor de schatting. In hoofdstuk 9 is het hele traject vereist. De massa wordt eerst in verschillende stadia van een vlucht berekend. Vervolgens, met empirische kennis van de massa van de vlucht als a priori informatie, wordt de initiële massa bepaald als de maximale a posteriori schatting. Hoofdstuk 10 maakt gebruik van een kort vluchtsegment aan het begin van de klim en past een Bayesiaanse methode toe op het niveau van de vluchtdynamiek. Met een deeltjesfilter worden vliegtuigtoestanden geschat met behulp van observaties uit surveillancegegevens. Deze toestanden omvatten positie, snelheid, verticale snelheid, wind, temperatuur, massa en stuwkrachtinstelling. Het deeltjesfilter kan niet alleen de massa afleiden, maar geeft ook inzicht in de onzekerheid van de schattingen. De studie geeft het niveau aan datakwaliteit aan dat nodig is om een betrouwbare schatting te kunnen maken. Dit is een van de eerste onderzoeken die de onzekerheid van massaschatting kwantificeert, op basis van surveillancegegevens in luchtverkeersstudies.

In het laatste hoofdstuk van dit proefschrift worden definitieve conclusies en aanbevelingen gedaan. Met het doel om toekomstige luchtverkeersstudies transparanter, vergelijkbaar en reproduceerbaar te maken, zijn de modellen en de broncode die in dit proefschrift worden voorgesteld vrij beschikbaar gemaakt, waaronder bijvoorbeeld de pyModeS, OpenAP, Meteo-Particle en deeltjesfilterbibliotheken. Een groot deel van het open prestatie-model is al geïmplementeerd in de BlueSky-luchtverkeerssimulator. Het prestatie-model dat in dit proefschrift wordt voorgesteld, is een efficiënt open-source alternatief gebleken voor de huidige closed-source modellen. Het heeft ook het potentieel om te worden gebruikt in andere studies, bijvoorbeeld die onderzoeken naar trajectoptimalisatie en voorspelling. Deze onderzoeksgebieden kunnen worden beschouwd als interessante paden om dit doctoraatsonderzoek uit te breiden.

1

Introduction

1.1. Background

As of 2018, all the top 500 supercomputers in the world use Linux as their operating systems. Compare this to 20 years before, in 1998, when only one of the top 500 supercomputers operated on Linux. The success of Linux is a representation of the wider success of the open-source movement and a recognition of the value created by open collaborations. The open-source philosophy has not only created a powerful and flexible operating system, but it has also enabled people from all over the world to work collaboratively towards the same goals. The benefits of sharing code and data publicly apply to studies in all scientific domains, and the air transportation domain is no exception.

Air transportation science deals with different subsystems supporting all segments of air transportation, such as terminal areas, departures, and arrivals, as well as en-route flights. Centered on passenger service, it also involves numerous stakeholders, such as airlines, regional regulators, air navigation service providers, air traffic controllers, and pilots. Studies that address challenges and solutions in the domain of air traffic management often involve these different stakeholders and a baseline of shared knowledge is necessary to ensure that results can be reproduced and compared.

As one example, many air transportation studies involve (fast-time) simulations. To perform such studies, several closed-source commercial tools exist, including the commercial simulators, AirTOP, ATMOS, NARSIM, and SIMMOD. While these commercial tools offer convenient simulation capabilities, their proprietary licenses make it difficult for third-parties to freely undertake comparative studies. Moreover, as source code is not available to users, it is often difficult to evaluate the underlying algorithms or to extend their functionalities. The closed-source tools could also contain software bugs that users are not aware of and that could alter the true nature of air transportation research outputs.

In the field of air traffic management (ATM), it is not uncommon to see multiple individual approaches to address a specific problem, where all claim to provide

the optimal solution [54]. Without a common foundation, validations and verifications of different research approaches are often limited. This situation can be improved by introducing open tools and open models in the air traffic management community.

With the aim of making repeatable and comparable studies available for ATM researchers, the open-source air traffic simulator BlueSky [54] was developed by a team from TU Delft. Its goal is to provide a fully portable, open-source simulator. Since its first release on GitHub in 2015, the BlueSky simulator has become a joint open-source effort supported by many institutes in the global air traffic management community. One of the core aspects of this simulator is its aircraft performance model.

To study how aircraft fly, performance models of flight mechanics have been developed. There are different categories of performance models, with varying levels of details. The most detailed, non-linear six-degree-of-freedom models are commonly used in aircraft control studies. Air traffic management research often assumes a stable aircraft and neglects fast rotational dynamics. This assumption means that a point-mass aircraft performance model is sufficient in most use cases. Such a point-mass model is used throughout this entire dissertation.

The most common aircraft performance model used in ATM research is the Base of Aircraft Data (BADA) model, which was developed by Eurocontrol. BADA is a proprietary performance model that forbids sharing of the model's data. Although BlueSky is compatible with the BADA performance files, its final goal is to include an open model that can be distributed freely with BlueSky. A preliminary open model based on textbook data was developed when BlueSky was first released [93]. Since the model that was first released with BlueSky relies on the literature data based on old aircraft models, its capabilities are limited. The need for a more comprehensive open performance model initiated the studies that are conducted in this dissertation. Furthermore, an open aircraft performance model is not only critical to the success of the BlueSky simulator but also an advantage for other air traffic management related studies.

With this goal in mind, the simple key research question to start our investigation is: *how can we model aircraft performance using only open data?*

This dissertation takes a data-driven approach and makes use of the abundance of openly available aircraft surveillance data. Unlike conventional data-driven approaches (e.g.: [60]) which often ignore underlying physical relationships, most of the modeling efforts proposed in this dissertation are centered on the point-mass model of flight dynamics. In general, this dissertation is built upon three cornerstones: data, models, and estimations, as shown in the diagram of Figure 1.1.

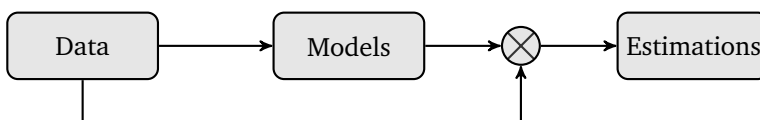


Figure 1.1: The relationships among the parts of the dissertation

Data represents aircraft surveillance data. *Models* represent models that are related to aircraft performance, such as thrust, fuel flow, and drag polar. *Estimations* represent the estimated aircraft parameters in individual flights. The arrow between *data* and *model* indicates the effort to construct the open model from data. The junction of *data* and *models* to *estimations* indicates the effort to derive aircraft parameters for individual flights. These parameters include, for example, mass, thrust setting, and bank angles, which are reflected in the estimation part of this dissertation.

This dissertation consists of three parts that correspond to the cornerstones. In the following sections of the introduction, these main concepts are further divided into different components. These components, in turn, reflect the nine chapters of this dissertation.

1.2. Data

As one of the foundations of this dissertation, almost all of its chapters depend heavily on data, specifically aircraft surveillance data. Traditionally, aircraft surveillance is performed by primary surveillance radar. The resulting surveillance data are not openly transmitted. A rotating radar determines its distance to aircraft by measuring the time difference of the emitted signal and the reception of the reflection of the signal from the aircraft. The primary radar is able to obtain the distance and azimuth of the aircraft. However, without the altitude, the position cannot be accurately determined. The secondary surveillance radar was introduced to address these problems.

The secondary surveillance radar implements different interrogation technologies (Mode-A/C/S) that are able to inquire additional information from aircraft. Mode A is used to interrogate a 4-digit octal identification code (or squawk code), while Mode-C is used to interrogate the barometric altitude of the aircraft. Mode-C is commonly combined with Mode-A as Mode-A/C, which provides both the aircraft identity and altitude. Mode-S (select) is a newer technology that is able to selectively interrogate a more comprehensive set of information from an aircraft.

In addition to interrogation-based information downlinked to air traffic control, recent technology has been designed to extend the Mode S capacity, allowing periodic broadcast of aircraft state information without the need for interrogation. The technology is called Automatic Dependent Surveillance-Broadcast (ADS-B), an implementation of the Mode-S Extended Squitter. Aircraft equipped with an ADS-B transponder periodically broadcast essential state information of the flight. These states include, for example, identity, position, altitude, velocity, and operation status. Since the information is broadcast omnidirectionally and un-encrypted, these messages can be decoded by anyone according to pre-defined standards.

These Mode-S communications are documented by *ICAO Annex 10 - Aeronautical Telecommunications* [64]. In total, there are 24 message formats, which can be identified by a 5-bit *Downlink Format (DF)* code at the start of any message. Among all 24 possible formats, 11 downlink formats are actively used. The contents of these messages are defined in Table 1.1, where the downlink format, con-

tent, and message length (in bits) are indicated.

Table 1.1: Mode-S downlink formats defined by ICAO Annex 10

DF	Message content	Length
0	Short air-air surveillance (ACAS)	56
4	Surveillance, altitude reply	56
5	Surveillance, identity reply	56
11	All-call reply	56
16	Long air-air surveillance (ACAS)	112
17	Extended squitter (ADS-B)	112
18	Extended squitter/non transponder	112
19	Military extended squitter	112
20	Comm-B, altitude reply	112
21	Comm-B, identity reply	112
24	Comm-D, extended length message	112

In this dissertation, our main source of data is Comm-B (DF 20/21) and ADS-B (DF 17) messages described in the previous table. The *Technical Provisions for Mode-S Services and Extended Squitter* [66] and its second revision [67] define the content of ADS-B and Comm-B messages.

Information transmitted through ADS-B is not dependent on interrogations. Each message contains the identification of the aircraft, given by a 24-bit unique ICAO transponder code. Each transponder code can be related to a specific aircraft. In all ADS-B messages, a *Type Code* (TC) is included. Based on the *Type Code*, we can identify the message format and then decode transmitted data accordingly.

The information contained in ADS-B messages is not sufficient for all studies in this dissertation. For instance, the speed information contained in ADS-B messages refers to the ground speed ¹ of the aircraft. In many chapters of the dissertation, we need to obtain the true airspeed of the aircraft to more accurately model the performance of the aircraft. In these cases, we try to make use of the airspeed data contained in the Comm-B messages originating from Mode-S interrogations.

As a third-party observer, the decoding of Comm-B messages is not as simple as that of ADS-B messages. Critical indicators necessary to identify aircraft ² and message types are not included in these messages. Identifying information in Comm-B messages has been a complex but fundamental part of this dissertation. We have developed new methods to decode the data in the absence of the interrogation information transmitted by the surveillance radar.

Most of the data used in this dissertation are collected using the receiver mounted on the top of the Aerospace Faculty building at the TU Delft. Thanks to the flat terrain in the Netherlands, the receiver has a large coverage area of more than 400 km radius. In Figure 1.2, the coverage and normalized density of traffic over one day are illustrated.

In addition to surveillance data, other public data sources are also considered.

¹Airspeed is broadcast only when the accurate position cannot be determined using the global navigation system. This only happens in rare cases.

²In general, an aircraft is identified by its 24-bit transponder address in this dissertation.

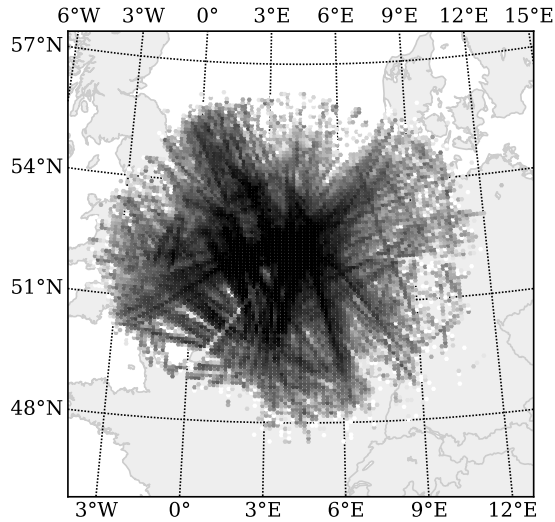


Figure 1.2: Coverage of the ADS-B antenna at TU Delft. The different shades of gray indicate the ADS-B data density. Common flight routes are distinguished by darker shades of gray.

For example, weather data from numerical weather prediction models [74] is used in Chapter 4, where the wind information is integrated with the ground speed to estimate the airspeed of the aircraft. In Chapter 6, we construct a model for estimating fuel consumption, which is based on the ICAO engine emission databank [63]. In Chapter 7, data on basic aircraft characteristics are gathered from published aircraft manufacture data sheets [105].

1.3. Models

The second foundation of the dissertation consists of different models that are involved in calculating aircraft performance. They are the kinematic model, dynamic model³, weather model, thrust and fuel flow models, and drag polar model.

1.3.1. Kinematic model

There are two different types of point-mass models: kinematic and dynamic. The primary difference is that while a dynamic model focuses on forces and energy, a kinematic model deals only with aircraft motions. BADA, for instance, employs both model types: the BADA aircraft performance operation file (OPF) models the dynamic properties of the aircraft, while the airline procedures file (APF) models the kinematic aspects of flights. The General Aircraft Modelling Environment (GAME) [15], also developed by Eurocontrol, is another example of a kinematic performance model.

The kinematic model describes the motions of aircraft without involving the

³Some other studies also refer to the dynamic model as the kinetic model.

forces. It is a simplified form of the model, which is used to describe the aircraft motion at different flight phases, such as takeoff, initial climb, climb, cruise, descent, final approach, and landing. In each flight phase, the kinematic properties (velocity, acceleration, or distance) can be modeled based on ADS-B data.

1.3.2. Dynamic model

When aircraft forces are considered, most studies require a more complicated model to describe performance than a kinematic model can provide. Commonly, the total energy model has been used in air traffic management related studies. The total energy model describes the conservation of total energy that is generated by the engines to compensate drag and the change of kinetic and potential energy. In this model, velocity and altitude can be obtained through aircraft surveillance data, leaving thrust, drag, and mass to be determined.

Thrust

Thrust is produced by the engines of the aircraft, and modeling aircraft engine performance is a complicated research area. In air traffic management studies, the thrust model is simplified. Instead of trying to model the performance of engines, we are interested in the net force that is produced by the aircraft in different stages of the flight. For example, in BADA v3 [103], thrust is modeled as a polynomial model related to the aircraft altitude.

Aircraft thrust is a parameter that cannot be derived using surveillance data. We have to rely on open models that are created by other researchers. In several chapters of the dissertation, we use an empirical model for two-shaft turbofan engine thrust calculation proposed by [8]. The model is constructed and evaluated based on real engine performance data. Thus, in this dissertation, thrust is modeled as functions of both altitude and speed, as well as the vertical rate. This offers a more accurate interpolation than the BADA v3 model, in which the thrust is only dependent on aircraft altitude.

Drag and drag polar

To compute aircraft drag in a point-mass performance model, we need knowledge of the drag polar. The challenge here is to accurately estimate the drag polar parameters for different aircraft types using other available components of the performance model.

Based on the deterministic total energy equation, we introduce a stochastic total energy (STE) model in this dissertation. The stochastic model treats the parameters of the total energy model as random variables. Introducing these stochastic components allows us to construct a hierarchical model that can be solved using Bayesian computing. Numerical solvers based on Markov Chain Monte Carlo sampling are used. In the end, this new technique allows us to derive drag polar models for different aircraft types.

1.3.3. Weather

In order to obtain a better estimation of aircraft performance parameters such as airspeed, it is crucial to understand weather conditions during flight, specifi-

cally wind and temperature. Traditionally, air traffic management studies rely on weather data generated from the numerical weather forecast system. The most common datasets are from the National Centers for Environmental Information (NCEI) and the European Centre for Medium-Range Weather Forecasts (ECMWF). Studies that focus on predictions require forecasting datasets that predict weather conditions in the future, while performance studies often take advantage of more accurate re-analysis data.

Re-analysis datasets are generated using data assimilation techniques that combine information from different sources, such as multiple satellites, weather balloons, and weather stations. The advantage of these datasets is that a wide (often global) coverage is guaranteed. However, they are not without their own limitations. Due to the global grid representation of the weather information, the resolution is relatively low. For example, the Global Forecast System Analysis dataset from NCEI offers the highest resolution of 0.5 degrees. In the context of the Netherlands, this indicates that local wind variation in an area of approximately 35 by 55 kilometers is smoothed out. For performing studies that predict short flight segments, a higher resolution is often desired.

In this dissertation, we introduce a new model called the Meteo-Particle model, which allows us to re-construct weather conditions in real time using aircraft surveillance data. We are able to estimate wind and temperature field using observations derived from the combination of ADS-B and Comm-B information.

1.4. Estimations

Once we know the data and models, it is possible to estimate aircraft performance parameters. In this dissertation, aircraft turn performance, aircraft mass, and thrust settings are estimated.

Turn performance is often overlooked in studies related to air traffic management. Air traffic controllers often rely on track and turn reports from aircraft Mode-S secondary surveillance data to gather performance indicators such as roll angle and track rate. However, this data has a low update rate and is not always available. An accurate estimation method for roll angles would be beneficial for performance analysis.

The mass is commonly considered as an input parameter for most air traffic management related studies. With a known mass, we can compute the drag of the aircraft in the point-mass model. If we are able to model the thrust, drag, speed, the derivative of the speed (i.e., acceleration), and vertical rate, it is possible to estimate the mass of an aircraft. Since mass is a crucial parameter for predicting a trajectory or performing fuel optimization, an accurate method to determine mass could bring welcome insights to air traffic management studies.

Previous studies were conducted aiming to estimate mass using flight data such as radar data [2, 3, 4] or flight recorder data [19, 17]. However, low accuracy and a lack of knowledge of the uncertainty of the estimation remain issues for these existing methods. Based on ADS-B and Comm-B data, we investigate two improved methods to address the estimation of mass. We also propose a new way to study the uncertainties in estimations.

1.5. Research questions

In the previous sections, we discussed the details of the core concepts of the dissertation: data, models, and estimations. With these connections in mind, our main research question is posed as follows:

How can we use aircraft surveillance and other open data to improve aircraft performance modeling and parameter estimation?

Based on the above discussion, this question can be divided into the following sub-questions:

1. *How to extract valuable information from open aircraft surveillance signals?*
2. *How to turn scattered broadcast data into continuous and segmented flights?*
3. *How to use surveillance data to improve aviation weather models?*
4. *How to construct a kinematic model using surveillance data?*
5. *How to model thrust and fuel flow with open data?*
6. *How to use open data to construct aircraft drag polar models?*
7. *How to estimate performance parameters using open surveillance data?*

1.6. Structure

In this dissertation, with the exception of the introduction and conclusion, the chapters focus on answering one (or a part) of the sub-research questions. Most of the chapters are based on journal or conference articles published or submitted during the period of the Ph.D. research, but with a consolidated structure to emphasize the main arguments of the dissertation.

Figure 1.3 illustrates the inter-connectivity between all parts of this dissertation. Arrows indicate dependent relationships among the components. Dashed arrows illustrate the modeling and estimation efforts. In addition, circled numbers indicate the chapter where the corresponding components are discussed.

1.6.1. Part One: Data

Chapter 2 deals with the foundation of the dissertation, which are aircraft surveillance data. Two types of Mode-S data are introduced: ADS-B (DF=17) and Comm-B (DF=20/21). The chapter starts with the fundamental concepts of Mode-S data and its importance in aircraft performance modeling research. The detailed message formats of ADS-B and Comm-B are laid out in this chapter. Decoding ADS-B messages is straightforward. However, as a third-party observer, decoding Comm-B messages is challenging. Most of the chapter is focused on developing a set of methods to infer and decode the information contained in these Comm-B messages with unknown types. In addition, an error checking mechanism is introduced that allows us to evaluate the accuracy of the Comm-B messages. At the end of this chapter, we are able to offer a stable solution to infer and decode Mode-S (ADS-B and Comm-B) data. An open-source tool, pyModeS [135], is developed along with the chapter.

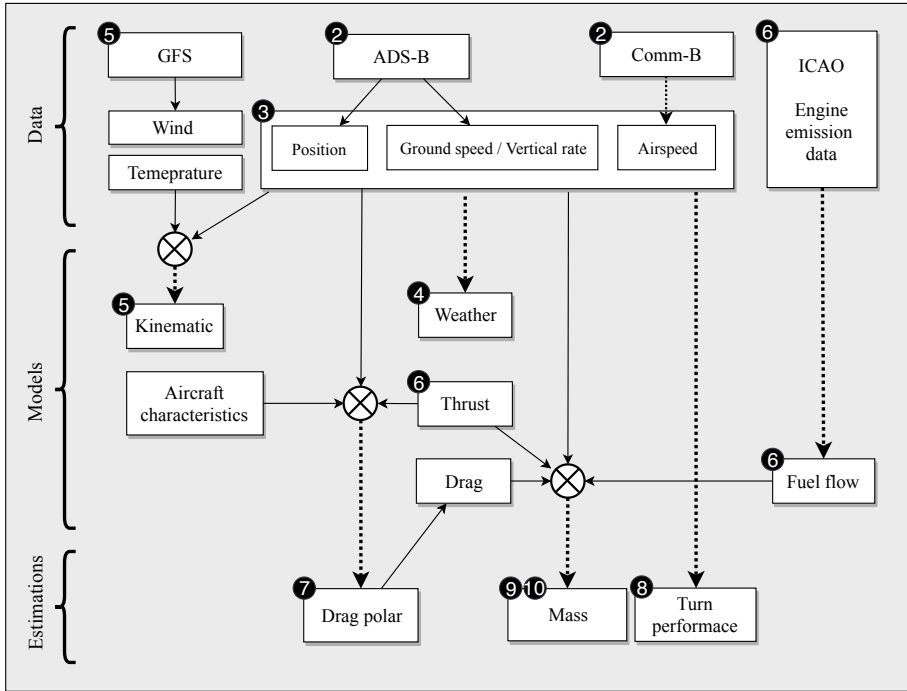


Figure 1.3: Connections between the aircraft performance components in this dissertation

In Chapter 3, following the decoding of scattered surveillance data in the previous chapter, we continue to extract useful information related to flights. This chapter first introduces a clustering algorithm that allows us to automatically group data points from the same flights and export them into continuous flight trajectories. Once a trajectory is given, a fuzzy logic identification method is used to segment trajectories into different flight phase segments. By the end of this chapter, we are able to extract ground, climb, descent, cruise, and level flight segments from scattered surveillance data.

1.6.2. Part Two: Models

Chapter 4 proposes a new model to reconstruct aviation weather fields. Due to the limitations of the weather data as provided by numerical forecast models, we identify the valuable information contained in Mode-S data as a new source of data. If derived correctly, Mode-S data can be used to accurately model the weather. In this chapter, the novel Meteo-Particle model allows us to construct real-time wind and temperature field using only Mode-S data. Using the wind field, we can further derive true airspeed of aircraft based on their ADS-B ground speed. This enables us to model aircraft performance more accurately for later chapters.

In Chapter 5, the kinematic performance models for common commercial aircraft are constructed. In this chapter, we use a larger dataset gathered by a crowd-

sourced ADS-B network (FlightRadar24). Combined with the segmentation algorithm from Chapter 1, aircraft motion in seven flight phases is modeled. The results of this chapter provide a comprehensive open-source model on kinematic performances for common commercial aircraft.

In Chapter 6, the thrust and fuel flow model are studied. The thrust model is primarily based on the two-shaft turbofan model proposed in [8]. In combination with data from [136], this chapter constructs a thrust model that can be used to calculate the maximum net thrust of common aircraft types. Based on the ICAO emission databank [63], we also construct a fuel flow model that can be used to calculate the fuel consumption of the aircraft based on Mode-S data. The model is verified with the flight recorder data from the TU Delft Cessna Citation II aircraft.

In Chapter 7, the aerodynamic model (drag polar) is studied. This chapter focuses on deriving the drag polar of different aircraft types using Mode-S data. Other open data sources, such as the characteristics of the aircraft published by aircraft manufacturers, are used. In order to cope with the large dimension of the parameters, Bayesian computing is introduced to solve a hierarchical stochastic total energy model. Using this new method, the drag polar models for the 20 most common commercial aircraft types are computed and provided.

1.6.3. Part Three: Estimations

Chapter 8 investigates the possibility of using aircraft surveillance data to study turn performances of aircraft. In this chapter, we develop a set of methods to infer hidden parameters such as turn radius, bank angle, and load factor.

Chapter 9 proposes using Bayesian inference to estimate aircraft initial mass, taking into account empirical knowledge about an aircraft. This method first computes the mass at different phases of a flight. Then, a maximum likelihood estimate is derived based on a prior probability density function. The drawback of this method is that it requires the complete trajectory in order to perform the estimation.

To overcome this drawback, Chapter 10 introduces a new approach. The principle is to integrate Bayesian philosophy in the performance system. This concept is comparable to the stochastic total energy model used in Chapter 7. Here, a Sequential Monte Carlo estimator (also known as the particle filter) is developed to estimate the mass and thrust settings of a flight at the initial segment of the climb. In addition to providing the estimation, the particle filtering approach allows us to better understand and study the accuracy of estimation that is inherited from ADS-B uncertainties.

1.6.4. Conclusion

Finally, Chapter 11 revisits the research questions and findings. It summarizes the studies conducted in this dissertation. Recommendations for future research are given based on studies undertaken and the conclusions derived throughout the chapters of this dissertation.

Part I

Data

2

Mode-S data decoding

The availability of low-cost Automatic Dependent Surveillance-broadcast (ADS-B) receivers has given researchers the ability to make use of large amounts of aircraft state data. This data are being used to support air transportation research in performance study, trajectory prediction, procedure analysis, and airspace design. However, aircraft states contained in ADS-B messages are limited. More performance parameters are downlinked as Mode-S Comm-B replies, upon automatic and periodic interrogation of air traffic control secondary surveillance radar. These replies reveal aircraft airspeed, turn rate, target altitude, and so on. They can be intercepted using the same 1090 MHz receiver that receives ADS-B messages. However, a third-party observer does not know the interrogations, that originated the Comm-B replies. Thus, it is difficult to decode these messages without knowing the type and source aircraft. Furthermore, the parity check also cannot be performed without knowing the interrogations. In this study, we propose a new heuristic-probabilistic method to decode Comm-B replies and to check the correctness of the messages. Based on a reference dataset provided by air traffic control of the Netherlands, the method yields a success rate of 97.68% with an error below 0.01%. The performance of the proposed method is further examined with data from eight different regions of the world. The implementation of the inference and decoding process, pyModeS, is shared as an open-source library.

This chapter is based on the following publication:
Sun, J., Vũ, H., Ellerbroek, J. and Hoekstra, J.M., 2019. PyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research. IEEE Transactions on Intelligent Transportation Systems [133].

2.1. Introduction

In air transportation research, studies related to aircraft performance are often dependent on the airspeed of the aircraft. This speed information is used in the dynamic model of the aircraft to perform, state estimations and trajectory predictions for example. In addition to airspeed, the performance model also takes into account other trajectory state information, such as positions, ground speeds, and altitudes.

Many of these states in the dynamic model can be openly observed using modern aircraft surveillance technology, for instance, the *Automatic Dependent Surveillance-Broadcast* (ADS-B). ADS-B provides information on aircraft position, speed, and vertical rate. The speed contained in ADS-B refers to the ground speed rather than the airspeed, unless in (rare) cases when the location cannot be determined from the Global Navigation Satellite Systems. The advantage of ADS-B is that the signals can be openly intercepted and decoded using a simple ground receiver set-up.

When airspeed is not available, there are two ways to adopt the ground speed for performance analysis. The first simple approach is to assume the ground speed as airspeed by ignoring the wind. This may cause errors in performance calculations when a strong wind is present. The second approach is to integrate the wind from numerical forecast models. However, wind data from these models often cannot accurately reflect local wind variations.

At the same time, air traffic controllers are also interested in these performance parameters, which are constantly interrogated by surveillance radars under the *Enhanced Mode-S Surveillance* (EHS) technology. Corresponding messages are downlinked using the *Comm-B* protocol. Air traffic controllers make use of this data to better monitor and predict flights and to make better traffic control decisions. Within these downlinked messages, information such as true airspeed, indicated airspeed, Mach number, and true heading of the aircraft are transmitted.

The main difference between the two signals is that ADS-B is automatically broadcast by the aircraft, while Comm-B is transmitted upon interrogation. If available, direct access to air traffic control data would provide the most accurate information. However, due to licenses and data agreement processes, obtaining this data can be challenging for third-party researchers. Even when the access is granted, the information is often extracted from historical data archives, which makes it difficult to perform real-time performance analysis.

Nevertheless, it is possible to obtain the downlinked Comm-B data with the same ground receiver used for ADS-B data. However, many difficulties arise when trying to decode these reply messages. The biggest barriers for decoding are the unknown aircraft source represented by the ICAO transponder address and the interrogation type represented by the *Comm-B Data Selector* (BDS) code. Even though the structure of messages follows open standards [66, 67], without knowing the ICAO code and BDS type, useful information from these messages cannot be extracted.

The goal of this study is to enable open and real-time access to these Mode-S messages. The main research questions of this study are defined as:

1. How to determine the source aircraft of a Comm-B message?
2. How to identify the BDS type and decode a Comm-B message without knowing the original interrogation?
3. How to detect errors in Comm-B messages when information is incomplete?

To illustrate these inference efforts, Figure 2.1 shows the process related to the methods proposed in this study. In this figure, the downlink surveillance signal (containing ADS-B and Comm-B) from aircraft is received by a software-defined radio (SDR) receiver first. The signal is converted to a raw binary data stream, which is then further decomposed into a sequence of message frames. For ADS-B messages, information can be decoded directly. For Comm-B replies, we first use the inference methods proposed in this study to determine the BDS code, source aircraft, and errors. Finally, the information contained in these messages is decoded.

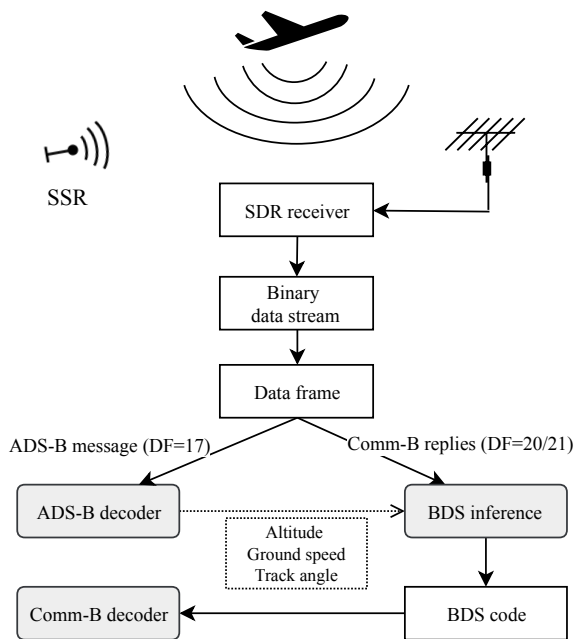


Figure 2.1: The Mode-S inference and decoding pipeline

In the remaining sections of this chapter, we first provide the background information on ADS-B and EHS. Next, we discuss the identification processes and error detection in detail. Several tests are proposed in this chapter, and methods are validated with a reference dataset provided by Air Traffic Control the Netherlands (LVNL). Finally, we discuss the use cases, implementation, and recommendations, as well as the conclusion at the end of the chapter.

2.2. Background

2.2.1. Fundamentals of Mode-S, ADS-B, and Comm-B

As the demand for air transportation increases, the airspace over the world is becoming more crowded. To efficiently make use of the airspace and increase the traffic capacity, air traffic controllers need to rely on accurate flight trajectory predictions. Communications between ground and aircraft are becoming more frequent in order to obtain accurate updates regarding aircraft states beyond the basic position provided by the primary radar. Since the beginning of aircraft surveillance, different methods and standards have been developed for downlinking aircraft data. The two most common methods are based on interrogation and broadcast.

Traditionally, when an air traffic controller requires information in addition to the aircraft position, the Mode-S selective interrogation [139] is used. This is performed by the *secondary surveillance radar* (SSR). Numerous aircraft states can be interrogated by the SSR. The most common downlinked messages are Comm-B replies. The content of the interrogation is identified by the BDS code, which is a two-digit hexadecimal code (8 bits) that indicates the information desired by the air traffic controller. In total, 255 BDS codes can be defined. The reply data is encoded in a 112-bit Comm-B downlink message. Among all these BDS codes, several BDS codes are grouped and identified as Mode-S enhanced surveillance (EHS), which consists of the *selected intention report* (BDS 40), *track and turn report* (BDS 50), and *heading and speed report* (BDS 60).

The simpler ADS-B is an implementation of the Mode-S extended squitter. It is a newer technology compared to the interrogation-based Mode-S. It allows the automatic broadcast of the aircraft state information at a constant rate. In many regions, aircraft are required to be equipped with Mode-S transponders that are compatible with ADS-B. When it is enabled, ADS-B allows aircraft to automatically report the identification, location, speed, and operational status. The common update interval of critical states (such as position and speed) is designed to be around 0.5 seconds.

Both ADS-B messages and Comm-B replies are transmitted using the 1090 MHz transponder. Downlinked signals can be intercepted freely using low-cost commercial off-the-shelf ground receivers. Several crowd-sourced initiatives have been constructing global networks of ground receivers, for example, *ADS-B Exchange*, *FlightRadar24*, *FlightAware*, and *OpenSky-network*. The quantity of data gathered by these networks is enormous, which leads to great potential in air transportation researches. For example, in recent research, this data has been used for operational performance studies [77] and trajectory prediction [140]. The ground receiver networks also enable the possibility to determine aircraft location by using multilateration [76].

ADS-B is designed as an independent communication protocol, where the message itself contains all information needed for decoding, while Comm-B communication is designed as a dependent protocol. Only the air traffic controller who initiated the interrogation can identify the source aircraft and decode the content of the replies. To this extent, third-party observers have no information on

the interrogated aircraft or corresponding BDS code. However, earlier research has been shown that some information can be extracted, for example, to provide meteorological observations [50].

2.2.2. Regulation and availability

Several Mode-S capabilities are mandatory for aircraft flying in European airspace since 2009. Two different categories of Mode-S surveillance are defined, which are elementary surveillance (ELS) and enhanced surveillance (EHS) [46]. According to European regulation, all aircraft that fly Instrument Flight Rules (IFR) in general air traffic (GAT) must be ELS compliant. In addition, all fixed-wing aircraft flying IFR in GAT with a maximum takeoff mass greater than 5.7 ton or a maximum cruising true airspeed greater than 250 knots must be EHS compliant. ADS-B is a newer surveillance technology and has also been adopted broadly. Regulators in both Europe and the United States have set the agenda for obligatory compliance.

Since ADS-B does not require active interrogations from surveillance radar, the messages are broadcast and available at all times everywhere. These messages can be received by ground receivers or satellites [101].

In Mode-S ELS, only a limited number of parameters are reported, including aircraft identity, altitude, flight status, and related supporting parameters. In Mode-S EHS, more aircraft states are interrogated, such as indicated airspeed, Mach number, vertical rate, magnetic heading, track angle, roll angle, selected altitude, and ground speed.

Depending on the location of the (third-party) ground receiver, the number of received replies varies. The availability and quantity of Comm-B messages also depends on air traffic density and the number of secondary surveillance radars in the area, as well as the rate of interrogation.

2.2.3. Data structure

In this chapter, we focus on two types of messages, which are ADS-B messages and EHS Comm-B messages. The structures of ADS-B and Comm-B messages are defined in ICAO Annex 10 [64]. ADS-B and Mode-S data are constructed using the data frame shown in Figure 2.2, with a total message length of 112 bits. The number of bits of each segment is indicated with parentheses in this figure. Each message starts with the *downlink format* (DF), followed by a 27-bit header with different components. Then, the crucial 56-bit data is appended with the downlink information encoded. Lastly, 24 bits are dedicated to the parity checksum.

ADS-B messages are identified by a DF number of 17 (10001 in binary format). In the header of an ADS-B message, the address of the aircraft transponder is indicated. This is a 24-bit address assigned by ICAO and categorized according to geographic region and country. The leading 3 bits are sub-type or category in different types of ADS-B messages. The *Type Code* (TC) is set using the first 5 bits of the 56-bit data segment. It defines the general type of message, for example, airborne position, airborne velocity, surface position, identification, etc.

In a Comm-B reply message, the DF number can be either 20 or 21 (10100 or 10101 in binary format). In the case of DF=20, the last 15 bits of the header

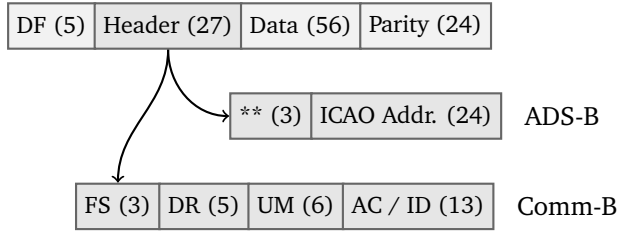


Figure 2.2: The structures of ADS-B and Mode-S Comm-B messages

2

indicates the *Altitude Code* (AC). When DF=21, the last 15 bits represent the *Identification Code* (ID) (a.k.a: the *squawk code*). The leading three segments in the header are *Flight Status* (FS), *Downlink Request* (DR), and *Utility Message* (UM). Unlike ADS-B, there is no indication of ICAO transponder address nor the BDS code in a message, except for a few cases.¹

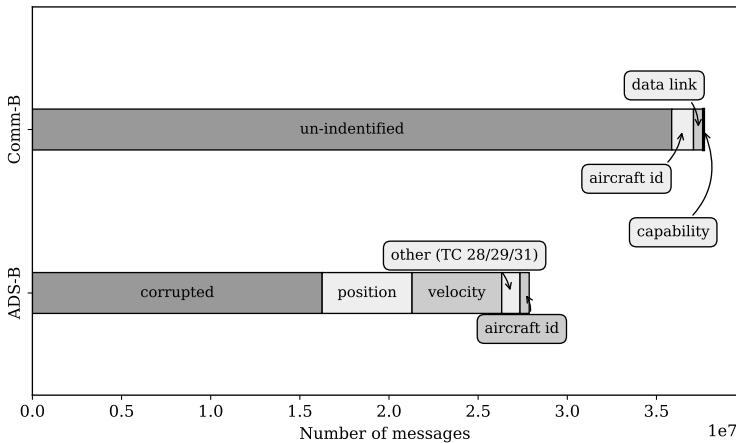


Figure 2.3: Number of ADS-B message and Mode-S Comm-B replies received in 24 hours at TU Delft, on May 30, 2018

Figure 2.3 illustrates the distribution of ADS-B and Comm-B replies, together with their distinct types, for a 24-hour period of data from a ground receiver situated in Delft, the Netherlands. We can see there are more Comm-B messages than ADS-B messages. About 35 out of 38 million EHS messages are not directly identifiable. Unlike ADS-B, none of the Comm-B messages can be checked for corruption due to the incomplete information on aircraft source and BDS code.

It is worth pointing out that more than half of the ADS-B messages (16 out of 28 million messages) are corrupted. The corruption of messages is also analyzed in the later sections of this chapter.

¹The exception cases are BDS 10, 20, and 30 messages. The BDS code is indicated in these messages. However, together, they only represent a very small percentage of all messages.

2.3. Basic decoding of ADS-B data

As mentioned earlier, the content of an ADS-B message is directly identified by its Type Code. The primary parameters downlinked by each type are listed in Table 2.1.

Table 2.1: Basic ADS-B message types (version 0, 1, 2)

TC	Content	Primary parameters
1 - 4	Aircraft identification	Call sign
5 - 8	Surface position	Latitude Longitude Speed Track angle
9 - 18 20 - 22	Airborne position	Latitude Longitude Altitude
19	Airborne velocity	East-west component North-south component Vertical rate

Within the same group, Type Code values also have their own indications. For example, in the identification group, a TC of 1 to 4 indicates different aircraft categories defined by the Mode-S standard according to [66]. In position groups, TC defines the level of accuracy and uncertainty.

Since ADS-B was first introduced to the present, there have been different versions of implementation: version 0, version 1, and version 2. In version 1 and 2, the accuracy and uncertainty are re-classified using two additional supplement bits. They offer more levels of *Navigation integrity category* (NIC). In addition, from ADS-B version 1 onward, three more types are introduced, as shown in Table 2.2.

Table 2.2: Additional ADS-B message types (version 1, 2)

TC	Content	Primary parameters
28	Emergency/priority status ACAS RA Broadcast	Status value Resolution advisories Threat identity
29	Target state and status	Target altitude Target heading / track Navigation accuracy category Surveillance integrity level
31	Operational status	Airborne capacity class Surface capacity class ADS-B version number NIC supplement Navigation accuracy category Surveillance integrity level

The version number 1 or 2 can be retrieved in TC=31 messages. However, for

version 0 transponder, no information regarding the version number is transmitted. In order to identify the exact version of the ADS-B transponder, we have to first check whether the aircraft is broadcasting any message with a TC of 28, 29 or 31. If not, then the ADS-B version is assumed to be version 0. When a TC 31 message is received, the ADS-B version (1 or 2) can be found at the 73th to 75th bit of the message.

Knowing the Type Code, in combination with the ADS-B version, it is possible to identify the uncertainties in broadcast aircraft states. At the same time, payload information can be decoded according to the *Technical Provisions for Mode Service and Extended Squitter*, which is published by ICAO [66, 67]. Some of the ADS-B states, such as ICAO address and speed, are used for the Comm-B BDS inference.

2

2.4. Interference of the Comm-B downlink parameters

Without knowledge of the original interrogation request, most of the Comm-B replies cannot be identified directly due to the lack of BDS information. It is also not possible to know whether a message is corrupted because the source aircraft ID (ICAO address) is unknown. This is because the checksum in Comm-B reply is overlaid with the 24-bit ICAO address. In some cases, it is overlaid again with the BDS code.

The two-digit hexadecimal BDS code can support up to 255 different types of messages. In practice, only a small portion of these types are interrogated. According to the European ELS and EHS mandate, the most commonly interrogated BDS codes are 10, 17, 20, 30, 40, 50, and 60. In Table 2.3, parameters contained in these reply messages are listed.

Table 2.3: Common Mode-S Comm-B BDS code and content

Type	BDS	Content	Parameters
ELS	10	Data link capability	Transponder capabilities
	17	GICB capability	Mode-S data capabilities
	20	Aircraft identification	Call sign
	30	ACAS resolution	Resolution advisories Threat identity
EHS	40	Vertical intention	Select altitude Barometric pressure setting
	50	Track and turn	Roll angle Track angle Ground speed Track angle rate True airspeed
	60	Heading and speed	Magnetic heading Indicated airspeed Mach number Vertical velocity

In order to decode these messages, we first need to recover the ICAO address of the source aircraft and examine whether the messages are corrupted without

full knowledge of parity. In parallel, we also implement the process of identifying the BDS codes, which is used for further error detection. As a result, the decoding of a message can be accomplished.

2.4.1. Source aircraft identification and error detection

Thanks to a process called *Address Parity* [44], we can recover the ICAO addresses for most of the Comm-B messages. In the last 24 bits of the message, a parity checksum is inserted. The checksum is computed using a *cyclic redundancy check* (CRC) algorithm [121], which is a common error-detecting scheme in telecommunications.

The generator code used for Mode-S CRC encoding is specifically designed in the following binary format:

$$G = 1111111111111010000001001 \quad (2.1)$$

which is used to compute the checksum by the encoder and to validate the message by the decoder. In the polynomial form, the generator is expressed as:

$$\begin{aligned} G(x) = & x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} \\ & + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} \\ & + x^{13} + x^{12} + x^{10} + x^3 + 1 \end{aligned} \quad (2.2)$$

Similarly, the binary format of the first 88 bits of the message can also be written in the polynomial format with the highest order of 87 (x^{87}). The message polynomial is denoted as $M(x)$. The checksum is computed using polynomial division between the message and generator. By combining the checksum with parity $P(x)$, we can compute the (possible) ICAO address:

$$\begin{aligned} M(x) &= \sum_{i=0}^{87} a_i x^i, \quad a_i \in \{0, 1\} \\ R(x) &= M(x) \% G(x) \\ A(x) &= R(x) + P(x) \end{aligned} \quad (2.3)$$

where $R(x)$ is the remainder (checksum) of the division $M(x)$ by $G(x)$. In Figure 2.4, we illustrate this ICAO address reversal process with an example message.

After the reversal process, the resulting 24 bits would be one of the following possibilities:

- Candidate ICAO address (CA): The correct ICAO address of an aircraft.
- Modified ICAO address (MA): The ICAO address with the first 8 bits overlaid with the BDS code.² The parity under this condition is identified as *Data*

²Only a small percentage of current aircraft to date are equipped with this capability.

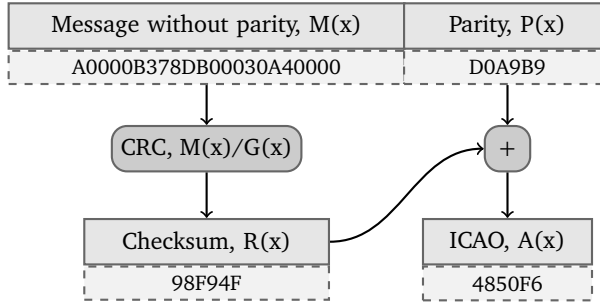


Figure 2.4: The ICAO address recovery logic

Parity.

- Impossible ICAO address (IA): An ICAO address that is not assigned. This indicates an error in the message.

If a corruption occurs in a reply, the reversal process will generate an incorrect ICAO address. This address can still be either CA, MA, or IA. Since we do not know the actual aircraft that has been interrogated, it is not possible to detect the error just using the parity. Instead, we have to identify new methods to identify corrupted messages.

Our proposed error detection mechanism consists of three checks. The first check is to identify impossible ICAO addresses. The unassigned blocks of addresses are listed in Table 2.4. If the resulting ICAO belongs to one of these blocks, it is likely that the message is corrupted. However, the possibility of Data Parity cannot be ruled out in this case.

Table 2.4: Un-assigned ICAO address block

Starting bits	Address block	Geographic region
00100	200000 - 27FFFF	Africa-Indian Ocean
00101	280000 - 28FFFF	South America
0101	500000 - 5FFFFF	Europe and North Atlantic
01100	600000 - 67FFFF	Middle East
01101	680000 - 6F0000	Asia
1001	900000 - 9FFFFF	North America and Pacific
111011	B00000 - BFFFFF	Caribbean
1101	D00000 - DFFFFF	Reserved
1111	F00000 - FFFFFF	Reserved

Next, the address is cross-validated with the ICAO addresses included in all ADS-B messages from the same time period. Since ADS-B messages can be properly error checked with the CRC process, we are confident about the obtained addresses. After this step, the correct messages are identified. An unidentified ICAO address indicates either corruption in a message or that the aircraft is not equipped with ADS-B capability.

The third error check is the ICAO-to-Squawk comparison. For a Comm-B reply message with DF=21 (identity reply), the squawk code of the transponder can be obtained. By comparing the squawk code and the ICAO address, we can identify error messages, such as the ones associated with low ICAO-to-Squawk combination frequency. Since it is still possible that the message is overlaid with a BDS code, we also use the inferred BDS code to compare the overlaid ICAO address and squawk code.

Summarizing all previous steps, we can construct an error detection model, as illustrated in Figure 2.5.

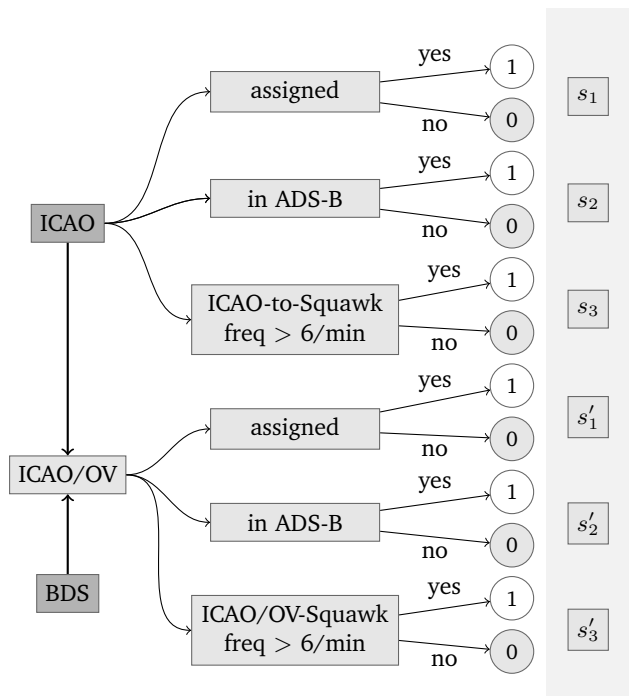


Figure 2.5: EHS message error identification process

In this model, we first compute three binary scores based on the following conditions:

1. The ICAO address is assigned.
2. The ICAO address appears in the pool of ADS-B addresses.
3. The ICAO address has the ICAO-to-Squawk frequency for more than a certain number times per minute (six).

The resulting scores are denoted as s_1 , s_2 , and s_3 respectively. Then, the in-

ferred BDS code ³ is overlaid with the ICAO address to examine the possibility of a Modified ICAO Address. If the MA is applied, the resulting ICAO address with an overlapping BDS code (ICAO/OV) satisfies some of these conditions. Similarly, three more scores are computed, denoted as s'_1 , s'_2 , and s'_3 respectively. The total correctness score of S is computed as follows:

$$S = \left[(s_1 \vee s'_1) \vee (s_2 \vee s'_2) \right] \wedge (s_3 \vee s'_3) \quad (2.4)$$

where \vee is the logic OR operation and \wedge is the logic AND operation. The error messages are identified with $S = 0$, while correct messages are identified with $S = 1$. It is important to point out that the error model takes into consideration the situations of 1) aircraft not being equipped with an ADS-B transponder and 2) the Modified ICAO Address being used in a parity checksum.

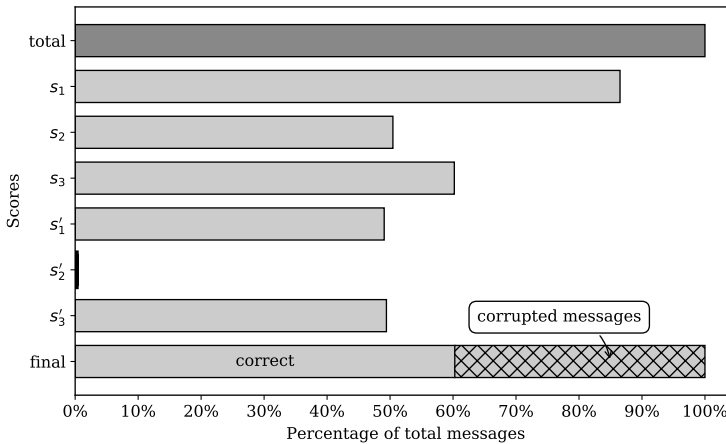


Figure 2.6: Error identification statistics, based on Comm-B replies received at TU Delft from 12:00 to 13:00 UTC, September 7, 2017. A similar level of corrupted messages can also be seen in Figure 2.10.

Figure 2.6 shows the resulting decoding statistics based on applying the error detection model to the same one-hour dataset as before. In total, 60% of the messages are identified as correct messages. The remaining 40% are corrupted. We can also see that there are only a few messages that satisfy the condition $s'_2 = 1$, which implies that the BDS overlay is infrequently requested at the present time by ATC in the European airspace. In addition, only aircraft with overlay capability are able to support this feature. Such information can be found in the BDS 17 (GICB capability) messages of the aircraft.

2.4.2. BDS inference using the Heuristic-Probabilistic method

Not all aircraft have the same BDS capabilities enabled. To determine which BDS capabilities are available for an aircraft, the SSR would first initiate a Common

³The BDS code is inferred in parallel using the methods described in the following section. This is possible because the deterministic BDS identification can be performed independently.

usage GCIB capability interrogation (BDS 17). There are 24 common BDS capabilities that are reported in the BDS 17 message. SSR will only interrogate the ones that are enabled for this aircraft.

To infer the BDS code request by ATC, a two-step inference process is designed. It consists of a heuristic step and a probabilistic step. The heuristic logic is inspired and developed based upon the method proposed by [50], which gives the first estimation of possible BDS codes. The probabilistic identification is introduced to identify messages with both BDS 50 and 60 codes from the previous step. Together, the Heuristic-Probabilistic (HP) process is able to deal with all common ELS and EHS replies.

Heuristic logic

All Comm-B payloads consist of multiple data blocks. Many of these blocks are combined with respective status bits. A status bit must be set to zero when no information is present in the data block. While evaluating the possibility of a specific BDS code, if any of the data blocks violate this rule, this BDS code is discarded.

In the structures of some Comm-B types, some bits can be reserved (or not used). These bits are required to be zeros at all times. If any of these predefined zero bits are set to one, the corresponding BDS code is also discarded.

The heuristic logic also checks parameter values that are decoded for different assumed BDS codes in a parallel fashion. These values need to be within their physical boundaries. For example, in BDS 60, the Mach number cannot be higher than one (for commercial aircraft). If a decoded Mach number is larger than one, BDS 60 is rejected as a BDS code. All evaluations are performed for each BDS code. Deterministic conditions for each parameter in all seven common BDS codes are listed in Table 2.5.

Probabilistic identification

Of all available surveillance messages, BDS 50 (track and turn report) and BDS 60 (heading and speed report) messages are most frequently interrogated. However, they have very similar structures, as shown in Table 2.5. Due to this similarity, some messages can be considered as both BDS 50 and 60 after the heuristic logic. In order to differentiate these two codes, the probabilistic step is designed.

The principle is to construct a probability density function (PDF) using current aircraft states that are observed from ADS-B data. The probabilities of BDS 50 and 60 are computed with this function. The BDS code that results in a higher probability is considered to be the correct one.

From ADS-B, we use aircraft ground speed (v_g) and track angle (χ) to construct the x and y components of the velocity. They are treated as the means for the joint probability density function. These two components, denoted as v_{gx} and v_{gy} , are calculated as:

$$\begin{aligned} v_{gx} &= v_g \cdot \sin \chi \\ v_{gy} &= v_g \cdot \cos \chi \end{aligned} \tag{2.5}$$

Table 2.5: Status bit and parameter bits

BDS	Bits	Parameter	Logic rule	Value rule
10	1 - 8 10 - 14	BDS Reserved	Bits equal to 00010000 Bits must all be zeros	
17	7 29 - 56	BDS 20 enabled ^I Reserved	Bit equals to 1 Bits must all be zeros	
20	1 - 8 9 - 56	BDS Call sign	Bits equal to 00100000	Only contain 0-9, A-Z, or space
30	1 - 8 29 - 30 16 - 22	BDS Threat type ACAS III	Bits equal to 00110000 Bits must not equal to 11	Less than 48
40	1 : 2-13 ^{II} 14 : 15-26 27 : 28-39 40 - 47 52 - 53	MCP/FCU selected altitude FMS selected altitude Barometric pressure setting Reserved Reserved	Status and value bits consistent ^{III} Status and value bits consistent Status and value bits consistent Bits must all be zeros Bits must all be zeros	
50	1 : 2-11 12 : 13-23 24 : 25-34 35 : 36-45 45 : 46-56	Roll angle True track angle Ground speed Track angle rate True airspeed	Status and value bits consistent Status and value bits consistent Status and value bits consistent Status and value bits consistent Status and value bits consistent	Between -60 and 60 (degrees) Between 0 and 600 (knots) Between 0 and 500 (knots)
60	1 : 2-12 13 : 14-23 24 : 25-34 35 : 36-45 46 : 47-56	Magnetic heading Indicated airspeed Mach number Barometric vertical rate Inertial vertical rate	Status and value bits consistent Status and value bits consistent Status and value bits consistent Status and value bits consistent Status and value bits consistent	Between 0 and 500 (knots) Between 0 and 1 Between -6000 and 6000 (feet/minute) Between -6000 and 6000 (feet/minute)

^I BDS 20 is the code that has to be enabled for all transponders to provide the minimum Mode-S capabilities.

^{II} This format b1:b2-b3 indicates the status bit at b1 with value bits from b2 to b3.

^{III} Consistency indicates that when a status bit is zero, all value bits must also be zeros.

Then, we construct a bi-variate normal probability density function considering these values as mean values. To simplify the problem, we assume there is no correlation between the x and y speed components, the function of the probability (the Gaussian PDF without normalization) can be expressed as follows:

$$p(v_{ax}, v_{ay}) = \exp \left\{ -\frac{1}{2} \left[\frac{(v_{ax} - (v_{gx} - v_{wx}))^2}{\sigma_{vx}^2} + \frac{(v_{ay} - (v_{gy} - v_{wy}))^2}{\sigma_{vy}^2} \right] \right\} \quad (2.6)$$

where v_{ax} and v_{ay} are the x and y components of airspeed. v_{wx} and v_{wy} are wind speed components. σ_{vx}^2 and σ_{vy}^2 are the variances. Accurate wind speed is not required for the purpose of BDS identification. For example, it can be obtained using weather forecast data. In case the wind information is not available or at calm wind conditions, these terms may be set to zero. However, this assumption may affect the accuracy of the identification.

For the variances, both are empirically set to be 20 knots in our experiments. This choice is based on the common magnitude of accuracy in speed that is included in ADS-B data [111]. Using Equation 2.6, the probabilities of BDS 50 and BDS 60 are compared. Finally, the corresponding BDS code to the higher $p(v_{ax}, v_{ay})$ value is accepted. Figure 2.7 shows an example of the identification based on the possible speeds decoded as BDS 50 and BDS 60.

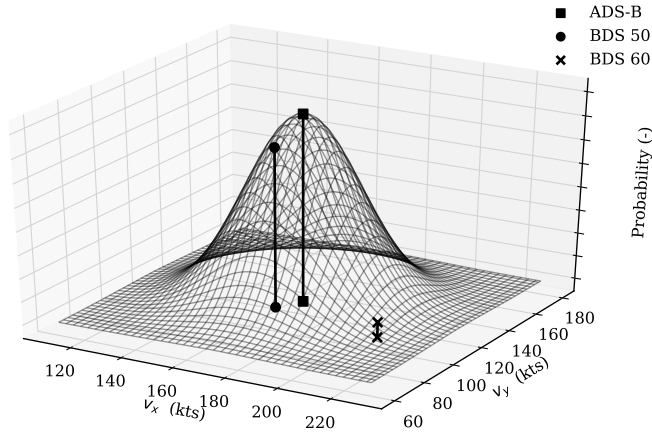


Figure 2.7: Example of BDS 50/60 identification

2.5. Experiment and validation

In this section, we use three different datasets to verify, validate, and analyze the Mode-S Comm-B replies. The first two datasets are from the same hour, with one collected by our receiver (located at the TU Delft) and the other provided by the Air Traffic Control the Netherlands (LVNL). The last one is a global dataset provided by the ADS-B Exchange receiver network.

2.5.1. Experiment I: Examining the heuristic logic

This experiment focuses on examining the effectiveness of the HP process of the BDS identification presented in this chapter. A one-hour dataset collected by our receiver is used as a test set (from 12:00 to 13:00 UTC, September 7, 2017). The result of the heuristic logic is shown in Figure 2.8. In total, 1.5 million Comm-B replies are received during this one-hour period. The most common three BDS codes are 40, 50, and 60. In each bar plot of Figure 2.8, the hatch pattern indicates the corrupted messages.

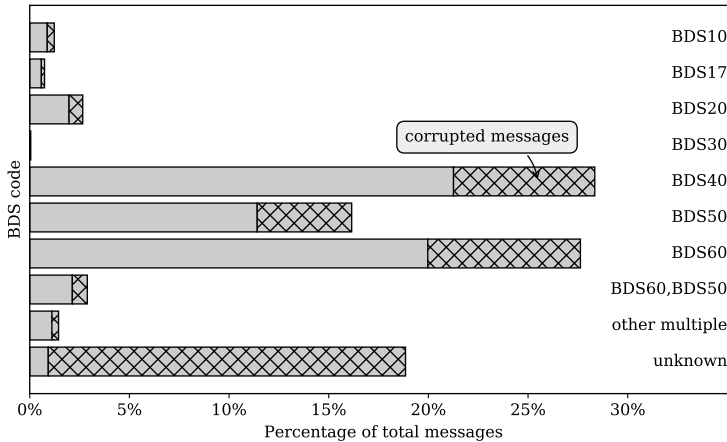


Figure 2.8: Heuristic BDS detection, based on Comm-B replies received from 12:00 to 13:00 UTC, September 7, 2017, Delft, the Netherlands

In this figure, we can see that around 20% of the messages are not identified (with unknown status). It is also noticeable that around 90% of these unidentified messages are corrupted.

For all other messages with a BDS code identified, BDS 10, 17, and 20 combined represent around 5% of the total number of messages. BDS 40 and 60 account for about 27% each and BDS 50 around 16%. Due to the similar message structures, around 4% of the messages identify as both BDS 50 and 60 codes. Less than 0.1% of messages are BDS 30. Since BDS 30 encodes emergency status and ACAS resolutions, it is expected that this category is less frequently interrogated during normal operations.

2.5.2. Experiment II: Examining the identification accuracy

To examine the accuracy of the HP process for BDS identification, we obtain a true reference dataset of the Dutch airspace from the Air Traffic Control the Netherlands (LVNL). This data is collected in the same hour as the previous test dataset. The original data is presented in ASTERIX format, from which we extract the raw messages. As such, raw messages only contain messages that are of BDS 40, 50 and 60 (Enhanced Mode-S only). Raw messages for other BDS types are not included in this ASTERIX dataset.

After applying both heuristic and probabilistic steps, we compute the accuracy at each stage. The results are shown in Table 2.6.

Table 2.6: Validation statistics of Mode-S BDS code identification using proposed *HP process* for a total of 253,059 messages. Unidentifiable messages refer to the ones with multiple BDS codes.

Result	Heuristic logic	Probabilistic id
Correct	235,772 (93.169%)	247,192 (97.682%)
Unidentifiable	17,281 (6.829%)	5847 (2.311%)
Incorrect	6 (0.002%)	20 (0.008%)

Using only heuristic logic, we achieve a correctness rate of 93.2% for a total of 250 thousand messages. 6.8% of the messages have been identified with more than one BDS code. Using additional probabilistic identification, we can further reduce the rate of uncertain messages to 2.3% and increase the success rate to 97.7%.

In each step, the number of incorrectly decoded messages is almost negligible. We have counted that only 6 (heuristic step) and 20 (probabilistic step) out of 253,059 messages were incorrectly identified. Regarding the significance of the probabilistic step, we found that among all 11,434 identifiable messages (with BDS 50 and BDS 60 combination) only 14 errors occurred. Upon further investigation, we notice that errors happen often when part of the flight information is missing from the message.

2.5.3. Experiment III: Analyzing the global data

Using multiple receivers provided by the ADS-B Exchange receiver network, we are able to examine the decoding performance of Mode-S Comm-B data across different regions of the world. A set of data consisting of one hour (local time 10:00 AM to 11:00 AM, on 24 or 25 August of 2018) is collected at eight different locations around the world. Using the method proposed in this chapter, we decode the messages and show the composition of Mode-S responses for these different regions. In Figure 2.9, the percentages of BDS codes are illustrated.

Mode-S data transmitted to ADS-B Exchange network is filtered by default. Only messages with s_2 state in Figure 2.5 are kept. These are Comm-B messages with ICAO addresses that appear in ADS-B data. In Table 2.7, the exact percentages of BDS codes in Comm-B responses are listed. In all eight regions, the unidentifiable messages account for between 1% to 6% of received messages. This result is in line with the test performed using our own receiver after corrupted messages are discarded (as shown in Figure 2.8). In most of these regions, common interrogated BDS codes are BDS40, BDS50, and BDS60, with the exception of Dallas, USA and Tel Aviv, Israel. The diversity of interrogations is one of the most challenging elements in the inference process. However, the proposed methods in this chapter are able to cope with interrogation variations and produce a large percentage of identifiable messages.

Table 2.7: Mode-S Comm-B response summary in different regions

Country	City	BDS10	BDS17	BDS20	BDS30	BDS40	BDS50	BDS60	EMPTY	MULTIPLE	UNKNOWN
Australia	Queenland	0.3%	0.4%	1.1%	-	31.7%	31.5%	32.9%	0.1%	0.3%	1.7%
Australia	Wollongong	0.3%	0.2%	0.5%	-	28.9%	31.6%	32.5%	0.3%	0.6%	5.0%
Belgium	Brussels	2.3%	1.2%	4.0%	0.0%	36.0%	17.1%	34.5%	0.7%	1.2%	3.0%
Israel	Tel Aviv	4.5%	1.6%	28.5%	-	22.4%	21.4%	16.2%	0.7%	1.1%	3.7%
Russia	Pushkino	2.3%	3.1%	8.6%	0.8%	28.6%	27.2%	20.5%	-	4.2%	4.6%
UK	London	1.0%	1.7%	6.6%	0.0%	32.8%	20.6%	30.9%	0.5%	2.6%	3.1%
USA	Dallas	20.8%	59.2%	10.9%	-	-	1.0%	-	6.5%	-	1.6%
USA	San Francisco	9.3%	7.4%	6.5%	-	23.0%	19.9%	23.0%	4.1%	1.3%	5.5%

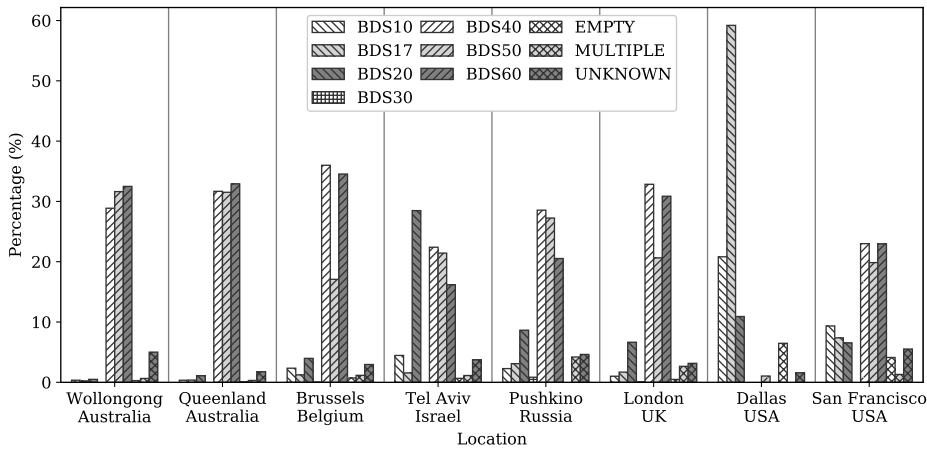


Figure 2.9: Statistics of BDS codes in Comm-B replies at different geographic regions

2.6. Discussion

This study aims at efficiently and accurately making use of open aircraft surveillance data. The focus has been on designing a path to deciphering Comm-B replies that are part of the Mode-S enhanced surveillance and cannot be decoded directly. The experiments and validations demonstrate that the identification process and error detection strategy that are proposed in this chapter are highly accurate. These refined identification methods provide the possibility to conduct aircraft performance studies with better accuracy when using open aircraft surveillance data.

Accurate identification of BDS 50 and 60 enables precise airspeed observations from Comm-B data. One possible extension of this research is in the area of atmospheric modeling. By combining ADS-B ground speed with Mode-S true airspeed, wind and temperature can be computed. This idea has been proposed in earlier research [49], where data are supplied by air traffic controllers. The methods and tools from this chapter allow anyone to gain such ability. In the Chapter 4 of this dissertation, different models are constructed based on wind and air temperature derived from this open data.

Figure 2.3 and 2.6 in earlier sections show the percentage of corrupted messages found in this study. This strongly suggests a frequency congestion problem on the 1090 MHz channel. Of all the ADS-B and Comm-B messages that we received, more than half are corrupted. Based on the percentage of errors, severe message corruption during daily operations is shown. In Figure 2.10, the percentage of corrupted ADS-B messages and the number of aircraft flying in a 24-hour period are plotted.

During the night time, when the airspace is less saturated, the percentage of corrupted messages decreases. However, the decrease in corrupted messages (~10%) is not proportional to the decrease in the number of flights (~65%). These corrupt messages not only present a challenge for obtaining more accurate

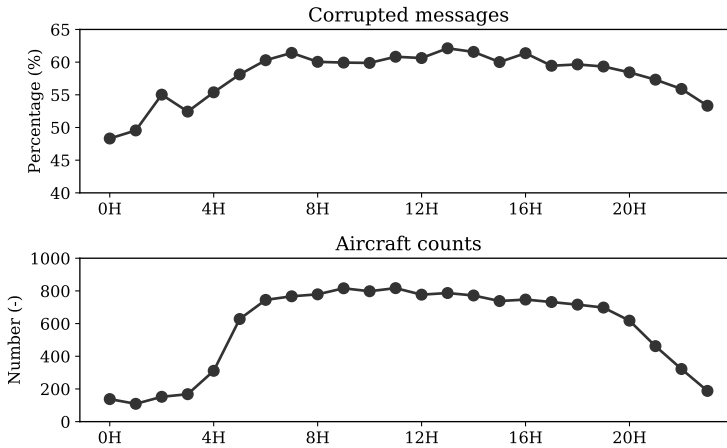


Figure 2.10: Percentage of corrupted ADS-B messages and number of aircraft per hour on May 30, 2018, in Delft, the Netherlands

data but also indicate constant frequency congestion in busy airspace.

Other than third-party researchers, air traffic controllers may also benefit from the methods proposed in this chapter. With a high level of frequency congestion as shown in Figure 2.8, air traffic controllers tend not to over-interrogate. This creates a dilemma where an ATC has to reduce the number and frequency of interrogations, but at the same time, more information is preferable for making better traffic control decisions. Using the proposed BDS identification process, one ATC can intercept replies that originate from interrogations by another ATC center. As one example, based on the one-hour reference dataset, we found that the replies from one specific secondary radar from the air traffic control of the Netherlands account for only around 17% of the total number of interrogation replies. The proposed identification method can therefore significantly increase the amount of information available to each ATC directly, even for aircraft that are outside of their airspace.

Finally, it is worth emphasizing that Mode-S Comm-B messages are passive replies originating from SSR interrogations. For oceanic and remote areas where secondary surveillance radars are not present, only ADS-B data is available. In this chapter, not all BDS codes are addressed. Instead, only the types related to ELS and EHS are investigated. However, this is the situation for many regions around the world, as illustrated in Figure 2.9 using the global Mode-S data.

2.7. Conclusion

In this chapter, we proposed a set of inference methods that allow third-party observers to identify and decode Mode-S Comm-B replies. The inference was based solely on surveillance replies without any knowledge of Mode-S interrogations. This chapter contributed to a missing area of knowledge on handling interrogation-based surveillance data. It gave researchers broader access to ac-

curate aircraft state updates that are transmitted through Enhanced Mode-S. The implementation was based on existing low-cost commercial off-the-shelf ADS-B receivers, with no additional hardware required. Using a reference dataset, the proposed process revealed a correctness rate of around 97.7%, with 2.3% unidentifiable and 0.008% error for Enhanced Mode-S messages.

Furthermore, the process proposed in this chapter could also be beneficial for air traffic controllers, since it would enable them to collect data for aircraft that are out of their controlled airspace, without the need for increasing interrogation frequency. Finally, we have made the decoding process into an open-source programming library, *pyModeS*, which includes the decoding and inferences discussed in this chapter. We hope this will also enable other researchers to make use of this valuable aircraft surveillance data source.

2.8. Acknowledgment

We would like to thank Air Traffic Control the Netherlands (LVNL) for providing us the Enhanced Mode-S validation dataset. We would also like to thank James Stanford from ADS-B Exchange for providing the data for global data analysis.

3

Flight trajectory processing

The previous chapter investigated the availability of surveillance data. We have decoded information from both ADS-B and Comm-B messages. With these large amounts of flight data gathered daily, we can apply them to a wide range of operational and aircraft modeling studies. Many of the studies in this dissertation rely on flight data from specific flight phases. This, therefore, requires methods to detect individual flight trajectories in the large amounts of recorded ADS-B data and divide them into data segments representing separate flight phases. The aim of this chapter is to derive a set of data mining methods for handling the large quantity of air traffic surveillance data. We first introduce an unsupervised machine learning algorithm that allows us to extract continuous flights from the scattered data points. Then, a fuzzy logic method is proposed to segment the flight into different flight phases. At the end of the chapter, a verification method is developed to examine the results of the flight phase identification process.

This chapter is based on the following publications:

- 1) Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2016, June. Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods. In 7th International Conference on Research in Air Transportation [127].
- 2) Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2017. Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance-Broadcast Datasets. Journal of Aerospace Information Systems, pp.566-572 [126].

3.1. Introduction

Automatic Dependent Surveillance-Broadcast (ADS-B) has been widely adopted in modern commercial aircraft. Flight state information such as the position, velocity, and vertical rate are broadcast by tens of thousands of aircraft around the world constantly. Identified by a 24-bit transponder address assigned by the ICAO, ADS-B data can be received and decoded with simple ground station set-ups. This large amount of open data enables many interesting topics in ATM research.

Many studies that rely on aircraft flight data require the knowledge of a continuous flight trajectory. Numerous cases focus on a specific segment of the flight, for example, during the climb, cruise, or descent. In this dissertation, most of the studies fall into this category, and therefore, an efficient method for extracting flights from ADS-B data, as well as segmenting these flights accordingly is necessary.

When dealing with a large flight dataset that can contain tens of thousands of flights, deterministic methods, such as identification of flight phases by fixed altitude or speed indicators, are often overgeneralized. Due to the large variety of aircraft types, exceptions to deterministic rules are common. At the same time, large fluctuations exist in climb rate, altitude, velocity, or a combination of these. This factor also needs to be handled properly.

In this chapter, instead of using deterministic logic to process and extract flight data, a set of more robust and versatile identification algorithms are proposed. A two-step method is designed and tested. Firstly, to extract individual flights from the loosely structured ADS-B data, a clustering algorithm is introduced, which is able to handle a large volume of data. Secondly, to segment these flights according to the flight phase, a fuzzy logic identification step is designed. It can be used to handle flight data generated by different aircraft types and flight procedures.

The structure of the chapter is designed as follows. Section two focus on flight extraction using the clustering algorithm. Section three describes the flight phase identification based on fuzzy logic. Section four uses a reference dataset and a heuristic method to verify the clustering and identification process respectively. Finally, discussion and conclusion are given in section five and six.

3.2. Flight extraction from large ADS-B datasets

3.2.1. Data fields

ADS-B information collected from aircraft is usually stored as loosely structured data points, which represent the states of aircraft at different times. Regardless of the data storage method, the data structure usually consists of the elements listed in Table 3.1.

For this study, we use a non-relational database, MongoDB, to store the ADS-B and flight data. It is an open-source data architecture frequently used for document-based big data processing [53].

When extracting continuous flights from a scattered ADS-B dataset, using the features in Table 3.1, two variables play a major role. These are the aircraft identification (ICAO address) and the timestamp. This is because a single flight can only

Table 3.1: Features of ADS-B flight data

Field	Type	Value range	Unit
ICAO address	string	-	-
Callsign	string	-	-
Timestamp	numeric	-	s
Latitude	numeric	[-180, 180]	deg
Longitude	numeric	[-90, 90]	deg
Altitude	numeric	[0, 40000]	ft
Speed	numeric	[0, 500]	kts
Heading	numeric	[0, 360]	deg

be carried out by a single aircraft. Each aircraft often carries out multiple flights during the cycle of a dataset, which is 24 hours in our study.

3.2.2. Simple flight extraction

Based on these characteristics, a straightforward *ad hoc* approach is developed. The method consists of following steps:

1. Filter and group data per aircraft by the unique ICAO address
2. Sort data of each group by timestamp
3. Examine time gaps between data points
4. Identify a general time gap which split flights carried by the same aircraft
5. Extract flights in each aircraft group

This method is simple to implement. However, since the data needs to be sorted and processed sequentially, the efficiency can decrease when dealing with very large datasets using this *ad hoc* method. In addition to this simple approach, a machine learning based clustering approach is designed.

3.2.3. Machine learning flight extraction

Using unsupervised machine learning, also known as clustering, on a large ADS-B dataset can have two significant benefits. Firstly, it can increase efficiency when dealing with many aircraft simultaneously. Secondly, it can handle outliers caused by irregularities in the flight data.

Pre-processing

Pre-processing steps are often required before applying any machine learning algorithm. First of all, any non-numerical data needs to be converted into numerical values. In addition, different features need to be scaled to a reasonable range and missing values need to be computed to complete the dataset.

Most clustering algorithms compute the Euclidean distances [94] between data points. A large range of values in the input features can cause the poor clustering performance. A simple method to mitigate this is to scale each feature

$X = \{x_0, x_1, \dots, x_n\}$ into a common range $[0, 1]$, where all values can be converted to $X' = \{x'_0, x'_1, \dots, x'_n\}$ as:

$$x'_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (3.1)$$

A numerical label encoder is used for text features such as ICAO addresses. However, converting these text features into numerical features implies a finite distance between any two different labels, which can affect clustering. In this study, to distinguish data from different aircraft, the numerical representation of the ICAO address is scaled to a different range $[0, N]$ where N is greater than 1.

Clustering

Unsupervised clustering algorithms group data into subsets (or clusters) based on the feature differences among data points. Many well-known algorithms, such as K-Means, DBSCAN, BIRCH, and Mean-Shift, have been developed in the computer science domain. Each of these algorithms has its own advantages for solving data particular feature sizes and geometries.

In this study, DBSCAN (density-based spatial clustering of applications with noise) [36] was selected due to its ability to handle an unknown number of clusters. It is also found to be efficient in handling outliers in the dataset. DBSCAN is a density-based clustering method, which separates data into areas of high and low density.

Two fundamental parameters are used in DBSCAN, which are *EPS* and *MinPTS*. The algorithm makes distinctions among three types of data points. These types are core points, reachable points, and outliers. *EPS* defines the maximum distance between two data samples for them to be considered in the same neighborhood. *MinPTS* is the number of data samples in the neighborhood of a core point. Clusters are formed as follows:

1. If more than *MinPTS* points are within a distance of *EPS* to p , then p is considered as a core point. These points are all defined as directly density-reachable from p .
2. A point q is reachable from p if a directly density-reachable path d_1, d_2, \dots, d_n exists, where d_1 and d_n are p and q .
3. From all above points, a cluster is formed.

Although it is not explicitly expressed in the original DBSCAN study [36], data points that are not density-reachable can be considered as outliers in our case. The ability to identify outliers offers a considerable advantage in processing ADS-B data, insomuch as it is preferable to exclude trajectories with low data quality systematically. This a key advantage over other types of clustering methods.

Figure 3.1 shows an example of the results of the DBSCAN method implemented on a small test dataset. From the first to last plot, the increasing *EPS* leads to increasing average cluster size. The increasing *MinPTS* eliminates clusters with

a few samples. The clustering process can be optimized by tuning the combination of these two variables. Performance benchmarking and parameter tuning is discussed in section 3.4.1.

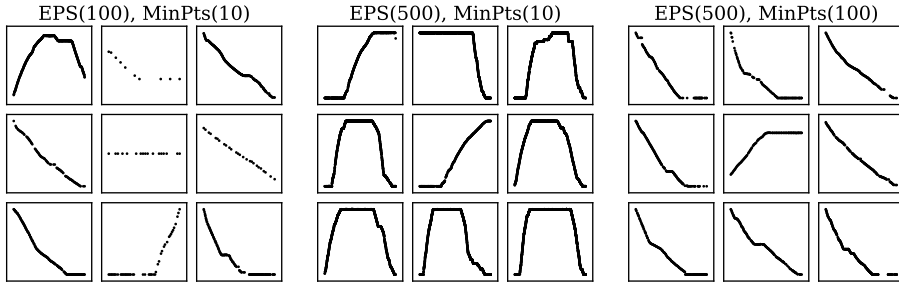


Figure 3.1: Clustering with DBSCAN. In each plot, the altitude profile of 9 random trajectories from all resulting trajectories are shown. They represent how the DBSCAN algorithm considers a set of data as a continuous trajectory with different parameter settings.

3.3. Flight phase identification

The outcome of the clustering process provides a set of continuous flights, representing either full or partial trajectories depending on the coverage of the ADS-B receiver. To segment a flight into different phases, previous clustering methods may still be used to create sub-clusters based on the characteristics of time-series data [41]. However, two problems arise when using the clustering method to group data by flight phases.

1) Each data point is relatively close to its neighbors in terms of the Euclidean distance that computed from timestamps, altitudes, velocities, and positions. A classic clustering method cannot produce sub-clusters with a sufficient level of consistency.

2) Due to differences between aircraft types and their divergent flight procedures, flight behavior may vary. This can result in, for example, aircraft climbing at different rates, flying at different cruise altitudes, and traveling at different speeds, even within the same flight phase.

In this study, these two problems are solved by a fuzzy logic identification method. Fuzzy logic, also known as fuzzy sets theory [146], has been introduced to express real-world objects or concepts where no precise definition of criteria exists. It uses membership functions to define the degree of truth for different features. Logic operators AND, OR, and NOT are defined as the minimum, maximum, and complement operators. Different output states are activated by user-defined logic.

3.3.1. Flight phase identification process

The entire segmentation process is presented in Figure 3.2. Continuous flight data is used as the input to the process. Then, it is sliced according to a small time

window (one minute in this study). The flight phase of each time window is identified by the fuzzy logic. Finally, the output that consists of a series of flight phase labels is produced sequentially or in parallel.

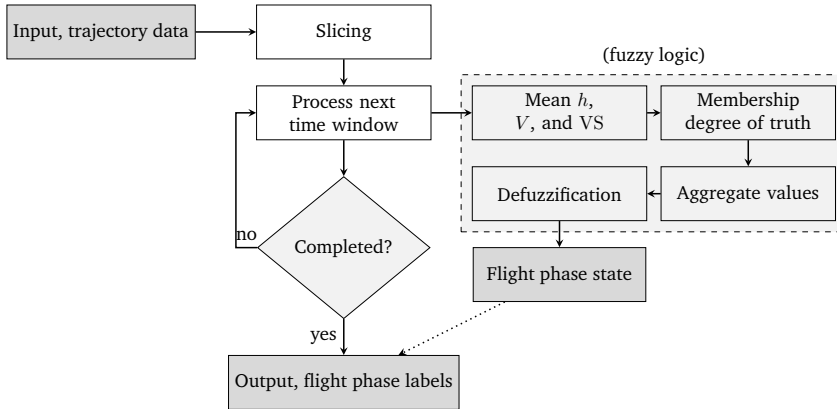


Figure 3.2: Flight segmentation with fuzzy logic

In this process, we can see that the trajectory data is first sliced into multiple time windows. The mean values of the altitude, speed, and vertical rate are used for the data points that belong to the same time window. There are two advantages of this simple slicing strategy. Firstly, it reduces the fluctuations in data by averaging. Secondly, it reduces the number of iterations for the entire trajectory and improves the speed of computation.

3.3.2. Membership functions

In this study, three inputs are used (i.e., altitude, speed, and vertical rate) to determine the flight phases. The membership functions for these input parameters and output flight phase is illustrated in Figure 3.3

Most of the membership functions are defined as Gaussian functions (denoted as \mathcal{G}), where the mean μ and standard deviation σ reflect the reasonable value and range of uncertainty:

$$\mathcal{G}(x; \mu, \sigma) = \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right) \quad (3.2)$$

Other membership functions used are Z-shaped membership functions (denoted as \mathcal{Z}) and S-shaped membership functions (denoted as \mathcal{S}), and are defined as follows:

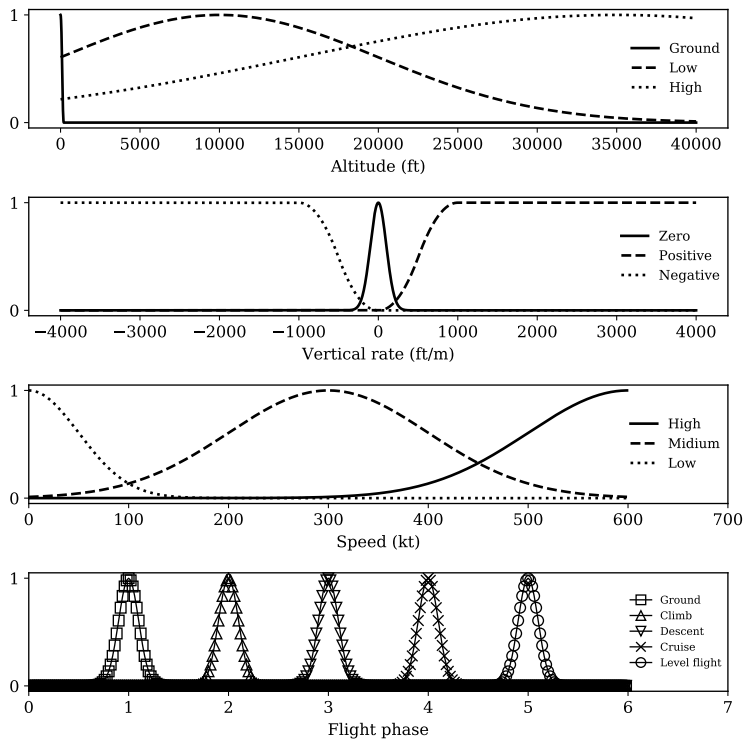


Figure 3.3: Fuzzy logic membership functions

$$\mathcal{Z}(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2 \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2 \left(\frac{x-b}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b \end{cases} \quad (3.3)$$

$$\mathcal{S}(x; a, b) = \begin{cases} 0, & x \leq a \\ 2 \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2 \left(\frac{x-b}{b-a} \right)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases} \quad (3.4)$$

Here, a and b (or b and a) are the high and low extremes of the sloped part of the function curve. Each membership function from Figure 3.3 is constructed with appropriate parameters as follows:

$$H_{\text{gnd}}(x) = \mathcal{Z}(x, 0, 200) \quad (3.5a)$$

$$H_{\text{lo}}(x) = \mathcal{G}(x, 10000, 10000) \quad (3.5b)$$

$$H_{\text{hi}}(x) = \mathcal{G}(x, 35000, 20000) \quad (3.5c)$$

$$VS_0(x) = \mathcal{G}(x, 0, 100) \quad (3.5d)$$

$$VS_+(x) = \mathcal{S}(x, 10, 1000) \quad (3.5e)$$

$$VS_-(x) = \mathcal{Z}(x, -1000, -10) \quad (3.5f)$$

$$V_{\text{lo}}(x) = \mathcal{G}(x, 0, 50) \quad (3.5g)$$

$$V_{\text{mid}}(x) = \mathcal{G}(x, 300, 100) \quad (3.5h)$$

$$V_{\text{hi}}(x) = \mathcal{G}(x, 600, 100) \quad (3.5i)$$

$$P_{\text{gnd}}(x) = \mathcal{G}(x, 1, 0.2) \quad (3.5j)$$

$$P_{\text{clb}}(x) = \mathcal{G}(x, 2, 0.2) \quad (3.5k)$$

$$P_{\text{cru}}(x) = \mathcal{G}(x, 3, 0.2) \quad (3.5l)$$

$$P_{\text{des}}(x) = \mathcal{G}(x, 4, 0.2) \quad (3.5m)$$

$$P_{\text{lvl}}(x) = \mathcal{G}(x, 5, 0.2) \quad (3.5n)$$

where H_{gnd} , H_{lo} , and H_{hi} represent the membership functions for ground, low altitude, and high altitude. VS_0 , VS_+ , and VS_- represent the membership functions for zero, positive, and negative vertical rates. V_{lo} , V_{mid} , and V_{hi} represent the membership functions for low, medium, and high speeds. Finally, P_{gnd} , P_{clb} , P_{cru} , P_{des} , and P_{lvl} represent the flight phases of ground, climb, cruise, descent, and level flight.

3.3.3. Identification logic

With the membership functions for altitude, speed, and vertical rate defined, the following relationships can be used to map the flight phases:

$$H_{\text{gnd}} \text{ and } V_{\text{lo}} \text{ and } VS_0 \Rightarrow \text{Ground} \quad (3.6a)$$

$$H_{\text{lo}} \text{ and } V_{\text{mid}} \text{ and } VS_+ \Rightarrow \text{Climb} \quad (3.6b)$$

$$H_{\text{hi}} \text{ and } V_{\text{hi}} \text{ and } VS_0 \Rightarrow \text{Cruise} \quad (3.6c)$$

$$H_{\text{lo}} \text{ and } V_{\text{mid}} \text{ and } VS_- \Rightarrow \text{Descent} \quad (3.6d)$$

$$H_{\text{lo}} \text{ and } V_{\text{mid}} \text{ and } VS_0 \Rightarrow \text{Level flight} \quad (3.6e)$$

Fuzzy logic takes such relationships between inputs and output to identify the five different flight phases (ground, climb, cruise, descent, and level flight during climb and descent), for a given data point, denoted as (h_i, v_i, vs_i) , and all possible discrete flight phase states P ($0 < P_i < 6$), as shown in the last plot of Figure 3.3. Each fuzzy value (the numerical representation of phase) can be calculated

as follows:

$$S_{\text{gnd}}(\mathbf{P}) = \min \left\{ \min \left[H_{\text{gnd}}(h_i), V_{\text{lo}}(v_i), \text{VS}_0(\text{vs}_i) \right], P_{\text{gnd}}(\mathbf{P}) \right\} \quad (3.7a)$$

$$S_{\text{clb}}(\mathbf{P}) = \min \left\{ \min \left[H_{\text{lo}}(h_i), V_{\text{mid}}(v_i), \text{VS}_+(\text{vs}_i) \right], P_{\text{clb}}(\mathbf{P}) \right\} \quad (3.7b)$$

$$S_{\text{cru}}(\mathbf{P}) = \min \left\{ \min \left[H_{\text{hi}}(h_i), V_{\text{hi}}(v_i), \text{VS}_0(\text{vs}_i) \right], P_{\text{cru}}(\mathbf{P}) \right\} \quad (3.7c)$$

$$S_{\text{des}}(\mathbf{P}) = \min \left\{ \min \left[H_{\text{lo}}(h_i), V_{\text{mid}}(v_i), \text{VS}_-(\text{vs}_i) \right], P_{\text{des}}(\mathbf{P}) \right\} \quad (3.7d)$$

$$S_{\text{lvl}}(\mathbf{P}) = \min \left\{ \min \left[H_{\text{lo}}(h_i), V_{\text{mid}}(v_i), \text{VS}_0(\text{vs}_i) \right], P_{\text{gnd}}(\mathbf{P}) \right\} \quad (3.7e)$$

$$S(\mathbf{P}) = \max \left[S_{\text{gnd}}(\mathbf{P}), S_{\text{clb}}(\mathbf{P}), S_{\text{cru}}(\mathbf{P}), S_{\text{des}}(\mathbf{P}), S_{\text{lvl}}(\mathbf{P}) \right] \quad (3.7f)$$

where $S(\mathbf{P})$ is the combined fuzzy value computed according to the membership logic.

The last step is known as *defuzzification*, where the most likely flight phase state \hat{P} can be found at:

$$\hat{P} = \text{round} \left[\arg \max_P S(P) \right] \quad (3.8)$$

where \hat{P} represents the final output where the highest combined fuzzy value occurs. Finally, the numerical flight phase representation can be converted to human readable flight phase text.

To visualize the outcome, an example flight trajectory is applied with this flight phase identification process. By inspecting the results shown in Figure 3.4, we can confirm the correctness of the phase identification.

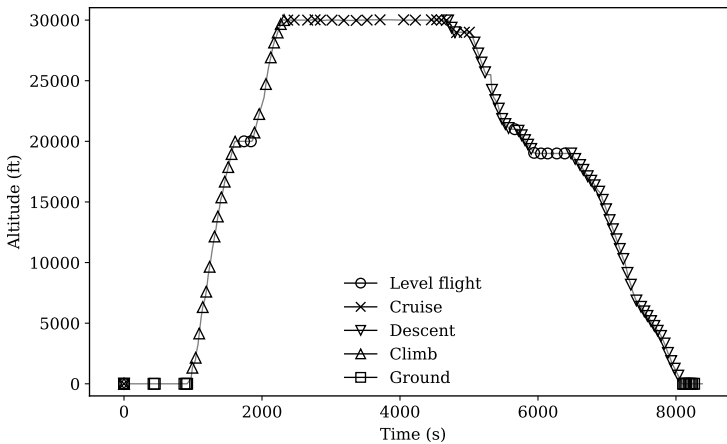


Figure 3.4: Example of the fuzzy logic flight phase identification

3.4. Verification of the algorithms

3.4.1. Benchmarking of the clustering methods

The quality of a clustering outcome not only depends on choosing the right machine learning method but also on setting the proper parameters. To evaluate the algorithm thoroughly, a grid of parameters is used to benchmark the outcome of DBSCAN clustering algorithms. A test dataset is constructed using an aggregated dataset that consists of 518 flights extracted from *FlightRadar24*. By searching the parameter grid, it is possible to locate the best pair of EPS and $MinPTS$ of DBSCAN that produces the corresponding result.

Figure 3.5 shows the benchmark results. The two axes of the figure represent the parameters that are to be tuned. The circular areas represent the number of clusters found by using different parameter pairs.

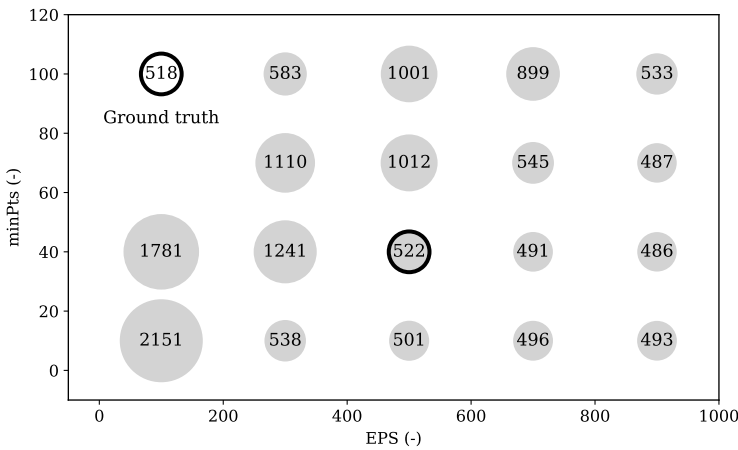


Figure 3.5: DBSCAN benchmark. The best parameter $EPS/minPts$ pair (500/40) is identified after comparing the resulting number of flights with the ground truth.

In this figure, the parameter pair yielding the best performance can be easily identified. By comparing the outcomes with this small ground truth dataset, identified parameters can be applied on other large datasets.

3.4.2. Examination of flight phases

To validate the quality of the flight-phase identification process, a heuristic method is processed. Two proposed indicators are:

1) The number of phase transitions (N_{Trans}): This indicator is calculated by comparing the phase of two adjacent data points and summing the number of differences. Statistics of such a parameter on a large number of trajectories are used as a first evaluation.

2) The number of invalid transitions (E_{Trans}): A transition can only occur between certain phase states. The state diagram in Figure 3.6 shows the possible transitions. Transitions that are not connected by arrows are considered as invalid

transitions. GND, CL, DE, CR, and LVL represent the flight phase of ground, climb, descent, cruise, and level flight respectively.

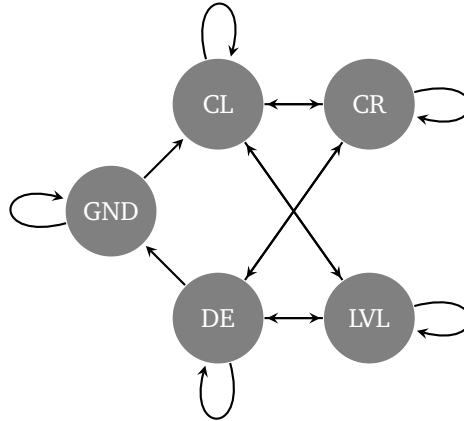


Figure 3.6: Flight phase state diagram

A second test dataset of 500 complete end-to-end flights is drawn from the database. N_{Trans} and E_{Trans} are calculated for all segmentation labels, shown in Figure 3.7. The majority of flights contain less than 10 transitions.

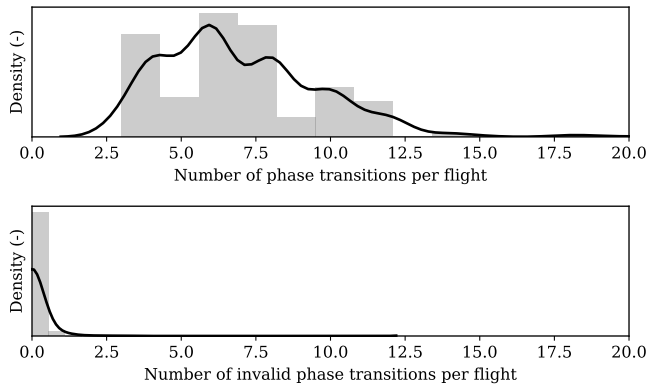


Figure 3.7: Evaluation of flight phase identification for 500 flights. Normalized density is shown in each plot. The values indicate the number of phase transitions of each flight in the dataset.

By analyzing the results, we find that most of the flights have zero invalid phase transitions, accounting for more than 95% of all flights. The total number of E_{Trans} is as low as 0.007%, which represents around 40 out of nearly 600,000 data points.

3.5. Discussion

3.5.1. Data storage

An operational system relies on a reliable data storage infrastructure. The best choice to store flight trajectories is up for debate. Options range from the simple comma separated plain text file to distributed data system (for example, Apache Hadoop [120]) with integrated data mining and machine learning plug-ins. The final choice, in the end, depends on the scope of the study or application, as well as the amount of data processing tasks.

In this dissertation, the plain text files are used to store raw and decoded ADS-B data, while MongoDB is used to store the trajectories. This choice is a tradeoff between portability and computational efficiency.

3.5.2. Tuning of clustering algorithms

Unsupervised machine learning algorithms like clustering methods require no training. However, their performance can vary not only depending on the actual clustering method but also depending on their respective tuning parameters.

In DBSCAN, the *EPS* parameter indicates the maximum distance between two samples for them to be considered in the same neighborhood. A larger *EPS*, in general, produces fewer clusters. *MinPTS* defines the number of samples in the neighborhood of a data point to be considered as the core point, making it a good parameter to eliminate the noise in a dataset.

Very often, empirical knowledge is used to determine values for these parameters. Applying a parameter grid search on a reference dataset can be a good approach to find out the best parameter values.

3.5.3. Flight phase identification limitations

One limitation of the segmentation process is that the system cannot currently separate flight data into further detailed flight phases, such as taxiing, takeoff, landing, initial climb, and final approach. This is because of the short time duration for these flight phases. In the rest of the dissertation, more deterministic approaches are applied to further extract data from these segments.

The fuzzy logic method that is introduced in this chapter relies on a small time window of data to average out the noise. In situations with low data frequency or higher noise levels, the performance may decrease.

3.6. Conclusion

In this chapter, we presented a two-step machine learning approach for processing ADS-B flight data. A clustering method and a fuzzy logic method were proposed to extract flights and segment them in individual flight phases respectively. The proposed methods were robust in handling different aircraft types and flight patterns. We verified this two-step approach using a reference dataset and a heuristic method to ensure a reasonable and expected result.

It is worth noting that takeoff and landing were both identified as the ground

phase in this chapter. However, considering the change in speed, one could easily further identify the takeoff and landing phase.

One issue that could influence the performance of the segmentation is noise in the measurement data. Features such as speed and vertical speed can show large fluctuations. A median filtering [72], Savitzky–Golay filter [115], or piece-wise polynomial splines could be used to smooth the data before applying the flight phase identification.

To reduce the fluctuations of variables and to shorten the computation time, we introduced the trajectory slicing with a short fixed time window. It was found to be an effective technique to improve the efficiency and accuracy of the fuzzy logic identification process.

Based on the results, we can conclude that the two-step machine learning approach proposed in this chapter is able to conveniently handle large amounts of ADS-B flight data. In addition to the method depicted in this chapter, we also published an open-source toolkit ¹ for the convenience of those undertaking future studies.

¹Available at: <https://github.com/junzis/flight-data-processor>

Part II

Models

4

Meteorological model

Wind and temperature data are important parameters in aircraft performance studies. The lack of accurate measurements of these parameters forces researchers to rely on numerical weather prediction models, which are often filtered for a larger area and thus offer decreased local accuracy. Aircraft, however, also transmit information related to weather conditions, in response to interrogation by air traffic controller surveillance radars. Although not intended for this purpose, aircraft surveillance data contains information that can be used for weather models. This chapter presents a method to reconstruct a weather field from surveillance data. Throughout the chapter, we answer two main research questions: how to accurately infer wind and temperature from aircraft surveillance data, and how to reconstruct a real-time weather grid efficiently. We consider aircraft as moving sensors that measure wind and temperature conditions indirectly at different locations and flight levels. To address the first question, aircraft barometric altitude, ground velocity, and airspeed are decoded from downlinked surveillance data. Then, temperature and wind observations are computed based on aeronautical speed conversion equations. To address the second question, we propose a novel Meteo-Particle (MP) model for constructing the wind and temperature fields.

This chapter is based on the following publications:

- 1) Sun, J., Vũ, H., Ellerbroek, J. and Hoekstra, J.M., 2017. Ground-based Wind Field Construction from Mode-S and ADS-B Data with a Novel Gas Particle Model. Proceedings of the Seventh SESAR Innovation Days, 28, p.30th [132].
- 2) Sun, J., Vũ, H., Ellerbroek, J. and Hoekstra, J.M., 2018. Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model. PLoS One, 13(10), p.e0205029 [134].

4.1. Introduction

Using aircraft as weather sensors is a recent development in air traffic management and meteorological research. Traditionally, aircraft obtain weather updates from Air Traffic Services, which they use to optimize their trajectory and speed, to best adapt to wind conditions, and to avoid areas of extreme conditions. These meteorological updates come mostly from ground-based observations, such as radar surveillance, weather observation stations, or forecast systems [113] [74]. In addition to this, local meteorological conditions are also computed by the aircraft, using observations from onboard air data sensors. Existing technologies such as Aircraft Meteorological Data Relay (AMDAR) [96] and Meteorological Routine Air Report (MRAR) allow aircraft to downlink these meteorological data either through the Aircraft Communications Addressing and Reporting System (ACARS) or using a technology called Selective Interrogation (Mode-S).

Both AMDAR and MRAR are unencrypted broadcast data, which means that anyone can set up receivers to intercept these data. However, as part of ACARS, the legality of intercepting AMDAR is questionable in certain countries. As for MRAR, the number of aircraft that broadcast this information is limited, since most aircraft choose not to enable the MRAR capability. Furthermore, this information is not always interrogated by surveillance radars.

In addition to the explicitly transmitted meteorological data, aircraft state information can also be used to infer local meteorological conditions. The state information is traditionally acquired using the primary surveillance radar for aircraft position, complemented by the secondary radar, which interrogates aircraft to obtain onboard data. Several studies have proposed to use such flight data to estimate wind conditions at the location of an aircraft. We can categorize these studies into three categories:

1) Estimation of wind from ground-based trajectory observations: This concept assumes a quasi-constant wind velocity and aircraft airspeed during turning maneuvers. Under these assumptions, the wind velocity vector can be estimated dynamically using observations of aircraft ground speed in combination with Bayesian filtering. A method using radar track data, based on this concept, was first proposed in 1989 [58]. Later on, variations and extensions of the method were implemented [29] [31] [30]. Recently, the use of ADS-B data for weather determination is being explored [81] [71].

2) Estimation of aircraft local weather conditions from interrogated aircraft data: To provide more aircraft state information to air traffic controllers, Mode-S secondary radar surveillance was developed as a complementary source of information to radar. Mode-S is designed to interrogate specific aircraft states individually, such as airspeed, intentions, and turn performance. A series of studies conducted by the Dutch Meteorological Institute presented wind data constructed from Mode-S and MRAR [49] [48]. Other research combined MRAR data and Kalman filtering to construct weather conditions [61]. In addition to the direct wind information in MRAR (which is rarely requested), the airspeed of aircraft is downlinked upon interrogation using Mode-S. This information can be used to compute wind as the difference between aircraft airspeed and ground speed [50]. Temperature, on the other hand, can be derived from ADS-B data alone, based

on the difference between Global Navigation Satellite System (GNSS) height and barometric altitude, which are both broadcasted with ADS-B [122].

3) Estimation of wind field based on multiple wind measurements: While most of the above studies focused on deriving the local meteorological conditions of an aircraft, other studies tried to extend such methods to wind field or multiple aircraft scenarios. For example, a Hidden Markov model was used to update a wind grid, based on measurements from multiple aircraft by [58]. [31] and [81] used non-linear Kalman filters on radar and ADS-B data respectively. The least-squares method was also employed to construct wind fields from multiple aircraft measurements by [62]. Finally, a concept using machine learning based on Gaussian Process was also proposed in [75].

The methods described above reveal the potential for using aircraft surveillance data, but they are not without limitations. For example, some studies considered only the wind, while other studies were based on data from air traffic controllers. Rarely has the potential of large quantities of streaming aircraft surveillance data been exploited to its full potential. Moreover, it is hard to identify a method that is fast to compute and easily reproducible.

In this chapter, we focus on two main research questions: 1) how to accurately infer both wind and temperature measurements from Mode-S and ADS-B data; 2) how to reconstruct the real-time weather grid efficiently. In addition to answering these two main questions, we also construct an open-source model that can be re-used for future research.

Weather data assimilation is a well-established area of research. Existing methods include Kalman filtering [59] and 3D/4D variational assimilation [7] [110]. Unlike traditional weather observations such as those from ground stations and weather balloons, there is an abundant amount of observation data being generated from flights. In our proposed model, temperature and wind are first computed based on the ADS-B and Mode-S downlink messages. Within the 400 kilometers radius of a typical ground receiver setup, it is common to obtain more than 50 weather observations per second with normal operational air traffic density. The characteristics of weather observations from aircraft are:

1. Aircraft are moving objects. The measurements derived from air traffic data vary both in time and position.
2. Aircraft often fly along predefined routes. As such, measurements are concentrated along a limited number of flight paths. This creates a highly uneven distribution of measurements in space.
3. The interval between successive measurements is small, which can be in the order of a single second.
4. Different sources, such as measurement error (aircraft), transmission error (data link), and identification error (decoder), cause errors in individual wind calculations.

Considering these characteristics of the data source, in this chapter, we propose a new mode: the Meteo-Particle (MP) model, which is able to construct accurate

local wind and temperature fields. The model is aimed at providing estimation within aviation airspace with a confidence indicator using only aircraft surveillance data. The fundamental idea of the model is to use a stochastic process to extend weather information (modeled as particles) from high-density flight paths to areas without aircraft observations. The system works on a short timescale, which can be from several minutes to an hour. It is constructed and maintained with a probabilistic nature incorporating both historical and recent weather information. The model is first validated with weather data from numerical forecast models. The variance and stability of the MP model are tested, and the error tolerance is also examined.

Before introducing the details of the MP model, two existing concepts using similar terminology need to be distinguished. These concepts are particle filtering [97] and Lagrangian transportation modeling [82]. The particle filter, also called Sequential Monte Carlo, is a system state estimator, and particles are numerical approximations of probability density functions of the states. The Lagrangian transportation model is commonly used to simulate atmospheric chemistry, where the particles' propagation follows atmospheric dynamics, such as fluid and heat. In the proposed MP model, the particles can be considered as the information medium, which propagates the wind and temperature measurements to surrounding areas. The propagation of MP particles is a stochastic random walk process.

The remainder of the chapter is structured as follows. Section two describes the process of obtaining wind observations. Section three focuses on the Meteorological Particle model with examples. Section four discusses the ability to make short-term predictions based on the model. Section five and six detail experiments and validations and provides an analysis of the MP model. The discussion and conclusions are presented in sections seven and eight.

4.2. Meteorological observations from Mode-S data

The Automatic Dependent Surveillance-Broadcast (ADS-B) is an aircraft surveillance technology that enables aircraft to automatically broadcast flight states such as location, altitude, and ground speed. In contrast to conventional surveillance technologies, ADS-B enables information exchange without the need for interrogation from air traffic controllers. Information is broadcast approximately every half of a second, providing a fast update rate of the aircraft states. A downside from the perspective of weather estimation, however, is that within ADS-B messages, only ground speed is transmitted. For the computation of wind, we also need to obtain the airspeed of the aircraft. Here, ground speed refers to the relative speed between the aircraft and the earth, while the airspeed refers to the relative speed between the aircraft and the air. Without wind these two speeds are equal. When the wind is present, it can be calculated as the difference between ground speed and airspeed.

The airspeed can be acquired from the Comm-B response messages that are generated by selective interrogations (Mode-S) from the secondary surveillance radar. However, unlike ADS-B, Mode-S Comm-B replies do not contain any information on their message types. This is because only the interrogating radar knows the target aircraft and what to expect in the downlinked messages. This

lack of transparency in the Mode-S design poses the biggest challenge in making use of this type of open data. In Chapter 2, methods were developed to extract information from Comm-B messages.

4.2.1. Accurate models of temperature and wind speed

With ADS-B messages and Mode-S Comm-B replies, one can derive multiple features of any aircraft state. This information can be used to compute the meteorological conditions (temperature and wind) indirectly but accurately. The necessary aircraft states are:

1. **Aircraft barometric altitude** (H_p): broadcast through ADS-B (Type code: 9 to 18), high update rate, high certainty.
2. **Aircraft ground speed** (V_{gs}): broadcast through ADS-B (Type code: 19), high update rate, high certainty.
3. **Aircraft true airspeed** (V_{tas}): transmitted in Mode-S Comm-B BDS 50 message, low update rate, lower certainty than V_{gs} .
4. **Aircraft indicated airspeed** (V_{ias}): transmitted in Mode-S Comm-B BDS 60 message, medium update rate, lower certainty than V_{gs} .
5. **Aircraft Mach number** (M): transmitted in Mode-S Comm-B BDS 60 message, medium update rate, lower certainty than V_{gs} .

Denoting the p , ρ , T , and V_w as the atmospheric pressure, air density, air temperature, and wind speed, the inference procedure is shown in the flow diagram of Figure 4.1.

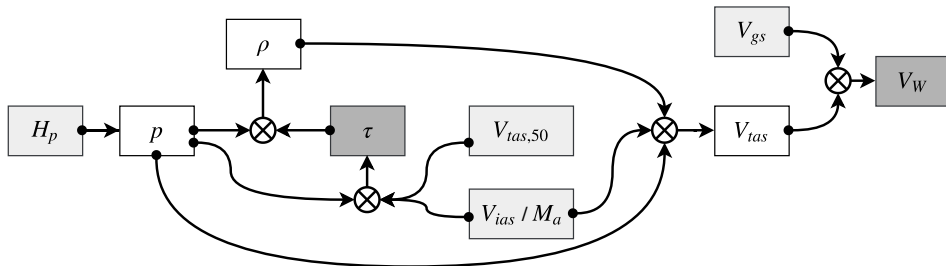


Figure 4.1: Relationships among different aircraft performance parameters, wind, and temperature

The light gray blocks represent the observable aircraft states, and the white blocks are intermediate atmospheric or aircraft states. Since the update rate of true airspeed from BDS 50 messages is low, true airspeed converted from indicated airspeed in BDS 60 messages is also used. However, such a conversion requires knowledge of the air temperature, which can be computed using true airspeed from BDS 50 messages in combination with the air pressure. The air pressure can be derived using the barometric altitude from ADS-B using the following equation:

$$p = \begin{cases} p_b \left[\frac{\tau_b + \lambda(h - h_b)}{\tau_b} \right]^{\frac{-g}{\lambda R}} & \lambda \neq 0 \\ p_b \exp \left[-\frac{g}{R\tau_b}(h - h_b) \right] & \lambda = 0 \end{cases} \quad (4.1)$$

where h is the pressure altitude of the aircraft. λ is the lapse rate representing the ISA temperature gradient (-6.5 K/km for troposphere). h_b is the base altitude, which is 0 km or 11 km for troposphere and stratosphere respectively. p_b is the base pressure, which is 101,325 Pa or 22,632 Pa for troposphere and stratosphere respectively. T_b is the base temperature, which is 288.15 K or 216.65 K for troposphere and stratosphere respectively.

The temperature can then be computed based on the speed conversion models among equivalent airspeed¹, Mach number and true airspeed:

$$T = \begin{cases} \frac{V_{\text{tas},50}^2 \cdot p}{V_{\text{ias}}^2 \cdot \rho_0 \cdot R} & M < 0.3 \\ \frac{V_{\text{tas},50}^2 \cdot T_0}{M^2 \cdot a_0^2} & M \geq 0.3 \end{cases} \quad (4.2)$$

where $V_{\text{tas},50}$ is the true airspeed from the BDS 50 message. Note that the indicated airspeed from BDS50 is used for low-speed flights, while the Mach number is used for high-speed flights. Once the temperature has been derived, the true airspeed can be computed similarly:

$$\begin{aligned} \rho &= \frac{p}{R T} \\ V_{\text{tas}} &= V_{\text{ias}} \sqrt{\frac{\rho_0}{\rho}} & M < 0.3 \\ V_{\text{tas}} &= M a_0 \sqrt{\frac{T}{T_0}} & M \geq 0.3 \end{aligned} \quad (4.3)$$

In summary, the above process first combines the data from BDS 50 (less frequent) and BDS 60 (more frequent) to derive the temperature of the air outside aircraft, which is a relatively constant value for a for the same flight level with a short range. Knowing the temperature and barometric altitude, we can compute more true airspeed samples using the more frequent BDS 60 messages.

After the true airspeed is obtained, together with the heading, ground speed, and track angle, the wind vector is derived. Denote χ_g , χ_a , and χ_w are the track angle, aircraft heading, and wind direction with respect to the true north. The ground speed vector is then computed as:

¹Mach number 0.3 is chosen considering the compressibility of the air. This is the condition where it is safer to assume indicated airspeed is equal to equivalent airspeed.

$$\vec{V}_w = \vec{V}_{gs} - \vec{V}_{tas} \quad (4.4)$$

where the wind vector is the subtraction of true airspeed from ground speed. Figure 4.2 shows the relationship between true airspeed, ground speed, and wind.

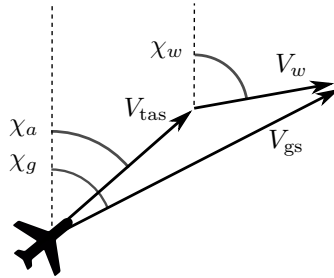


Figure 4.2: Relationship between true airspeed, ground speed, and wind vector

4.3. The Meteo-Particle model

In the proposed Meteo-Particle (MP) model, the particles are modeled to represent the states of wind and temperature measurement. Particles are first generated when a new observation of wind and temperature is obtained. They propagate and decay over time according to separate models. Wind fields are constructed by combining the weighted states of all neighboring particles. The propagation of particles allows for wind to be computed at areas with low measurement density. The following section will be dedicated to a more detailed explanation of the model, methods, and exponential functions used to compute wind field and confidences levels. In Figure 4.3, the general steps are illustrated.

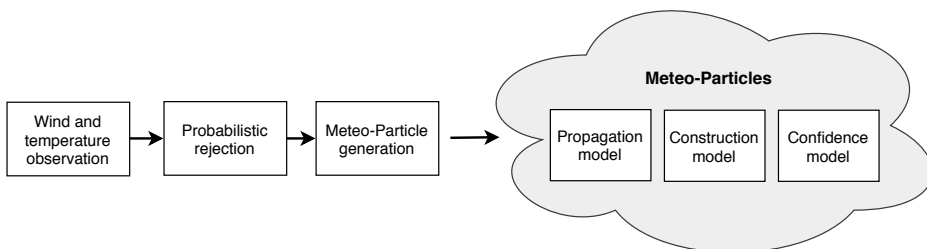


Figure 4.3: Steps and components of the MP model

4.3.1. Assumptions

The proposed Meteo-Particle model is developed based on the following assumptions:

1. *The true states of wind and temperature are generally stable in a local area (on the level of a few kilometers).* This assumption allows us to estimate weather conditions based on observations made in adjacent areas, with the exception of extreme phenomena such as turbulence, which cannot be represented accurately in the model.
2. *The true states of wind and temperature of a given location are stable at the level of minutes.* This assumption ensures that the dynamics of field states do not change too rapidly. This is usually true because aircraft avoid extreme atmospheric conditions as much as possible. Nevertheless, the MP model is able to track rapid local changes by reducing the aging parameter, but at the cost of stability for the larger airspace.
3. *The burst error rate of observations from a single aircraft is reasonably not too high.* The burst error is a continuous sequence of wrong measurements from incorrect aircraft states. This error is hard to eliminate due to the uncertainty in Mode-S Comm-B reply decoding. However, with the probabilistic measurement rejection of the MP model, we can reduce the effect of burst errors.

4

4.3.2. Measurements and probabilistic rejection

Measurements are done at aircraft position (x, y, z) , which is converted from the longitude, latitude, and altitude reported by the aircraft. Wind measurement is a vector represented by a west-east component (u) and a south-north component (v) at a specific location. The temperature is a scalar denoted as τ . The measurement array consists of all wind measurements from different aircraft at the time interval of one second in the observed airspace. It is denoted as $[\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}, \tau]$.

When wind and temperature samples are derived using ADS-B and Mode-S Comm-B messages, there is a chance that the wind measurements are incorrect. This often occurs due to incorrect decoding. Although the chance of incorrect data is low, such wrong information can cause sudden variations in instantaneous wind fields. To solve this problem, a probabilistic rejection mechanism is applied.

For each new measurement $\mathbf{x} : (u, v, \tau)$, a probability function is constructed based on the current field. First, the mean and variance of wind and temperature states from existing particles from the same vertical level (± 500 meters) are computed. These are denoted as (μ_u, σ_u^2) for the u component of the wind, (μ_v, σ_v^2) for the v component of the wind, and $(\mu_\tau, \sigma_\tau^2)$ for the temperature. The probability function is expressed as:

$$\begin{aligned}
 p &= \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T (k_1 \boldsymbol{\Sigma})^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right] \\
 \boldsymbol{\mu} &= (\mu_u, \mu_v, \mu_\tau) \\
 \boldsymbol{\Sigma} &= \begin{bmatrix} \sigma_u^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_\tau^2 \end{bmatrix}
 \end{aligned} \tag{4.5}$$

Hence, any new sample will be accepted with a probability of p . This extra step ensures a low probability of acceptance for extreme wind samples. On the other hand, due to its probabilistic nature, it will also decrease the number of correct samples that are accepted. As a trade-off, one can increase the parameter k_1 for a higher tolerance. We propose to choose k_1 as a value between 2 to 4.

4.3.3. Particles

A particle is defined as a point object that carries the state of the wind and temperature. Particle states consist of position (x_p, y_p, z_p) , origin (x_0, y_0, z_0) , horizontal wind components (u_p, v_p) , temperature, τ_p , and age (α) .

Particles are generated when new wind measurements are observed (computed). For each new measurement $[x, y, z, u, v, \tau]$, N number of particles are created at the location of the aircraft:

$$\begin{bmatrix} x_{p,i} \\ y_{p,i} \\ z_{p,i} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad i = 1, 2, \dots, N \quad (4.6)$$

The age of all particles is set to zero during the initialization. The carried states of the particles are assigned a small variance that represents the uncertainty of the wind measurement and temperature:

$$\begin{pmatrix} u_{p,i} \\ v_{p,i} \\ \tau_{p,i} \end{pmatrix} \sim \mathcal{N} \left(\begin{bmatrix} u \\ v \\ \tau \end{bmatrix}, \begin{bmatrix} \sigma_{u0}^2 & 0 & 0 \\ 0 & \sigma_{v0}^2 & 0 \\ 0 & 0 & \sigma_{\tau0}^2 \end{bmatrix} \right) \quad i = 1, 2, \dots, N \quad (4.7)$$

As an example, Figure 4.4 displays the wind vectors in solid arrows, as well as the generated particles in thin vectors (after the first propagation step). The plot shows the 2D projection of the X-Y plane, and only a small percentage of all particles are illustrated. The dashed circles indicate the variance of particle positions in relation to the measurement location.

4.3.4. Particle propagation

Particle motion follows a Gaussian random walk model. At each update step, the particle age (α) increases. The following equation describes the motion model of a particle:

$$\begin{pmatrix} x_{p,i,t+1} \\ y_{p,i,t+1} \\ z_{p,i,t+1} \end{pmatrix} = \begin{pmatrix} x_{p,i,t} \\ y_{p,i,t} \\ z_{p,i,t} \end{pmatrix} + \Delta P_t dt \quad i = 1, 2, \dots, N \quad (4.8)$$

$$\Delta P_t \sim \mathcal{N} \left(\begin{bmatrix} k_2 u_p \\ k_2 v_p \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{px}^2 & \sigma_{pxy} & 0 \\ \sigma_{pxy} & \sigma_{py}^2 & 0 \\ 0 & 0 & \sigma_{pz}^2 \end{bmatrix} \right)$$

The step factor ΔP is different in horizontal and vertical direction. Horizon-

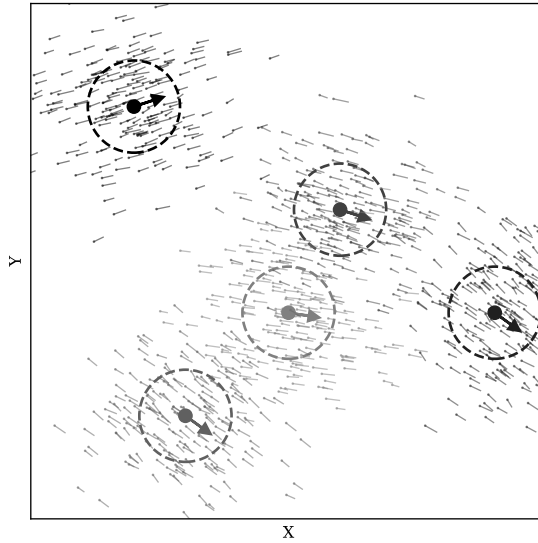


Figure 4.4: Wind measurements and particle initialization

tally, the terms $k_2 u_p$ and $k_2 v_p$ allow the random walk to be executed with a small bias along the direction of the wind, with a scaling factor k_2 . Choosing a larger k_2 allows the propagation to become more biased toward the wind direction. Vertically, the propagation follows a zero-mean Gaussian walk. The particle motion model is illustrated in Figure 4.5, where two projections (X - Y and X - Z) of a possible particle update are shown. The dot represents the position $(x_{p,t}, y_{p,t}, y_{p,t})$, while the probability of the next position $(x_{p,t+1}, y_{p,t+1}, y_{p,t+1})$ is shown by the contour plot. The vector equals to $\mathbb{E}[\Delta P_t]$.

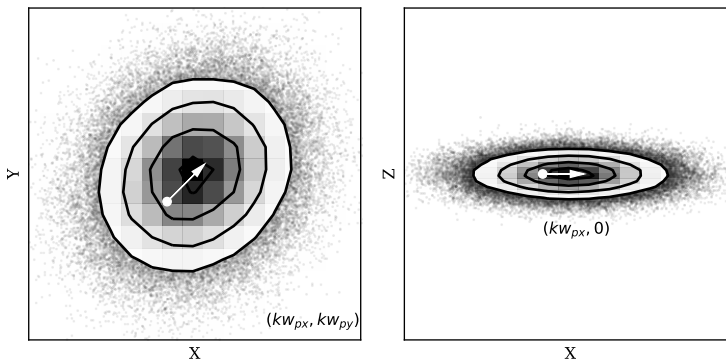


Figure 4.5: Random update of a particle position in a simulated scenario

The updates of particles follow the Gaussian random walk as shown in Figure 4.6, where several possible 100-step random walks of a particle (with origin

$[0, 0, 0]$) are illustrated. Different trajectories are distinguished by different shades of gray.

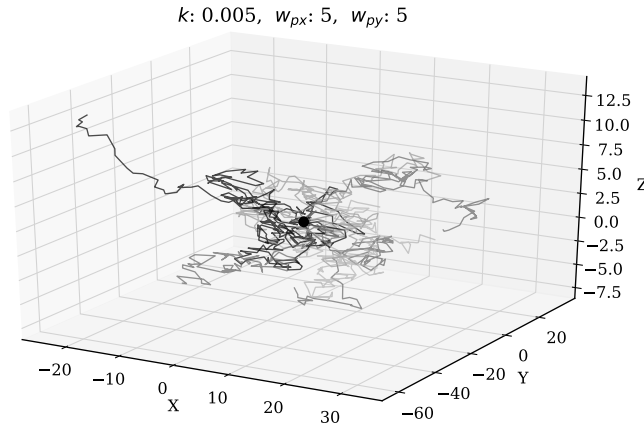


Figure 4.6: Particle random walks in 3D

4.3.5. Probabilistic re-sampling

At the end of each update, the particles are re-sampled. First, all particles that have propagated outside of the horizontal and vertical boundaries are removed. Then, all particles are sampled by age according to the probability computed in Equation 4.9:

$$p(\alpha) = \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right) \quad (4.9)$$

where α represents the age of the particles, and σ_α is the control parameter. This re-sampling maintains the number of particles in the system according to age. There are always more new particles than older particles in the entire system.

4.3.6. Information reconstruction

At any location, the wind and temperature information can be reconstructed using surrounding particles. Since wind and temperature are distributed differently, different formulas are used to compute their values. Let position (x, y, z) be the location where wind and temperature are to be computed. The wind is constructed using the weighted wind state values from the neighboring particles, denoted as P . A particle p with location (x_p, y_p, z_p) is considered in the set P if it is within the boundary of:

$$\begin{aligned}
x - x_b &\leq x_p \leq x + x_b \\
y - y_b &\leq y_p \leq y + y_b \\
z - z_b &\leq z_p \leq z + z_b
\end{aligned} \tag{4.10}$$

Subsequently, the wind at location (x, y, z) is computed as the weighted sum of the wind state information carried by the considered particles:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\sum_{p \in P} W_p} \cdot \sum_{p \in P} \left(W_p \cdot \begin{bmatrix} u_p \\ v_p \end{bmatrix} \right) \tag{4.11}$$

For temperature, we introduce an additional condition to the particle selection, in addition to Equation 4.10, to ensure that the origin of the particles is also at a similar altitude:

$$z - z_b \leq z_{p0} \leq z + z_b \tag{4.12}$$

After obtaining the reduced set of particles P^* , the computation of the temperature at location (x, y, z) also becomes a weighted sum of the temperatures from the considered particles:

$$\tau = \frac{1}{\sum_{p \in P^*} W_p} \cdot \sum_{p \in P^*} (W_p \cdot \tau_p) \tag{4.13}$$

In Equation 4.11 and 4.13, W_p is the weight of each particle that is computed based on the product of two exponential functions. Function $f_d(\cdot)$ expresses an exponential relationship the weight with the distance between the particle and the coordinate where wind and temperature need to be calculated. Function $f_0(\cdot)$ defines the weight with the distance of the particles from their origins:

$$W_p = f_d(d) \cdot f_0(d_0) \tag{4.14}$$

$$f_d(d) = \exp\left(-\frac{d^2}{2C_d^2}\right) \tag{4.15}$$

$$f_0(d_0) = \exp\left(-\frac{d_0^2}{2C_0^2}\right) \tag{4.16}$$

Here, d represents the spatial distance between a particle and the location of interest. C_d and C_0 are control parameters for the functions $f_d(\cdot)$ and $f_0(\cdot)$.

Figure 4.7 displays a re-constructed wind field from previously generated particles, at time-step zero. At each grid point, the wind vector is shown in solid arrows. Grid points with no information yet are presented in scattered circles.

It is worth pointing out that the MP model does not use a pre-defined grid. Particles are generated as weather measurements and are propagated independently. Values can be computed at any point or any set of points at the current time using

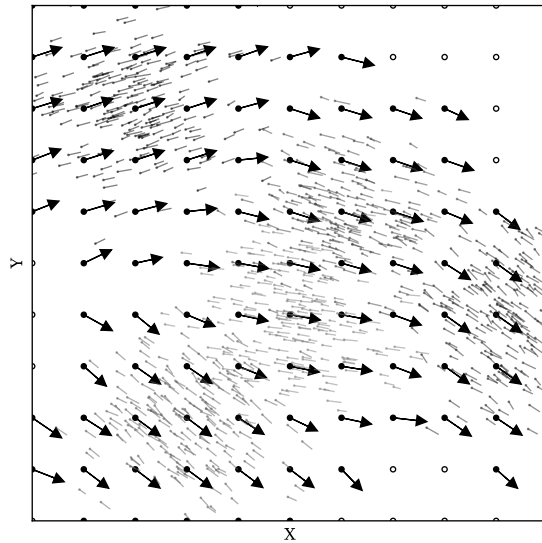


Figure 4.7: Wind field constructed from particles in a simulated scenario

Equation 4.11, as long as a sufficient number of particles exist in the neighborhood of these locations. In later experiments of this chapter, we chose a minimum of 10 particles.

4.3.7. Representation of confidence level

Once wind and temperature are reconstructed, it is also important to evaluate the confidence level of the estimates. The confidence level is computed as the combination of confidence functions that are based on several independent factors. These factors are:

1. The number of particles in the vicinity of the location of interest (N).
2. The mean distances between particles and the location of interest (D).
3. The homogeneity of states carried by particles (H).
4. The strength of the particles due to aging function (S).

Particle numbers and distances

The number of particles N represents the number of particles that are present within the calculation area. The mean distance to the point of interest is denoted as D . Higher confidence values are assigned to locations where more particles are present nearby. Areas that are far from flight trajectories tend to have fewer reachable particles and should yield lower confidence values.

Homogeneity of carried states

The level of homogeneity refers to the similarity between particle states. It essentially indicates whether different measurements propagated from a nearby area indicate similar evidence of wind vectors. The homogeneity of wind (H_w) is computed as the spectral norm of the covariance matrix of the wind vector components of the particles. For the temperature state, the homogeneity H_τ is simply represented by the variance of the temperature of the particles:

$$\begin{aligned}\Sigma &= Cov(\mathbf{u}_p, \mathbf{v}_p) \\ H_w &= \|\Sigma\|_2 = \sqrt{\lambda_{\max}(\Sigma^T \Sigma)} \\ H_\tau &= Var(\tau_p)\end{aligned}\quad (4.17)$$

Here, λ_{\max} represents the largest eigenvalue of a matrix, and \mathbf{u}_p , \mathbf{v}_p , and τ_p are wind and temperature states of the particles within the computing bound.

4

Particle strength

From the creation of a particle, its age (α) increases at each step of propagation. Since the particles are sampled at each update according to age, the strength parameter can simply be calculated as the fraction of the mean particle ages:

$$S = \frac{1}{\bar{\alpha}_p} \quad (4.18)$$

Normalized and combined confidence

Values from all four confidence factors have a distinct range. It is important to normalize these factors into the same range. Linear scaling is applied to convert all values of each factor into the (0, 1) range:

$$s(x) = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (4.19)$$

At any given time, the confidence vectors that represent all wind grid points are N , D , H , and S . Then, the combined confidence is expressed as:

$$\begin{aligned}C_w &= mean\{s(N), s(D), s(H_w), s(S)\} \\ C_\tau &= mean\{s(N), s(D), s(H_\tau), s(S)\}\end{aligned}\quad (4.20)$$

Figure 4.8 illustrates the confidence contour plot based on previously defined calculations. Areas in the plot in darker grays represent higher levels of confidence.

The confidence indicator is a relative value that can be compared within the field at any time instance, but it is not comparable between different time steps due to the normalization undertaken during its calculation.

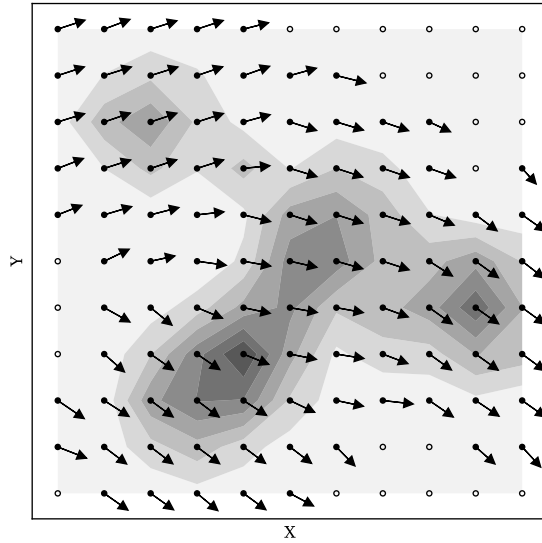


Figure 4.8: Wind field construction in a simulated scenario. Higher confidence is represented in darker gray.

4.4. Short-term prediction with the Meteo-Particle model

The previously MP model can rapidly construct the wind and temperature fields. However, the model alone is not able to make predictions of wind and temperature. In this section, we construct an additional predictor layer for wind (u , v component) or temperature as a univariate regression function that is based on particles in the MP model. The regression predictor requires the construction of a statistical model that is a function of time. For each location, surrounding particles are grouped by age, and the means are computed for each existing time step. Thus, we have the input data t and y_t .

In previous research, the Gaussian process regression (GPR) has been used in a similar fashion for the interpolation of wind conditions [75, 26]. A similar approach can be applied in the proposed MP model for short-term predictions. From a Bayesian perspective, the Gaussian process regression (also known as kriging in geostatistics) is an interpolation method. It can be considered as a form of Gaussian process prediction, which is based on a prior over functions and fitted over the observed data [144].

One way to understand the Gaussian process regression is to view functions as infinite-length vectors. The Gaussian process describes joint distributions over these infinite dimension vectors. The Gaussian process regression considers functions to be drawn from a prior that is defined by a mean function and a kernel (covariance) function. The prior can be formulated as follows:

$$f(t) = \mathcal{N}\{\mu(t), K(t, t')\} \quad (4.21)$$

where t and t' are two time instances. $\mu(t)$ and $K(t, t')$ are the mean and kernel functions respectively. Commonly, the mean function is assumed to be zero ($\mu(t) = 0$). The kernel function can be chosen from a wide range of options [109]. The kernel function defines the underlying property of the function $f(t)$. For example, it can act as a constant component or represent the smoothness and periodic properties. Different types of kernels can be combined as summation or multiplication. In this chapter, we use a summation of three kernels to describe the covariance function. Denoting K_C , K_{SE} , and K_{WN} as a constant kernel, squared exponential kernel, and white noise kernel, the combined kernel is defined as:

$$\begin{aligned}
 K(t, t') &= K_{SE}(t, t') + K_C(t, t') + K_{GN}(t, t') \\
 K_{SE}(t, t') &= \sigma_{se}^2 \exp \left\{ -\frac{1}{2\ell^2} (t - t')^2 \right\} \\
 K_C(t, t') &= C \\
 K_{WN}(t, t') &= \sigma_{wn}^2 \delta(t - t')
 \end{aligned} \tag{4.22}$$

where σ_{se} , ℓ , C , and σ_{wn} are hyper-parameters for the kernel function. Denoting $\Theta = \{\sigma_{se}, \ell, C, \sigma_{wn}\}$ as the vector of hyper-parameters, we can compute the optimal Θ by maximizing the log marginal likelihood function:

$$\begin{aligned}
 \log p(y_t | t, \Theta) &= \frac{1}{2} y_t^T K(t, t)^{-1} y_t - \frac{1}{2} \log |K(t, t)| - \frac{n}{2} \log 2\pi \\
 \hat{\Theta} &= \underset{\Theta}{\operatorname{argmin}} \{ \log p(y_t | t, \Theta) \}
 \end{aligned} \tag{4.23}$$

Once an optimal set of hyper-parameters is obtained, the probabilistic prediction of future states can be computed as follows:

$$\begin{aligned}
 p(f(t^*) | t^*, y_t, t) &= \mathcal{N}\{A, B\} \\
 A &= K(t^*, t) K(t, t')^{-1} y_t \\
 B &= K(t^*, t) - K(t^*, t) K(t, t')^{-1} K(t^*, t)^T
 \end{aligned} \tag{4.24}$$

where t^* is an unseen or future time instance. The training and prediction require the computationally expensive inversion of $K(t, t')$, which can be calculated using Cholesky factorization [109, Chapter 2]. The computational complexity of Gaussian regression is $O(n^3)$. In Figure 4.9, using the GPR predictor, a 30-minute prediction based on a 30-minute observation of one location is illustrated.

In this figure, solid dots represent values computed by the MP model for the first 30 minutes. The black line is the mean prediction of the GPR model for the entire hour. From darker to lighter gray areas are σ , 2σ , and 3σ of prediction coverage respectively. The circles are values computed by the MP model for the second 30 minutes, which are not used for constructing the GPR predictor. In this specific example, we can see that the majority of the estimates in the second 30

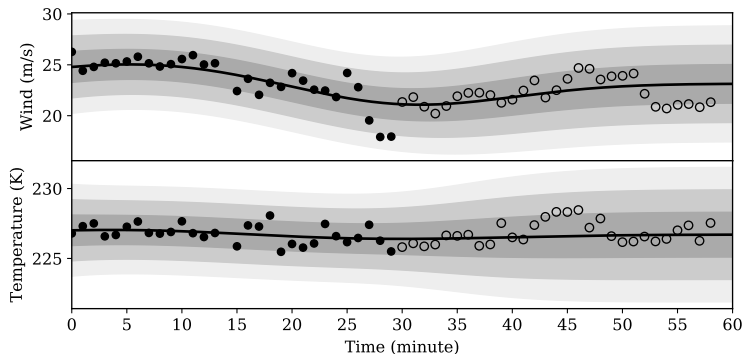


Figure 4.9: Gaussian process regression predictor example, based on wind and temperature computed at an example location from an Eulerian point of view.

minutes are within one σ of the prediction interval.

The GPR predictor can be constructed at any position of interest to provide short-term predictions. However, a short history of estimated states from the MP model needs to be recorded for the predictions.

4.5. Experiments and validations

An ADS-B/Mode-S receiver is installed at the faculty of Aerospace Engineering at the TU Delft. This device provides a constant stream of signals obtained from aircraft that are within line of sight of this receiver. Using the open-source decoding library pyModeS, ADS-B and Mode-S Comm-B replies can be used to derive wind observations for the MP model. The area of the experiment is between 300 to 400 kilometers in radius, centered around Delft, the Netherlands.

At first, to demonstrate the Meteo-Particle model, a small dataset (with 30 minutes data) from ADS-B and Mode-S is used to compute the wind observations. The observations are used to construct the wind and temperature fields. The results of the wind and temperature fields are illustrated. Later on, to validate the MP model, we compare the results with two public numerical weather prediction datasets, which are the Global Forecast System (GFS) reanalysis data and the European Centre for Medium-Range Weather Forecast (ECMWF) ERA5 reanalysis data.

4.5.1. Construction of sample wind and temperature fields

As a first experiment, we want to reconstruct the real wind and temperature grid, based on a small set of aircraft surveillance data. The dataset consists of 30 minutes of obtained wind data, from 9:00 to 9:30 UTC on January 01, 2018. In total, around 90,000 wind measurements were generated during this time period. In Figure 4.10, the distributions of wind observations are displayed both horizontally and vertically. The plot on the left-hand side illustrates the ground projection of all observations. On the right-hand side, the plot shows a histogram with the num-

ber of observations per 1 km altitude. Horizontally, the measurements are highly concentrated along flight routes. Vertically, the majority of the observations are at cruise altitudes and lower approaching altitudes. The statistic of wind at different altitudes in this dataset is computed. The distributions grouped by altitude are shown in Figure 4.11.

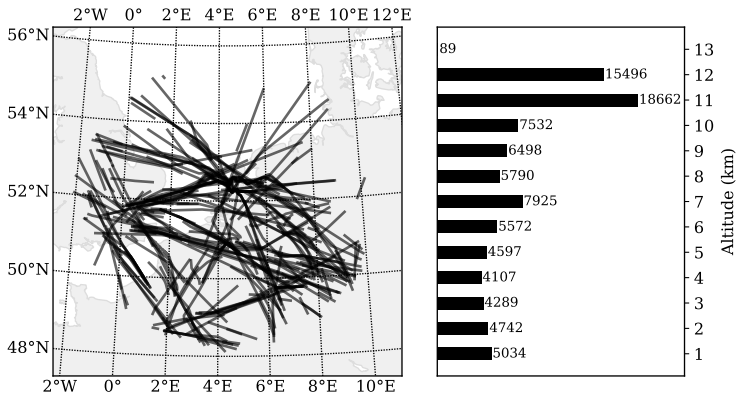


Figure 4.10: Wind data ground projections and vertical distributions

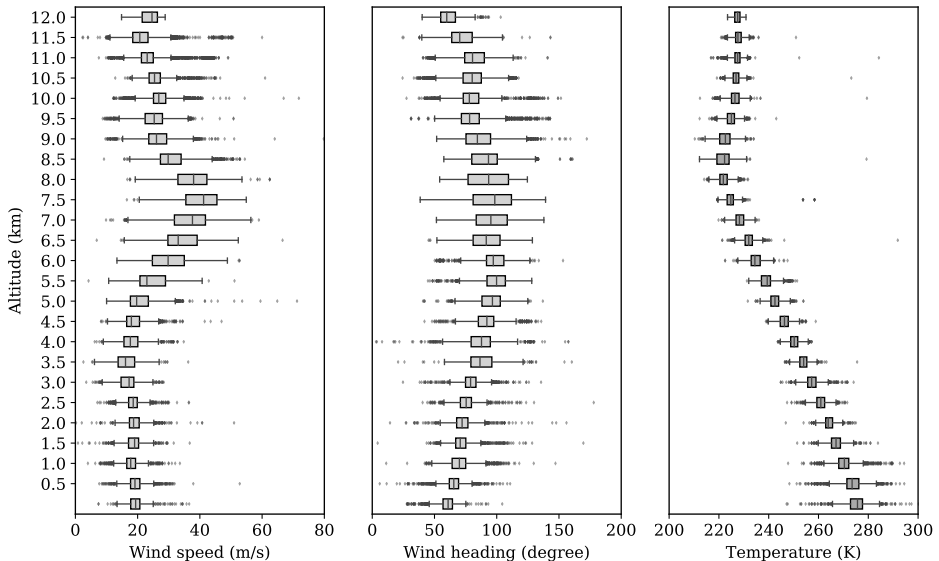


Figure 4.11: Wind speed and direction distributions grouped by altitude

During this time period, wind generally comes from a westerly or south-westerly direction. The spatiotemporally varying wind also leads to variability in both wind velocity and direction, even within this short period of time. Figures 4.10 and 4.11

reflect the two challenges proposed earlier in this chapter. They are 1) highly non-uniformly distributed and varying observation data, and 2) the observation errors (shown as outliers).

To simulate a real-time run of the model, these recorded wind data are streamed to the MP model using the original sequence based on the data time-stamp. A snapshot of the wind grid at 09:00 UTC is shown in Figure 4.12, while the temperature grid is shown in Figure 4.13. In these two figures, the entire airspace is represented by a field consisting of $10 \times 10 \times 12$ grid-points. It is centered around the location of the receiver (latitude: 51.99°N , longitude: 4.37°E). The horizontal grid spacing is approximately 60 km. Vertical grid spacing is 1 km. We can observe visually that both wind and temperature are consistent with the observation distributions from Figure 4.11. The average wind speed is around 20 to 40 m/s at different vertical levels, while the average wind direction is around 90 degrees (West-East wind). At the same time, the average temperatures at different vertical levels are also aligned in these figures.

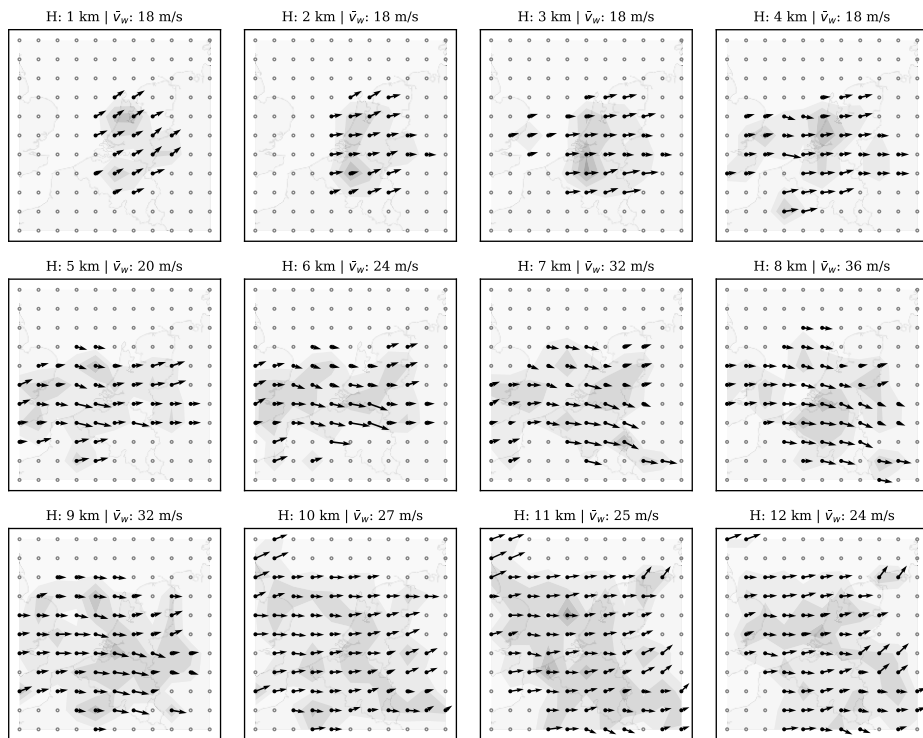


Figure 4.12: Wind grid at 12 different altitude levels

4.5.2. Validation of MP model with NWP data

In this section, we focus on validating the correctness of the MP model output. The level of correctness can be examined against data from existing meteorolog-

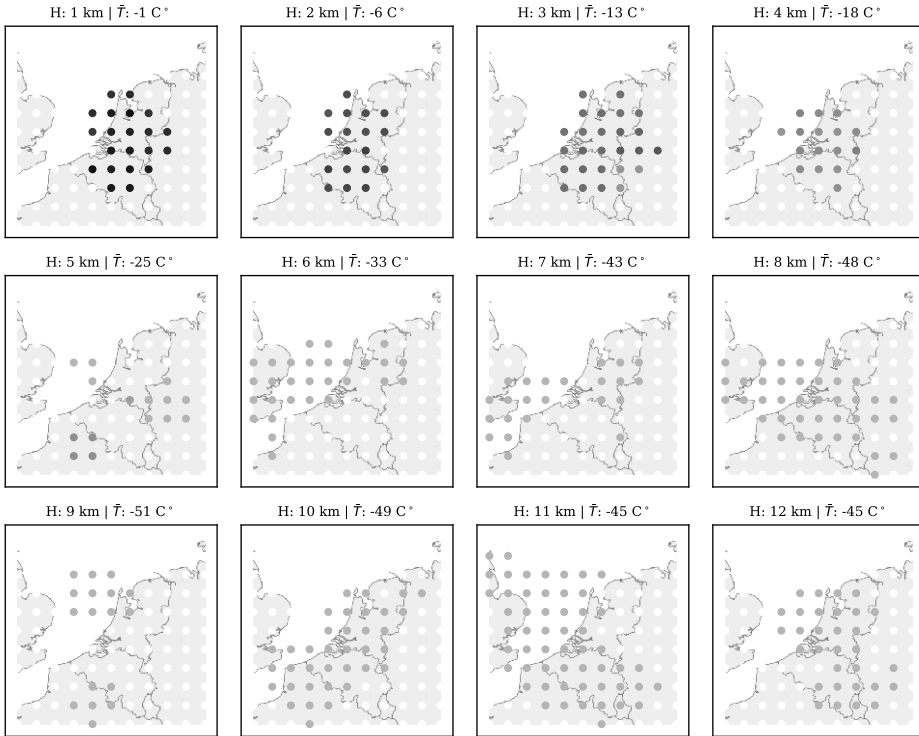


Figure 4.13: Temperature grid at 12 different altitude levels

ical models. GFS reanalysis data and ECMWF ERA5 reanalysis data are used to examine the MP model result. A period of 10 days (from the 1st to the 10th of January 2018) is used in this experiment.

GFS reanalysis data provide a global atmospheric reanalysis of all altitudes at the highest available resolution of 0.5 degrees in latitude and longitude. Meteorological conditions (including wind and temperature) are computed at 00:00, 06:00, 12:00, and 18:00 hour each day. The ECMWF ERA5 can provide higher resolution reanalysis data. We use the 0.25-degree resolution data as the point of comparison.

The wind and temperature observations are computed based on aircraft surveillance data, which contains 10 days of one-hour data around the four hours indicated in GFS reanalysis grid data.

At each GFS hour (00:00, 06:00, 12:00, or 18:00), spot values and average values are computed. The spot value is the wind grid computed from the MP model at the exact GFS hour. The average values are computed as the mean of the hour around GFS hour (per minute, +/- 30 minute of wind grids). Spatially, we extract wind/temperature fields at the grid indicated in the GFS and ECMWF ERA5 data, with the resolution of 0.5 and 0.25 degrees respectively.

In order to compare the difference in wind vectors between MP model and

NWP data, two distance matrices are used, which are *angular difference* and *magnitude difference*. The angular difference is computed as follows:

$$\Delta\theta = \arccos\left(\frac{\mathbf{V}_{\text{pm}} \cdot \mathbf{V}_{\text{gfs}}}{\|\mathbf{V}_{\text{pm}}\| \cdot \|\mathbf{V}_{\text{gfs}}\|}\right) \quad (4.25)$$

where V_{pm} and V_{gfs} are the two wind vectors computed by the MP model and extracted from GFS respectively. $\Delta\theta$ is the angle in degrees between two wind vectors with a range of $[0, 180]$. The smaller the $\Delta\theta$, the smaller the angular difference between the two wind vectors. The magnitude difference is computed as the absolute difference of wind vectors:

$$\Delta V = \text{abs}(\|\mathbf{V}_{\text{pm}}\| - \|\mathbf{V}_{\text{gfs}}\|) \quad (4.26)$$

Similarly, the temperature difference is computed as:

$$\Delta T = \text{abs}(T_{\text{pm}} - T_{\text{gfs}}) \quad (4.27)$$

The spot value and average value are first computed using the MP model. The difference with the GFS data is illustrated in Figure 4.14, showing the results of a total of around 62,000 point comparisons during the 10-day period. The difference between the MP result and the ECMWF data is illustrated in Figure 4.15. The statistics are computed based on around 300,000 data points due to the higher grid resolution. In each figure, the three plots represent the differences in wind magnitude, wind direction, and temperature respectively.

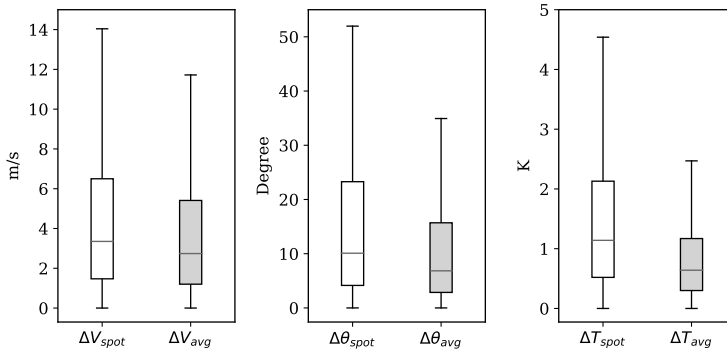


Figure 4.14: Mean wind and temperature differences between MP model and GFS reanalysis data, with 0.5 degree resolution. Spot values are values computed at the exact GFS hour. Average values are computed using values within 30 minutes before and after the GFS hour. The outliers are not shown in the boxplots.

We can see that the result of the MP model is closer to the higher resolution ECMWF ERA5 data. The exact differences are summarized in Table 4.1. It can also be observed that when using a one-hour average (mean of 60 snapshots on each minute), the differences become smaller. This could reflect the averaging effects of NWP models. It could also be caused by the lack of interpolation accuracy due

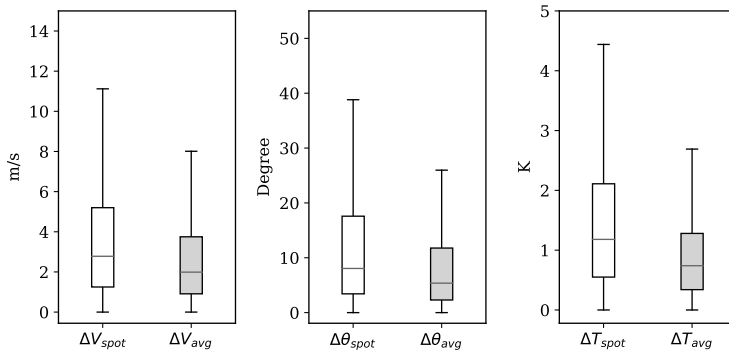


Figure 4.15: Mean wind and temperature differences between MP model and ECMWF ERA5 reanalysis data, with 0.5 degree resolution.

4

to the gaps in source data of these NWP models.

Table 4.1: Comparing MP with NWP models (mean absolute errors)

	ΔV_{spot}	ΔV_{avg}	$\Delta \theta_{spot}$	$\Delta \theta_{avg}$	ΔT_{spot}	ΔT_{avg}
GFS (0.5°)	3.35 m/s	2.74 m/s	10.09 °	6.84 °	1.14 K	0.64 K
ECWMF (0.25°)	2.78 m/s	1.99 m/s	8.05 °	5.37 °	1.18 K	0.74 K

4.5.3. MP model estimation accuracy

Accuracy poses one of the common drawbacks of using a NWP model (such as the previously used GFS) for aircraft performance studies. Due to the fixed grid and the large update time interval, interpolation models generated based on an NWP analysis dataset are often over-smoothed. In the low-resolution meteorological dataset, local variations are often absent. When studying aircraft performance, accurate information on the local wind conditions is important and sometimes critical. In this experiment, our goal is to study the accuracy of the proposed MP model compared to the interpolated model from NWP analysis data. The same NWP data source, the Global Forecast System reanalysis data (0.5° resolution), is used for this purpose.

To compare the accuracy of the two models, we use the same set of the 30-minute wind data as shown in the earlier section. The approximately 90,000 data points are split into the training (60%) and testing (40%) datasets randomly.

To examine the accuracy of the MP model estimation, we use the training dataset to construct the wind and temperature fields using the MP model. Then, we estimate the wind and temperature for all points (4D) that appear in the test set. The estimations and actual values are compared to calculate the accuracy metrics. To calculate the accuracy metrics based on NWP data, a linear interpolation model is generated using two GFS analysis datasets (06:00 and 12:00 UTC). In this way, we can compute the wind and temperature of each point that appears in the test dataset.

Results from the two estimation models are compared with the true value in the test dataset. They are illustrated in Figures 4.16 (wind speed) and 4.17 (temperature).

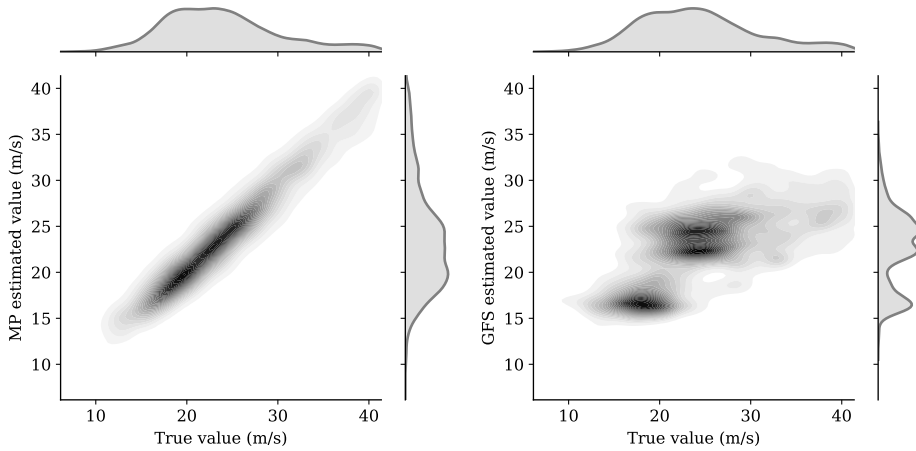


Figure 4.16: Comparison of MP and GFS model for local wind estimation

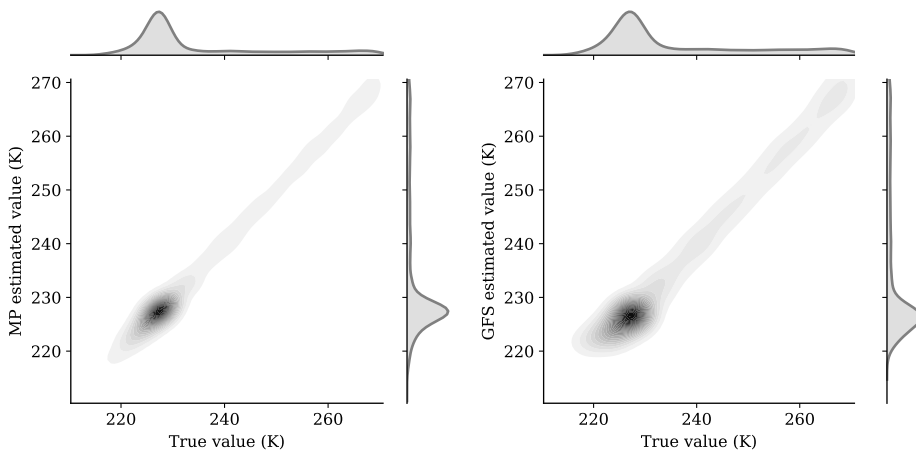


Figure 4.17: Comparison of MP and GFS model for local temperature estimation

We can see that the MP model displays a significantly higher accuracy than the GFS interpolation model when inferring the wind. In terms of temperature, the MP model also shows better accuracy. To quantify the differences, different training/testing splits are chosen for the prediction. Quantitative metrics *Mean Squared Error* (MSE) and *Mean Absolute Error* (MAE) are computed for wind and temperature under each configuration. The results are shown in Table 4.2 (wind speed) and Table 4.3 (temperature).

Table 4.2: Estimation errors in wind calculation (unit: m/s)

Train-Test split	MSE(GFS)	MAE(GFS)	MSE(MP)	MAE(MP)
60%-40%	27.64	3.92	3.69	1.30
70%-30%	27.60	3.92	3.71	1.30
80%-20%	28.46	3.98	3.63	1.29
90%-10%	29.05	4.03	3.26	1.26

Table 4.3: Estimation errors in temperature calculation (unit: K)

Train-Test split	MSE(GFS)	MAE(GFS)	MSE(MP)	MAE(MP)
60%-40%	4.81	1.63	3.11	1.20
70%-30%	4.82	1.63	3.00	1.18
80%-20%	4.76	1.62	2.93	1.17
90%-10%	4.75	1.61	2.87	1.14

4

From the above results, when comparing the absolute error, we can conclude that the wind error decreases around 67%, from around 4 m/s in the GFS model to around 1.3 m/s in the MP model. This represents a threefold increase in wind accuracy. At the same time, the temperature error also decreases around 26%, from 1.6K in the GFS model to around 1.2K in the MP model. This represents a 1.3 fold increase in temperature accuracy.

4.5.4. Short-term prediction accuracy

A short-term prediction (up to 30 minutes) is constructed based on the Gaussian process regression predictor defined in Section 4.4. The experiment is carried out using the well-established open-source machine learning library Scikit-Learn [106]. It includes the implementation of the proposed Gaussian Process Regression models. The validation experiment is based on the 10x10x12 grid (as show in Figures 4.12 and 4.13).

A new dataset consisting of 30 minutes from 9:30 to 10:00 UTC on January 01, 2018, is used for testing these predictions. The reference wind and temperature values are computed based on this dataset using the MP model. The prediction of 9:31 to 10:00 is performed at 9:30 based on the wind and temperature fields that have been constructed using the MP model.

At each minute from 9:31 to 10:00, the differences between the prediction and the true value of all points are plotted in Figure 4.18.

In this figure, we can see that the differences between reference wind and predicted wind are within 10 m/s, whereas the differences of temperatures are within 5 K. The exact error metrics are listed in Table 4.4.

Table 4.4: Error metrics for the prediction of 30 minutes

	ME	MAE	RMSE
Wind speed (m/s)	0.37	3.71	24.71
Temperature (K)	0.02	1.23	2.80

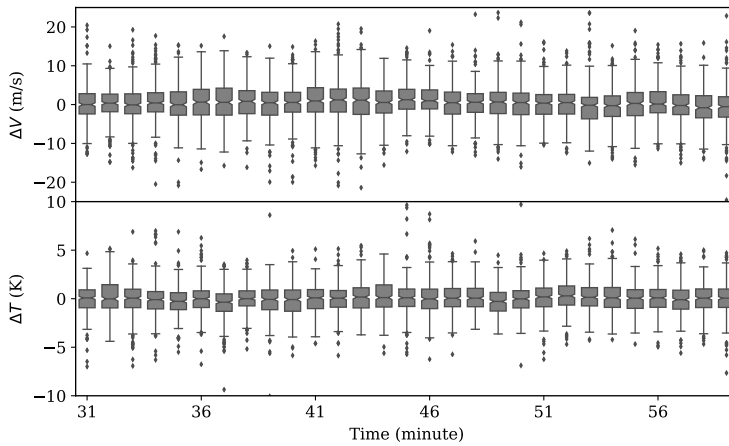


Figure 4.18: Prediction error of wind and temperature in 30 minutes

4.6. Uncertainty analysis

The variability of the system depends on the uncertainty in the MP model and observation data. The uncertainty of the MP model is caused by the stochastic process involved, such as the probabilistic observation rejection and the particle position update. On the other hand, the uncertainties of wind and temperature observations are caused by inaccurate information downlinked from aircraft or due to decoding errors.

In this section, we focus on the study of variation caused by the randomness in the stochastic process and data, as well as the error tolerance of the MP model. All analyses are based on the same dataset used in the earlier experiments, from 9:30 to 10:00 UTC on January 01, 2018, collected by our ADS-B receiver located at Delft, the Netherlands.

4.6.1. Model uncertainty

Randomness exists in the MP model due to the stochastic processes, probabilistic rejection, and sampling. To study whether the combined randomness could affect the wind and temperature field results, as well as the level of the influence, we conduct multiple (100) runs of the model based on the same input data. The wind field at 09:00 (as shown in Figure 4.12) is measured at the end of each run. Combining all 100 results, we can understand the variation caused by the stochastic elements in the MP model.

In Figure 4.19, the distribution of standard deviations of wind-grid speed and heading among 100 runs is displayed. Among these runs, the difference is almost negligible: less than 2 m/s for wind speed and 1.5 Kelvin for the temperature.

Examining the resulting fields, the outliers are found to be the points that are far away from the flight paths, where fewer particles could reach. Based on the variations shown in Figure 4.19, we can conclude that despite the randomness in

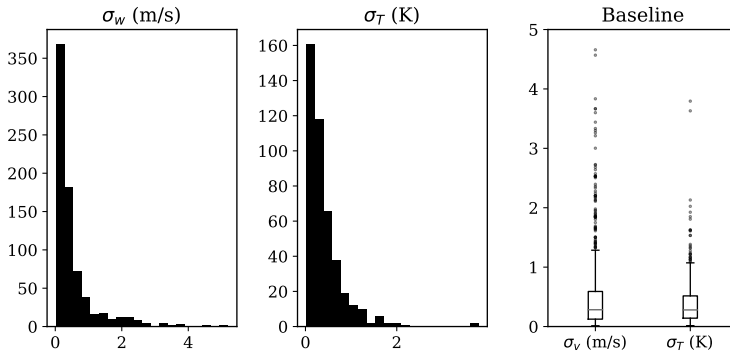


Figure 4.19: Standard deviation of wind speed and temperature of 100 runs

4

the model, the MP model always converges to a result with small uncertainties. The results here can be considered as the baseline model for comparison with other studies later in this chapter.

4.6.2. Data uncertainty

Another important validation is to determine how the quality of observation data affects the wind field estimation. More precisely, it is necessary to ascertain whether the wind grid could be different if some percentage of the observed data is not available. To study this effect, the previous dataset is randomly sampled into several new datasets that contain 80%, 40%, 20%, 10%, and 50% of the total wind observations. Then, the same processes are run to create five different wind fields at 12:00 hours.

Figure 4.20 illustrates the wind and temperature grids, estimated at an altitude of 8 km when different percentages of sampled data are used. From the first plot to the last, it is obvious that with increasing observation data samples, the size of the estimated field is increased. At the same time, with increasing sample size, the difference between wind fields is smaller.

In order to quantify the differences, the absolute mean differences between wind and temperature are calculated. In Figure 4.21, the distribution of the entire grid (including all altitude levels) of all wind vectors are compared with the results from the complete data.

Compared to the baseline variance of the model as shown in Figure 4.19, we can infer that with a loss of up to 90% of the total data (10% sample), the differences are still small. This test indicates that there is an abundant number of observations from aircraft to sustain a stable meteorological field reconstruction. At the same time, even with a large amount of data loss, the MP model can still obtain stable and correct results. By examining the differences of wind and temperature under different sample rates, we can observe that the results of MP model are consistent with the results using the full dataset, even though the size of the fields may be reduced.

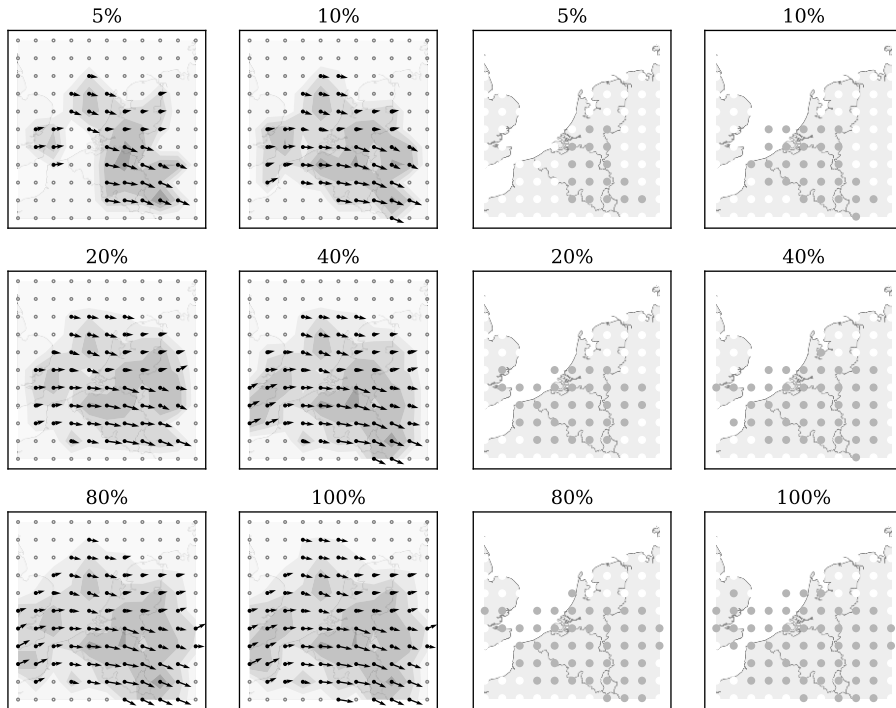


Figure 4.20: Wind and temperature field at a 8 km altitude under different samples

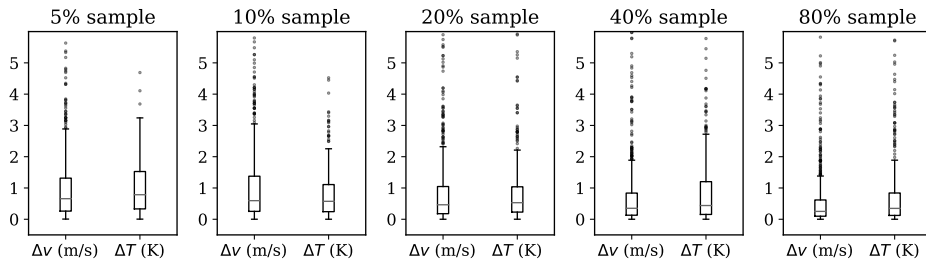


Figure 4.21: Grid magnitude and heading difference due to sampling

4.6.3. Error tolerance

Measurement errors in raw data affect the stability and correctness of the MP model results. We want to quantify the percentage of the errors in data that would produce significant divergence in the results. In this experiment, a percentage of the dataset is replaced with random wind vectors that are uniformly distributed between the minimum and maximum wind speeds with headings between 0 and 360 degrees. Temperatures are altered randomly within ± 20 Kelvin of the originally computed temperature.

In Figure 4.22, wind grid differences between no assumed error and data error rates of 2%, 4%, 6%, 8%, 10%, and 15% are shown.

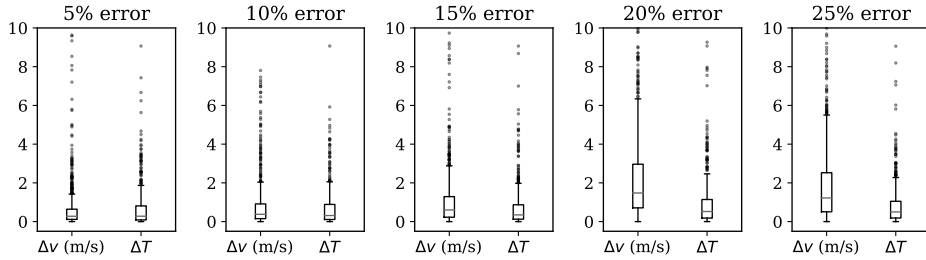


Figure 4.22: Grid magnitude and heading difference due to error

4

With such an aggressive error model, the MP model can maintain a reasonably correct wind field with an error rate of up to approximately 15% (in comparison to the baseline). As a visualization reference, wind and temperature field (8 km altitude) under different error rates are shown in Figure 4.23.

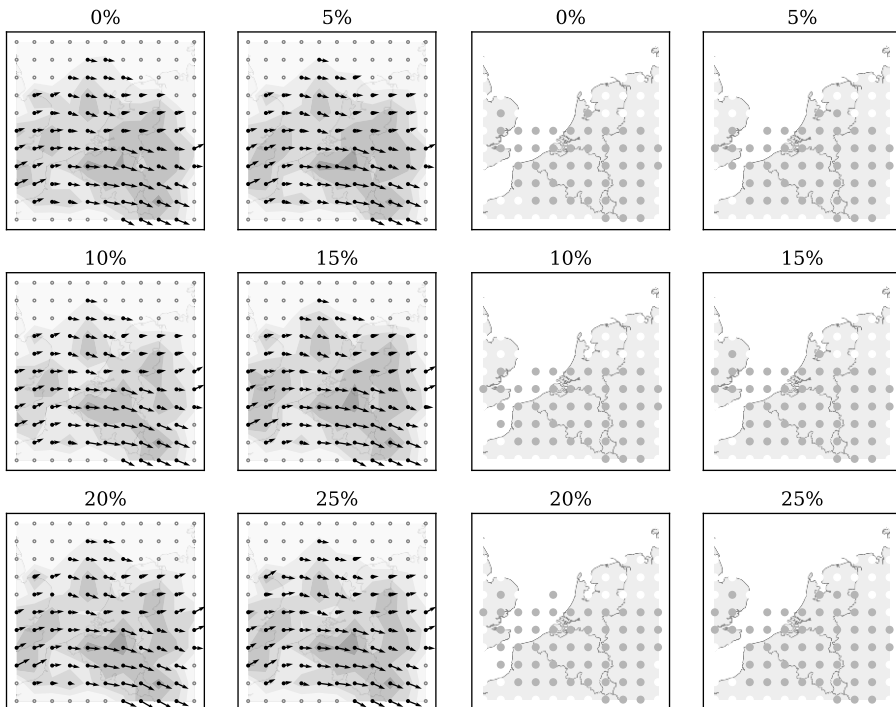


Figure 4.23: Wind and temperature field at a 8 km altitude under different error rates

4.7. Discussion

Throughout the chapter, we describe the Meteo-Particle model and related methods to construct a wind and temperature field from aircraft position and speed broadcasts. We also propose a Gaussian regression predictor for short-term wind and temperature prediction.

The first contribution of this chapter is the method to compute weather information using the aircraft broadcast data. The challenge is not only the decoding of BDS 50 and 60 messages but rather the complete identification process for the entire Mode-S family of messages. As a third-party observer without the knowledge of Mode-S interrogations, this decoding process can be complex. In Chapter 2, we developed the pyModeS decoder to solve this problem, which is used in this chapter to obtain accurate airspeed and temperature measurements.

The core of this chapter focuses on constructing a model that is able to cope with the chaotic nature of wind, moving aircraft, and non-uniformly distributed observations. The MP model proposed in this chapter addresses the stochastic characteristic of wind through particles while maintaining stability through the use of a large number of particles. The MP model uses a stochastic update to propagate wind information to surrounding areas using imaginary particles. With these propagated particles, wind and temperature fields can be estimated in areas where fewer or no measurements are available. Parameters on particle propagation and decay can be tuned in order to control performance. Recommended parameters used in this chapter are given in Table 4.5. They are based on empirical knowledge but not necessarily optimized for all receiver setups.

Table 4.5: Meteo-Particle model parameters used in this chapter

Notation	Parameter	Value	Unit
σ_{u0}	Particle initialization wind variation (u component)	0.2	m/s
σ_{v0}	Particle initialization wind variation (v component)	0.2	m/s
$\sigma_{\tau0}$	Particle initialization temperature variation	0.1	K
σ_{px}	Particle random walk (x direction)	5	km
σ_{py}	Particle random walk (y direction)	5	km
σ_{pz}	Particle random walk (z direction)	0.1	km
x_b	Selection boundary (x direction)	20	km
y_b	Selection boundary (y direction)	20	km
z_b	Selection boundary (z direction)	500	m
k_1	Measurement acceptance probability factor	3	-
k_2	Particle random walk factor	10	-
σ_α	Particle aging parameter	180	s
σ_d	Weighting parameter (distance to point of interest)	30	km
σ_d	Weighting parameter (distance to origin)	30	km

When reviewing the objectives of the Meteo-Particle model, it can be viewed as a type of data assimilation method. In the introduction of this chapter, we mentioned a few existing variational assimilation methods in NWP models, for example, 3DVAR [7] and 4DVAR [110] used by ECMWF. These methods are better suited for large spatiotemporal modeling with data from different observation sources. The MP model focuses on a rapid local real-time weather reconstruc-

tion, based specifically on aircraft measurements. Without minimizing the cost function as in variational methods, the meteorological grid constructed based on the MP model displays less smoothness than NWP model. For supporting general air traffic studies, this lower level of continuity is not a concern. Sometimes this is not even necessary when only weather along a specific trajectory needs to be constructed.

When comparing the Monte-Carlo based particle approach in the MP model to the Gaussian weighted interpolation, there are several advantages. The MP model always includes past observation information without the need for storing historical measurement. When constructing wind and temperature at a location, we only need to consider a small number of the particles in adjacent areas, which is faster to compute. Most importantly, the probabilistic observation acceptance mechanism introduced in the MP model ensures that erroneous observations are not used in the system. Thus, they have little influence on the current and future estimations.

4 In this chapter, three assumptions were made before constructing the MP model. The first two assumptions stated that the variation of wind and temperature are small temporally (at the level of hours) and locally (at the level of kilometers). Only under these two conditions, the aggregated states from propagated particles can represent the true weather conditions. Later on, from the example dataset and results produced, we can confirm the validity of these two hypotheses. However, we have to be cautious when applying the model to near-surface scenarios, such as constructing very low altitude wind fields using aircraft data from the takeoff phase. The wind dynamics can be far from locally steady in this situation. In this chapter, the lowest altitude was set at 1 km to avoid such conditions, regardless of the actual planetary boundary layer.

The last assumption is that the error rate in original measurement data is not too large. This is generally guaranteed with the accuracy of the pyModeS decoder. Using one-hour ground truth data from local air traffic controllers, we were able to find out that the error rate of pyModeS on BDS 50 and BDS 60 message identification is below 0.1%. With this accuracy, only the original aircraft speed measurement error and data transmission error acted as the causes of inaccuracy. However, with the probabilistic error rejection, we showed that the model can handle up to 15% of the artificial error in the data (as shown in Figure 4.22 and 4.23). This is well beyond any normal error rate in aircraft surveillance data.

One of the limitations in analysis of the chapter is the time period of the data used for validation with GFS data. Ideally, prolonged of periods validation would give a more confident statement. Nevertheless, as illustrated in Figure 4.16-4.17 and Table 4.2-4.3, the randomly chosen dataset already displays a large improvement in terms of accuracy.

4.8. Conclusion

Based on aircraft surveillance data, in this chapter, we proposed a fast, real-time model, the Meteo-Particle (MP) model, to construct real-time wind and temperature fields using aircraft as sensors. Using raw temperature and wind states de-

coded from ADS-B and Mode-S downlink data, the MP model constructed wind and temperature fields within the range of the receiver. The range has been demonstrated to be around 300 to 400 kilometers in radius by our experiment.

The results obtained from the MP model were close to NWP reanalysis data. For example, when comparing with GFS reanalysis (0.5-degree resolution) data, the absolute mean difference in wind speed, wind direction, and temperature were 2.74 m/s, 6.94 °, and 0.64 K respectively. When compared to ECMWF ERA5 (0.25-degree resolution) data, the differences were 1.99 m/s, 5.37 °, and 0.74 K respectively. We also studied the accuracy of inference using the MP and GFS model. The mean absolute error of wind speed and temperature estimations are reduced by 67% and 26% when the MP model is applied. This increased accuracy indicates the potential benefits for air traffic management studies.

The Meteo-Particle model demonstrates the validity of using aircraft as large sensor networks to construct a global scale real-time meteorological measurement system for the open research domain. This model and the results proposed in this chapter are fully open. The implementation of the MP model in Python programming language is shared,² as is the experiment data.³ Without the need for any new equipment or communication protocols, the implementation of such a system can be enacted using existing technology and data sources. Based on the single receiver demonstrated in this chapter, we believe that future research can offer a meteorological monitoring capability with a large coverage by using data from existing crowd-sourced receiver networks.

²Available at: <https://github.com/junzis/meteo-particle-model>

³Available at: <https://doi.org/10.6084/m9.figshare.6970403>

5

Kinematic performance model

In Chapter 2 and 3, a number of methods are proposed allowing us to efficiently process aircraft surveillance data. With large quantities of trajectory data collected from a wide range of different aircraft types, it is possible to extract accurate aircraft performance parameters. In this chapter, a set of more than thirty parameters from seven distinct flight phases are extracted for common commercial aircraft types. We use various data mining and estimation methods to generate parametric models for these performance parameters. All parametric models combined can be used to describe a complete flight that includes takeoff, initial climb, climb, cruise, descent, final approach, and landing. Both analytical results and summaries are shown. When available, parameters from these models are also compared with the Base of Aircraft Data and the Eurocontrol aircraft performance database. In summary, the research conducted in this chapter presents a comprehensive set of methods for extracting different aircraft performance parameters. It also provides the first open dataset of kinematic performance models for common aircraft types.

This chapter is based on the following publications:

- 1) Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2017. Modeling aircraft performance parameters with open ads-b data. In Twelfth USA/Europe Air Traffic Management Research and Development Seminar [128].
- 2) Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2019. WRAP: An open-source kinematic aircraft performance model. Transportation Research Part C: Emerging Technologies, 98, pp.118-138 [130].

5.1. Introduction

Fast-time air traffic simulation requires accurate aircraft performance models (APM). These performance models play an important role in modeling and optimizing air traffic flows, especially in managing 4D trajectories. Traditionally, aircraft performance models for air traffic management research are built based on manufacturer performance reports and/or proprietary flight data. These data sources are either costly or have certain license limitations. For example, the most commonly used APM, Base of Aircraft Data (BADA), has license limitations [102, 42]. Besides BADA, aircraft manufacturers also provide performance models. Unfortunately, very few of these are available as open-access models.

A different approach to model aircraft performance is to use flight data. A recent study [60] proposes the use of radar data and machine learning methods to obtain performance parameters and patterns. At the same time, thanks to the mandates on aircraft Automatic Dependent Surveillance-Broadcast (ADS-B) transponders from different regulatory agencies around the world, an increasing number of commercial aircraft are being equipped with this capability. Low-cost ground receivers are being installed around the world in a crowd-sourced fashion. With ADS-B, certain surveillance data such as position, altitude, velocity, and vertical speed are broadcast and received constantly around the world by a number of ground receiver networks.

Several leading crowd-sourced ground networks are providing free access to aggregated ADS-B data. Commercial networks like FlightRadar24 and FlightAware have large infrastructure and coverage. Research networks such as ADS-B Exchange and the OpenSky Network [116] are also expanding. Each network has a different policy on data sharing. However, many of these networks grant the rights of data usage to network contributors. These large amounts of raw flight data provide a new source for modeling aircraft performance.

How we can use these large quantities of data in order to construct aircraft performance models is the core research question of this chapter. This chapter aims to construct an accurate kinematic performance model for common aircraft types and, then, employ such a model for air traffic simulation and analysis.

There are different categories of performance models. The most detailed six degree-of-freedom model is commonly used in aircraft control studies, while the point-mass model is often used in studies concerning air transportation. In the point-mass model, aircraft roll, pitch, and yaw motions are ignored, leaving only horizontal and vertical motions. There are two different types of point-mass models: dynamic and kinematic. The primary difference is that while a dynamic model focuses on forces and energy, a kinematic model deals only with aircraft motions.

This chapter proposes a kinematic aircraft performance model based entirely on open data, called the WRAP performance model.¹ In addition to common procedural values, a parametric statistical model of each performance indicator is constructed. The use of parametric statistical models enables probabilistic sampling in simulations and produces realistic scenarios. Thanks to this unique property,

¹Speaking of which, WRAP is not meant to be an acronym. If one must put a longer name to it, then it is short for: *WRAP Represents Aircraft Performance*.

the model can cover a wide range of use cases.

The remainder of this chapter is structured as follows. Section two introduces the model parameters, definitions, and maximum likelihood estimations for three different distribution functions. Section three explains the data and pre-processing methods. Section four is focused on methods and algorithms for constructing parameters of each flight phase. Detailed results on a single aircraft type and summaries of 17 different aircraft types are shown in section five. Finally, discussion and conclusion are addressed in sections six and seven.

5.2. The WRAP model

In this section, the kinematic aircraft performance parameters of the WRAP model are first presented. Based on flight data, each parameter is evaluated under three parametric statistical model assumptions. We then choose the best statistical model according to the Kolmogorov-Smirnov test [89], from which the most common and limitation values are computed.

5.2.1. Model parameters

This study aims to derive performance parameters for the following flight phases: takeoff, initial climb, climb, cruise, descent, final approach, and landing, which closely correspond to the flight phase definitions the ICAO Common Taxonomy Team [24], with the exception that the definitions of the initial climb and final approach are slightly different in the WRAP model. This is because the WRAP model tries to generalize the parameters over different air traffic control procedures.

The significance of each performance indicator differs from one flight phase to another. For example, vertical speed is only of interest for climb and descent, while, during takeoff and landing, the runway distance, acceleration, and takeoff/touchdown speed are studied. Specifically, during the climb (or descent) phase, a constant CAS/Mach climbing profile is extracted. The parameters included in the WRAP model are listed as follows per flight phase:

- Takeoff:
 - liftoff speed (V_{lof})
 - takeoff distance (d_{tof})
 - mean takeoff acceleration (\bar{a}_{tof})
- Initial climb:
 - calibrated airspeed ($V_{\text{cas,ic}}$)
 - vertical rate (VS_{ic})
 - cutoff altitude (h_{ic}), fixed at 457m / 1500 ft
- Climb:
 - range to the top of climb ($R_{\text{top,cl}}$)
 - altitude where constant CAS starts ($h_{\text{cas,cl}}$)
 - constant CAS ($V_{\text{cas,cl}}$)
 - vertical rate during constant CAS climb ($VS_{\text{cas,cl}}$)
 - constant Mach climb crossover altitude ($h_{\text{mach,cl}}$)
 - constant Mach number (M_{cl})

- vertical rate during constant Mach climb ($VS_{\text{mach,cl}}$)
- vertical rate before constant CAS climb ($VS_{\text{precas,cl}}$)
- Cruise:
 - cruise range (R_{cr})
 - maximum cruise range ($R_{\text{max,cr}}$)
 - initial cruise altitude ($h_{\text{init,cr}}$)
 - cruise altitude (h_{cr})
 - maximum cruise altitude ($h_{\text{max,cr}}$)
 - cruise Mach number (M_{cr})
 - maximum cruise Mach number ($M_{\text{max,cr}}$)
- Descent:
 - range from the top of descent ($R_{\text{top,de}}$)
 - constant Mach number (M_{de})
 - constant Mach descent crossover altitude ($h_{\text{mach,de}}$)
 - vertical rate during constant Mach descent ($VS_{\text{mach,de}}$)
 - constant CAS ($V_{\text{cas,de}}$)
 - altitude where constant CAS starts ($h_{\text{cas,de}}$)
 - vertical rate during constant CAS descent ($VS_{\text{cas,de}}$)
 - vertical rate after constant CAS descent ($VS_{\text{postcas,de}}$)
- Final approach:
 - calibrated airspeed ($V_{\text{cas,fa}}$)
 - vertical rate (VS_{fa})
 - cutoff altitude (h_{fa}), fixed at 300m / 1000 ft
 - path angle (\angle_{fa})
- Landing:
 - touchdown speed (V_{tcd})
 - braking distance (d_{ind})
 - mean braking deceleration (\bar{a}_{ind})

One of the rationales for the selection of these parameters is to ensure compatibility and comparability with existing kinematic performance models, such as the Aircraft Performance Database of Eurocontrol [37], as well as the kinematic parameters of BADA [102]. The WRAP model parameters are also designed with the aim of supporting the open air traffic simulator BlueSky [54]. The parameters are chosen so that the kinematic flight envelope of different aircraft models can be constructed according to the flight phases.

5.2.2. Parametric statistical models and selection

Three continuous probability density functions (PDF) are proposed to describe each of the performance parameters: the Normal, Gamma, and Beta distributions. Many of the performance parameters, for example, mean speed or mean vertical rates can be modeled using Normal distribution. Empirical observations with flight data support this claim, as does the central limit theorem in the probability theory. The theorem states that when independent random variables are drawn independently, the distribution of the averages tends to follow Normal distributions. In cases where human decisions are involved, however, the normal distribution may

not be the best choice to describe a parameter. For example, with cruise range or crossover altitude of constant CAS/Mach climb, the distribution can be skewed. Therefore, we need to introduce skewed distributions to model these parameters. The Gamma distribution and Beta distribution are used for right-skewed and left-skewed distributions respectively. Finally, the maximum likelihood estimation (MLE) [1] is used to estimate the best-unbiased model parameters under each statistical model assumption.

Normal distribution

For a Normal distribution, the PDF is expressed as:

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.1)$$

where the parameters μ and, σ^2 represent the mean and variance. The MLE estimator of $\hat{\mu}$ and $\hat{\sigma}^2$ can be computed as the mean and variance of the observations.

Gamma distribution

For a standardized Gamma distribution with support of $(0, +\infty)$, the PDF is expressed as:

$$p(x|\alpha) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} \quad (5.2)$$

where α ($\alpha > 0$) represents the shape of the distribution. As a general form, the location (μ) and scale (k) parameters are introduced to allow the support of $(\mu, +\infty)$ for the PDF:

$$p(x|\alpha, \mu, k) = \frac{1}{k \cdot \Gamma(\alpha)} \left(\frac{x - \mu}{k}\right)^{\alpha-1} \exp\left(-\frac{x - \mu}{k}\right) \quad (5.3)$$

The MLE estimate $(\hat{\alpha}, \hat{\mu}, \hat{k})$ does not have a closed-form solution, but it can be solved numerically [21].

Beta distribution

For a standardized Beta distribution with support $(0, 1)$, the PDF is given as:

$$p(x|\alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (5.4)$$

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$$

where α and β represent the shape of the distribution. As a general form, the location (μ) and scale (k) parameters can also be introduced to allow the support of $(\mu, +\infty)$ for the PDF:

$$p(x|\alpha, \beta, \mu, k) = \frac{1}{k \cdot B(\alpha, \beta)} \left(\frac{x - \mu}{k}\right)^{\alpha-1} \left(1 - \frac{x - \mu}{k}\right)^{\beta-1} \quad (5.5)$$

Similar to the Gamma distribution, the Beta distribution also does not have a closed-form solution for the MLE estimate $(\hat{\alpha}, \hat{\beta}, \hat{\mu}, \hat{k})$. Therefore, it also needs to be solved numerically [45].

Selection of the statistical model

After all possible optimal parametric models are computed based on test data, the best of all three are identified using the D score from the one-sample Kolmogorov-Smirnov (KS) Test [89]:

$$D = \max_x |F_n(x) - F(x)| \quad (5.6)$$

where $F(x)$ is the cumulative distribution function (CDF) of a statistical model, and $F_n(x)$ is the the empirical distribution function (EDF) from the data. The D score is essentially the largest distance between two functions, which is calculated according to Equation 5.6. A smaller D score reveals a better fit of the statistical model to the test data.

Of all three KS statistics, D_N , D_Γ , and D_B , the probability density that yields the minimum D will be selected. The KS-test is performed on a different dataset that was used for the parameter estimation. In this chapter, the experiment data is split into two equal parts (one for fitting and one for testing) to construct and validate the best parametric statistical model.

An example dataset of one parameter is shown in Figure 5.1, where the CDF of all three models obtained from MLE is displayed against the EDF computed from the data. The best model (beta) is obtained, as it yields the lowest D score in this example.

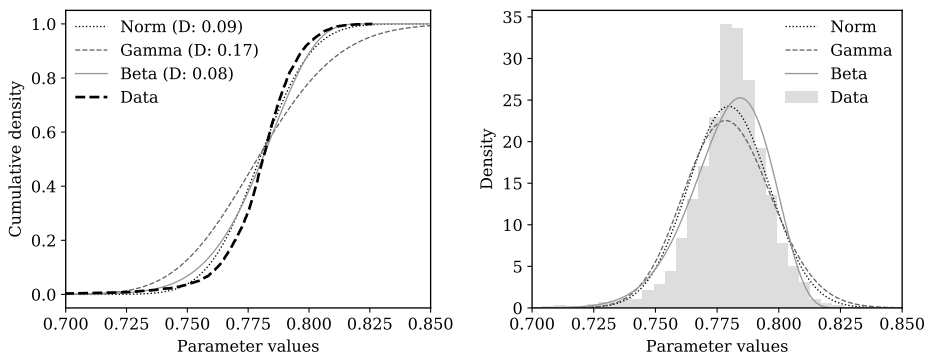


Figure 5.1: Cumulative distributions and probability distributions of fitted model with data. The example parameter is the mean cruise Mach number of A320. The best model describing the data is the Beta distribution that yields the minimum D value of 0.08.

5.2.3. Interpretation of the model

In the resulting WRAP model, each parameter is described with three parts using the following convention:

$$\left\{ \hat{\psi} \mid \psi_{\min}, \psi_{\max} \mid *pdf \right\} \quad (5.7)$$

The first part, $\hat{\psi}$ is the most common operational value. This value is computed as the mode of the probability distribution function that corresponds to the most frequent value in reality.

The second part, ψ_{\min} and ψ_{\max} , are minimum and maximum values obtained from different confidence intervals (CI). The CI is defined based on the parameter type:

- CI=0.85: Parameters associated with speed during takeoff, initial climb, final approach, and landing.
- CI=0.99: Parameters associated with ranges.
- CI=0.9: Default value of other parameters.

Here, the choice of a low confidence interval for velocity is made to reduce the impact of wind uncertainties in flight phases that are close to the ground. Even though the wind data has already been integrated, wind information is less certain for near-surface conditions. Thus, the low CI is chosen. The choice of a higher CI for flight ranges is made to take into account flights with a range as close to their maximum operational range as possible. The hypothesis is that extreme values are likely not outliers when dealing with flight range.

The third part in Equation 5.7 is the optimal statistical model that is obtained from the KS test. It is defined as:

$$*pdf = \begin{cases} [norm', \mu, \sigma] & \text{for } x \sim \mathcal{N} \\ [gamma', \alpha, \mu, k] & \text{for } x \sim \Gamma \\ [beta', \alpha, \beta, \mu, k] & \text{for } x \sim B \end{cases} \quad (5.8)$$

Three types of values defined by the WRAP model provide different use cases. For example, in the case of conducting air traffic simulations [54]:

1. The most common value can be used for air traffic simulations with a fixed performance profile. It can be considered as a standard airline procedure profile, which is equivalent to the APF in BADA model.
2. The minimum and maximum values define the limits of the kinematic model, which implements the flight envelopes.
3. The PDF provides the possibility of constructing Monte-Carlo simulation or study where parameters need to be sampled according to a realistic probability distribution.

5.3. The data

When dealing with aggregated ADS-B flight data, errors often occur. Trajectory data need to be pre-processed and segmented by flight phase accordingly. At the same time, not all parameters can be directly observed. To infer hidden parameters, certain models and methods need to be designed. In the following two sections, we discuss the methods used for data processing and performance modeling.

5.3.1. Data source

For this research, the input data are primarily based on ADS-B messages that are broadcast by aircraft through Mode-S transponders. However, even with good installation and placement, a single ground-based receiver only has a maximum reception range of around 250 nm (~ 500 km). Considering Mode-S line-of-sight availability, it is not possible to capture large quantities of completed flight data with a single ground station. This is especially challenging when dealing with long-range flights. Thanks to networks of ADS-B ground receivers, such as FlightRadAR24 [39], it is possible to gain access to a much larger collection of flight data obtained from ground stations from contributors around the world.

It is also worth noting that in addition to ADS-B data, using the same receiver setup, aircraft positions, and velocities can be also obtained from multiple ground stations using Mode-S multilateration. This technology is employed by some ground receivers within the FlightRadAR24 network. It is useful for those aircraft that are not yet equipped with ADS-B compatible transponders. However, the availability is limited to areas where air traffic controllers are present and not usable when aircraft are close to the ground.

5.3.2. Trajectory processing

Data collected from ADS-B are usually scattered. The methods proposed in Chapter 3 have made it possible to extract flight trajectories easily. This method can only segment flights into five different phases: on-ground, climb, cruise, descent, and level flights.

The study in this chapter requires further processing of those segments. To this end, an altitude threshold is applied to the climb and descent trajectory to extract the initial climb and final approach. We consider the climbing section up to 1500 ft to be the initial climb. The final approach starts from 1000 ft toward the end of the descent. An evaluation process is used to examine the data of all flight segments, ensuring a certain level of completeness and continuity in the given time series data.

5.3.3. Atmospheric conditions and speed conversions

The velocity provided in an ADS-B message gives the ground speed of an aircraft. However, the accuracy of the model may be compromised as wind is not considered. For example, the seasonal wind and jet streams cause a great bias when modeling the cruise speed, if only the ground speed is considered. In this chapter, wind speeds are computed using an interpolation model from the Global Forecast

System (GFS) Reanalysis data [113].

The GFS Analysis model cycles every six hours, producing atmospheric grids using global data assimilation. The grid data are stored at two resolutions (0.5 and 1 degree). Vertically, isobaric levels from 1000hPa to 10hPa are provided and divided into 26 layers. In this chapter, the U component (west-east direction) and V component (south-north direction) of wind, as well as the temperature of all grid points, are extracted from the datasets. Then, 4-D (latitude, longitude, altitude, and time) grids for wind and temperature are constructed. Figure 5.2 shows an example of the wind grid snapshot.

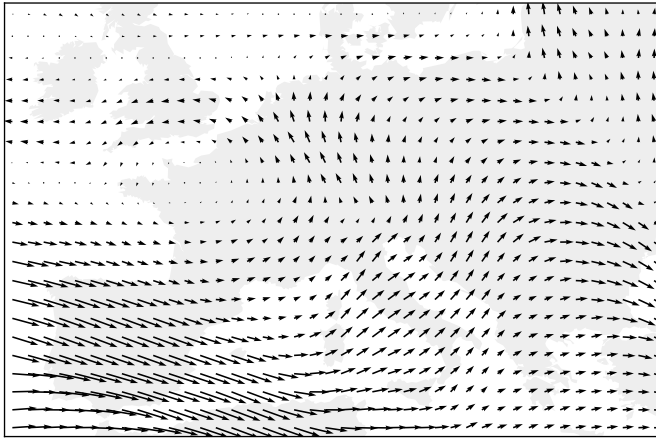


Figure 5.2: Example of wind over Europe at 400hPa pressure level of GFS model (23.5k ft, ISA+0°C)

After that, an N-dimensional multi-linear interpolation [141] is used to construct a linear 4-D interpolated model. As such, wind and temperature can be produced at any position and time. ADS-B positions of all aircraft are mapped with wind vectors from the model to produce the corresponding airspeeds. Figure 5.3 illustrates an example of the airspeed versus ground speed of aircraft over the Iberian Peninsula, where the difference is large.

The WRAP model uses Mach number (M) and calibrated airspeed (V_{cas}) when appropriate. Mach number can be computed using the following equation:

$$M = \frac{V_{tas}}{a} = \frac{V_{tas}}{a_0 \sqrt{\frac{T}{T_0}}} \quad (5.9)$$

where T_0 and a_0 are temperature and speed of sound at ISA sea level. T and a are the air temperature and speed of sound at the location of the aircraft. The air pressure can be derived from the barometric altitude obtained using ADS-B according to Equation 4.1 from Chapter 4. Knowing the pressure and Mach number, we can compute the calibrated airspeed as follows [145]:

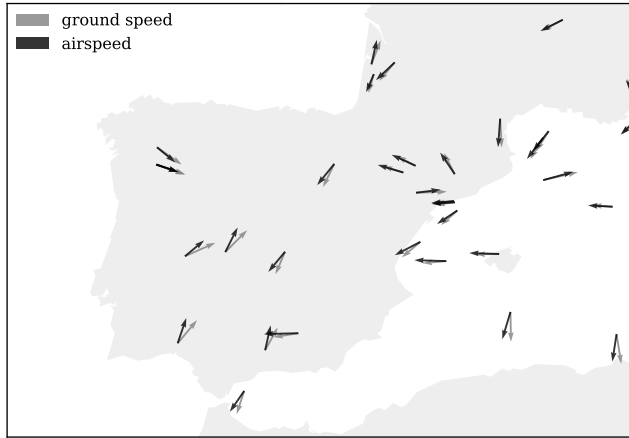


Figure 5.3: Example aircraft ground speed and airspeed difference (cruising aircraft above 20k ft)

5

$$V_{\text{cas}} = a_0 \sqrt{5 \left\{ \left[\frac{p}{p_0} \left\{ (1 + 0.2M^2)^{3.5} - 1 \right\} + 1 \right]^{\frac{1}{3.5}} - 1 \right\}} \quad (5.10)$$

5.4. Construction of WRAP parameters

After the flight trajectories are sorted and segmented to proper flight phases, they can be used to construct the desired model parameters. For each aircraft type, at least five thousand trajectories are used to give a good level of confidence in the model. This section discusses the methods for extracting these parameters from the trajectory data.

5.4.1. Takeoff

During the takeoff phase, the parameters of most interest are takeoff distance (D_{tof}), mean acceleration (\bar{a}_{tof}), and lift-off speed (V_{tof}). To overcome the large noise in the velocity measurements during takeoff, a second-degree spline is applied to obtain a smoothed velocity sample set. The distance parameters can be derived from aircraft surface position at the starting and ending positions of the takeoff, using the spherical law of cosines, based on the dot product of the vectors from the center of the earth to the positions:

$$D_{\text{tof}} = R \cdot \arccos[\sin \phi_1 \cdot \sin \phi_N + \cos \phi_1 \cdot \cos \phi_N \cdot \cos(\lambda_N - \lambda_1)] \quad (5.11)$$

where ϕ and λ represent the latitude and longitude in radians. 1 and N represent the first and last recorded position during takeoff. The average acceleration \bar{a}_{tof} is obtained from a second-order polynomial fitting, based on the time and speed

measurements during the takeoff.

Compared to takeoff distance and average acceleration, the lift-off speed is more complicated to estimate, due to the low data update rate in an aggregated ADS-B dataset. There is usually a gap of a few seconds between the last on-ground data sample and the first in-air observation.

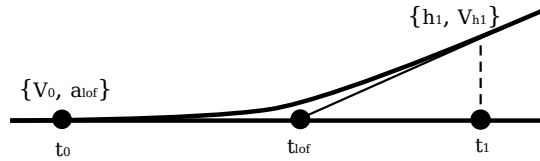


Figure 5.4: Approximation of the moment of lift off

To estimate the exact moment of lift-off, as shown in Figure 5.4, an interpolation model is used. Firstly, the vertical rate VS and time stamp at the first in-air observation are used to estimate the time of lift-off t_{lof} . Combining this result with previously calculated \bar{a}_{tof} , the lift-off speed can be obtained as follows:

$$t_{lof} = t_1 - \frac{h_1}{VS_1} \quad (5.12)$$

$$V_{lof} = V_0 + \bar{a}_{tof} \cdot (t_{lof} - t_0) \quad (5.13)$$

5.4.2. Initial climb

The initial climb phase is defined as the segment from 35 ft until the aircraft reaches around 1500 ft. There are several major configuration changes (retraction of landing gear and flaps) during this short period of time that can affect the performance of the aircraft. However, because the initial climb segment lasts for only a short time, it suffers from having relatively few data samples, compared to the takeoff phase. The parameters to be studied are aircraft calibrated airspeed ($V_{cas,ic}$) and vertical rate (VS_{ic}). Both parameters can be computed directly from ADS-B data.

5.4.3. Climb

The climb segment starts when the aircraft reaches a clean configuration and lasts until the moment when it reaches cruise altitude. As a common practice, aircraft first accelerate to a target CAS and then fly according to this constant CAS. Mach number increases with the increasing true airspeed, as well as the effect of the decreasing speed of sound. When a certain Mach number is reached, an aircraft will fly according to this constant Mach number until its cruising altitude. During the Mach climb segment, a decreasing CAS will be observed due to the decreasing pressure and temperature.

The challenge is to identify the crossover between the constant CAS and constant Mach climb. Knowing the general profile of CAS and Mach number during the climb, it is possible to design a two-step and two-pieceswise estimator for ex-

tracting this feature from CAS and Mach number profiles.

The crossover estimator consists of two parts: an increasing (quadratic or linear) segment and a constant segment. The quadratically increasing part is designed so that it resembles the general observation that velocity increases with a decreasing acceleration. The constant segment approximates constant CAS or Mach values. The estimator is expressed as follows:

$$f_{\text{mach}}(t) = \begin{cases} k_1 \cdot (t - t_{\text{mach}}) + y_{\text{mach}} & t \leq t_{\text{mach}} \\ y_{\text{mach}} & t \geq t_{\text{mach}} \end{cases} \quad (5.14)$$

$$f_{\text{cas}}(t) = \begin{cases} -k_2 \cdot (t - t_{\text{cas}})^2 + y_{\text{cas}} & t \leq t_{\text{cas}} \\ y_{\text{cas}} & t_{\text{cas}} \leq t \leq t_{\text{mach}} \end{cases} \quad (5.15)$$

To illustrate the modeling process, we use a two-step piecewise least-squares fitting as illustrated in Figure 5.5.

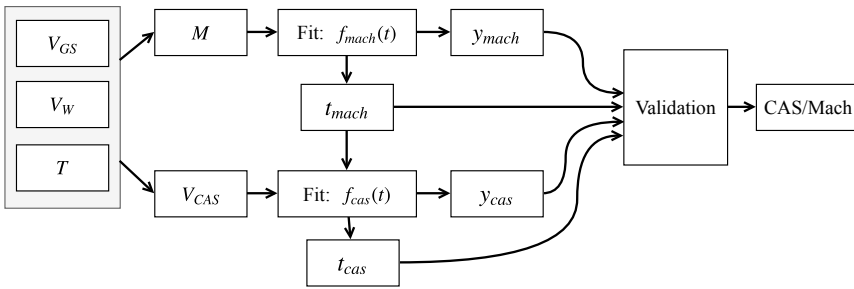


Figure 5.5: Constant CAS/Mach climb identification process

The entire climbing phase in terms of Mach number can be divided into two parts (increasing and constant). The first estimator applies on the entire climb data and uses a least-squares fitting to extract the best CAS/Mach crossover time t_{mach} and constant Mach climb number y_{mach} .

The second estimator uses t_{mach} as cutoff time, fitting the start of the time series until t_{mach} . This estimator is designed to obtain the constant CAS speed y_{cas} and its starting time t_{cas} . The crossover altitude is identified by extracting the actual altitude of the aircraft at time t_{mach} . We can also compute the theoretical crossover altitude (\tilde{h}_{mach}) from the y_{mach} and y_{cas} [145, section 17.2.2]:

$$\begin{aligned} \tilde{h}_{\text{mach}} &= \frac{T_0}{L} \left(\delta^{\frac{-RL}{g_0}} - 1 \right) \\ \delta &= \frac{\left[0.2 \left(\frac{y_{\text{cas}}}{a_0} \right)^2 + 1 \right]^{3.5} - 1}{(0.2 y_{\text{mach}}^2 + 1)^{3.5} - 1} \end{aligned} \quad (5.16)$$

where T_0 and a_0 are temperature and speed of sound at sea level. R and L are specific gas constant and lapse rate. In this chapter, we use this theoretical model to validate the obtained result. The complete validation model is defined as follows:

$$\begin{aligned}
 k_1 &> 0 \\
 k_2 &> 0 \\
 t_{cl,end} - t_{mach} &> 0.05 (t_{cl,end} - t_{cl,start}) \\
 t_{mach} - t_{cas} &> 0.05 (t_{cl,end} - t_{cl,start}) \\
 \text{VAR}[V_{mach}] &< 0.02^2 \\
 \text{VAR}[V_{cas}] &< 10^2 \\
 \frac{|h_{mach} - \tilde{h}_{mach}|}{\tilde{h}_{mach}} &< 0.05
 \end{aligned} \tag{5.17}$$

where $t_{cl,start}$ and $t_{cl,end}$ are the timestamps of start and end of the climb, and VAR represents the variance of the parameter based all the data points in a constant segment. The first two equations ensure the correct model parameters k_1 and k_2 are identified. The second two equations ensure the constant segments last for more than 5% (an empirical value) of the climb duration. The last two conditions examine the variance of all V_{cas} or V_{mach} during the constant segment. They ensure the variations of data values remain below certain thresholds.

To illustrate the entire process, this method is applied on one flight and shown in Figure 5.6. The Mach and CAS profile are shown during the climb, both with a constant segment that has been identified by the model.

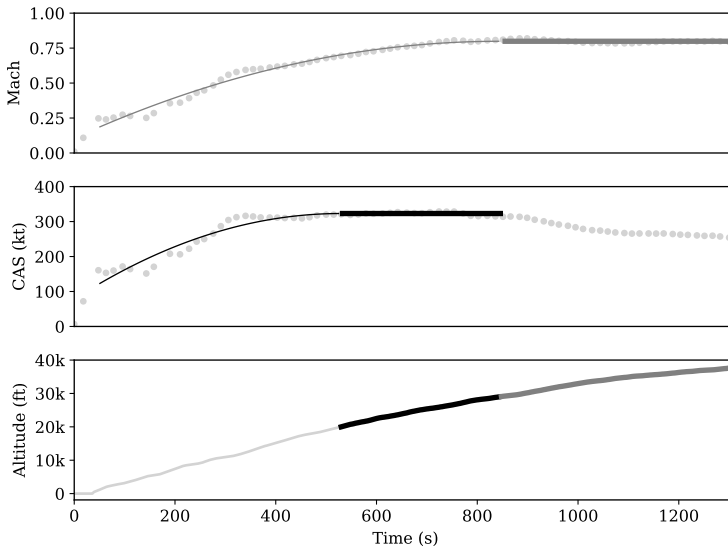


Figure 5.6: Constant CAS/Mach climb identification example

After t_{cas} and t_{mach} are determined, crossover altitudes $h_{\text{cas,cl}}$, $h_{\text{mach,cl}}$, at which the constant CAS and constant Mach climbing commence, can be obtained. Vertical rates $VS_{\text{precas,cl}}$, $VS_{\text{cas,cl}}$, and $VS_{\text{mach,cl}}$ that correspond to these three segments can also be computed. In addition, range from takeoff to the top of climb $R_{\text{top,cl}}$ is calculated.

5.4.4. Cruise

Aircraft generally cruise at selected optimal flight levels that are often the most fuel efficient. Optimal cruise altitude changes as aircraft weight decreases and atmospheric conditions alter. The calculation of the optimal cruise altitude requires the knowledge of thrust and drag forces, atmospheric conditions, as well as the mass of the aircraft. Since the WRAP model deals with the kinematic parameters of the flight, we have to limit the scope of our model. The model focuses on describing the common and maximum values of altitude, speed, and range based on our data.

The common parameters to be modeled are the cruise altitude h_{cr} and cruise Mach number M_{cr} . These two parameters are computed as the mean values respectively during the flight. The initial cruise altitude $h_{\text{init,cr}}$ is also an interesting parameter that defines the top of the climb. It is computed as the average altitude during the first minute of the cruise.

Aircraft operational boundary conditions such as their maximum cruise altitude $h_{\text{max,cr}}$ and maximum cruise Mach number $M_{\text{max,cr}}$ are first computed as the maximum altitude and Mach number per flight. Then, the maximum values of the flights from the same aircraft type are used to model the final boundary conditions in the WRAP model. It is worth noting that the obtained parameters representing operational boundary conditions are the maximum values occurring during the operation. They are often within the performance limitations provided by the aircraft manufacturer.

In addition, the cruise range R_{cr} is extracted as an operational reference parameter. Since aircraft rarely fly a direct route between the origin and destination airport, the flight range cannot be calculated as the great circle distance between origin and destination. Due to the noise inherited from onboard GPS receivers, position reports in ADS-B can contain errors [95]. When integrating positions along the trajectory, the accumulated error may grow larger. Hence, a spline filter can be used in cases where large position errors exist. Then, the trajectory can be re-sampled and integrated to calculate the cruise range. This process is illustrated in Figure 5.7.

5.4.5. Descent

The descent phase of the aircraft is comparable to the climb phase. From the top of descent, the aircraft undergoes a constant Mach and constant CAS descent segment before reaching the approach altitude.

The essential parameters to be modeled are: range from top of descent to destination $R_{\text{tod,de}}$, Mach number during constant Mach descent $v_{\text{mach,de}}$, CAS during constant CAS descent $v_{\text{cas,de}}$, crossover altitudes $h_{\text{mach,de}}$ and $h_{\text{cas,de}}$, and

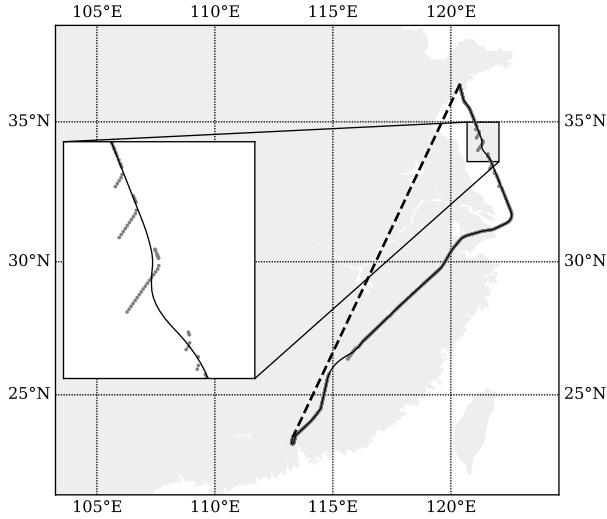


Figure 5.7: Cruise range spline filtering

vertical rates $VS_{\text{mach,de}}$, $VS_{\text{cas,de}}$, and $VS_{\text{postcas,de}}$.

Similar to climb, the constant Mach/CAS descent performance can be modeled by piecewise estimators. It is possible to use the same process as described in Figure 5.5. The Mach profile is described using two linear pieces. The CAS profile consists of a linear and quadratic piece due to the high non-linearity in speed for the final approach segment. The model can be described as follows:

$$f_{\text{mach}}(t) = \begin{cases} y_{\text{mach}} & t \leq t_{\text{mach}} \\ -k_1 \cdot (t - t_{\text{mach}}) + y_{\text{mach}} & t \geq t_{\text{mach}} \end{cases} \quad (5.18)$$

$$f_{\text{cas}}(t) = \begin{cases} y_{\text{cas}} & t_{\text{mach}} \leq t \leq t_{\text{cas}} \\ -k_2 \cdot (t - t_{\text{cas}})^2 + y_{\text{cas}} & t \geq t_{\text{cas}} \end{cases} \quad (5.19)$$

Similarly, the results of such a model as applied on one flight is shown in Figure 5.8.

At lower traffic densities, it can occur that aircraft follow a so-called Continuous Descent Approach (CDA), which eliminates level-flight segments to reduce fuel consumption and engine noise. Such an approach affects the result of the aircraft performance in descent. This effect is discussed in section 5.7.5 in this chapter.

5.4.6. Final Approach

Due to different control procedures at each airport, it is not easy to generalize the entire approach segment solely based on flight data. However, the final approach

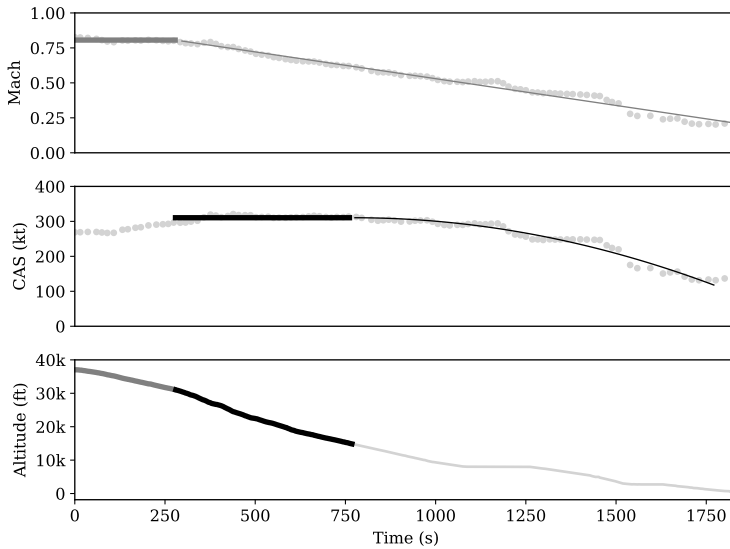


Figure 5.8: Constant CAS/Mach descent identification example

5

segment is considered to start from around 1,000 ft until landing.

The segment of the final approach represents the end of a descent, where aircraft operate at a nearly constant airspeed and rate of descent. The approach speed $V_{cas,fa}$ and rate of descent VS_{fa} are modeled. In addition, the final approach path angle \angle_{fa} is calculated.

5.4.7. Landing

The landing model is comparable with the takeoff model. Parameters such as touchdown speed V_{tcd} , braking distance d_{lnd} , and average braking deceleration a_{lnd} of the aircraft are modeled similarly to the takeoff phase. Approach speed V_{tcd} can be observed from the last in-air velocity. The braking distance can be calculated using Equation 5.11, and average deceleration can be calculated similarly to takeoff acceleration.

5.5. Results

For this chapter, the proposed methods have been applied to 17 common aircraft types. Sufficient data were collected for all aircraft types, approximately 5000 sets of data per flight phase. In order to better illustrate the modeling of each individual parameter, detailed results from a single aircraft type (Airbus A320) are described according to the previous model specification. A summary of the results for other aircraft types is shown in Table 5.1.

5.5.1. An Airbus A320 example

For each parameter, all three probability density functions (Normal, Gamma, and Beta) are applied using MLE. The best model is chosen and represented by different line styles in all figures, shown in solid, dashed, and dotted-dashed line respectively. $\hat{\psi}$, ψ_{\min} , and ψ_{\max} are also marked in each of the plots accordingly.

Takeoff

Three performance parameters D_{tof} , V_{tof} , and \bar{a}_{tof} are shown in Figure 5.9, where the most common values of these parameters are 1.65 km, 165 kt, and 1.93 m/s² respectively.

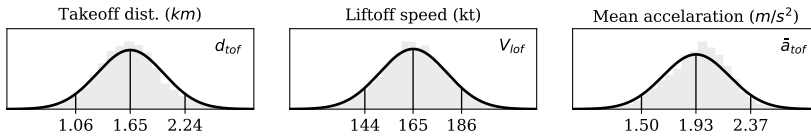


Figure 5.9: Takeoff parameters

Initial climb

Two performance parameters $V_{\text{cas,ic}}$ and VS_{ic} of the initial climb, up to the altitude of 1500 ft, are shown in the first two plots of Figure 5.10, where the most common values are 161 kt and 2477 ft/min.

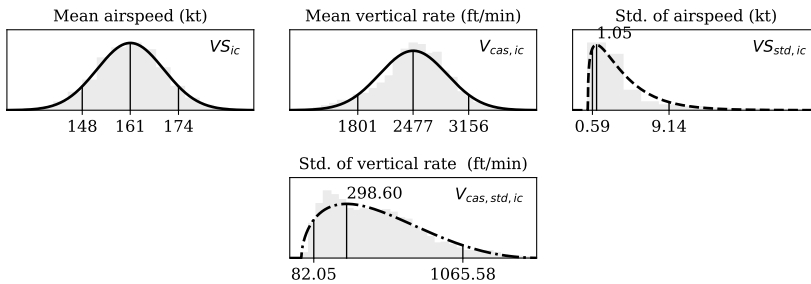


Figure 5.10: Initial climb parameters

The evidence for quasi-constant airspeed and vertical rate assumption can be seen from the standard deviations per trajectory, shown in the last two plots. In general, the vertical rate has larger variances due to the fact that data sources for vertical rate commonly contain a certain degree of uncertainty [51].

Climb

Within the climb phase, the objective is to model the constant CAS/Mach climb profile, as well as the vertical rates in each of the segments of the profile. All parameters are shown in Figure 5.11. Most commonly, before the aircraft accelerates to a constant calibrated airspeed of 293 kt, it has a mean climbing rate

of 2016 ft/min. Once reaching the altitude of approximately 12k ft, the aircraft then climbs at 1600 ft/min until reaching a constant Mach number of 0.78 at the crossover altitude of 28.9k ft. After that, the aircraft climbs at 1039 ft/min until reaching the cruise altitude. The flight range of the climb phase is also shown in the last plot, which is typically around 257 km.

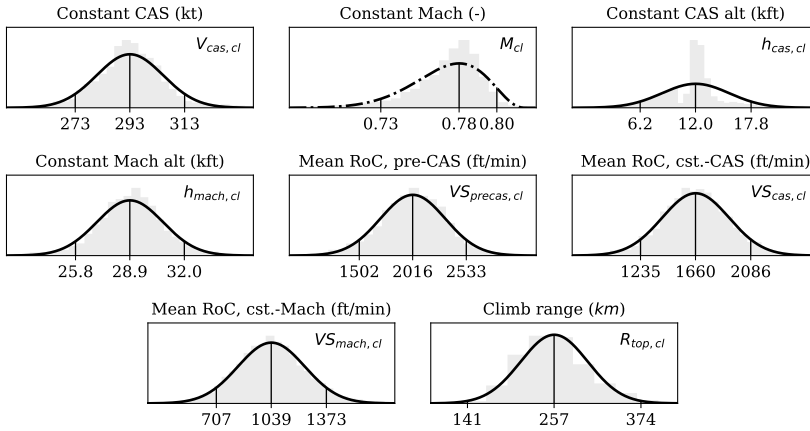


Figure 5.11: Climb parameters

Cruise

During the cruise phase, cruise Mach number M_{cr} , altitude h_{cr} , and cruise range R_{cr} are shown in Figure 5.12 respectively. Maximum cruise Mach number $M_{max,cr}$ and maximum cruise altitude $h_{max,cr}$ can be obtained as the maximum value of M_{cr} and h_{cr} .

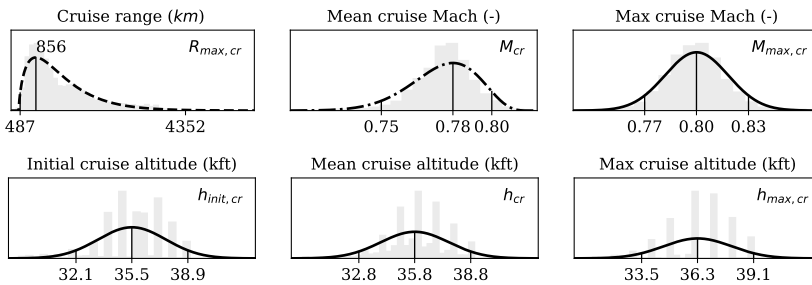


Figure 5.12: Cruise parameters

Unlike other performance parameters, the cruise range R_{cr} is spread very widely, where 99% of flights globally range from 487 km to 4352 km. This reflects the broad usage of the A320. However, most common flights for this aircraft type cruise with a range of less than 1000 km in our global dataset.

Descent

Similar to climb, the descent phase can also be modeled as a constant Mach descent segment and a constant CAS descent segment. The parameters are shown in Figure 5.13. Most commonly, the aircraft starts its initial descent at a constant Mach number of 0.77 and a vertical rate of -1128 ft/min, until reaching an altitude of 31.6k ft. It then starts a constant CAS descent of 281 kt and vertical rate of -1974 ft/min until reaching the altitude 18.9k ft. Then, the aircraft descends with a vertical rate of -1196 ft/min until final approach. The last plot shows the range from top-of-descent to destination, typically around 234 km.

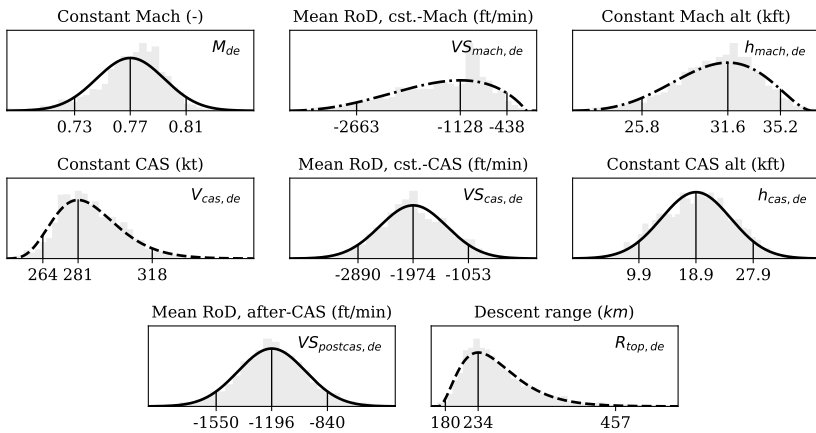


Figure 5.13: Descent parameters

Final Approach

At final approach, airspeed $V_{cas, fa}$ and vertical rate VS_{fa} are shown in Figure 5.14, which are 140 kt and -698 ft/min respectively. The last two plots show the variance of these two parameters within each trajectory. Similar to the initial climb, the quasi-constant airspeed, and vertical rate assumption are still valid. The path angle obtained during the final approach is around 3 degrees.

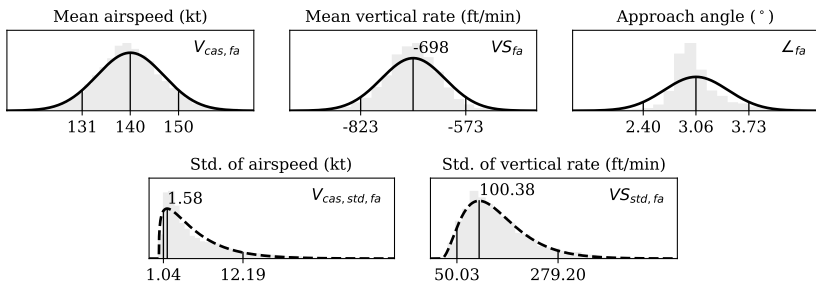


Figure 5.14: Final approach parameters

Landing

For the landing phase, approach speed V_{tcd} , braking distance d_{lnd} , and mean deceleration \bar{a}_{lnd} are shown in Figure 5.15, with values of around 134 kt, 1.08 km, and -1.22 m/s^2 . Braking distance shows a large variance, ranging from around 600 meters to more than 3 km. Different factors such as aircraft weight, wind direction, and runway conditions would all cause such large deviations.

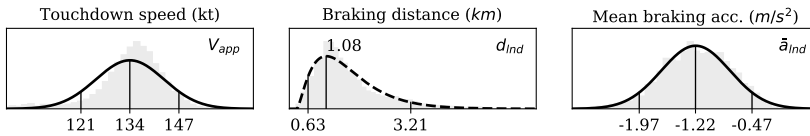


Figure 5.15: Landing parameters

5.5.2. Correlations among parameters

There is usually an interdependence between parameters within each phase. For instance, with higher acceleration, the takeoff distance would be reduced. In order to reveal these relationships, a correlation matrix is computed for each combination of parameters and illustrated using a correlation plot.

In the following graphs, correlations are computed and visualized. The Pearson correlation coefficient (denoted as r) of two variables x and y is computed as:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \tag{5.20}$$

where \bar{x} and \bar{y} are the means of two vectors. When r is close to 1, the variables are positively correlated. When r is close to -1 , variables are negatively correlated. When r is close to 0, there is no correlation between the two variables. In all the following correlation plots, other than displaying computed Pearson correlation coefficient values, the direction of the ellipse shows the negative or positive correlation. The darkness of the ellipse shows the level of correlation.

In Figure 5.16, acceleration is highly correlated with the distance and velocity parameters. This is to be expected as they are always closely related during takeoff.

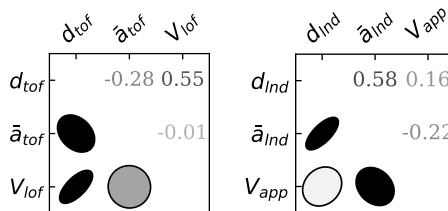


Figure 5.16: Correlations among parameters of takeoff and landing

While in Figure 5.17, during the cruise, for instance, it is possible to see a negative correlation between cruise altitude and cruise range.

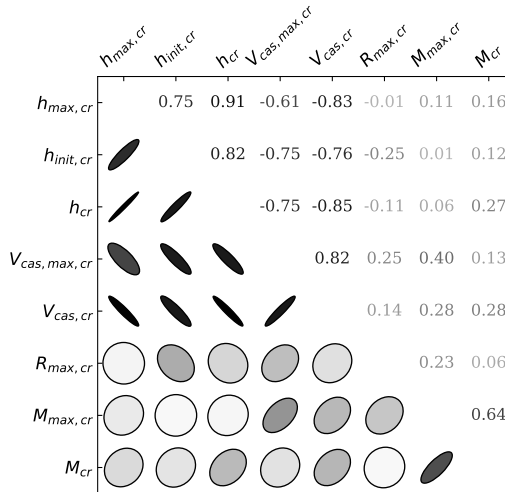


Figure 5.17: Correlations among parameters of cruise

From Figure 5.18, it possible to conclude that during the initial climb, the two parameters that are of interest (horizontal and vertical speed) are not significantly correlated. During the final approach, however, they show a strong negative correlation, which indicates a shallow angle for higher speed approaches.

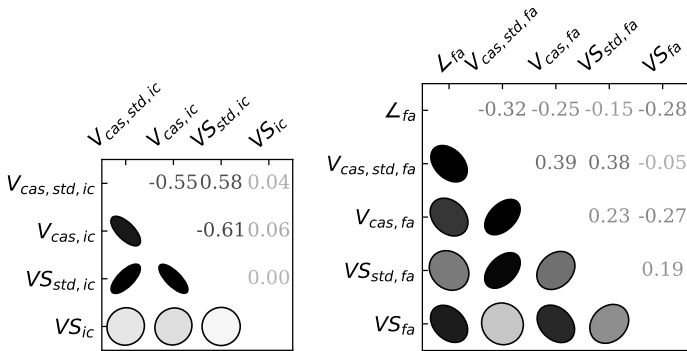


Figure 5.18: Correlations among parameters of initial climb and final approach parameter

During the climb phase, as shown in Figure 5.19, strong positive correlations appear between constant CAS and Mach crossover altitudes, vertical speeds, and velocities and range. Strong negative correlations appear between vertical speeds, range, and velocities, which firmly indicates the law of energy rates during the climbing performance.

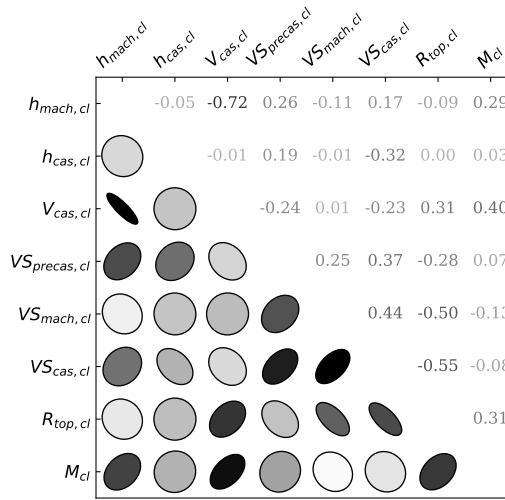


Figure 5.19: Correlations among parameters of climb

5

In Figure 5.19, regarding the descent performance parameters, we can also see similar correlations as described in the climb phase, with the exception that the crossover altitude of constant CAS descent to decreasing CAS demonstrates low correlations with all other parameters. This is possibly due to variation in approach procedures among airports around the world.

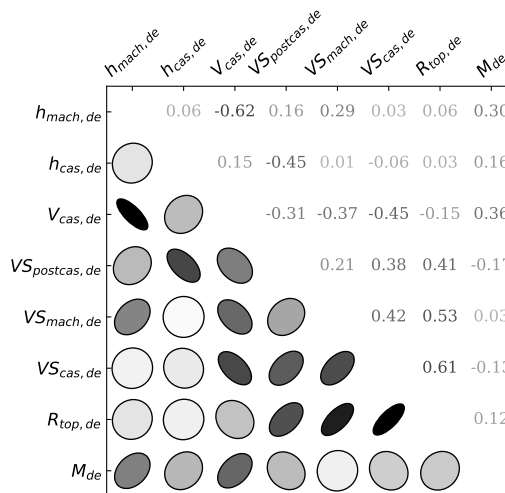


Figure 5.20: Correlations among parameters of descent

5.5.3. Summary based on common aircraft types

The same methods and analysis are applied on a much larger dataset that contains multiple aircraft types. Due to the complexity of displaying all distribution and models for all parameters and aircraft types, the most common parameter values of 17 aircraft types are shown in Table 5.1. In this summary table, we follow the International System of units instead of aeronautical units, so they are consistent with the ones in WRAP dataset. Corresponding aircraft types of the ICAO designators can be found in [65].

5.6. Comparison with existing models

The performance parameters in the WRAP model are all derived from a large amount of flight data, resembling the best estimates of how aircraft truly perform. There are a few existing performance models that can be used as references to examine the validity of the estimated parameters. BADA (version 3.12) [103] and Eurocontrol Aircraft Performance Database [37] are used for comparison. However, not all parameters present in the WRAP model are available in these two existing models. Therefore, only the commons parameters are compared.

Within the BADA (version 3.12), similar performance parameters are contained in the OPF (Operational Performance File) and APF (Airline Procedure File). The differences between BADA and WRAP on ten performance parameters (from 14 aircraft types) are shown in Figure 5.21.

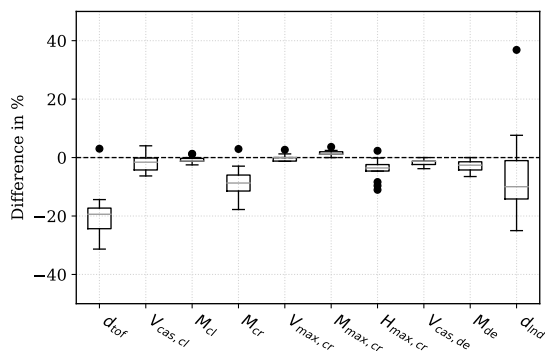


Figure 5.21: Comparison with BADA

Most parameters from different aircraft types are very close to what is presented in BADA, with the exception that takeoff distance (d_{tof}) at around 20% lower than for BADA. This is likely due to the maximum takeoff weight assumed in the BADA model. The same applies to the landing distance. In BADA, the landing distance is also based on the maximum landing weight, for dry, hard, sea level runways under ISA conditions with no wind.

In a similar fashion, the Eurocontrol Aircraft Performance Database can also be used as a source for comparison. It provides more performance parameters, which are used primarily for training purposes on a wide range of commercial

Table 5.1: Most common performance parameters for various commercial aircraft

Phase	Param	Unit	A319	A320	A321	A332	A333	A343	A388	B737	B738	B739	B744	B752	B763	B77W	B788	B789	E190	
TO	V_{lof}	m/s	82.7	85.3	90.8	89.9	89.9	87.0	89.9	83.6	86.5	91.3	92.4	90.5	92.4	96.4	90.3	96.1	88.1	
	d_{lof}	km	1.51	1.65	1.85	1.97	1.86	2.16	2.56	1.50	1.64	1.91	2.29	1.70	1.82	2.21	2.16	2.49	1.80	
	$\dot{\alpha}_{lof}$	m/s ²	1.84	1.93	1.95	1.75	1.72	1.40	1.35	1.77	1.82	1.87	1.67	1.91	1.97	1.89	1.61	1.63	1.83	
IC	V_{Sic}	m/s	80	83	86	86	87	84	88	81	87	91	91	88	88	98	87	92	78	
	$V_{cas,ic}$	m/s	12.27	12.59	13.21	12.48	12.28	6.44	5.65	11.93	12.27	12.36	9.24	12.99	14.28	13.02	10.65	10.60	11.19	
CL	$R_{top,cl}$	km	209	236	250	309	315	305	296	203	217	231	222	227	209	222	294	264	226	
	$V_{cas,cl}$	m/s	147	154	156	156	157	160	165	151	153	159	172	158	163	165	163	168	144	
	M_{cl}	-	0.78	0.78	0.78	0.81	0.80	0.79	0.84	0.79	0.78	0.79	0.84	0.79	0.80	0.83	0.85	0.85	0.76	
	$h_{cas,cl}$	km	3.5	3.5	3.9	3.8	3.7	3.7	4.2	3.5	3.4	4.0	4.2	4.1	4.1	4.7	4.7	4.1	3.5	
	$h_{mach,cl}$	km	10.4	8.6	9.4	10.4	10.1	9.4	10.2	10.4	10.3	9.4	9.6	10.0	9.5	9.8	9.8	10.9	10.5	10.2
	$V_{Sprecas,cl}$	m/s	11.05	10.15	9.25	9.41	8.84	6.76	6.73	8.17	11.60	10.79	10.21	8.49	10.61	10.38	10.07	9.56	8.89	10.85
	$V_{S_{cas,cl}}$	m/s	8.96	8.15	7.09	7.13	6.76	6.55	6.90	10.62	9.29	8.00	9.27	8.57	9.73	8.81	8.28	7.96	7.91	
	$V_{S_{mach,cl}}$	m/s	5.25	5.28	4.73	3.92	4.07	4.07	4.71	5.78	5.29	4.95	5.71	5.44	5.63	4.64	4.94	4.90	4.13	
CR	$R_{max,cr}$	km	4019	4352	5338	11012	8911	13929	20565	4929	4953	4458	12725	6087	9544	14177	11619	14555	2710	
	$V_{gas,cr}$	m/s	125	132	138	129	131	138	136	122	128	138	146	131	138	149	132	139	131	
	$V_{cas,max,cr}$	m/s	128	136	140	135	141	149	143	126	133	145	157	138	146	160	139	148	133	
	M_{cr}	-	0.77	0.78	0.78	0.81	0.81	0.81	0.81	0.84	0.78	0.78	0.84	0.79	0.80	0.84	0.84	0.85	0.85	0.77
	$M_{max,cr}$	-	0.80	0.80	0.80	0.84	0.83	0.83	0.86	0.80	0.80	0.81	0.87	0.82	0.83	0.86	0.87	0.87	0.81	
	$h_{mit,cr}$	km	11.54	10.82	10.34	11.33	11.10	10.01	11.55	12.52	11.13	10.43	10.28	11.00	10.65	9.63	11.65	11.30	10.99	
	h_{cr}	km	11.54	10.92	10.41	11.67	11.49	10.86	11.73	11.74	11.23	10.52	10.81	11.17	10.81	10.35	12.01	11.58	11.05	
	$h_{max,cr}$	km	11.64	11.06	10.56	11.95	12.53	11.70	12.06	11.90	11.38	10.74	11.29	11.64	11.17	11.00	12.28	11.99	11.16	
DE	$R_{top,de}$	km	233	234	236	283	283	281	309	245	241	245	262	247	245	257	293	294	254	
	M_{de}	-	0.76	0.77	0.77	0.81	0.81	0.81	0.83	0.78	0.78	0.78	0.82	0.79	0.79	0.82	0.84	0.84	0.77	
	$V_{cas,de}$	m/s	143	145	149	152	153	152	156	145	147	148	153	151	152	158	155	156	147	
	$h_{mach,de}$	km	9.2	9.6	9.1	10.2	10.2	9.9	10.1	9.6	9.7	9.4	10.0	9.4	9.6	9.7	10.4	10.5	9.3	
	$h_{cas,de}$	km	5.4	5.8	6.3	6.3	6.3	5.8	6.7	5.8	5.9	6.5	6.6	6.4	6.7	7.0	7.2	6.9	5.5	
	$V_{S_{mach,de}}$	m/s	-5.85	-5.76	-5.53	-6.30	-6.15	-5.85	-6.07	-5.52	-5.79	-4.73	-5.40	-6.73	-6.32	-7.00	-7.59	-7.03	-5.24	
	$V_{S_{cas,de}}$	m/s	-9.85	-10.03	-9.42	-9.37	-9.02	-9.27	-8.36	-9.36	-9.93	-8.96	-9.15	-9.56	-10.00	-9.02	-9.54	-9.14	-9.07	
	$V_{S_{postcas,de}}$	m/s	-5.93	-6.08	-6.04	-5.82	-5.75	-5.51	-5.48	-5.89	-6.22	-6.03	-6.28	-6.17	-6.28	-6.29	-6.18	-6.18	-5.95	
FA	$V_{cas,fa}$	m/s	67	72	75	72	73	74	73	70	77	78	79	69	74	78	75	77	70	
	$V_{S_{fa}}$	m/s	-3.42	-3.55	-3.61	-3.62	-3.74	-3.64	-3.71	-3.58	-3.84	-3.92	-4.03	-3.41	-3.76	-4.03	-3.99	-4.15	-3.57	
LD	$\dot{\alpha}_{fa}$	°	3.15	3.06	3.09	2.98	2.97	3.04	2.90	3.08	3.02	3.03	2.98	3.09	2.98	2.95	2.88	2.89	2.92	
	V_{app}	m/s	66.2	69.4	72.9	70.7	71.5	72.2	70.0	68.8	75.4	76.8	77.9	66.7	72.1	76.9	71.3	74.0	66.9	
	d_{hd}	km	1.38	1.08	1.55	1.54	1.48	1.65	2.26	1.91	1.25	1.45	1.64	1.21	1.51	1.49	2.08	2.49	2.10	
	$\dot{\alpha}_{hd}$	m/s ²	-0.87	-1.22	-1.21	-1.18	-1.20	-1.01	-1.01	-0.83	-1.36	-1.33	-1.18	-1.03	-1.10	-1.33	-1.12	-1.11	-1.08	

and military aircraft. A total of 17 parameters from 14 aircraft that exist in both models are compared. The result is shown in Figure 5.22.

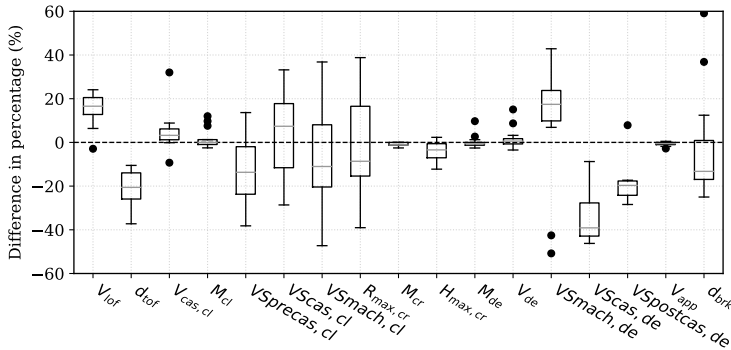


Figure 5.22: Comparison with Eurocontrol Aircraft Performance Database

The parameters that show large differences are the vertical rates during the descent phase. In the Eurocontrol database, it is common that the second descent segment has a much higher vertical rate. However, this trend is not significant in the flight data as observed. It may also be the case that the crossover altitudes obtained from data in the WRAP model are different from the fixed crossover altitude from the Eurocontrol database.

Furthermore, compared to both BADA and the Eurocontrol database, the WRAP model also includes the minimum value, maximum value, and a parametric model for each parameter.

5.7. Discussion

5.7.1. Geo-correlation

The flights used for this chapter are distributed all over the world, which means the obtained distributions of parameters reflect combinations of the performance of all airspaces that are present in the dataset. The advantage of such a dataset is that it gives a complete view of performance from different areas globally. On the other hand, some of these distributions may not be able to reflect certain airspaces, countries, or airports accurately due to factors such as controller procedures or airport limitations. These geographically different situations may especially affect data from the departure and arrival segments of flights. The correlation between performance indicators and locations are not studied in this chapter. Hence, no conclusion can be drawn as to whether the distributions would be significantly different between different locations. This could, however, be an interesting topic for future research.

5.7.2. Point of lift-off and touch-down

During the takeoff and landing phases, because of the low data update rate, the estimation method illustrated in Figure 5.4 is used to locate the most likely lift-off

or touch-down moment, from which related parameters are interpolated.

Statistically, there are relatively large gaps between the last on-ground data point and the first in-air data point. Figure 5.23 shows the time gaps (Δt) observed in the dataset from this study (takeoffs and landings of around 7000 flights).

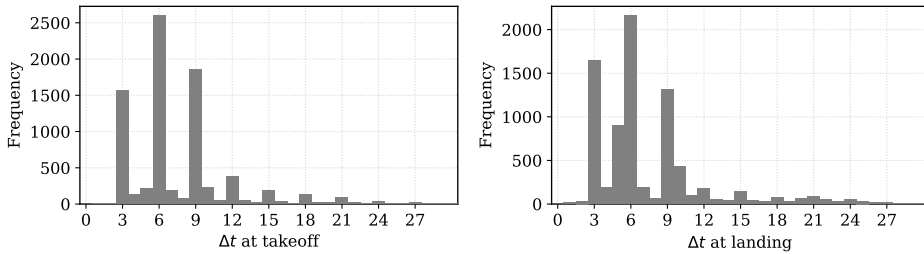


Figure 5.23: Time gap between two data points at takeoff and landing

5

Most commonly, the Δt is about three to nine seconds during which the aircraft would have flown, climbed, or descended for a fairly large distance. Thus, when dealing with such data, using the above method improves the accuracy of parameter estimations during the phases where fewer samples are available. However, the extrapolated liftoff time and speed would diverge from actual values.

It is also interesting to note that there is a three-second time gap in both figures for takeoff and landing. This is likely due to the update rate of aggregated ADS-B data used in this chapter. When considering raw ADS-B data, this gap can be reduced to a half second if receivers are in a good line-of-sight range of the taking off or landing aircraft.

5.7.3. Speed restrictions

It is not uncommon that speed restrictions are imposed by air traffic control for aircraft below a certain altitude. In many airspaces, this restriction is set to be 250 kt below 10,000 ft. In the WRAP model, such a restriction is left out due to the fact that it is hard to generalize this rule for all aircraft models and different airspaces. In Figure 5.24, the calibrated airspeed of Airbus aircraft are shown. We can see that for a majority of the flight, such speed restrictions exist, especially for smaller aircraft types. However, for heavier aircraft types like A380 or A340, the speed restriction cannot always be maintained.

This procedure limitation does influence parameter modeling. In this chapter, we used a piece-wise model (as shown in Equation 5.14) to extract the climb profiles. The quadratic function allows us to handle data under different operating procedures.

Even though the WRAP model does not define these speed restrictions, it is not difficult for future users to implement similar control logic in their own scenarios. Then, from 10,000 ft above, the WRAP model parameter can be used.

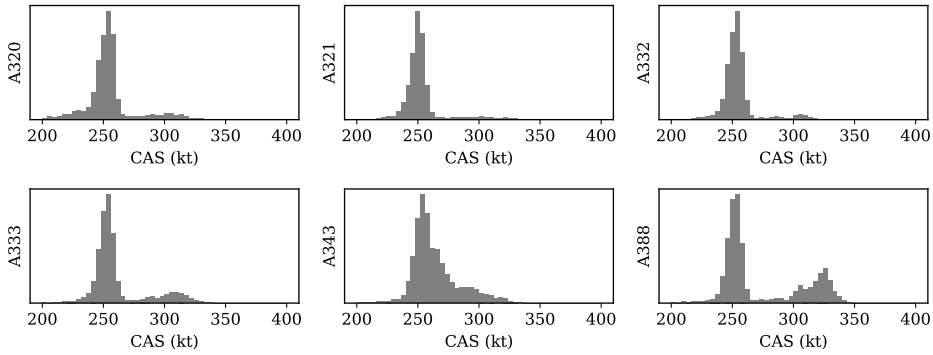


Figure 5.24: Calibrated airspeed at 10k ft per aircraft type

5.7.4. Cruise altitude

It is apparent from Figure 5.12, that the cruise altitude cannot be well described by a continuous distribution. A discrete distribution would be better placed to capture the selection of discrete cruise flight levels. However, for the uniformity of the model parameters, the continuous density is assumed. Nevertheless, the mean and maximum values found from the Normal distribution can still be used in most scenarios.

Another common procedure that is not described in this chapter is the step climbs for long-haul flights. With the decreasing aircraft mass due to fuel consumption, the optimal cruise altitude increases. Airliners often perform step climbs to avoid continuous changes in aircraft altitude, though it is less efficient as pointed out by [27]. It is difficult to model the step climb when compared to other parameters in the WRAP model. Step climbing is a discrete process and happens only for long-haul flights. For long-haul flights, there are often large missing segments in the cruise phase due to limitations in ADS-B receiver coverage. As a result, important climbing data may be omitted/missing. Given these complexities, we have left step climbs out of this chapter. However, by using the existing parameters $h_{init,cr}$, h_{cr} , and $h_{max,cr}$, a continuous cruise climbing profile from initial cruise altitude to the top of descent can be constructed.

5.7.5. Continuous descents

To maximize fuel efficiency and when it is allowed by air traffic controllers, aircraft often perform continuous descent approaches (CDA) in order to save fuel and reduce noise. However, when modeling the aircraft performance, CDA has an impact on some parameters during the entire descent phase. To study the influence of this factor, a similar number of CDA and non-CDA flights (around 3000 flights each) are used. All parameters related to descent are computed and compared in parallel. There are different ways of defining CDA. In this chapter, we considered CDA for flights that do not contain any level-off during the descent.

As shown in Figure 5.25, eight parameters in CDA and non-CDA flights are shown. Crossover altitudes and velocities of constant Mach/CAS descent do not

change between these two approaches, but other parameters show large differences. In the first sub-figure, it is clear that performing a CDA decreases the range of top-of-descent drastically. From sub-figures six to eight, it is also apparent that the descent rates are all increased under CDA. All of these differences are indeed aligned with the definition of CDA. However, other than the range of descent, by separating the cases of CDA and non-CDA, the accuracy for other parameter models do not show significant increases. As a result, the generic models are still proposed for the descents phase. Nevertheless, as an additional precaution, such operational influences should not be neglected when it comes to flight data mining and analytics.

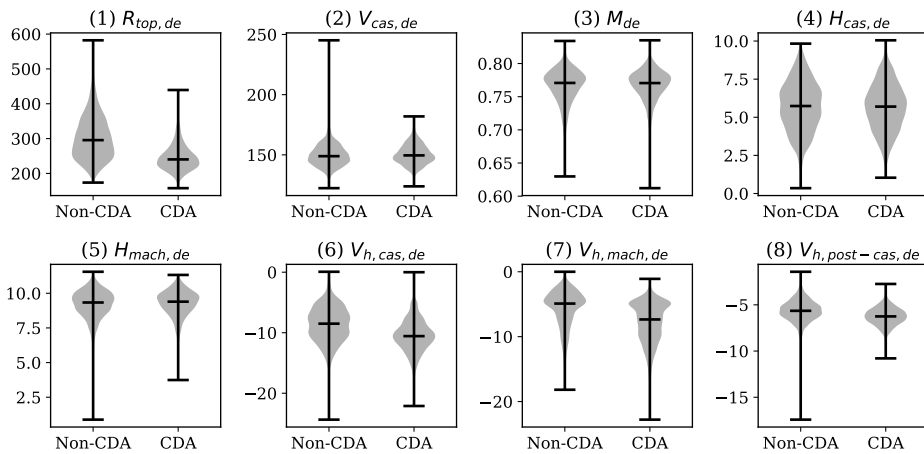


Figure 5.25: Comparison of descent parameters between CDA and Non-CDA

5.7.6. Data and model use cases

ADS-B data are not distributed equally per aircraft type. The same amount of data from most common aircraft types such as Airbus-320 and Boeing-737 can be collected within a week, while for other less common aircraft types such as Airbus-380 and Boeing-777, this process may take several months. While completing this research, a total of 1.6 million flights have been captured during a period of five months for around 30 commercial aircraft types. However, with the constant flow of new ADS-B data, the models can be continuously improved and extended.

The current WRAP model and future updates are published under the GNU open source license. Parameters from the model have been adopted in the BlueSky Open Air Traffic Simulator [54], as one of the core performance models of flight envelopes. At a broader scale, both methods and models addressed in this chapter can be beneficial for other use cases, for example, probability density function based Monte-Carlo simulations and aircraft performance comparison studies.

5.8. Conclusion

In this chapter, we primarily discussed the process of utilizing Automatic Dependent Surveillance-Broadcast (ADS-B) data collected by large crowd-sourced networks to construct aircraft kinematic performance models. The entire process consisted of flight data processing, aggregating atmospheric data using external sources, developing methods that extract performance parameters, and designing parametric statistical models, as well as examining the final WRAP model. As a result, a total of 32 parameters per aircraft type were presented in the current version of the WRAP model, covering the 17 most common aircraft types. The models for the 17 aircraft types were summarized in this chapter and shared under an open-source license.² Comparison studies were performed against existing models such as Eurocontrol Base of Aircraft Data (BADA) and Eurocontrol Aircraft Performance Database.

The WRAP aircraft performance model was constructed based on global flight data observed over an extended period of time. It was not designed to reflect different local variations in the air traffic management sectors. Short-term variations were also not considered in this chapter, which could pose a limitation for the resulting model. However, the set of methods proposed in the study could allow the creation of similar models using user-defined spatial and temporal filtering.

Like other kinematic models, WRAP is not designed for the purpose of single trajectory prediction or optimization. This is the main limitation of the WRAP model. It aims to serve as a realistic model for applications where only kinematic performance models are needed.

It is worth emphasizing that the WRAP model not only defines procedural parameters but also proposes parametric statistical models for these parameters. These statistical models can be used conveniently to conduct Monte-Carlo simulations. The maximum and minimum parameter values in WRAP can be used as performance limitations in simulations. This property enables extended capabilities for aircraft performance based studies, such as large-scale 4D trajectory simulations and air traffic optimizations. With the increase in data gathered by ADS-B receiver networks, the accuracy and aircraft type varieties of the model can be gradually improved.

5.9. Acknowledgment

We would like to thank FlightRadar24 and its network contributors for the global ADS-B dataset used in this Chapter.

²The WRAP model is available at: <https://github.com/junzis/wrap>.

6

Thrust and fuel flow model

The previous chapter discusses the kinematic performance of the aircraft. From this chapter onward, we start to dive into the more complicated dynamic performance models. Thrust and fuel flow are the first two important components that need to be addressed, as many air traffic management studies are dependent on thrust and fuel consumption models. The commonly used BADA model provides the capability to compute maximum thrust and thrust specific fuel consumption. However, its usage is limited by its closed-source license restrictions. As a contribution to the open aircraft performance model, this chapter describes an open approach to compute maximum thrust and fuel flow for aircraft equipped with turbofan engines. First, a thrust model is constructed. It is based on a simplified thrust model proposed by Bartel and Young and a data-driven empirical model proposed by Svoboda. Next, the fuel flow is handled by a new model using the engine test data from the ICAO emission data bank. All data required for the model are open and publicly available. Coefficients required to compute the fuel flow are derived and listed in Appendix C of this dissertation. At the end of the chapter, a verification of the open modeling approach is made with flight management system data from the TU Delft Cessna Citation II aircraft.

Part of this chapter is published in:

Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2018. Aircraft initial mass estimation using Bayesian inference method. *Transportation Research Part C: Emerging Technologies*, 90, pp.59-73 [123].

6.1. Introduction

Any air transportation studies which make use of dynamic performance models require a model that describes engine thrust. Unlike other parameters addressed in this dissertation (drag polar or mass), thrust and fuel consumption are two parameters that are difficult to estimate based on aircraft surveillance data alone.

Aircraft thrust and aircraft mass are two hidden parameters that influence the flight performance significantly. This can also be seen in the total energy equation:

$$(T - D)V = mV \frac{dV}{dt} + mg \frac{dh}{dt} \quad (6.1)$$

We can re-arrange the equation so that the dynamic and kinematic components are on different sides of the equation, which is expressed as:

$$\frac{T - D}{m} = \frac{dV}{dt} + \frac{g}{V} \frac{dh}{dt} \quad (6.2)$$

From Equation 6.2, we can infer that the same trajectory (represented by horizontal and vertical speeds) can be constructed with different combinations of thrust and mass. In the later chapters of the dissertation, we will focus on using this basic concept to estimate aircraft mass. In order to derive the thrust of the aircraft, a different modeling approach has to be adopted.

Nowadays, most airliners are equipped with turbofan engines. As such, turbofan engines are considered in most air transportation studies. The net thrust of the aircraft is dependent on the drag. However, the maximum available thrust, defined by the engine performance, is dependent on the airspeed and altitude of the aircraft. For example, in [5], the relationship between the maximum thrust, airspeed, and altitude are discussed. The maximum takeoff thrust at sea level is modeled as:

$$\frac{T}{T_0} = 1 - K_1 V + K_2 V^2 \quad (6.3)$$

where T_0 is the maximum thrust at zero speed at sea level. K_1 and K_2 are engine specific coefficients.

During the climbing, the maximum thrust decreases with increasing airspeed. It can be considered as a function of the Mach number:

$$\frac{T}{T_0} = AM^{-n} \quad (6.4)$$

where A and n are functions of altitude.

When aircraft is cruising with high airspeed, the maximum thrust can be seen as constant with regards to the speed. In this condition, the relationship between the maximum thrust (T) and maximum static thrust at sea level (T_0) can be simplified as:

$$\frac{T}{T_0} = \left(\frac{\rho}{\rho_0} \right)^m \quad (6.5)$$

where ρ and ρ_0 are air density at the altitude of aircraft and sea level, and m is an engine dependent parameter.

The models that describe the thrust profile are straightforward. However, most of the coefficients are engine dependent. Without data from engine manufacturers, it is hard to obtain specific model coefficients for each aircraft type. In this chapter, we rely on previous literature that studied these coefficients.

A number of turbofan engines are studied by [8] in 2008. The study has led to an open model that describes the coefficients in the thrust equations mentioned earlier. In the second section of the chapter, we focus on explaining the details of this model.

In order to complete the thrust model proposed by [8], one crucial piece of information is required, which is the engine thrust at cruise altitude. In [8], such information is assumed to be known. However, not all engines have their cruise thrust published. In combination with another study proposed by [136], we are able to use an empirical equation to approximate the cruise thrust, when the data is not available.

Knowing the thrust is the first step in computing fuel consumption. The fuel to energy conversion is commonly given as thrust specific fuel consumption (TSFC), which specifies the amount of fuel burnt per unit of time and per unit of force. Engine manufacturers in some cases publish TSFC values of the engines at cruise conditions. This can be used as an indicator to evaluate engine performance during the cruise. However, computing fuel flow in all flight conditions requires a more sophisticated model.

There are generally two ways to compute fuel consumption in the literature. The first and most common approach is to first estimate the net thrust of the aircraft based on the trajectory data (assuming certain mass). The fuel flow can then be computed based on the TSFC model given by the BADA. This method can be seen in [20], [34], and [28]. The second approach is to skip the physical model of flight dynamics and BADA and to instead use a supervised machine learning model constructed from flight data recorder information. These machine learning based methods using Gaussian process regression can be seen in [18] and [16].

Both approaches, however, present drawbacks. As mentioned earlier, the restricted license is the limitation of the BADA. On the other hand, the black-box machine learning model does not easily express the fundamental relationships between aircraft states and fuel consumption. The construction of the black-box model also requires proprietary and often closed-source data from aircraft flight management systems.

In the third part of this chapter, we focus on explaining the open fuel flow model constructed based on the ICAO emission data bank. In section four, both thrust model and fuel flow model are verified with flight recorder data from TU Delft's Cessna Citation II aircraft.

6.2. Model thrust of turbofan engines

In [8], three simplified thrust models for turbofan engines are proposed. The first one is used for calculating the maximum takeoff thrust at sea level. The second model takes the first model as a basis and also considers the altitude, so that maximum takeoff thrust can be computed at non-sea-level conditions. The third model calculates the maximum thrust for en-route conditions.

All proposed models contain empirical coefficients that are obtained using existing engine data. Though the original data were not published, most of the model coefficients are given. In this section of the chapter, we reproduce the equations that were proposed by [8], with some alterations to improve the computational efficiency.

In the proposed simplified thrust model, many numerical coefficients exist in the equations that are listed in the rest of this section. According to [8], these values are obtained from the model fitting using available engine data.

6.2.1. Takeoff thrust at sea level

According to Equation 6.3, the takeoff thrust can be considered as a (second-degree polynomial) function of the speed. The coefficients of the polynomial function are engine specific. In [8], the coefficients are further modeled as functions of the bypass ratio. Thus, the relationship between the maximum takeoff thrust and the maximum static thrust can be described as:

$$\frac{T}{T_0} = 1 - \frac{0.377(1 + \lambda)}{\sqrt{(1 + 0.82\lambda)G_0}}M + (0.23 + 0.19\sqrt{\lambda})M^2 \quad (6.6)$$

where λ is the bypass ratio of the engine. Coefficient G_0 is also a function of the bypass ratio:

$$G_0 = 0.0606\lambda + 0.6337 \quad (6.7)$$

According to the validation based on different engine performance data used in [8], the error of thrust produced by Equation 6.6 is within $\pm 1\%$ regarding the true thrust, with the speed up to Mach 0.4 during the takeoff.

6.2.2. Takeoff thrust with the influence of altitude

The previous model assumes the condition of being at sea-level. However, not all runways are at sea-level. Taking into consideration the altitude of the airport, the maximum takeoff thrust can be modeled similarly:

$$\frac{T}{T_0} = A - \frac{0.377(1 + \lambda)}{\sqrt{(1 + 0.82\lambda)G_0}}ZM + (0.23 + 0.19\sqrt{\lambda})XM^2 \quad (6.8)$$

where A , Z , and X are newly introduced coefficients that are related to the pressure ratio of δ . Let δ be:

$$\delta = \frac{p}{p_0} \quad (6.9)$$

Coefficients A , Z , and X in Equation 6.8 can be computed as:

$$\begin{aligned} A &= -0.4327\delta^2 + 1.3855\delta + 0.0472 \\ Z &= 0.9106\delta^3 - 1.7736\delta^2 + 1.8697\delta \\ X &= 0.1377\delta^3 - 0.4374\delta^2 + 1.3003\delta \end{aligned} \quad (6.10)$$

6.2.3. Maximum thrust for en-route flights

To compute the thrust, the climbing phase is divided into three segments: 0-10k ft, 10k-30k ft, and 30k-40k ft. In each of these segments, the thrust is computed by different equations. Instead of computing the thrust ratio of the net static thrust at sea level, the thrust is modeled in relation to the net thrust at cruising altitude (of 30k ft), denoted as T_{cr} . In addition, the calculation of thrust also depends on a reference Mach number, M_{cr} , and calibrated airspeed, $V_{cas,cr}$, at the cruising altitude. The thrust is computed in three steps:

1) For the en-route segment from 30k ft to 40k ft, the thrust ratio is modeled as follows:

$$\frac{T}{T_{cr}} = c_1 \ln \left(\frac{p}{p_{cr}} \right) + c_2 \quad (6.11)$$

where c_1 and c_2 are calculated as:

$$\begin{aligned} c_1 &= -0.4204 \left(\frac{M}{M_{cr}} \right) + 1.0824 \\ c_2 &= \left(\frac{M}{M_{cr}} \right)^{-0.11} \end{aligned} \quad (6.12)$$

2) For the segment from 10k ft to 30k ft, the thrust can be similarly computed. Instead of using the reference Mach number, the reference calibrated airspeed is used. The thrust in this segment is calculated as:

$$\frac{T}{T_{cr}} = c_3 \left(\frac{p}{p_{cr}} \right)^{c_4} \quad (6.13)$$

where c_3 and c_4 are calculated as follows:

$$\begin{aligned}
 c_3 &= \left(\frac{V_{\text{cas}}}{V_{\text{cas,cr}}} \right)^{-0.1} \\
 c_4 &= -0.335 \left(\frac{V_{\text{cas}}}{V_{\text{cas,cr}}} \right) + c_5
 \end{aligned} \tag{6.14}$$

In order to find the value of c_5 , a look up table is used in [8]. To simplify the computation, a linear approximation for c_5 is adopted in this dissertation, which is expressed as follows:

$$c_5 = 2.667 \cdot 10^{-5} \text{ VS} + 0.8633 \tag{6.15}$$

where the unit of vertical speed (VS) is feet/minute.

3) For the first climbing segment up to 10k ft, a linear model is used to approximate the thrust ratio, with respect to the thrust ratio at 30k ft. The relationship is as follows:

$$\frac{T}{T_{\text{cr}}} = c_6 \left(\frac{p}{p_{\text{cr}}} \right) + \left[\frac{T_{10}}{T_{\text{cr}}} - c_6 \left(\frac{p_{10}}{p_{\text{cr}}} \right) \right] \tag{6.16}$$

where T_{10} is the thrust at 10k ft. It is first computed using the Equation 6.13. Coefficient c_6 is introduced to incorporate the variations of speeds and vertical rates. In [8], a look up table is used to find the value of c_6 . However, to increase the computation efficiency, a function is constructed to approximate m lookup values from Table 4 in [8]:

$$c_6 = -1.2043 \cdot 10^{-1} \left(\frac{V_{\text{cas}}}{V_{\text{cas,cr}}} \right) - 8.8889 \cdot 10^{-9} \text{ VS}^2 + 2.4444 \cdot 10^{-5} \text{ VS} + 4.7379 \cdot 10^{-1} \tag{6.17}$$

where c_6 can be considered as a bi-variate polynomial function that is dependent on the Mach number and the vertical rate.

6.2.4. Reference cruise thrust

One of the parameters used in this empirical two-shaft turbofan engine thrust model is the reference thrust at 30k ft altitude, denoted as T_{cr} . However, this information is not always published by aircraft or engine manufacturers. In Appendix C, a know list of cruise thrust for different aircraft and engine types is summarized based on available information.

When T_{cr} is not available for a specific engine, we rely on the empirical model proposed in [136], where the cruise thrust model is constructed with respect to the maximum static thrust as:

$$T_{\text{cr}} = 0.2T_0 + 890 \tag{6.18}$$

6.2.5. Summary

Combining all previous equations, it is possible to see that the model of net thrust involves multiple parameters of the flight. The details of the model are also different in each flight phase. To generalize the model, in the remaining chapters, the following equation is used to represent the model to calculate the maximum thrust:

$$T = f_{\text{thr}}(h, V, \text{VS}) \quad (6.19)$$

6.3. Model engine fuel flow

One of the most comprehensive public data sources on aircraft engines is the *ICAO aircraft engine emission data-bank* [63]. The database is designed to provide emission information of aircraft engines concerning their environmental impact. In addition to its quantification of greenhouse gas emissions, the database also provides several fuel flow indicators for each engine.

The ICAO engine emission data-bank defines fuel flow under four different modes, which are takeoff, climb-out, approach, and idle. The power settings of the engine are defined at 100%, 85%, 30%, and 7% of engine maximum power respectively. Under the static testing condition, these engine fuel flow data are obtained and published.

In order to create a model that can compute actual fuel consumption under different thrust settings, we construct a continuous fuel flow profile based on the four points from the ICAO engine emission data-bank. These four data points are fitted with a 3rd-degree polynomial model. The coefficients for the polynomial function are denoted as $C_{\text{ff}3}$, $C_{\text{ff}2}$, and $C_{\text{ff}1}$. In Figure 6.1, the results for some example aircraft types and engines are illustrated.

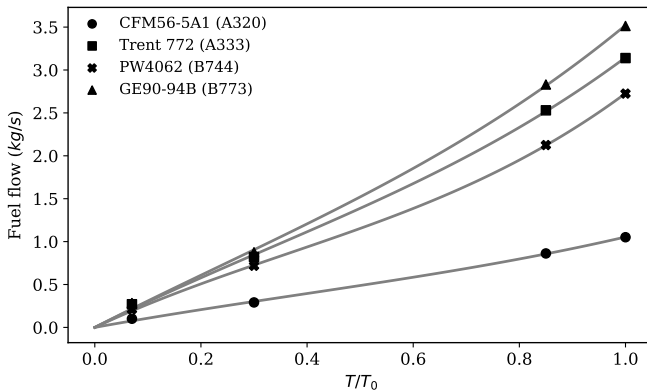


Figure 6.1: Interpolation of fuel flow based on ICAO emission data

The fuel flow profile defines the relationship between fuel flow and the percentage of the maximum static thrust. Denoting $f_{\text{fuel,SL}}$ as the function representing fuel flow (kg/s) at the sea level, the dynamic fuel flow model can be computed

knowing the net thrust:

$$f_{\text{fuel,SL}}(T) = C_{\text{ff3}} \left(\frac{T}{T_0} \right)^3 + C_{\text{ff2}} \left(\frac{T}{T_0} \right)^2 + C_{\text{ff1}} \frac{T}{T_0} \quad (6.20)$$

With the increasing altitude, the amount of fuel flow is increased in order to produce the same amount of the thrust. This difference can be seen by the thrust specific fuel consumption (SFC) at sea-level and cruise conditions. In Figure 6.2, the difference of SFC between sea level and cruise altitude for multiple engines are shown.

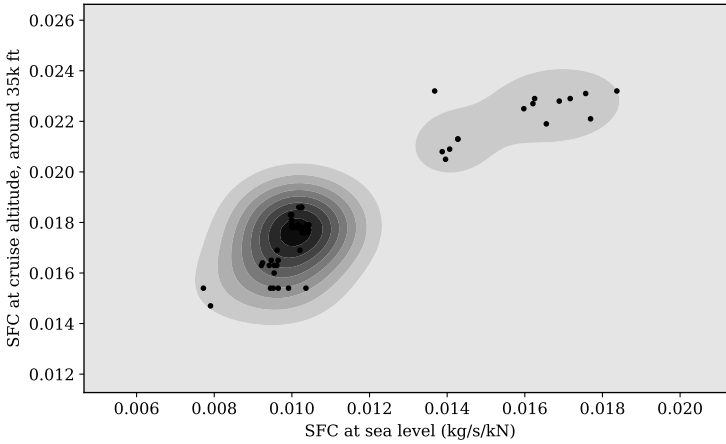


Figure 6.2: Difference in specific fuel consumption at sea level and cruise altitude

Based on the SFC at sea level and cruise altitude, we propose a linear correction factor ($C_{\text{ff,ch}}$) that takes into consideration the altitude effect. When SFC is available at the cruise altitude, the correction factor is calculated as:

$$\begin{aligned} C_{\text{ff,ch}} &= \frac{\text{SFC}_{\text{CR}} - \text{SFC}_{\text{SL}}}{h_{\text{CR}}} \\ &= \frac{\text{SFC}_{\text{CR}} - f_{\text{fuel,SL}}(T_0)/T_0}{h_{\text{CR}}} \quad \left(\text{unit: } \frac{\text{kg}}{\text{s} \cdot \text{kN} \cdot \text{m}} \right) \end{aligned} \quad (6.21)$$

where the h_{CR} is the cruise altitude for the corresponding SFC. Figure 6.3 illustrates the coefficient for engines with available SFC_{CR} . The mean value is found to be around $6.7 \cdot 10^{-7}$, which is used as a default value for engines where SFC_{CR} is not available.

Combining the fuel flow at sea level condition and the fuel flow altitude correction, the dynamic fuel flow can be computed. The final fuel flow model is expressed as:

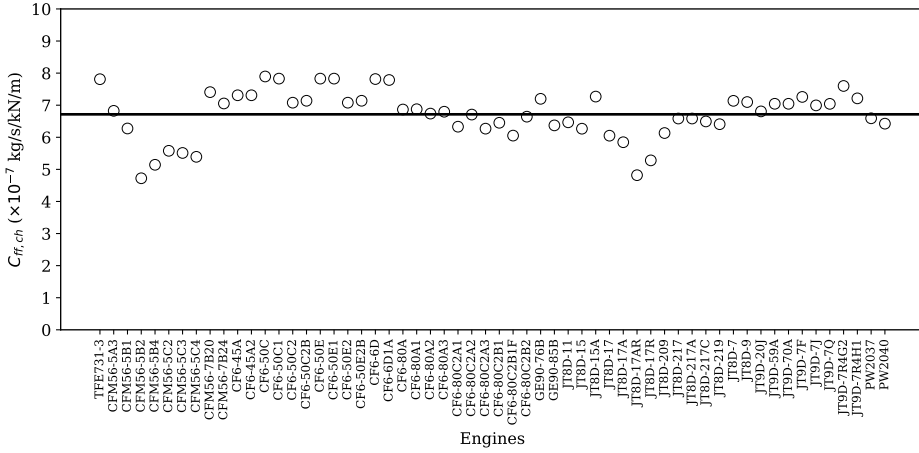


Figure 6.3: Difference of specific fuel consumption between sea level condition and cruise condition

$$\begin{aligned}
 f_{\text{fuel}}(T, h) &= f_{\text{fuel,SL}}(T) + C_{\text{ff,ch}} \cdot T \cdot h \\
 &= C_{\text{ff3}} \left(\frac{T}{T_0} \right)^3 + C_{\text{ff2}} \left(\frac{T}{T_0} \right)^2 + C_{\text{ff1}} \frac{T}{T_0} + C_{\text{ff,ch}} \cdot T \cdot h
 \end{aligned} \tag{6.22}$$

6.4. Discussion

In this section, we first evaluate the approximation made to two parameters from the two-shaft turbofan thrust model proposed in [8]. Next, based on the ICAO engine emission data, we compute and provide the fuel flow coefficient of the most common aircraft types (with available engine configurations). Last, based on flight management system data, we verify the proposed thrust and fuel flow models. In addition, the differences between the proposed models and BADA model are also measured.

6.4.1. Evaluation of thrust model coefficients

In [8], the parameters c_1 , c_5 , and c_6 are given by a lookup table. The disadvantage of the lookup table is that it provides discrete parameter values, which are not efficient for computation. It also does not efficiently handle values that are not in the table. In this chapter, two function based models are constructed to approximate these parameters (Equation 6.12, 6.15, and 6.17). An easy way to verify these low dimension functions is through visualization, where we can inspect values from the lookup table with the function based models.

In Figures 6.4, 6.5, and 6.6, values from the lookup table and the approximation for c_1 , c_5 , and c_6 are shown. From all these figures, we can see that the equations 6.12, 6.15, and 6.17 are able to accurately approximate the c_1 , c_5 , and c_6 . The non-linear relationships in Figure 6.6 is also shown. Using these functions,

we can further interpolate c_1 , c_5 , and c_6 at input values that are not described in the lookup table.

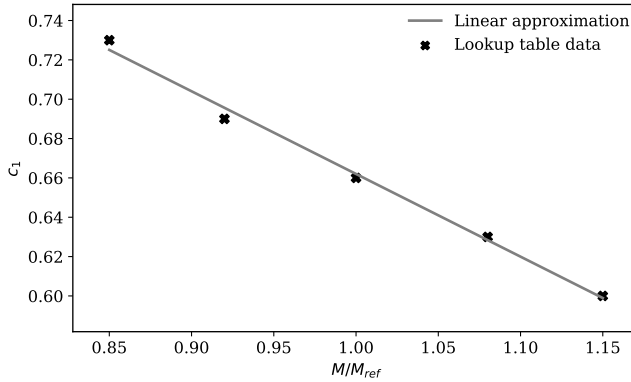


Figure 6.4: Comparison of the linear approximation with the lookup table data (c_5)

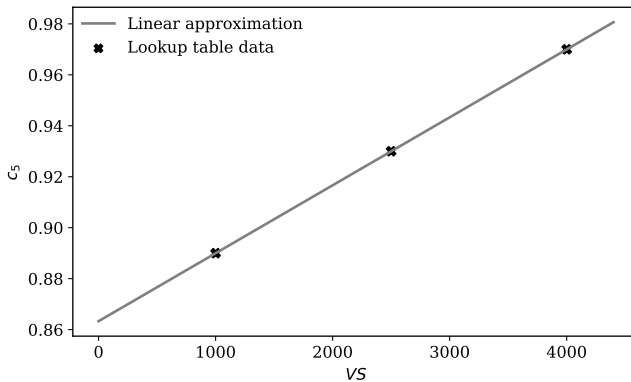


Figure 6.5: Comparison of the linear approximation with the lookup table data (c_5)

6.4.2. Coefficients of engine performance

In order to simplify the process for future studies, we pre-computed the coefficients from Equation 6.22 using the data from the ICAO engine emission databank. In Appendix C, the coefficients for common aircraft are listed. Since most aircraft can have multiple engine options, the fuel flow model can vary among aircraft within a fixed aircraft type. In this table, different engine options for the same aircraft type are available.

6.4.3. Comparison of OpenAP and BADA

To demonstrate the difference between the open models and BADA, we computed the maximum thrust of an aircraft (Airbus A320 with V2500-A1 engines) under

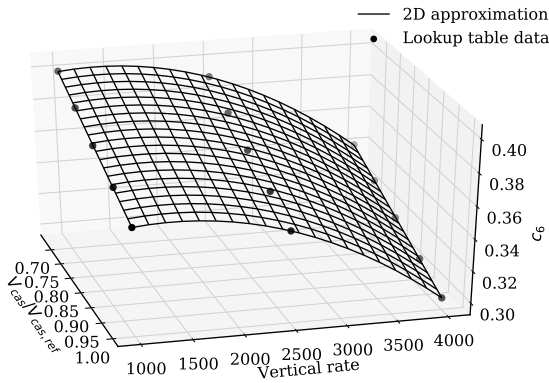


Figure 6.6: Comparison of the bi-variate approximation with the lookup table data (c_6)

different flight conditions. The result is shown in Figure 6.7. The maximum thrust is computed using different combinations of altitude and speed.

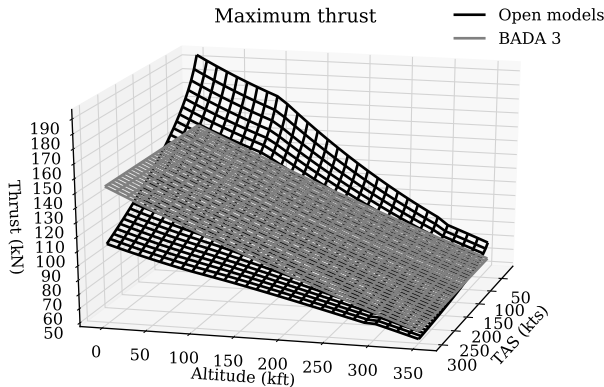


Figure 6.7: A320 maximum thrust (V2500-A1 engine)

A large difference is visible for the aircraft at low altitude. The difference between open models and BADA becomes smaller when the aircraft is at a higher altitude. One of the major reasons for the difference is that BADA 3 does not model thrust dynamic due to the change in airspeed. In contrast, the open models show a clear nonlinear relationship between aircraft speed and thrust. The performance is more realistic for a lower altitude at a lower speed.

We can also compare the results of fuel flow models. BADA's fuel flow model is dependent on the thrust and the speed, while the OpenAP fuel flow model depends on the thrust and the altitude. In Figure 6.8, the difference between the two fuel flow models is shown.

The performances of both fuel flow models are similar at a lower speed. However, while the aircraft is flying at a higher speed, the OpenAP fuel flow model

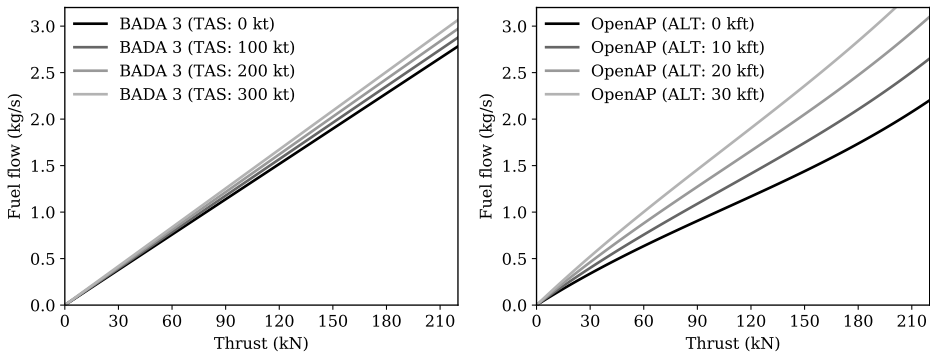


Figure 6.8: A320 fuel flow (V2500-A1 engine)

depends more on the altitude of the aircraft.

6.4.4. Comparison of OpenAP, BADA, and flight data

In this section, we compare the thrust and fuel flow model with data obtained from the flight management system of a Cessna Citation II flight. The flight was conducted for a student practice course at the TU Delft, on March 20, 2017. In Figure 6.9, the trajectory profile is shown.

6

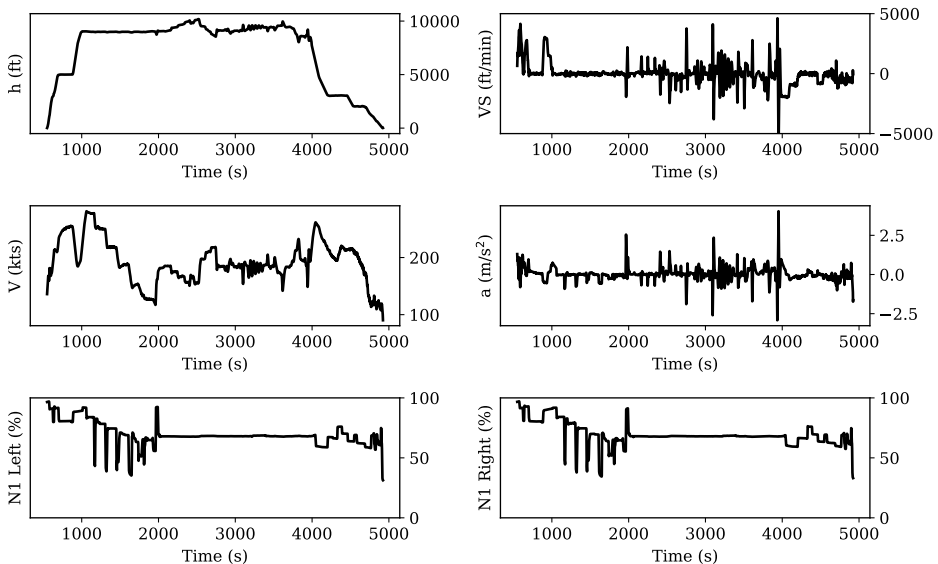


Figure 6.9: Trajectory profile of the flight

The first four plots show the altitude (h), vertical speed (VS), true airspeed (V), and acceleration (a) respectively. The last two plots illustrate the N1 profiles

for the left and right engines. N1 represents the rotational speed of the low-pressure spool of the turbofan engine, with 100% being the highest engine rotation speed. It is a parameter that is directly related to the power setting of the engine.

The distinct characteristics of this practice flight profile should be noted. First, the aircraft is cruising at a lower altitude (around 10,000 ft) instead of the optimal altitude (over 30,000 ft). Secondly, the flight procedure is designed to test the envelope of the flight performance. Thus, drastic changes in speed and altitude occur during the flight.

With the known aircraft mass, we can compute the true thrust of the aircraft based on these data. In Figure 6.10, the true thrust is compared with an estimated maximum and net thrust using the thrust model explained earlier. We also compute the maximum and net thrust using the BADA model in the same figure.

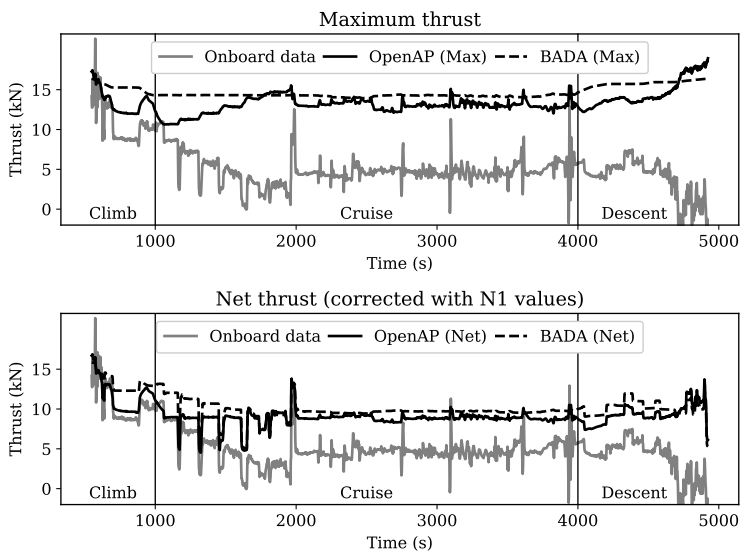


Figure 6.10: Thrust estimation comparison using two-shaft turbofan model, BADA, and FMS data

The flight is divided into three segments, which are the climb, cruise, and descent. In the upper plot, the solid and dashed black lines represent the estimated maximum thrust and BADA maximum thrust respectively. The gray solid line represents the true net thrust. Similarly, the lower plot illustrates the estimated net thrust based on the open model and BADA. The net thrust is computed as the product of the maximum thrust and percentage of the total power. The percentage of total power is computed as the averaged N1 values for left and right engines.

Looking at the comparison of the net thrusts in the lower plot, we can see that the open two-shaft turbofan thrust model is able to approximate the thrust to some degree of accuracy. For this particular flight carried out by our Cessna Citation II aircraft, the proposed open model yields a better approximation compared to BADA 3 model. When examining the top plot, we can see that the highest true net thrust corresponds to the maximum thrust during the period when N1 is close to

100%. This occurs at the start of the climb and around 2000 seconds during the cruise.

It is important to pay attention to the difference between the estimated thrust and true thrust computed using onboard data. This difference could be due to the nonlinear relationship between the fan rotation speed ($N1$) and the actual throttle setting.

Next, based on the net thrust (estimated and true), we can compute the fuel consumption using our fuel flow model. In Figure 6.11, the estimated fuel flow profiles are illustrated. The true fuel flow recorded in the flight management system is also shown in this figure. In the upper plot of the Figure 6.11, the estimated net thrust is used in the fuel flow model. In the lower plot of the figure, the true thrust is used in the fuel flow model. Estimations are drawn in solid black, while the true fuel flow is drawn in gray.

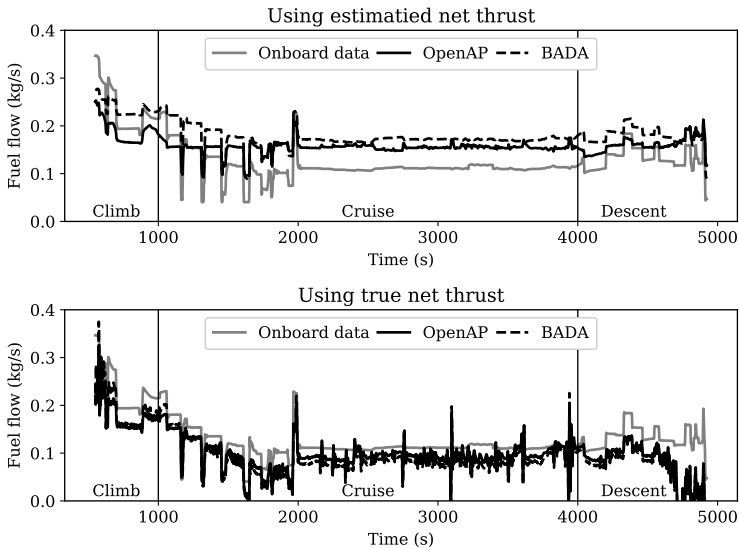


Figure 6.11: Fuel flow estimation comparison using two-shaft turbfan model, BADA, and FMS data

By comparing the difference between true and estimated fuel flow, we can compute the error metric for the open model and the BADA model. The results are shown in Table 6.1. We can see that when using our open model, with the estimated net thrust, the mean absolute error is reduced in respect to the BADA 3 model. When true net thrust is used, the open model still shows less error than the BADA model.

6.5. Conclusion

In this chapter, we described two important open models (thrust and fuel consumption) related to aircraft performance. The thrust model was created by combining two open models proposed in the literature. The fuel flow model was

Table 6.1: Fuel flow error metric

Thrust	Model	Mean absolute error	In percentage
Estimated thrust	Open model vs true	0.0400 kg/s	35.61
	BADA 3 vs true	0.0524 kg/s	46.56
True thrust	Open model vs true	0.0326 kg/s	25.66
	BADA 3 vs true	0.0381 kg/s	31.36

constructed based on the open data available in the ICAO engine emission data-bank. In addition to the equations that calculate the thrust and fuel flow model, related engine parameters are also listed in the Appendix C of this dissertation.

By using the recorded fuel consumption data from a Cessna Citation flight, we verified the fuel flow computed using the open model and BADA 3 model against the true values. The fuel flow estimated using our open models was found to be closer to true values than that of estimations made with BADA 3.¹

It is important to note that using one flight alone could not sufficiently validate the models. However, this could be considered as a quick verification for both thrust and fuel flow models. In order to fully test the models, ideally, multiple flights from different aircraft types shall be tested in the future.

¹Compared to BADA 3, the newer BADA 4 provides a more accurate fuel flow model. However, we lack a license to make any comparison in this dissertation.

7

Drag polar model

In the last chapter, a thrust and fuel model are proposed. In this chapter, another crucial element of the performance model, the aerodynamic model, is studied. In air traffic management research, the aerodynamic properties of an aircraft determine a crucial aspect of the aircraft performance model. Deriving accurate aerodynamic coefficients requires detailed knowledge of an aircraft's design. These designs and parameters are often protected by aircraft manufacturers and can rarely be used in public research. On the other hand, very detailed aerodynamic models are not always necessary in air traffic management studies, as a simplified point-mass aircraft performance model can be used. In these studies, a quadratic relation is assumed to compute the drag of an aircraft based on the required lift. This so-called drag polar describes an approximation of the drag coefficient based on the total lift coefficient. In this chapter, using surveillance data, we estimate the drag polar coefficients based on a novel stochastic total energy model using Bayesian computing. The method is based on a stochastic hierarchical modeling approach and is made possible given accurate open aircraft surveillance data and additional analytical models from the literature. The resulting drag polar model can be used to compute aircraft total drag under clean configuration, as well as when flaps and landing gear are deployed. In addition, the wave drag due to compressibility at high Mach number is also modeled.

This chapter is based on the following publication:

Sun, J., Hoekstra, J.M. and Ellerbroek, J.M., 2018. Aircraft Drag Polar Estimation Based on a Stochastic Hierarchical Model. Proceedings of the 8th SESAR Innovation Days [131].

7.1. Introduction

Since the invention of aircraft, researchers have been studying the aerodynamic properties of airfoils and aircraft. Examples of fundamental studies on aerodynamic drag are given by [55, 56]. From the start, the main goal has been to optimize the lift over drag ratio for the cruise flight. Much effort has been dedicated to creating designs that would reduce drag and thus increase the lift efficiency of aircraft. While the zero drag coefficient contains the parasitic drag of the whole aircraft, the wing is mainly responsible for the lift-induced drag. Besides the choice of the airfoil, the aspect ratio of the wing plays an important role. The wing can be seen as a drag to lift converter, of which the already high efficiency can be increased further. Current research on such topics as boundary layer suction, morphing wings, plasma control, and blended wing-body aircraft shapes all contribute to this effort.

Lift and drag are considered as functions of the wing area, dynamic airspeed, and air density, and the remaining effects of the flow for both the lift and drag are described with coefficients for both forces. The most challenging aspect is to accurately calculate the lift and drag coefficients. They are dependent on the Mach number, the angle of attack, the boundary layer, and, ultimately, the aircraft aerodynamic shape. For fixed-wing aircraft, these coefficients are often modeled as functions of the angle of attack, which is the angle between the aircraft body axis and the airspeed vector. In air traffic management research, however, the simplified point-mass aircraft performance model is mostly used. This model considers an aircraft as a dimensionless point, where the angle of attack, as well as the side-slip angle and the effect of the angular rates, are not explicitly considered. Instead, the *drag polar* is used. Drag polar expresses the relationship between the drag coefficient and lift coefficient, which is one of the main factors determining the performance of an aircraft. Knowledge of the drag polar is, therefore, essential for most ATM research such as trajectory prediction, fuel optimization, and parameter estimation.

Many methods exist to explore the aircraft performance during the preliminary design phase, often with a focus on modeling of the aerodynamics. Hence, one source of open information regarding drag polar models comes from textbooks [5, 112, 104, 138]. However, only older aircraft models and extrapolations of similar designs are available in the literature. In [35], an empirical model for estimating zero-lift drag coefficients was proposed using existing literature data based on several aircraft models. In general, open data on drag polar model is rare, especially for modern commercial aircraft.

The aircraft manufacturers do have accurate aerodynamic data that can be used to construct the drag polar model. However, these data are generally not publicly available due to commercial considerations. Currently, the most comprehensive collection of drag polar data is the Base of Aircraft Data (BADA). It contains the drag polar models for most aircraft types. BADA is the default “go-to” aircraft performance model for current ATM studies. However, the strict license makes it impossible to share and redistribute the BADA model data. Therefore, there is a need for a set of drag polar models that can be shared openly.

The goal of this chapter is to propose an alternative path for estimating the

drag polar models of modern commercial aircraft without relying on proprietary manufacturer data. We approach this estimation problem using a novel stochastic total energy (STE) model. The STE approach treats the parameters of the standard total energy model as random variables. Then, we try to solve the parameter estimation as a minimization problem using Bayesian computing, specifically, the Markov Chain Monte Carlo (MCMC) approximations. The application of this Bayesian data approach is slowly being brought to the engineering domain. In [99], we can see the application of the Bayesian hierarchical model and the use of the MCMC method for inverse problems in the aerospace domain. The STE model and solutions proposed in this chapter are closely related to this line of research. Finally, using this method, a database of drag polar models for common airliners are produced and made publicly available.

The structure of this chapter is as follows. In section two, the fundamentals of the point-mass drag polar model are introduced. In section three, we focus on the hierarchical stochastic model approach and the solution using MCMC. In section four, experiments are conducted to examine and obtain drag polar data of multiple aircraft types. Section five lists the results of drag polar parameters for all available aircraft. Finally, section six and seven are dedicated to the discussions and conclusions of this research.

7.2. Theory of point-mass aerodynamic model

7.2.1. Drag polar in point-mass models

When an aircraft flies, resulting aerodynamic force can be presented as two orthogonal components, which are lift and drag. The drag is produced by the air-flow interacting with the aircraft body. The lift is produced due to the pressure difference between the upper and lower surface of the lifting devices (wings). With the same airspeed and altitude conditions, control of lift is performed by re-configuring the aircraft angle of attack and/or modifying the surface shape of lifting devices. By changing the elevator settings, the pitch angle and the angle of attack can be controlled. On the other hand, a change of the lifting device surface is primarily performed by re-configuring flaps.

In general, the lift and drag forces of an aircraft that is traveling in the free stream can be computed as:

$$\begin{aligned} L &= C_L qS \\ D &= C_D qS \\ q &= \frac{1}{2}\rho V^2 \end{aligned} \tag{7.1}$$

where C_L and C_D are lift and drag coefficients respectively. q is the dynamic pressure. ρ , V , and S are air density, true airspeed, and the lifting surface area of the aircraft. In practice, C_L and C_D can be modeled as functions of the angle of attack (α), Mach number (M), and flap deflection (δ_f):

$$\begin{aligned} C_L &= f_{cl}(\alpha, M, \delta_f) \\ C_D &= f_{cd}(\alpha, M, \delta_f) \end{aligned} \quad (7.2)$$

In aerodynamic models, multi-dimensional table interpolation or higher order polynomials are often used to express these functions. However, in many ATM studies, a point-mass model is only required, where the angle of attack is not modeled.

In the point-mass performance model, the relation between the aerodynamic coefficients C_D and C_L is simplified to the drag polar. It is commonly represented using a quadratic function as:

$$C_D = C_{D0} + kC_L^2 \quad (7.3)$$

where C_{D0} is the *zero-lift drag coefficient* and k is the *lift-induced drag coefficient factor*. In the following sections, our efforts are focused on modeling and estimating these two parameters.

7.2.2. Analytical model for lift dependent drag coefficient

Studies [104] and [100] describe the effects of increasing vortex drag and skin friction. It is also common to describe the lift-induced coefficient k using the Oswald efficiency factor e ($e \leq 1$). It is a correction factor reflects the deviation from an ideal elliptical lift distribution ($e = 1$). The relationship between k and e is:

$$k = \frac{1}{\pi A e} \quad (7.4)$$

where A is the aspect ratio of the wing. e is the Oswald factor that can be model theoretically as follows:

$$e = \frac{1}{Q + P\pi A} \quad (7.5)$$

where Q and P represent the inviscid and viscous part of the induced drag coefficient. There are many ways to approximate the P and Q , which are summarized by [100].

In this chapter, we have adopted the model proposed by [78] to calculate the Oswald efficiency factor (e). In this model, Q is approximated as follows:

$$Q = \frac{1}{0.99 \left[1 - 2(d_F/b)^2 \right]} \quad (7.6)$$

where d_F/b is the fuselage diameter to wingspan ratio. P is considered to be dependent on the zero-lift drag coefficient as:

$$P = 0.38C_{D0} \quad (7.7)$$

where the coefficient 0.38 is obtained empirically based on several existing aircraft types. Combining previous equations, we can produce the closed-form model for lift-induced drag coefficient:

$$k = \frac{1}{0.99 \left[1 - 2(d_F/b)^2 \right] \pi A} + 0.38C_{D0} \quad (7.8)$$

and the Oswald efficiency factor:

$$e = \left(\frac{1}{0.99 \left[1 - 2(d_F/b)^2 \right] \pi A} + 0.38C_{D0} \right)^{-1} \quad (7.9)$$

7.2.3. Different aircraft aerodynamic configurations

In addition to the angle of attack, which affects the values of the lift and drag coefficients, the change in the shape of the aircraft can alter these values. The most notable change in aircraft involves flaps (and slats), speed brakes (or lift dumpers), and landing gear. Each structural setting also has its own corresponding drag polar model. In this chapter, we focus on the additional drag caused by flaps and landing gear.

Flaps

Flaps are common aircraft surfaces deployed in order to provide an increase in the maximum lift coefficient. They are deployed to allow aircraft to fly at lower speeds with a higher lift, typically at low altitudes (for example, during takeoff, initial climb, and approach). Different aircraft types have different configurations of flaps and flap settings. In general, an increase in flap angle leads to an increase in the lift coefficient under the same angle of attack, at the expense of higher drag. Slats are similar to flaps but located on the leading edge of the wing. They increase the maximum lift coefficient by increasing the stall angle of attack. Slats are automatically extended when selecting a flap setting and are considered as part of this configuration. Different flap designs have been adopted by aircraft manufacturers. In Table 7.1, a list of common flap options on airfoils and their approximated maximum lift coefficients are listed. These values are produced by [84]. It is worth noting that the $C_{L,max}$ values of an airfoil are larger than the values of the aircraft with the same shaped wing, especially for swept wings [5, p.263].

Based on the data from the literature [138, p.253], the increase in lift coefficient due to flap deflection can be calculated empirically. This accounts for the effect of the change in wing area, as well as the changes in shape and camber. The deflection angle of flaps (δ) and increased maximum lift coefficient $C_{L,max}$ are shown in Table 7.2, where IC and FA represent the initial climb and final approach configuration respectively.

Table 7.1: Example flap settings in airfoils [84, p.107]









Flap types	$C_{L,max}$	Illustration
airfoil only	1.4	
leading-edge slat	2.4	
plain flap	2.5	
split flap	2.6	
Fowler single-slotted flap	2.9	
Fowler multi-slotted flap	3.0	
with leading-edge slat	3.3	
with boundary layer suction	3.9	

Table 7.2: Increase of lift coefficient due to flaps with typical flap deflection angles

Trailing	Leading	δ_{IC}	δ_{FA}	$C_{L,max,IC}$	$C_{L,max,FA}$
plain flap		20°	60°	1.60	2.00
split flap		20°	40°	1.70	2.20
single-slotted flap		15°	40°	2.20	2.90
double-slotted flap		20°	50°	1.95	2.70
double-slotted flap	w/ slat	20°	50°	2.60	3.20
triple-slotted flap	w/ slat	20°	40°	2.70	3.50

With flaps deflected, the drag of the wing also increases. Using an empirical model [90, p.109], this increase in drag coefficient due to flaps deflection can be computed as:

$$\Delta C_{D,f} = \lambda_f \left(\frac{c_f}{c} \right)^{1.38} \left(\frac{S_f}{S} \right) \sin^2 \delta_f \quad (7.10)$$

where c_f/c and S_f/S are flap to wing chord ratio and flap to wing surface ratio respectively. δ_f is the flap deflection angle. When exact c_f/c and S_f/S are not available, they are both assumed to be 0.15, which is an empirical approximation based on existing aircraft data. λ_f is dependent on the type of flaps. According to [90], λ_f is set to be 1.7 for plain and split flaps, and 0.9 for slotted flaps.

The deflection of flaps also affects the Oswald efficiency factor (e). Based on data from several existing aircraft, a linear relationship is found by [104]. The model to calculate such increase in e is:

$$\begin{aligned} \Delta e_f &= 0.0046 \delta_f && \text{(rear-mounted engines)} \\ \Delta e_f &= 0.0026 \delta_f && \text{(wing-mounted engines)} \end{aligned} \quad (7.11)$$

Landing gears

The landing gear adds a significant amount of drag to the aircraft when it is extended. Hence, the landing gear is retracted as soon as the aircraft becomes airborne and only extended shortly before landing. There are limited studies that quantify the drag coefficient of aircraft landing gears. In this chapter, we adopted the model proposed by [86] to calculate the increased drag coefficient by landing gears:

$$\Delta C_{D,g} = \frac{W}{S} K_{uc} m_{\max}^{-0.215} \quad (7.12)$$

where W/S is the wing loading, m_{\max} refers to the maximum mass of an airplane, and K_{uc} is a factor that relates to the flap deflection angle. In principle, the value of K_{uc} is lower when more flap deflection is applied. This is because the flow velocity along the bottom of the wing decreases when flaps are deployed, which leads to a lower drag on the landing gear. For simplification, it is taken as 3.16×10^{-5} in this chapter based on data from [86].

7.2.4. Wave drag at high Mach numbers

When aircraft fly at a high Mach number, the compressibility of the air flow also needs to be considered. Under such conditions, the drag can increase drastically due to local supersonic flows and the shock wave. Modern airliners often cruise at transonic speeds, where the additional wave drag should not be ignored when calculating the total drag.

In order to model the wave drag, we first need to estimate the critical Mach number (M_{crit}), which is defined as the lowest Mach number at which the airflow over any part of the aircraft reaches the speed of sound. The critical Mach number can often be related to the drag divergence Mach number, which is the Mach number where the drag starts to increase rapidly with increased aircraft speed.

According to [88], the drag divergence Mach number can be approximated as:

$$M_{dd} = \frac{\kappa_A}{\cos \Lambda} - \frac{t/c}{\cos^2 \Lambda} - \frac{C_L}{10 \cos^3 \Lambda} \quad (7.13)$$

where Λ is the mid-chord sweep angle. t/c is the thickness ratio. κ_A is the Korn factor, which is set to be 0.95 for supercritical wing profiles and 0.87 for conventional wing profiles. C_L is the maximum lift coefficient available, which is chosen empirically to be around 1.3 in our chapter. When t/c of an aircraft type is not available, a default common value of 0.11 is used, according to the empirical data from [104].

Knowing the drag divergence Mach number, the critical Mach number can be calculated as follows, according to [47]:

$$M_{crit} = M_{dd} - \sqrt[3]{\frac{0.1}{80}} \quad (7.14)$$

The previous equation indicates that the critical Mach number is always lower

than the drag divergence Mach number by a constant of approximately $\sqrt[3]{0.1/80}$ (approximately 0.1077). Then, combining Equation 7.14 and Equation 7.13, the final close-form critical Mach number model is found:

$$M_{\text{crit}} = \frac{\kappa_A}{\cos \Lambda} - \frac{t/c}{\cos^2 \Lambda} - \frac{C_L}{10 \cos^3 \Lambda} - \sqrt[3]{\frac{0.1}{80}} \quad (7.15)$$

Finally, according to [68] and [47], the rise of the drag coefficient due to the wave drag is calculated as:

$$\Delta C_{D,w} = \begin{cases} 0 & M \leq M_{\text{crit}} \\ 20 (M - M_{\text{crit}})^4 & M > M_{\text{crit}} \end{cases} \quad (7.16)$$

7.2.5. Summary

Considering different structural variations such as the flap deflection and the landing gear, as well as the effect of the wave drag beyond critical Mach number, the dynamic zero-lift drag coefficient C_{D0}^* is modeled as:

$$C_{D0,\text{total}} = C_{D0} + \Delta C_{D,f} + \Delta C_{D,g} + \Delta C_{D,w} \quad (7.17)$$

Similarly, the dynamic Oswald factor considering the flap deflections is modeled as:

$$e_{\text{total}} = e + \Delta e_f \quad (7.18)$$

while the lift-induced drag coefficient becomes:

$$k_{\text{total}} = \frac{1}{1/k + \pi A \Delta e_f} \quad (7.19)$$

We can see that $\Delta C_{D,f}$, $\Delta C_{D,g}$, $\Delta C_{D,w}$, and Δe_f can all be related to the aircraft design, mostly the parameters of wings. The remaining challenge is to estimate C_{D0} , e , and k under the clean configuration at low Mach number flight condition.

7.3. Estimation of drag polar under clean configuration

In this section, we first explore a novel hierarchical model that describes the total energy model in a stochastic fashion, where the model parameters are considered as random variables. Using Bayesian computing, we then try to infer the drag polar based on open flight data from ADS-B and Enhanced Mode-S surveillance communications. The clearest benefit of this stochastic total energy (STE) model is that the process can be applied to any aircraft, as long as accurate flight data and the basic performance parameters of the aircraft are known. The following section describes the process in detail.

7.3.1. The hierarchical stochastic model

Commonly, in the total energy model, the change of energy is described by multiplying each force with speeds in the same direction. This results in the following equation:

$$(T_t - D_t) V_t = m_t a_t V_t + m_t g V S_t \quad (7.20)$$

where T and D are the thrust and drag of the aircraft and m is the aircraft mass. Parameters a , V , and VS are the acceleration, airspeed, and vertical speed respectively. These three variables can be derived from aircraft surveillance data. Subscript t indicates the data is a time series. In general, thrust can be modeled as a function of velocity and altitude. In this research, we use the model proposed by [8] and explained in Chapter 7. Let f_{thr} represent the function that gives the maximum thrust, and the net thrust of the engines can be expressed as:

$$T_t = \delta_T f_{thr}(V_t, h_t, V S_t) \quad (7.21)$$

where f_{thr} is the maximum thrust profile and δ_T represent the thrust setting. Combining the previous two equations, the drag coefficient is calculated as:

$$C_{D,t} = \frac{D_t}{q_t S} \quad (7.22)$$

$$= \frac{\delta_T f_{thr}(V_t, h_t, V S_t) - m_t a_t - m_t g V S_t / V_t}{q S} \quad (7.23)$$

On the other hand, it is also possible to derive the drag coefficient using the drag polar equation. First, from the equilibrium of forces in the direction perpendicular to the airspeed, assuming no banking angle and a small flight path angle, we find the relation between the lift coefficient and the mass to be:

$$C_{L,t} = \frac{L_t}{q_t S} = \frac{m_t g}{q_t S} \quad (7.24)$$

Then, combining Equation 7.8, where the lift-induced coefficient k is defined as a function of aspect ratio (A), fuselage diameter to wing span ratio (d_F/b), and the zero-lift drag coefficient (C_{D0}), the drag coefficient becomes:

$$\begin{aligned} C_{D,t}^* &= C_{D0} + k C_{L,t}^2 \\ &= C_{D0} + k \left(\frac{m_t g}{q_t S} \right)^2 \\ &= C_{D0} + \left(\frac{1}{0.99 [1 - 2(d_F/b)^2]} \pi A + 0.38 C_{D0} \right) \left(\frac{m_t g}{q_t S} \right)^2 \end{aligned} \quad (7.25)$$

Assuming a perfect system and perfect observations, the two drag coefficients (C_D and C_D^*), obtained in two different ways should be the same at each time step, with the following relationship:

$$\Delta C_{D,t} = C_{D,t} - C_{D,t}^* = 0 \quad (7.26)$$

Although noise is inevitably present, Equation 7.26 could be the general assumption of the estimator. In other words, we would consider minimizing $|\Delta C_{D,t}|$ in order to estimate the unknown parameters.

At this point, there are three sets of unknown parameters, which are C_{D0} , m_t , and $\delta_{T,t}$. In order to have an accurate estimation of C_{D0} , we need to know more about m_t and $\delta_{T,t}$. Unfortunately, this information is not available directly from the surveillance data. Even though we do not know the exact mass and thrust setting of each flight, there are some hypotheses that can be assumed:

1. C_{D0} and k are constant and the same for all flights of the same aircraft model, under clean configuration, in low-speed incompressible airflow.
2. Based on aerodynamic theory, it is possible to know the ranges of C_{D0} and k values.
3. The range of aircraft mass and thrust setting values can be found empirically.
4. Accurate surveillance data for a sufficient number of flights can be obtained, including trajectories, velocity, temperature, and wind conditions.

In the proposed STE model, we consider all parameters as random variables. The observable parameters are defined as follows:

$$\begin{aligned} V_t &\sim \mathcal{N}(\tilde{V}_t, \sigma_v^2) \\ a_t &\sim \mathcal{N}(\tilde{a}_t, \sigma_a^2) \\ VS_t &\sim \mathcal{N}(\tilde{V}S_t, \sigma_{vs}^2) \\ h_t &\sim \mathcal{N}(\tilde{h}_t, \sigma_h^2) \end{aligned} \quad (7.27)$$

where each parameter is assumed to be drawn from a normal distribution. \tilde{V}_t , \tilde{a}_t , $\tilde{V}S_t$, and \tilde{h}_t are the observed values at each time step respectively. σ_v^2 , σ_a^2 , σ_{vs}^2 , and σ_h^2 are the variances for each variable. The models for the three sets of unknown system parameters can be constructed similarly:

$$\begin{aligned} \delta_{T,t} &\sim \mathcal{U}(\delta_{T_{min}}, \delta_{T_{max}}) \\ m_t &\sim \mathcal{U}(m_{min}, m_{max}) \\ C_{D0} &\sim \mathcal{U}(C_{D0_{min}}, C_{D0_{max}}) \end{aligned} \quad (7.28)$$

where $\delta_{T,t}$, m_t , and C_{D0} are defined with uniform probability density functions. Random variables $\delta_{T,t}$ and m_t are defined as thrust setting and mass at time t , where different random variables are used at different time step. Finally, ΔC_D from Equation 7.26, can be expressed as:

$$\Delta C_D \sim \mathcal{N}(0, \sigma_\Delta^2) \tag{7.29}$$

With these variables defined, we have converted the deterministic total energy model into a hierarchical stochastic model. This STE model considers each parameter as a random variable and preserves the underlying structure among the observable parameters (a, V, VS, τ, h), forces (T and D), and system parameters (m, C_{D0} , and k), as defined in total energy model. In Figure 7.1, the dependencies among all those parameters are visualized. This hierarchical model includes ten random variables, which are indicated in gray.

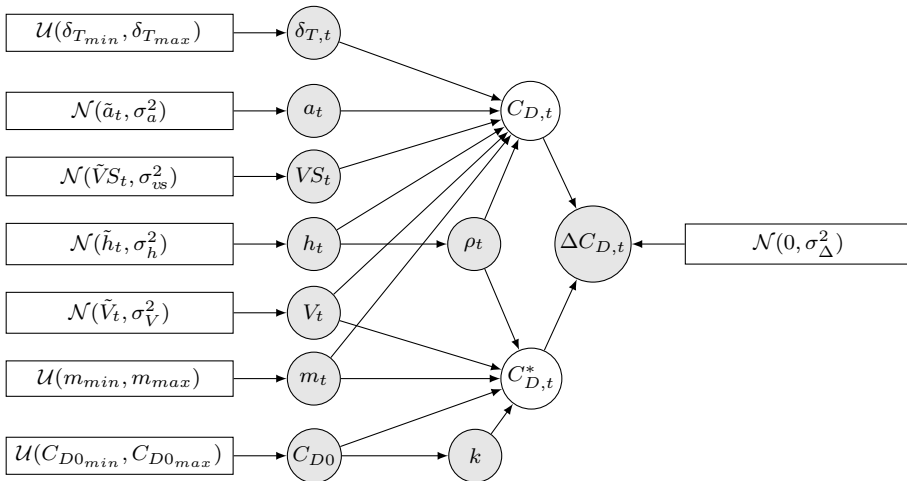


Figure 7.1: Hierarchical relationships of model parameters

Parameter values of the prior distributions defined in Equation 7.27 and 7.28 (also in Figure 7.1) are listed in Table 7.3. Variance for speed, vertical rate, and altitude are selected based on uncertainties defined by the ADS-B navigation accuracy category [111]. Mass is chosen to be uniformly distributed between the minimum operational weight and maximum takeoff weight. Other prior parameters are chosen empirically.

Table 7.3: STE model hierarchical parameters

Parameter	σ_v	σ_a	σ_{vs}	σ_h	$\delta_{T_{min}}$	$\delta_{T_{max}}$	m_{min}	m_{max}	$C_{D0_{min}}$	$C_{D0_{max}}$
Value	5	0.2	7.62	22.5	0.85	0.15	m_{OEWS}	m_{MTOW}	0	0.05
Unit	m/s	m/s ²	m/s	m	-	-	kg	kg	-	-

7.3.2. Bayesian computing - Markov Chain Monte Carlo

Instead of estimating the drag polar (C_{D0} and k) directly, we have converted the original estimation to a Bayesian optimization problem. Based on the prior probability of all model parameters, this approach allows us to compute the posterior probabilities based on the constraints in Equation 7.26. To simplify the expression, let us define θ and \mathbf{y} as follows:

$$\begin{aligned}\theta &= \left\{ C_{D0}, k, [\delta_{T,t}], [m_t], [V_t], [a_t], [VS_t], [h_t] \right\} \\ \mathbf{y} &= \left\{ [\Delta C_{D,t}] \right\}\end{aligned}\quad (7.30)$$

where parameters in brackets are with multiple dimensions. The goal of the estimation is to compute the joint posterior probability $p(\theta|\mathbf{y})$, which is difficult (and impractical) to obtain analytically due to the number of state variables. Instead, we use a Markov Chain Monte Carlo (MCMC) simulation to approach this numerically.

The MCMC method uses sequential sampling, drawing a large number of values of θ from approximate distributions and then correcting these draws based on the distribution constructed from previous values drawn. The final estimates can be obtained once the Markov chain converges to a unique stationary distribution.

The simplest form of the MCMC sampling method is the Metropolis algorithm [92], which can be applied to almost any Bayesian computation problem. The algorithm can be explained in simple steps, which also demonstrate the principles of MCMC sampling. These steps are as follows:

1. Choose a initial set of θ as θ^0 .
2. For iteration $i = 1, 2, \dots, N$:
 - (a) Sample a proposal θ^* from a *jumping distribution* $J(\theta^*|\theta^{t-1})$, which is a conditional probability dependent on the current set of values. The jumping distribution is usually chosen as a normal distribution with θ^{t-1} as the mean.
 - (b) Calculate the ratio using the Bayes' rule:

$$r = \frac{P(\theta^*|\mathbf{y})}{P(\theta^{i-1}|\mathbf{y})} = \frac{P(\theta^*)P(\mathbf{y}|\theta^*)}{P(\theta^{i-1})P(\mathbf{y}|\theta^{i-1})}\quad (7.31)$$

where $P(\theta^*)$ is defined by the prior distributions, and the likelihood $P(\theta^{i-1}|\mathbf{y})$ is computed based on the hierarchical model.

- (c) set θ^i as θ^* with a probability of $\min(r, 1)$, otherwise keep as θ^{i-1} .

When the MCMC chain reaches the desired number of iterations, the sampling process can be stopped. The values from each iteration stored in the Markov Chain are used to construct the posterior distributions of the model parameters.

The Metropolis sampler is simple to implement and can be applied to both continuous and discrete probability distributions. The drawback is that it can lead to a very long convergence runtime. This is due to the random-walk behavior. In this chapter, a new sampler called No-U-Turn Sampler (NUTS) [57] is used. The NUTS sampler is an improved Hamiltonian Monte Carlo which avoids the random walk behavior. It can automatically determine some of the hyper-parameters in the Hamiltonian Monte Carlo sampling and allows the chains to converge quickly with fewer sampling iterations.

7.3.3. The posterior estimates

Using NUTS sampler, the convergence of the Markov chains can usually be achieved within 1000 iterations. In our experiments, we have increased the number of iterations to 3000 to have a larger number of stable samples. Once the sufficient number of sampling iterations has been reached, it is possible to compute the posterior probability density function using the numerical approximation based on these samples. An example can be seen in Figure 7.4 from the later experiment section.

The mean of the samples representing C_{D0} and k are considered as the final estimates for drag polar parameters. The standard deviation for samples representing C_{D0} is also calculated to determine whether the chain has converged at the boundary conditions:

$$\hat{C}_{D0} - 2\hat{\sigma}_{C_{D0}} > C_{D0_{min}} \quad (7.32)$$

$$\hat{C}_{D0} + 2\hat{\sigma}_{C_{D0}} < C_{D0_{max}} \quad (7.33)$$

where \hat{C}_{D0} and $\hat{\sigma}_{C_{D0}}$ are mean and standard deviation from the posterior distribution. Since there are a large number of trajectories for estimating the drag polar, estimations from trajectories that do not satisfy these conditions are considered as invalid.

7.4. Experiment

The experiment consists of several parts. We start with an example of drag polar estimation based on a single trajectory. Then, the results from multiple trajectories and aircraft types are summarized. We also include comparison analysis between our result and drag polar obtained from CFD simulations.

A dataset consisting of around 100 climbing flights per aircraft type is gathered. All data are collected using the Mode-S receiver at the TU Delft. The flights observed were taking off from Amsterdam Schiphol airport during March, 2018. In Figure 7.2, the ground tracks and vertical profiles of a selected subset of flights are shown.

In addition to the ground speed obtained from ADS-B, real-time wind data are computed using the Meteo-Particle model proposed by [134]. Combining ground speed and accurate wind information, the airspeed of the aircraft is computed. The

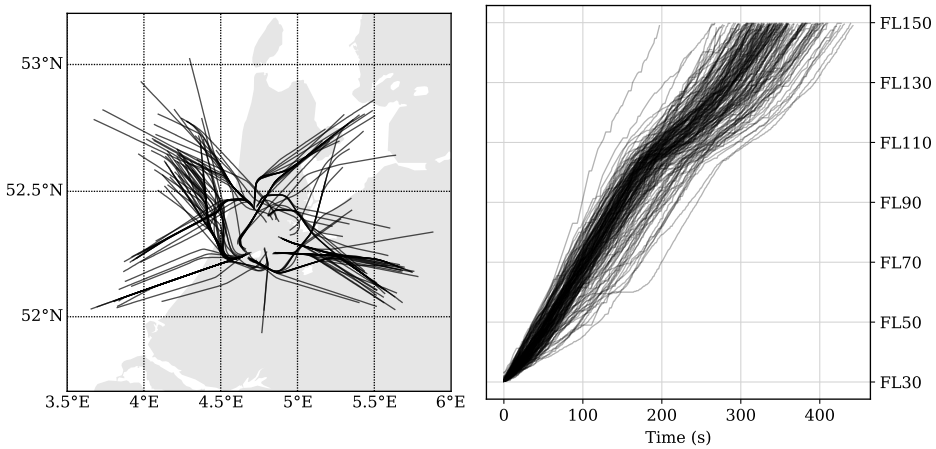


Figure 7.2: The trajectories of the Boeing 747-400 flights (during March, 2018, departing from Schiphol airport)

use of true airspeed instead of ground speed allows us to describe the performance more accurately.

For each flight, we apply MCMC sampling with the STE model to obtain the posterior probability distribution of the parameters C_{D0} and k . Using the NUTS MCMC implemented in the PyMC3 library [114], we can conveniently perform the sampling of our model with predefined prior distributions. To ensure consistent convergence, we utilize four independent chains, which are sampled in parallel.

7.4.1. Estimation of the drag polar using STE model

First, the results of one example flight are shown. In Figure 7.3, the trace of four chains (on the right-hand side) and the estimated posterior density (on the left-hand side) are shown. Each chain is marked with a different shade of gray. The stable convergence can be observed by comparing the posterior density of all four chains, which is shown in Figure 7.4. Combining all chains, we can obtain the mean value C_{D0} and k from the posterior distribution.

7.4.2. Multiple aircraft types

Next, we extend the experiment from one flight example to all flights gathered in our dataset mentioned earlier, which contains flights for 20 common aircraft types. Based on these flight, we are able to obtain C_{D0} and k for each aircraft type by combining the results of all flights. In Figure 7.5, the distributions of C_{D0} and k are shown.

Most of the C_{D0} values are between 0.02 and 0.04, while the values for k fall mostly between 0.035 and 0.055. In the later result section, all these values are listed in detail.

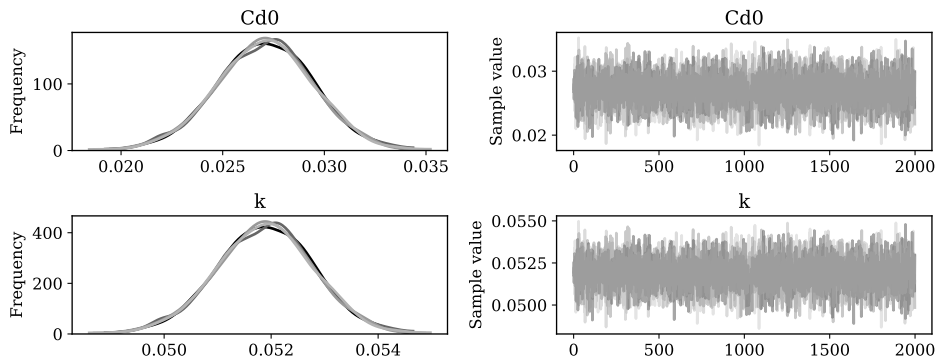
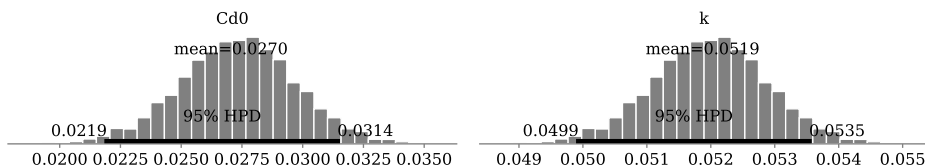


Figure 7.3: MCMC sampling of a Boeing 747-400 flight

Figure 7.4: Posterior distribution of C_{D0} and k from MCMC sampling

7.4.3. A drag polar using CFD simulation

In this experiment, we conduct some basic CFD calculations in order to study the drag polar results derived from the proposed MCMC methods. The details of the CFD simulation can be found in Appendix B. Three modified aircraft 3D models are constructed. They are 1) bare configuration without engines, 2) configuration with engines, but without fans, and 3) configuration with full engines. In Figure 7.6, these three different configurations are shown. Shades of gray indicate different levels of air pressure in CFD simulation.

With these engine configurations, CFD calculations are run under the different angle of attack, ranging from -5 degree to 30 degrees, with speed at approximately 0.3 Mach. For each scenario, obtained drag and lift coefficients (C_D and C_L) are computed. In Figure 7.7, results from all rounds of the simulations are shown.

If we correlate them via the angle of attack, we are able to plot the relationship between the drag and lift coefficient, which is shown in Figure 7.8. On the left-hand side, high order splines are used to fit the data points. We can see that the effect of additional drag caused by engines is significant. This is because the engines are considered as a block object in these CFD simulations. The drag produced by the engine is significantly larger than in actual operating situations.

On the right-hand side of Figure 7.8, quadratic drag polars are fitted to these data points with the angle of attack at less than 15 degrees. The drag polar obtained from the previous STE-MCMC approach is also shown in solid black lines. We can see that the actual drag polar computed based on flight data lies between the results of the full engine scenario and the fans-off scenario in the CFD the sim-

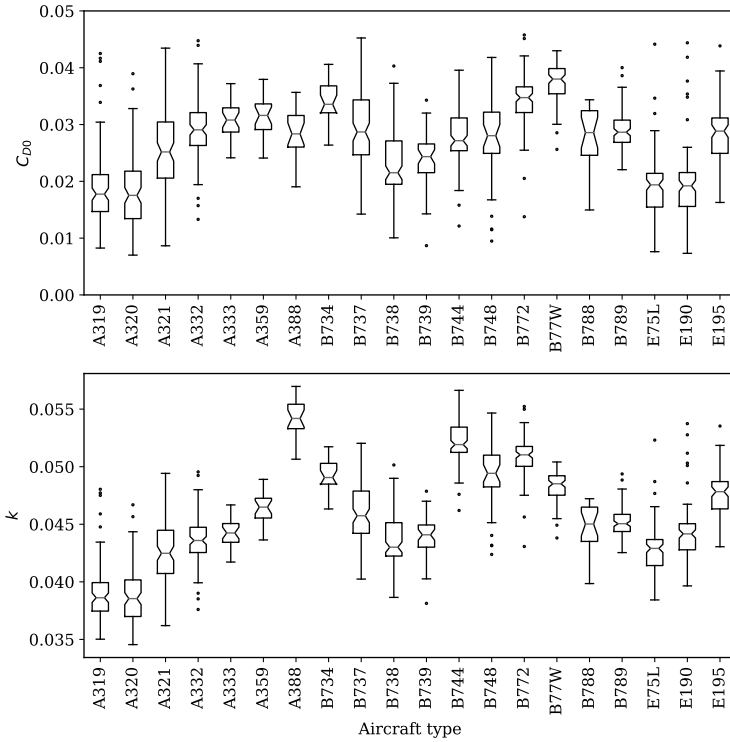


Figure 7.5: Drag polar of common aircraft models based on multiple flights

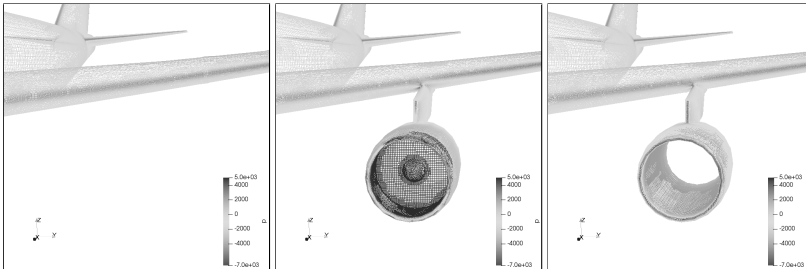


Figure 7.6: Three engine configurations in CFD calculations

ulations, which is a good confirmation for results obtained based on STE model.

7.4.4. Compressibility effect on drag polar

When an aircraft is flying at a high speed where the Mach number is higher than the estimated critical Mach number, the compressibility effect should be taken into account. According to the empirical models given by Equation 7.14, the increase of wave drag can be computed, which is dependent on the actual Mach number during the flight. In Figure 7.9, drag polar models of Boeing 747-400 aircraft

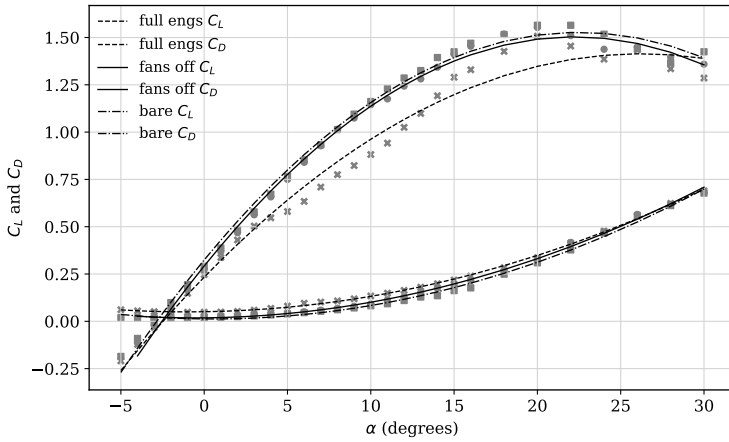


Figure 7.7: Relationships between angle of attack and lift/drag coefficients

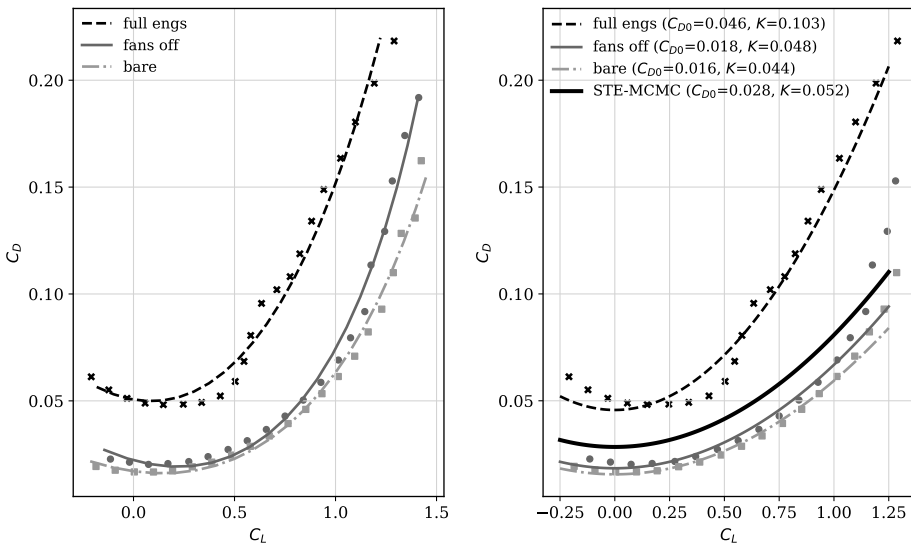


Figure 7.8: Drag polars under different conditions

under different Mach numbers are shown as an example.

7.5. Summary and discussion

7.5.1. Drag polar for different aircraft types

In Table 7.4, the drag polar of the 20 most common airliners are listed. In this table, the zero-lift drag coefficient (C_{D0}), lift-induced drag coefficient factor (k), and Oswald efficiency factor (e) are all estimated using the MCMC and STE model.

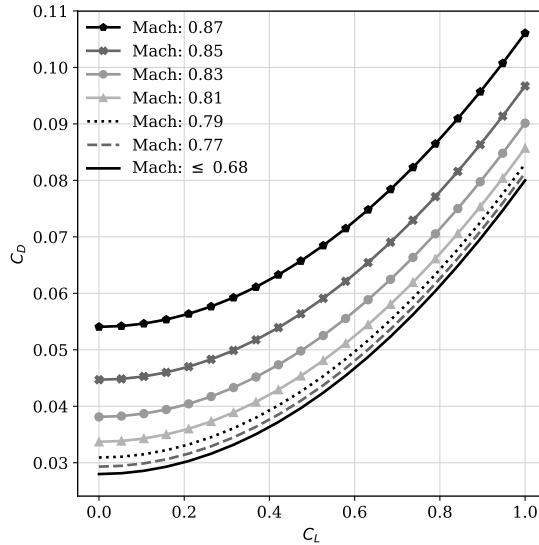


Figure 7.9: Changes in the drag polar due to compressibility

Table 7.4: Drag polar and related coefficients

Aircraft	\bar{C}_{D0}	k	e	M_{crit}	λ_f	c_f/c	S_f/S	$\Delta C_{D,g}$
A319	0.019	0.039	0.793	0.63	0.90	0.18	0.17	0.017
A320	0.018	0.039	0.798	0.63	0.90	0.18	0.17	0.017
A321	0.026	0.043	0.746	0.63	0.90	0.18	0.16	0.019
A332	0.029	0.044	0.728	0.64	0.90	0.15	0.15	0.014
A333	0.030	0.044	0.719	0.64	0.90	0.15	0.15	0.014
A359	0.031	0.046	0.725	0.65	0.90	0.15	0.15	0.013
A388	0.028	0.054	0.781	0.69	0.90	0.15	0.15	0.012
B734	0.034	0.049	0.705	0.61	0.90	0.15	0.15	0.021
B737	0.029	0.046	0.736	0.63	0.90	0.15	0.15	0.016
B738	0.023	0.044	0.775	0.63	0.90	0.15	0.15	0.017
B739	0.024	0.044	0.769	0.63	0.90	0.15	0.15	0.018
B744	0.028	0.052	0.774	0.68	0.90	0.20	0.15	0.015
B748	0.027	0.049	0.771	0.68	0.90	0.19	0.14	0.015
B772	0.034	0.051	0.723	0.65	0.90	0.17	0.16	0.014
B77W	0.037	0.048	0.687	0.65	0.90	0.16	0.15	0.016
B788	0.027	0.045	0.748	0.67	0.90	0.15	0.15	0.013
B789	0.029	0.045	0.737	0.67	0.90	0.15	0.15	0.014
E75L	0.019	0.043	0.803	0.63	0.90	0.15	0.15	0.017
E190	0.019	0.044	0.813	0.63	0.90	0.15	0.15	0.016
E195	0.028	0.048	0.752	0.63	0.90	0.15	0.15	0.017

For non-clean configuration situations, we can use previously introduced empirical models to incorporate changes of drag polar coefficients due to flaps or landing gears. Combining the coefficients from Table 7.4 and Equation 7.10, the increased drag coefficient at different flap deflection angles can be calculated. Similarly, the change in the Oswald factor can be calculated according to Equation 7.11. The increased values of drag coefficient due to landing gears are also listed

in this table. In addition, the approximated critical Mach number is given in Table 7.4 as well.

To facilitate future calculations, a pre-computed table with a set of drag polars under non-clean configuration with fixed flap deflection angles at the initial climb and landing phases are computed and listed in Table 7.5.

Table 7.5: Drag polar coefficients in non-clean configurations with default flap deflection angles

Aircraft	$\delta_{f,IC}$	$C_{D0,IC}$	k_{IC}	e_{IC}	$\delta_{f,FA}$	$C_{D0,FA}$	k_{FA}	e_{FA}
A319	20°	0.021	0.037	0.845	40°	0.025	0.035	0.897
A320	20°	0.020	0.036	0.850	40°	0.024	0.034	0.902
A321	20°	0.028	0.040	0.798	50°	0.034	0.036	0.876
A332	20°	0.030	0.041	0.780	50°	0.035	0.037	0.858
A333	20°	0.032	0.041	0.771	50°	0.036	0.037	0.849
A359	20°	0.032	0.043	0.777	50°	0.037	0.039	0.855
A388	20°	0.030	0.051	0.833	40°	0.033	0.048	0.885
B734	20°	0.036	0.046	0.757	40°	0.038	0.043	0.809
B737	20°	0.030	0.043	0.788	50°	0.035	0.039	0.866
B738	20°	0.024	0.041	0.827	50°	0.029	0.037	0.905
B739	20°	0.025	0.041	0.821	50°	0.030	0.038	0.899
B744	20°	0.030	0.049	0.826	40°	0.034	0.046	0.878
B748	20°	0.029	0.046	0.823	40°	0.032	0.043	0.875
B772	20°	0.036	0.047	0.775	50°	0.041	0.043	0.853
B77W	20°	0.039	0.045	0.739	50°	0.044	0.041	0.817
B788	20°	0.029	0.042	0.800	40°	0.031	0.039	0.852
B789	20°	0.030	0.042	0.789	40°	0.033	0.040	0.841
E75L	20°	0.020	0.040	0.855	50°	0.025	0.037	0.933
E190	20°	0.020	0.041	0.865	50°	0.025	0.038	0.943
E195	20°	0.029	0.045	0.804	50°	0.034	0.041	0.882

7.5.2. Uncertainties

It is important to understand that, in the proposed hierarchical model, all parameters are considered as random variables (described by probability density functions) instead of scalar values. Most parameters (except C_{D0} and k) are also expressed as time-varying random variables. The time series variables are constructed as multi-dimensional probability density functions. The solution of this hierarchical stochastic model is only possible thanks to numerical approximation using the Markov Chain Monte Carlo (MCMC) techniques.

In the proposed STE aircraft performance model, two major sources of uncertainty exist, which are the aircraft mass and the thrust. We consider the mass as a uniformly distributed random variable at each time step, which takes into account all possible masses. It is sampled together with all other parameters. When evaluating the sampling results, we are able to illustrate the posterior distribution of mass at different time steps, as shown in Figure 7.10. The goal is to estimate the drag polar coefficients C_{D0} and k , not the mass of an aircraft. The posterior distributions can be thought of as the results that optimize all different parameters at all time steps.

The aircraft thrust model is the factor that influences the results the most while using the STE model, as the thrust is strongly correlated to the drag of the aircraft. The uncertainty in thrust can be categorized in two ways. First, aircraft may be

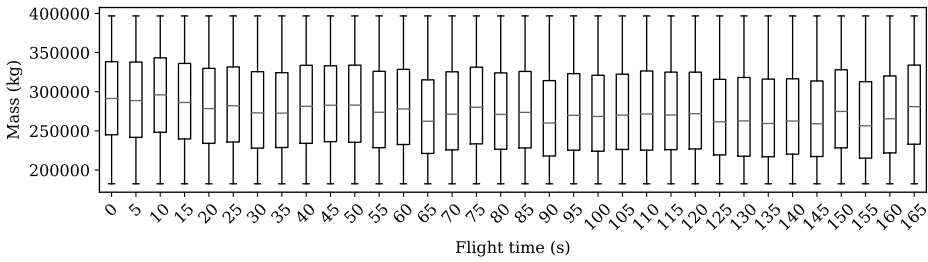


Figure 7.10: The distributions of mass at different flight time steps

equipped with different engine options within the same aircraft type family. This chapter uses a default engine per aircraft type, which can cause a small level of uncertainty. The second cause of uncertainty is related to the thrust model used in this chapter. We implement the thrust model based on two-shaft turbofan engines [8]. According to the analysis of [8], the model has an uncertainty of 4.5% of maximum thrust when compared to engine performance data. This uncertainty in thrust model would undoubtedly affect the final uncertainty of the drag polar estimates. That’s the reason why a uniform probability density function with relative large bounds are considered for the thrust setting (δ_T) in this chapter. In addition, the possible variations in thrust setting due to different procedures are also handled by this parameter.

7

7.5.3. Comparison with BADA drag polar

Due to the restriction of the BADA license, exposure of specific BADA coefficients is not permitted. Instead of showing the exact difference between BADA and our model for each aircraft type, the overall statistics for all available aircraft are shown in Figure 7.11. The mean absolute difference for C_{D0} and k are both found to be around 0.007.

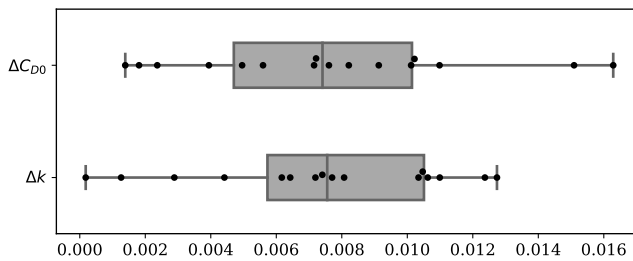


Figure 7.11: Absolute difference of drag polar between BADA (version 3.12) and the model derived in this chapter under clean configuration.

This gives a good indication of what the level of difference in the outcome of this study is, compared to well-established performance models. However, we should be cautious when interpreting this difference as the estimation accuracy.

Since neither the method nor the data used for constructing BADA drag polar are published, it is hard to identify the uncertainties in BADA model. Thus, the difference between the two models should not be directly used to quantify the accuracy of our model.

7.6. Conclusion

In this chapter, we proposed a method for deriving the aircraft drag polar based on the point-mass performance model. A novel stochastic total energy model was developed for estimating drag polar coefficients using open ADS-B and Mode-S surveillance data. Using these methods, the drag polar for aircraft with turbofan engines could be estimated under clean aerodynamic configuration.

By considering the states in the total energy model as random variables, we transformed the estimation to Bayesian domain, where the solution of such a minimization problem can be solved using the Markov Chain Monte Carlo method. Compared to other deterministic minimization methods, our approach not only provided the estimations but also the posterior probability densities for the states to be estimated. Due to the stochastic nature of the approach, the estimates were also less likely to result in local minima.

Based on studies from the literature, we included the models that can be used to compute additional drag caused by flaps, landing gear, and shock wave. Combining all knowledge together, the drag polar model proposed in this chapter described the complete phases of flight from takeoff to landing.

A comprehensive list of drag polar models was produced for the 20 most common airliners. To the best of our knowledge, this was the first time such models have been derived and shared publicly with the research community. The resulting drag polars proposed in this chapter have been implemented in the Open Aircraft Performance Model - OpenAP (see Appendix A), which is also used to power the open-source air traffic simulator BlueSky [54]. With the open-source license, we believe the drag polar coefficients produced by this chapter could contribute to improved transparency, accessibility, and repeatability among future air transportation studies.

Part III

Estimations

8

Turn performance estimation

From this chapter onward, the dissertation is focused on the estimation of performance parameters of individual flights. In this chapter, we will first take a look at the aircraft turn performance. The turning segments of a flight are often overlooked in studies related to air traffic management. Air traffic controllers often rely on track and turn reports from aircraft Mode-S secondary surveillance data to gather performance indicators such as roll angle and track rate. However, this data has a low update rate and is not always available. In this chapter, we construct methods that allow us to extract and analyze aircraft turn performance from ADS-B data. The chapter first discusses the fundamentals of the dynamics of turning aircraft. Based on ADS-B trajectory data, we are able to estimate the radius of the arc trajectory using the least-squares regression. Knowing the estimated radius of the turn and aircraft speed, we can then derive the bank angle and turn rate of the aircraft. The estimation results are validated with the aircraft track and turn reports whenever they are available. The median errors for bank angle and turn rate are found to be less than 2 degrees and 0.1 degrees/s respectively.

8.1. Introduction

Between two straight flight paths with different track angles, an aircraft needs to make a turning maneuver in order to change its heading. Earlier studies related to turning performance often focused on optimal control [23] or on the optimization of turns [11]. Unlike other segments of accelerated flight (e.g. takeoff and landing), turn performance is often overlooked in air transportation studies.

Aside from the short temporal scale of the turns, the simplified point-mass performance model used in most air transportation studies also accounts for the lack of research on this topic. The point-mass aircraft model often ignores the rotations of the aircraft along different axes (i.e. pitch, roll, and yaw). Without the roll angle, a minimum turn performance model cannot be established accurately. At the same time, knowledge of the roll (or bank) angle is important when analyzing the performance during turns. For instance, to maintain a steady flight, the total lift of the aircraft needs to be increased due to the rolling of the aircraft. The increase of lift will also lead to an increase in the drag of the aircraft and, subsequently, the thrust as well.

From an air traffic controller point of view, knowing the bank angle, in addition to the turn rate and airspeed, would provide better predictability of the trajectory. This is the reason why some of the secondary surveillance radar interrogations in Mode-S communications are designed to obtain such information. When aircraft fly in controlled airspace, the *track and turn reports* are selectively interrogated by the secondary surveillance radars. Messages of the track and turn report are identified with Comm-B data selector 50 (BDS 50) [66]. In our earlier research, we develop the method to identify and decode these messages. With the open-source pyModeS tool (from Chapter 2), these messages can be easily decoded. There are, however, significant drawbacks to using the track and turn reports. First of all, they have a much lower update rate than ADS-B messages. In addition, the data is not always available publicly.

In this chapter, our main goal is to estimate performance parameters related to individual turns using ADS-B data. These parameters include turn radius, roll angle, load factor, and turn rate. When it is available, the BDS 50 data of the aircraft is collected and decoded to validate the results.

The remainder of the chapter is designed as follows. In section two, we first address the fundamental principles of aircraft turns. In section three, the method used to extract turns from ADS-B flight trajectories is derived. In section four, processes for estimating different turn performance parameters are given. Then, in section five, we show the results of our method based on a large number of flights from different aircraft types. Finally, in section six and seven, discussions and conclusions are provided.

8.2. Coordinated turns

A turning flight is defined as a continuous change of path direction. Commonly, coordinated turns are performed to maintain passenger comfort and the safety of the flight. The maneuver requires the coordination of different control surfaces and thrust. When an aircraft turns, the ailerons are deflected to provide a desired

roll angle of the aircraft. The rudder of the aircraft also needs to be deflected to provide the corresponding yaw of the aircraft. The roll maneuver changes the direction of lift and leads to a decrease in the vertical lift component. Thus, the aircraft needs to increase its pitch angle (by deflecting the elevator) to increase the total lift. The increased lift causes an increase in total drag. Hence, the thrust also needs to be adjusted correspondingly to maintain the same speed.

During a coordinated turn, the aircraft flies at a certain bank angle ϕ and turn rate ω , depending on the velocity and combination of forces (lift L , weight W , and centrifugal force F_c). The kinematic and kinetic components are illustrated in Figure 8.1.

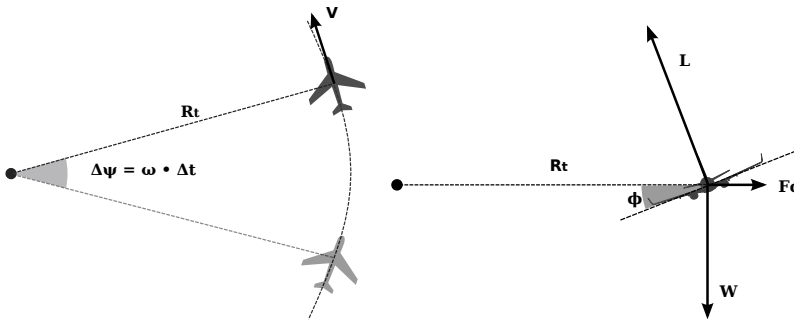


Figure 8.1: Aircraft banked turn

For a coordinated turn where the airspeed and altitude are constant, the relationship of different performance parameters can be described as follows:

$$F_c = L \sin \phi = \frac{W}{g} \frac{V^2}{R_t} \quad (8.1)$$

$$W = L \cos \phi \cos \gamma \quad (8.2)$$

where the V is the airspeed, R_t is the radius of the turn, and γ is the flight path angle. The load factor (denoted as n) is defined as the ratio of lift to the aircraft weight:

$$n = \frac{L}{W} \quad (8.3)$$

It can be calculated separately from Equations 8.1 and 8.2 as:

$$n = \frac{V^2}{g R_t \sin \phi} \quad (8.4)$$

$$n = \frac{1}{\cos \phi \cos \gamma} \quad (8.5)$$

By combining these two equations, the bank angle can be calculated from the aircraft turn radius, airspeed, and flight path angle (during climb or descent) as follows:

$$\phi = \arctan\left(\frac{V^2 \cos \gamma}{g R_t}\right) \quad (8.6)$$

8.3. Turn trajectory extraction

The first challenge we are facing is to extract the turning segments of the flight from the ADS-B trajectory data. To identify the turns, aircraft headings are used as an indicator. An example of a heading profile from one trajectory is illustrated in Figure 8.2. In this example, we can see several heading changes occur during a period of 500 seconds.

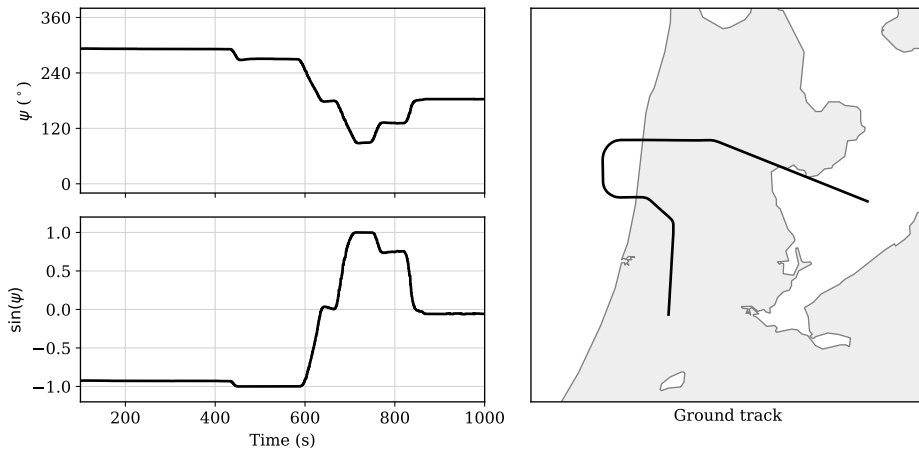


Figure 8.2: An example trajectory with turns (headings and ground track)

Aircraft track angle (χ) can be calculated from the ADS-B velocity reports using the south-north and west-east components of the ground velocity. Assuming zero (or calm) wind, the track angle is approximately the same as the heading (ψ):

$$\psi \approx \chi \quad (8.7)$$

The value of heading (or track) angle ranges from 0° to 360° in relation to the direction of true north. In order to represent the continuity of the heading (avoiding a jump between 0° and 360°), the sine function, $\sin(\psi)$, is used to represent the heading profile. To identify the start and end of a turn, the derivative of the function $\sin(\psi)$ is computed, denoted as $f'(\psi)$:

$$f'(\psi) = \frac{d \sin(\psi)}{dt} \quad (8.8)$$

A threshold is applied to filter out the segments where the changes in the heading are small. Then, the time windows of the turns can be identified individually. This identification process is illustrated in Figure 8.3. The plot on the left shows the $f'(\psi)$ and time windows in which turns occurred, whereas the plot on the right shows the ground projections of the turns.

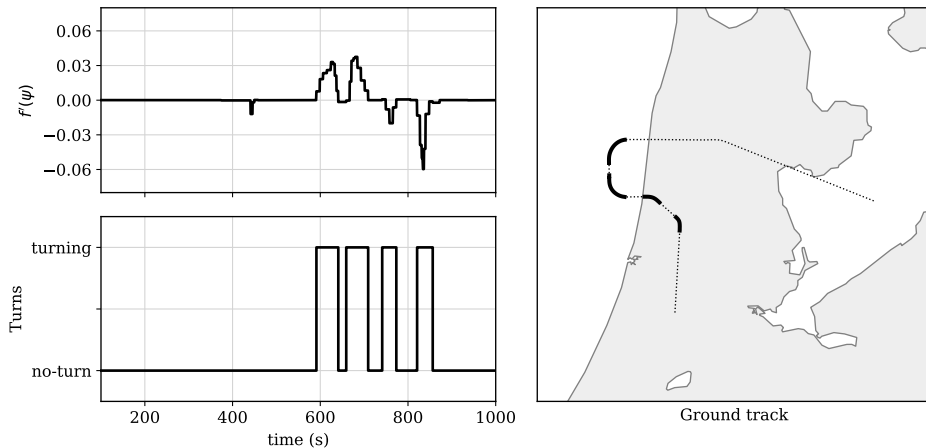


Figure 8.3: Extraction of turns from trajectory data

8.4. Turn performance estimation

Once the turns from the trajectories are extracted, performance parameters such as turning radius, bank angle, and load factors can be estimated individually.

The first step is to calculate the turning radius from the trajectory data. Trajectories consist of latitudes, longitudes, and altitudes. To simplify the calculation, we first convert the trajectories to three-dimensional Cartesian coordinates with the references to the center of the Earth using the spherical Earth model:

$$\begin{aligned} x &= (R_e + h) \cos(\text{lat}) \cos(\text{lon}) \\ y &= (R_e + h) \cos(\text{lat}) \sin(\text{lon}) \\ z &= (R_e + h) \sin(\text{lat}) \end{aligned} \quad (8.9)$$

where the R_e , h , lat , and lon are the radius of the Earth, altitudes, latitudes, and longitudes of the aircraft respectively.

Based on these coordinates, one can find a circle that best describes the trajectory using the least-squares regression. The radius of this circle should correspond to the aircraft turn radius. However, the variance of the least-squares fit of a circle to the three-dimensional points can be large. This sometimes prevents the best circle from being found. We first need to reduce the least-squares problem from three dimensions to two dimensions to help decrease the uncertainties in the regression

method.

In order to reduce the dimension of the positions, a two-dimensional plane in the three-dimensional space first needs to be found, where the squared-sum of the distances from all points is minimized (i.e., the plane that aligns best with the set of positions or the turn). After that, we can use the projection of the points on the plane to find the ideal circle.

We define the plane using any point \vec{p}_0 within the plane and a normal vector \vec{n} to the plane (see Figure 8.4):

$$\vec{p}_0 = [x_0, y_0, z_0]^\top \quad (8.10)$$

$$\vec{n} = [1, \alpha_p, \alpha_a]^\top \quad \text{Spherical} \quad (8.11)$$

$$= [\sin \alpha_p \cos \alpha_a, \sin \alpha_p \sin \alpha_a, \cos \alpha_p]^\top \quad \text{Cartesian} \quad (8.12)$$

where α_p is the polar angle and α_a is the azimuth angle. The first equation of \vec{n} is the spherical representation, while the second equation is the Cartesian representation. For all positions \mathbf{p} , the distances to the plane can be computed as the dot product of the distance between each position to \vec{p}_0 and the normal vector:

$$\mathbf{d} = \vec{n} \cdot (\mathbf{p} - \vec{p}_0) \quad (8.13)$$

The best plane can be found when the squared-sum of all distances to the plane is minimized:

$$\vec{p}_0, \vec{n} = \underset{\vec{p}_0, \vec{n}}{\operatorname{argmin}} \sum_{i=1}^n d_i^2 \quad (8.14)$$

Next, we transform the fitted plane to the reference x,y-plane. This step will transform the radius calculation from a 3D problem to a 2D problem. In order to do so, a transformation matrix needs to be found. The transformation matrix can be calculated using three points from the previously found plane (p_0 , u_0 , and v_0) and x,y-plane (p_1 , u_1 , and v_1), which satisfies the following relation:

$$\vec{u}_0 = \frac{\vec{m} \times \vec{n}}{\|\vec{m} \times \vec{n}\|} \quad (8.15)$$

$$\vec{v}_0 = \vec{u}_0 \times \vec{n} \quad (8.16)$$

$$\vec{p}_1 = [0, 0, 0]^\top \quad (8.17)$$

$$\vec{u}_1 = [1, 0, 0]^\top \quad (8.18)$$

$$\vec{v}_1 = [0, 1, 0]^\top \quad (8.19)$$

$$(8.20)$$

where \vec{m} can be any vector that is not parallel to \vec{n} . Then, the transformation (A) can be computed as:

$$\mathbf{A} = [\vec{p}_1, \vec{u}_1, \vec{v}_1] [\vec{p}_0, \vec{p}_0 + \vec{u}_0, \vec{p}_0 + \vec{v}_0]^{-1} \tag{8.21}$$

To convert all positions \mathbf{p} to x,y-plane points \mathbf{p}_{xy} , the following equations are applied:

$$\mathbf{p}_{xy} = \mathbf{A}\mathbf{p} \tag{8.22}$$

In Figure 8.4, such a transformation is illustrated.

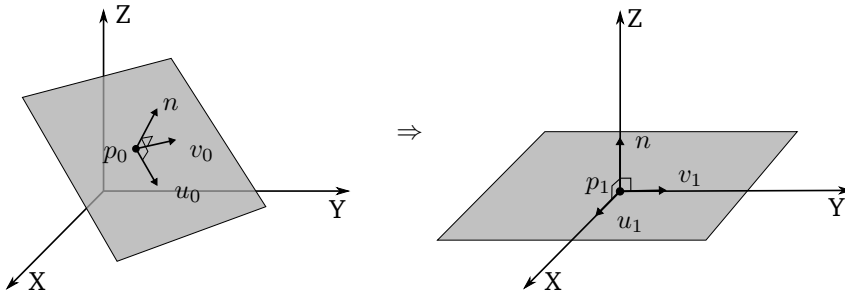


Figure 8.4: The rotation of the trajectory plane from 3D to 2D

It is worth noting that after the transformation, the points \mathbf{p}_{xy} maintain the same geometrical distances as the original three-dimensional point, except the small distances from the original points to the fitted two-dimensional plane are ignored. As an illustration, one of the trajectory transformations is shown in Figure 8.5.

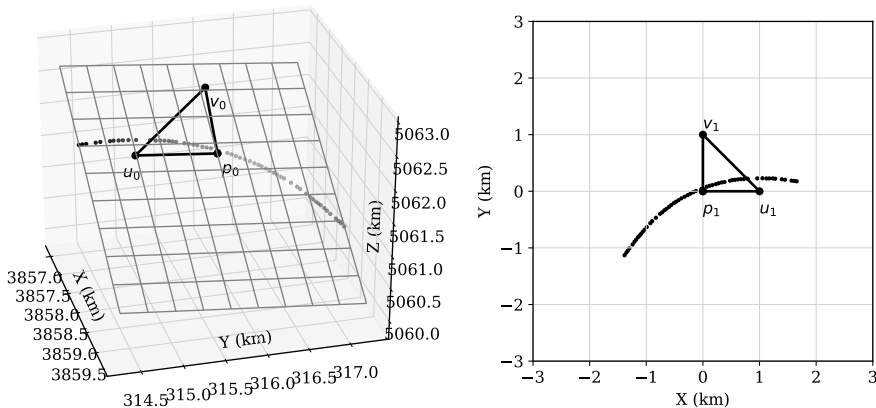


Figure 8.5: Transformation of 3D positions to 2D positions

Once positions are projected to the reference x,y-plane, it becomes simpler to calculate the turn radius. This can be achieved by using the least-squares regres-

sion to find the best circle to describe the trajectory.

First, define the circle as:

$$(x - x_0)^2 + (y - y_0)^2 = R_t^2 \quad (8.23)$$

For all positions (x_i, y_i) in \mathbf{p}_{xy} , the center of the turn (x_0, y_0) and the turn radius R_t can be found when the following condition is satisfied:

$$x_0, y_0, R_t = \operatorname{argmin}_{x_0, y_0, R_t} \sum_{i=1}^n [(x_i - x_0)^2 + (y_i - y_0)^2 - R_t^2]^2 \quad (8.24)$$

When turning radius and aircraft speed are known, bank angle (ϕ) , load factor (n) , and turn rate (ω) can be computed conveniently:

$$\phi = \arctan \left(\frac{V^2 \cos \gamma}{g R_t} \right) \quad (8.25)$$

$$n = \frac{1}{\cos \phi \cos \gamma} \quad (8.26)$$

$$\omega = \frac{d\psi}{dt} = \frac{V}{R_t} \quad (8.27)$$

Using these calculations, the example trajectory from Figure 8.3 yields a turn radius and turn rate, as shown in Figure 8.6.

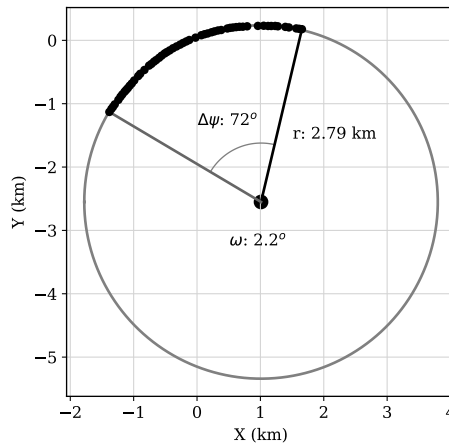


Figure 8.6: Computation of turn radius and turn rate for an example trajectory

8.5. Experiments and results

8.5.1. Experiment and validation data

We make use of a one-month long Mode-S dataset to evaluate the method proposed in this chapter. In this dataset, both ADS-B and Comm-B messages are recorded by a receiver set up at the Aerospace Engineering faculty of the TU Delft, during April, 2018.

The turning trajectories of aircraft are extracted from the dataset. When BDS 50 messages are available, we decode and save the roll and track rate information related to the respective turn. This allows us to have validation data alongside the trajectories that are to be used for performance analysis.

In total, 17 common aircraft types are considered in the experiment, where around 800 turns are extracted for each aircraft type. Because some aircraft types are more common than others, the final number of turns per aircraft type differs. For example, the data shows a higher number of Airbus A320 and Boeing 737 aircraft, and a lower number of Airbus A380 or Boeing 787 aircraft. The number of extracted turns per aircraft type is summarized in Table 8.1.

Table 8.1: Statistics of the turn trajectory dataset

Aircraft	Number of turns	Number of validations	
		Bank angle	Turn rate
A319	992	814	814
A320	999	805	805
A321	994	830	829
A332	998	798	799
A333	998	853	851
A343	82	70	70
A388	159	107	106
B737	998	753	753
B738	996	820	819
B739	1001	823	823
B744	1000	850	849
B752	721	596	596
B763	990	842	842
B77W	999	789	13
B788	577	465	465
B789	999	837	837
E190	974	728	728

8.5.2. Estimation results

For each turn trajectory extracted from the flights, we are able to estimate the bank angle, turn rate, and the load factor. The results are shown in Figures 8.7, 8.9, and 8.10 respectively.

In Figure 8.7, the common bank angle is shown to be around 20 degrees. During actual flight operations, the majority of bank angles range from approximately 12 degrees to 22 degrees.

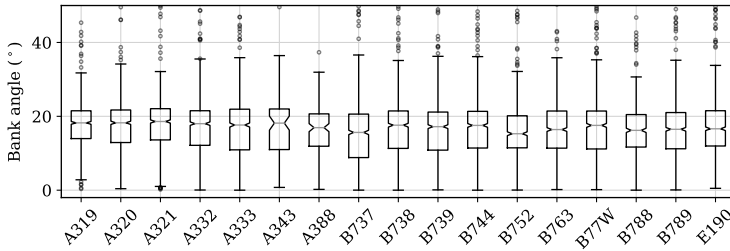


Figure 8.7: Distribution of estimated bank angles per aircraft type

In Figure 8.8, the corresponding flight path angles during the turns are shown. The majority of the flight path angles during the turns are below 6 degrees for all aircraft types.

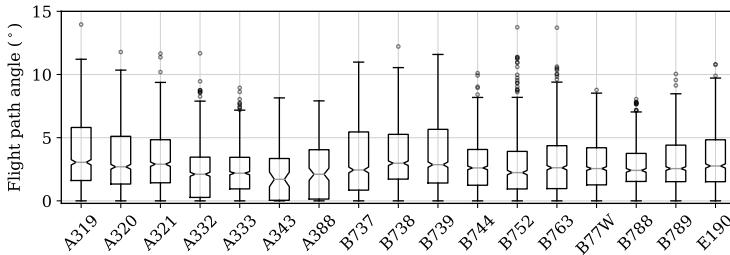


Figure 8.8: Distribution of flight path angles during the turns per aircraft type

8

In Figure 8.9, the common turn rates are shown to be around 1.5 degree/s. This result corresponds well to the standard *rate half* turn, which completes a 360° turn in four minutes.

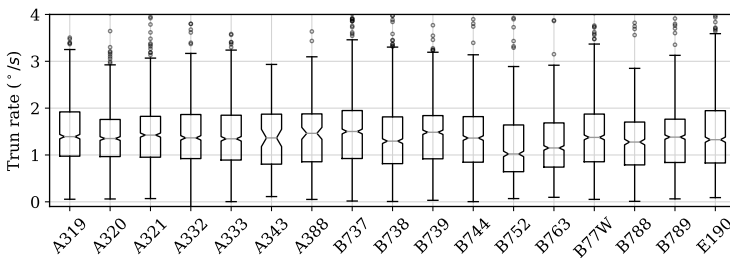


Figure 8.9: Distribution of estimated turn rate per aircraft type

In Figure 8.10, the common corresponding load factor for airliners is found to be around 1.05. For airliners, turning with low load factor is also a key factor to ensure the comfort of the passengers.

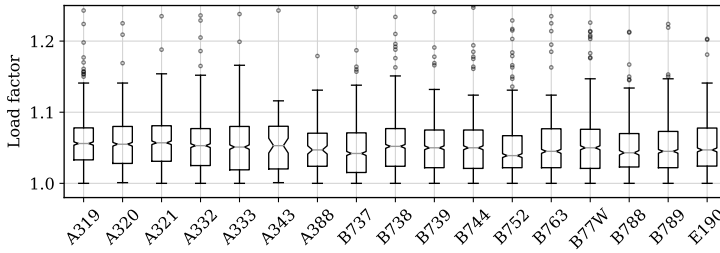


Figure 8.10: Distribution of estimated load factor per aircraft type

8.5.3. Estimation errors

Comparing the estimated bank angle and turn rate with the values obtained for Mode-S BDS 50 messages, it is also possible to examine the estimation error. The error is calculated as the difference between the median bank angle obtained from track and turn report and our estimation. The statistics of estimation errors are shown in Figures 8.11 and 8.12 for bank angle and turn rate respectively.

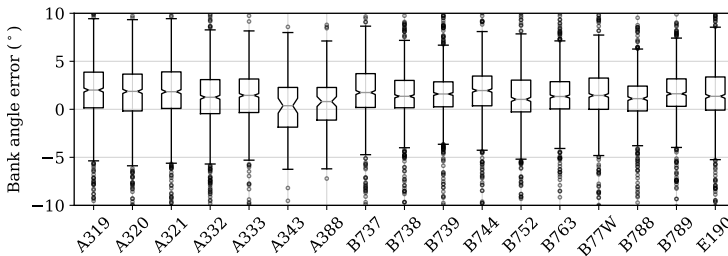


Figure 8.11: Distribution of bank angle estimation error per aircraft type

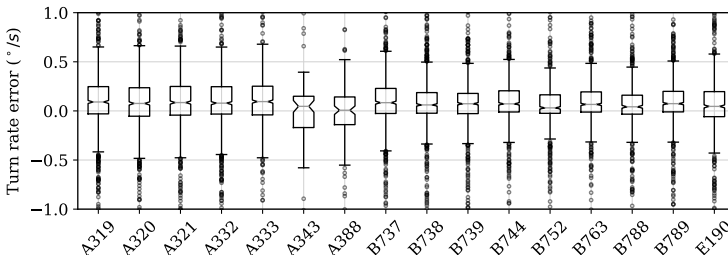


Figure 8.12: Distribution of turn rate estimation error per aircraft type

Based on these statistics, the median of bank angle error is calculated as less than 2 degrees among all available aircraft types. The median of turn rate error is less than 0.1 degrees/s among all available aircraft types. The overall positive median difference indicates a small but not significant under-estimation with the proposed method.

8.5.4. Correlation between speed and bank angle

Using the previous experiment data, we can study the correlation between the bank angle and speed. In Figure 8.13, the Pearson correlation coefficients are computed for each aircraft type. Based on the results of this figure, weak correlations are found for most of the aircraft types.

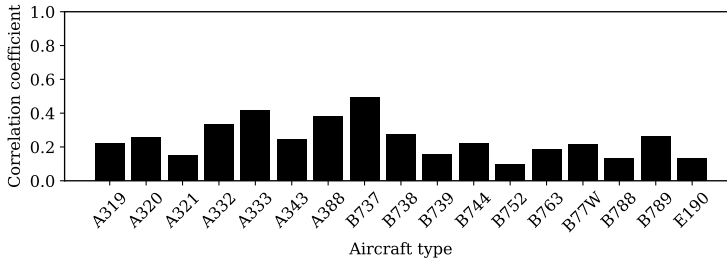


Figure 8.13: Correlation between speeds and bank angles

Aircraft maximum bank angle and speed are designed to be closely related to ensure the level of safety. Based on the weak correlation from this figure, it is possible to infer that large margins exist between the maximum allowed bank angle and the bank angle during flight.

8.6. Discussion

8.6.1. Mode-S track and turn report

In this study, one of the data sources for measuring turn performance is the track and turn report from the Mode-S secondary surveillance communications. The roll angle and track rate from the track and turn report are used to validate the estimated values of bank angle and turn rate.

Although it offers a direct method to obtain turn performance, the track and turn report has a much lower level of availability. These reports are selectively interrogated at a lower frequency than the ADS-B broadcast rate. For some trajectories, no track and turn report has been intercepted during the turns. The difference in the number of turns and the number of validations, as shown in Table 8.1, reflects this fact. Even when this information is interrogated, not all parameters are available in some cases. For example, most of the track rates are not included in Boeing 777-300ER (B77W) track and turn reports.

Other than the lower availability compared to ADS-B messages, the data in the track and turn reports are prone to large variations based on observations. Hence, based on the ADS-B data, we are able to provide a better source of measurement for air turn performance.

8.6.2. Least-square regression

In order to find the turn radius, the least-squares regression of a circle is applied to the 2D projections of the 3D positions on a *best* plane. This plane is also found

using the least-squares method in the first place, with which the distance from all positions are minimized.

Using such a two-step regression is intentional. Firstly, introducing such a plane allows us to reduce the variations in position, as well as to consider the turns with changing of altitude. Secondly, by reducing the dimension of the circle from 3D to 2D, the uncertainty of the estimation is decreased. This leads to a more accurate arc representing the turning trajectory.

8.6.3. Turn under heavy wind conditions

During the turn, aircraft commonly maintain a constant airspeed. Thus, wind can have a strong influence on the trajectory of an aircraft during a turn. This influence can be observed from the ground tracks as well. Strong wind can produce a non-circular ground track, as illustrated in Figure 8.14.

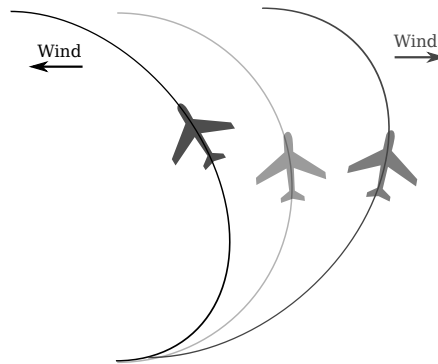


Figure 8.14: aircraft turn track under wind (not to scale)

In real flights, wind can be taken into consideration when planning an optimal turn [91]. Previous research has also used aircraft airspeed and ground speed during turns to estimate wind information [29, 30, 71, 81].

The non-symmetric ground track will affect the estimation of performance parameters using the regression method proposed in this chapter. Using these tracks could result in unrealistic turn radius estimations. For the experiments in the earlier section, we want to minimize the effect of wind. Thus, turns with large regression errors are filtered out.

8.6.4. Aircraft mass and turn performance

When an aircraft is turning at low speed, the bank angle needs to be smaller than the maximum allowed bank angle. This limitation leaves a safety margin ensuring that an aircraft does not enter stall condition. The maximum physical limitation of the bank angle is constrained by the maximum lift available and the weight of the aircraft. Knowing an aircraft's aerodynamic characteristic and airspeed, the mass boundary can be calculated.

Since the load factor of the aircraft can be computed based on the estimated bank angle, the maximum mass of the aircraft during the turn can be computed

as:

$$\begin{aligned}
 \tilde{m}_{max} &= \frac{W_{max}}{g} \\
 &= \frac{L_{max}}{ng} \\
 &= \frac{C_{L,max}}{ng} \frac{1}{2} \rho V^2 S
 \end{aligned} \tag{8.28}$$

where the $C_{L,max}$ is the maximum lift coefficient and S is the wing area. Given the maximum takeoff weight and the operational empty weight of an aircraft, we can reconstruct the mass boundary as:

$$m_{oew} < m < m_{max} \tag{8.29}$$

where m_{max} is the smaller value between m_{mtow} and \tilde{m}_{max} :

$$m_{max} = \min \left(m_{mtow}, \frac{C_{L,max}}{ng} \frac{1}{2} \rho V^2 S \right) \tag{8.30}$$

Using the same dataset from the experiment section, we can compute the maximum possible weight of the aircraft at each turn. In Figure 8.15, the distributions of the possible maximum mass are illustrated as the percentage of the maximum takeoff mass. Note that only turns with maximum masses lower than maximum takeoff mass are included. The number of trajectories compared to all trajectories are indicated in the lower left corner of each plot.

In this test, we set the maximum lift coefficient as a fixed value of 1.4 for all aircraft. In reality, the maximum lift coefficient differs according to the aircraft type and aerodynamic configurations. Regardless of the choice of maximum lift coefficient, this figure demonstrates the additional information on weight obtained by observing the turn.

8.7. Conclusion

In this chapter, we addressed the challenge of estimating aircraft performance parameters during the turning segments of flights. The method relied solely on open ADS-B data, without additional information necessary. It can also be used on radar data due to the similar nature of their properties.

At first, we developed a simple data mining algorithm that made it easy to identify the turn segments in the flight. Once an individual turn was extracted from the flight, we transformed the three-dimensional turning trajectory onto a two-dimensional plane, where the radius of the turn could be computed using least-squares regression.

In this study, we only considered the coordinated turns under light wind conditions. The coordinated turn was a safe assumption for most airliners. However, when heavy wind condition occurs, it is possible that our method produces an esti-

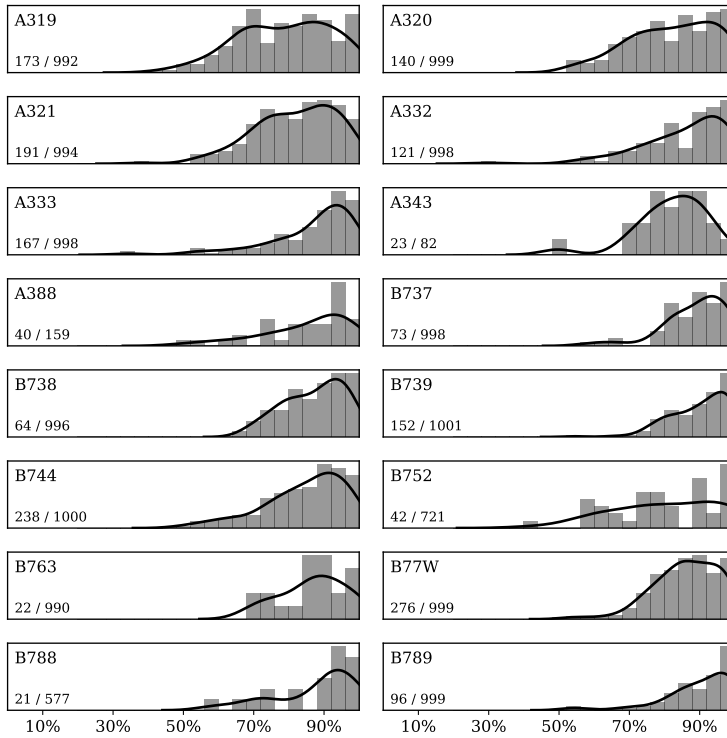


Figure 8.15: Distribution of maximum flight mass per aircraft type (only $m < m_{mtow}$ are considered, $C_{L,max} = 1.4$)

mate with a larger error. Knowing the wind information in addition to the ADS-B data would improve the calculation. It is worth noting that flight management systems from different manufacturers may perform the same turn procedures differently. For example, in research [52], fixed radius-to-fix experiments were carried out to test these flight management systems. Results showed differences in these test flights, with of the most noticeable ones been how the wind, as well as the climb gradient, were handled during the turns.

Combining the turn radius and the speed of the aircraft, we were able to compute the bank angle and load factor, as well as the turn rate of the aircraft during the turn. Using the data from secondary Mode-S communications (track and turn report), we were able to validate the estimations derived from ADS-B data. The median of the errors for bank angle and turn rate were around 2 degrees and 0.05 degree/s respectively.

Knowing the bank angle and turn rate would bring more accurate information for future ADS-B based aircraft performance studies. For instance, we demonstrated that knowing the actual load factor, we were able to increase our knowledge of the aircraft mass. This, in turn, would help to improve other performance parameter estimations.

9

Mass estimation using Bayesian inference

Aircraft mass is a crucial piece of information for studies on aircraft performance, trajectory prediction, and many other areas for research in aircraft traffic management. However, accessing this proprietary information is challenging for researchers and air traffic controllers. Previously, several studies have proposed methods to estimate aircraft weight based on specific segments of the flight. Due to inaccurate input data or biased assumptions, this often leads to less confident or inaccurate estimations. In this chapter, several aircraft masses are computed independently using the total energy model and reference model at different flight phases. We then adopt the Bayesian inference that uses a prior probability of aircraft mass based on empirical knowledge and computed aircraft initial masses to produce the maximum a posteriori estimation. Variations in results caused by prior assumptions, thrust, and wind are studied. The method is tested using 50 test flights of a Cessna Citation II aircraft, for which measurements of the true mass were available.

This chapter is based on the following publications:

- 1) Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2017. Bayesian inference of aircraft initial mass. Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar [125].
- 2) Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2018. Aircraft initial mass estimation using Bayesian inference method. Transportation Research Part C: Emerging Technologies, 90, pp.59-73 [123].

9.1. Introduction

Aircraft mass is a fundamental parameter for studies on aircraft performance and trajectory prediction. However, data concerning the mass of almost all modern commercial flights are treated as confidential information by airline operators, which poses a challenge for the research community. Studies [69] and [25] have shown that inaccurate aircraft mass estimations introduce a significant source of error and affect ground-based trajectory predictions. [137] implemented an adaptive aircraft weight algorithm to improve the accuracy of climbing predictions, while [40] illustrated the significant influence of aircraft mass on fuel burn during continuous descent operations.

Within the air traffic management research community, several methods have been developed to estimate aircraft mass based on flight data, either from radar data or more recently from ADS-B data. In two separate studies [2] and [4], a least-squares method and a machine-learning method were developed, which focused on the climb phase of aircraft. In a similar approach considering climbing aircraft [118], an adaptive estimation method is implemented for mass and thrust approximation. More recently, ADS-B data from takeoff is used to estimate the initial mass of an aircraft with two different analytical methods [129]. On the operational side, a different approach has been proposed by [80], which tries to calculate the weight of an aircraft based on an approximation of each individual weight component, i.e. aircraft empty weight, fuel weight, and payload weight. For most of these studies, the focus was on a specific part of the flight (takeoff or climb).

In this chapter, the use of multiple masses calculated during the different flight stages is identified as a potential improvement for mass estimation. We study not just one specific flight phase, but a combination of all phases. More information on aircraft mass can, thus, be included. When mass from different flight phases are computed, the final estimation problem is considered as a single parameter Bayesian inference problem [43, p.29]. Throughout the chapter, we consider masses computed along with the entire flight as *observations*.

Prior knowledge of weight can be used to improve the estimation. For instance, aircraft will never operate above their maximum takeoff weight or below their minimum operational weight. In practice, given an approximation of the number of passengers, one can even further constrain the weight estimation range by making an estimate of the aircraft payload. These kinds of prior knowledge can be very valuable for estimating the actual aircraft mass when applying Bayesian inference.

The remainder of this chapter is structured as follows: first, this chapter presents several existing methods to calculate aircraft mass independently in each flight phase. Then, in section three, a Bayesian inference approach is established to use these calculations as independent measurements, combining a priori knowledge of initial aircraft mass probability distribution to produce a maximum a posteriori estimation. Figure 9.1 offers an illustration of the entire estimation process.

Section four discusses the results for several aircraft types and parameter sensitivities. Section five uses a real flight dataset to test the Bayesian inference method. Finally, discussion and conclusions are presented in sections six and seven.

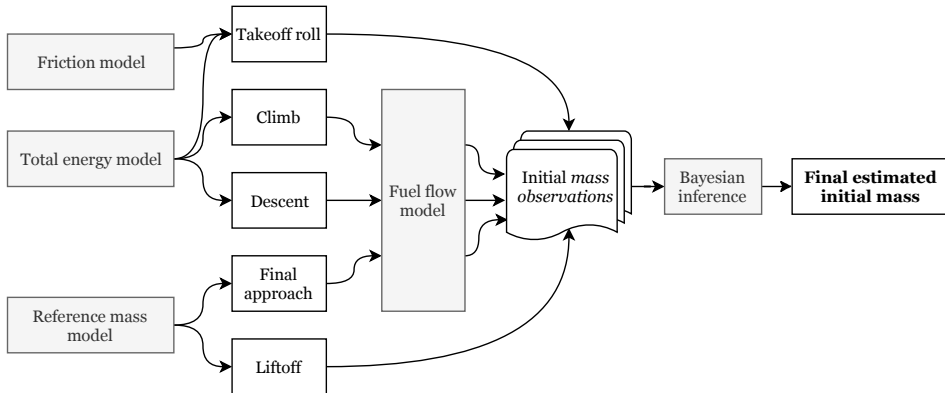


Figure 9.1: Flow chart of the estimation structure

9.2. Calculation of mass at different flight phases

This section describes five methods that can be used independently to compute aircraft mass at different flight phases. The total energy model (TEM) is used when the dynamic of the aircraft model is required. The model is described as:

$$\begin{aligned}
 (T - D)V &= maV + mgVS & (9.1) \\
 D &= C_D \frac{1}{2} \rho V^2 S \\
 L &= C_L \frac{1}{2} \rho V^2 S \\
 C_D &= C_{D0} + kC_L^2
 \end{aligned}$$

Here, T and D are the thrust and drag of the aircraft. V , a , and VS are the airspeed, acceleration, and vertical rate respectively. C_D , C_L , C_{D0} , and k are coefficients for drag, lift, zero lift drag, and lift-induced drag. ρ and S are the air density and the aircraft reference wing surface.

With the complete flight trajectory based on the total energy model, mass can be computed at different phases as illustrated in Figure 9.2, where the subscripts TO , LOF , CL , DE , and APP represent takeoff, liftoff moment, climb, descent, and final approach respectively. A fuel flow model is used together with these estimates to generate multiple initial takeoff mass calculations, which is used for the final inference.

In this chapter, the term *mass observation* is used for initial masses that are computed at different stages of the flight in combination with the fuel flow, as illustrated in Figure 9.1. It is worth noting that these *mass observations* are not directly observed but calculated using the deterministic total energy equations. The remainder of this section discusses how initial masses are computed deterministically.

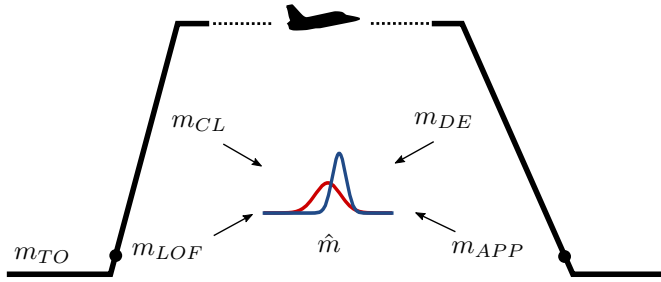


Figure 9.2: Bayesian inference using multiple mass observations

9.2.1. Computing mass at takeoff

While an aircraft is on the runway before liftoff, it only accelerates horizontally. Because it is still on the ground, ground friction needs to be considered in addition to the aerodynamic drag. This ground friction is proportional to the normal force on the ground, with a friction coefficient denoted by μ_1 . A common value is around 0.02 for concrete runways [143]. Equation 9.1 then can be re-written as follows:

$$T - D - D_g = ma \quad (9.2)$$

$$D_g = \mu_1(W - L) \quad (9.3)$$

Since BADA 3 does not have an accurate model for aircraft thrust during the takeoff, the empirical model form [8] is used. The detail for this thrust model is discussed in Chapter 6 of this dissertation. This thrust model can also be used for other phases during the flight. During the takeoff, the maximum takeoff thrust is expressed as a function of velocity:

$$T_{\max} = f(V) = T_0(1 - c_1V + c_2V^2) \quad (9.4)$$

where T_0 is the maximum static thrust at sea level. Parameter c_1 and c_2 are coefficients that are dependent on the engine bypass ratio. During takeoff, the optimal lift coefficient can be derived. Equation 9.2 is rewritten as follows:

$$\begin{aligned} T &= D + D_g + ma \\ &= \frac{1}{2}\rho V^2 S(C_{D0} + kC_L^2) + \mu_1(mg - \frac{1}{2}\rho V^2 SC_L) + ma \\ &= \frac{1}{2}\rho V^2 S(C_{D0} + kC_L^2 - \mu_1 C_L) + \mu_1 mg + ma \end{aligned} \quad (9.5)$$

For a given velocity V , T is minimum when C_L satisfies the following equation [87]:

$$C_L = \frac{\mu_1}{2K} \quad \left(\frac{dT}{dC_L} = 0 \right) \quad (9.6)$$

When the requirement of minimum excess thrust is satisfied, aircraft can take off with reduced thrust. This has become common practice for airliners to extend engine life and reduce costs. To accommodate this possibility in the model, a thrust coefficient δ_T is introduced that satisfies $\delta_T \leq 1$. Combined with Equation 9.6, Equation 9.5 becomes:

$$\delta_T T_{\max} - (\mu_1 g + a)m - \frac{1}{2} \rho V^2 S (C_{D0, \text{to}} - \frac{\mu_1^2}{4K_{\text{to}}}) = 0 \quad (9.7)$$

Let the left part of the equation be $f_1(m, \delta_T)$. The optimal \hat{m} and $\hat{\delta}_T$ that minimize the squared sum of f_1 of all takeoff data samples can be found as follows:

$$\begin{aligned} \hat{m}, \hat{\delta}_T &= \arg \min_{m, \delta_T} \sum_{i=1}^N f_1^2(m, \delta_T) \\ m &\in (0, c_m) \\ \delta_T &\in (c_{\delta_T}, 1) \end{aligned} \quad (9.8)$$

where the δ_T constraint represents the possible reduction of thrust during the takeoff, and the range of m represents a very large mass range. The coefficients c_m and c_{δ_T} are user-defined boundary conditions. The quality of the optimization solution depends on the accuracy of the velocity measurements and the number of data samples. It is possible that an optimal mass solution may not be found and that the optimization, instead, converges to the condition boundaries. In this chapter, a *mass observation* at the boundary conditions (zero or $2m_{\text{mtw}}$) is considered as an *impossible observation*, which is discarded for the final inference.

9.2.2. Computing mass at liftoff and final approach

At each liftoff and final approach, optimal speeds are usually selected, which are correlated to the stall speed. Even though it is a relatively weaker correlation at takeoff, both speeds can be used as indicators to infer aircraft mass.

This method first observes the aircraft liftoff speed (airspeed at which the aircraft first becomes airborne) and the speed of the final approach. Then, it infers the mass taking into account the relationship between the reference stall speed and the maximum takeoff (or landing) weight. This data can be obtained from the flight manual or BADA reference data.

At takeoff, the speed of an aircraft is often related with the mass of the aircraft. For example, according to FAA regulation, the takeoff speed has to be at least 20% over the stall speed. Assuming lift and weight are the same at liftoff, the following relation can be derived under certain assumptions:

$$\begin{aligned}
 W_{\text{to}} = L_{\text{to}} &\Rightarrow m_{\text{to}g} = \frac{1}{2}\rho S C_L V_{\text{lof}}^2 \\
 &\Rightarrow m_{\text{to}} \propto V_{\text{lof}}^2
 \end{aligned} \tag{9.9}$$

where the takeoff mass is proportional to the liftoff speed squared, under the assumptions of the same aircraft model and lift configurations. Knowing reference weight and corresponding stall speed (V_S), it is possible to calculate such a constant coefficient:

$$C = \frac{m_{\text{to}}}{V_{\text{lof}}^2} \tag{9.10}$$

$$= \frac{m_{\text{ref,lof}}}{V_{\text{ref,lof}}^2} \leq \frac{m_{\text{ref,lof}}}{1.2^2 \cdot V_{S,\text{ref,lof}}^2} \tag{9.11}$$

Then, at any given observed takeoff, the initial mass can be approximated using the following equation:

$$m_{\text{to}} \approx \left(\frac{m_{\text{ref,lof}}}{1.2^2 \cdot V_{S,\text{ref,lof}}^2} \right) V_{\text{lof}}^2 \tag{9.12}$$

A similar relationship can be obtained at landing. The approach speed empirically is around 30% above the stall speed. The landing mass, then, can be approximated as follows:

$$m_{\text{ld}} \approx \left(\frac{m_{\text{ref,app}}}{1.3^2 \cdot V_{S,\text{ref,app}}^2} \right) V_{\text{app}}^2 \tag{9.13}$$

Such approaches are also addressed in the previously mentioned research by [40]. When using ADS-B data, it is common that there is a certain gap between the on-ground data and in-air data. Hence, the liftoff speed and approach speed can be obtained using an interpolation method based on data points during takeoff or landing, as shown in Chapter 5.

9.2.3. Computing mass at climb and descent

During the climb and descent phases, horizontal speed and climb/descent rate observations can be employed to estimate aircraft mass. Assuming standard atmospheric conditions, the total energy model in Equation 9.1 can be expanded at each time step as follows:

$$(T_t - D_t) V_t = m_t a_t V_t + m_t g V S_t \tag{9.14}$$

$$m_t = m_0 - m_{\text{fuel},t} \tag{9.15}$$

$$T_t = \delta_T T_{\text{max},t} = \delta_T f(h_t, V_t) \tag{9.16}$$

$$D_t = C_D \frac{1}{2} \rho_t V_t^2 S \quad (9.17)$$

$$L_t = m_t g \cos(\gamma_t) = C_{L,t} \frac{1}{2} \rho_t V_t^2 S \quad (9.18)$$

$$C_D = C_{D0} + k C_{L,t}^2 \quad (9.19)$$

Here, γ represents the flight-path angle. Maximum thrust profile T_{\max} is a function of pressure altitude and airspeed, and δ_T is an assumed thrust coefficient for the entire climb or descent, describing the actual thrust setting as a percentage of maximum thrust. Given all observed and known variables, the above equations can be rewritten into the following equation:

$$\begin{aligned} \frac{2kg^2 \cos^2(\gamma_t)}{\rho_t V_t^2 S} \cdot m_t^2 + \left(a_t + g \frac{VS_t}{V_t} \right) \cdot m_t \\ + C_{D0} \frac{1}{2} \rho_t V_t^2 S - \delta_T \cdot T_{max,t} = 0 \end{aligned} \quad (9.20)$$

Fuel consumption can be estimated according to the fuel model defined in Chapter 6. The two remaining unknown variables m_0 and δ_T remain to be found. This process is similar to the takeoff method but includes more parameters in the equation. Let the left side of Equation 9.20 be $f_2(m_0, \delta_T, \theta_t)$, where $\theta_t = (m_{\text{fuel},t}, V_t, VS_t, \rho_t, \gamma_t)$. The task is to find the optimal \hat{m} and $\hat{\delta}_T$ that minimize the squared sum of f_2 of all N data samples. The mass is constrained by the aircraft Operational Empty Weight and the Maximum Takeoff Weight, and thrust reduction is no larger than 20% of the maximum thrust profile. The solution can be found as follows for climbing flights:

$$\begin{aligned} \hat{m}_0, \hat{\delta}_T &= \arg \min_{m_0, \delta_T} \sum_{t=1}^N f_2^2(m_0, \delta_T, \theta_t) \\ m &\in (0, c_m) \\ \delta_T &\in (c_{\delta_T}, 1) \end{aligned} \quad (9.21)$$

Although the total energy equation is the same for descent flights, their thrust profiles are different. In the above equations, $\delta_T T_{max,t}$ needs to be replaced with an idle thrust or appropriated thrust setting in the case of a powered descent. In this study, idle thrust is assumed during descent.

Compared to the takeoff phase, climb and descent usually last for a significantly longer time, where a large number of data samples can be gathered. Rather than using all data from the entire climb or descent for the calculation, these data can be split into smaller segments which produce multiple *mass observations*.

9.2.4. Computing fuel consumption

With the polynomial fuel flow model based on ICAO data-bank proposed in Chapter 7, the fuel consumption is now approximated as a continuous function in relation to the dynamic thrust of the engine. The initial mass can be calculated with the total fuel burn:

$$m_0 = m_t + m_{\text{fuel}} = m_t + \sum_{i=1}^t f_{\text{fuel}}(V_i, h_i, VS_i) \Delta t \quad (9.22)$$

9.3. Bayesian Inference

As shown in the last component of the estimation flowchart (Figure 9.1), the final estimate is computed using the Bayesian inference. The Bayesian inference uses Bayes' theorem [9] to update a hypothesis probability based on evidence (or observation). In the case of estimating aircraft mass, it uses computed mass at each flight phase to update an initial belief. This belief is called *prior* in Bayesian terms. The prior essentially describes the probability density function of a parameter before any observation is made.

To understand the process, first, let us assume n possible *mass observations* are obtained from different flight phases:

$$\mathbf{m} = m_1, m_2, \dots, m_n \quad (9.23)$$

These observations are possible values of initial aircraft mass. However, due to errors in aircraft flight data and estimations, those values can be very different from reality or even out of the physical boundaries. If, for simplicity, one assumes the observed aircraft mass follows a normal distribution:

$$m \sim \mathcal{N}(\mu, \sigma^2) \quad (9.24)$$

where the μ parameter refers to the underlying real aircraft mass and σ^2 is the variance of the observations.

The Bayesian inference method introduces a prior probability function for the parameters μ and σ^2 . The posterior probability function $p(\mu, \sigma^2 | \mathbf{m})$ is computed as the probability of μ based on observed measurements. The most likely value in the posterior distribution function (the mode) is denoted as the *Maximum a Posteriori Probability Estimate* (MAP).

To deduce the simplest closed form of the posterior function, the fixed variance (a chosen σ^2) assumption is used. Let the prior probability function for μ be defined as follows:

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0^2) \quad (9.25)$$

The posterior probability distribution of μ satisfies the following normal distribution:

$$\mu|\mathbf{m} \sim \mathcal{N}\left(\frac{n\sigma_0^2\bar{m} + \sigma^2\mu_0}{n\sigma_0^2 + \sigma^2}, \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right)^{-1}\right)$$

$$\bar{m} = \frac{1}{n} \sum_{i=1}^n m_i$$
(9.26)

where n is the number of the measurements and \bar{m} is the mean of all measurements.

9.4. Experiments and results

To examine and test the Bayesian estimation approach, data from a large number of flights are gathered for two common aircraft types (A320 and B738), which are based on ADS-B data from the FlightRadar24 network. These flights were carried out between the 10th and the 16th of October, 2016. The first 1500 complete trajectories for the experiment were selected. As an illustration, the great circle paths between the origins and destinations of all flights are drawn in Figure 9.3.

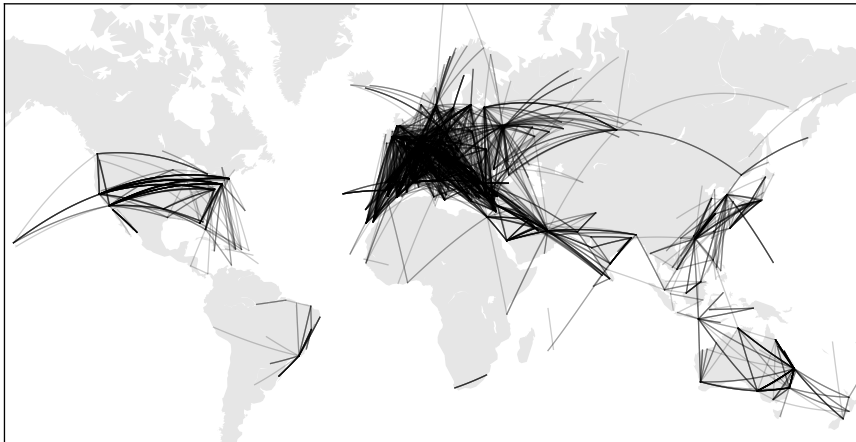


Figure 9.3: All flights in the dataset

For each flight, multiple initial *mass observations* are computed at different flight phases. After that, the Bayesian mass inference is applied. In this experiment section, three experiments are carried out: mass estimation for a single flight, distributions for mass estimated for all flights, and sensitivity studies. For the Bayesian inference, by default, the following values are used:

$$\begin{aligned}
 \mu_0 &= 65000 \text{ kg} \\
 \sigma_0 &= 10000 \text{ kg} \\
 \sigma &= 10000 \text{ kg}
 \end{aligned}
 \tag{9.27}$$

9.4.1. Single flight demonstrations

The Bayesian inference of two flights is illustrated in Figure 9.4. In the first plot, *mass observations* are computed at different flight phases. These numbers correspond to the solid probability density function in the second plot. The prior probability is illustrated as the dotted function. The posterior probability is computed and illustrated by the dashed function.

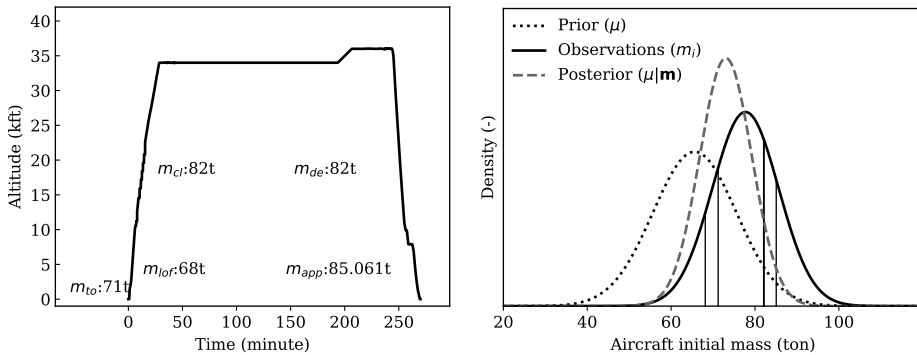


Figure 9.4: Example of Bayesian inference. Mass computed at different phase in the first plot corresponding to the vertical lines in the second plot. The final MAP estimate yields approximately 73 tons of initial mass.

9.4.2. Distributions for A320 and B737-800

Figure 9.5 shows the results of the inference method applied to all A320 and B738 flights in the dataset. The first five sub-plots illustrate the distribution of individual measurement results from methods that are applied at different flight phases, which are takeoff, liftoff, climb, descent, and final approach. The last plot shows the final estimated initial masses using Bayesian inference.

By studying the results for these two different aircraft types, it is possible to conclude that at different flight phases, the initial mass computed using deterministic methods contains large variances. Some of the results are outside of the boundaries of m_{ow} and m_{mtw} . The main causes are data inaccuracy, uncertainty in the thrust, and uncertainty in the airspeed. For liftoff and final approach estimates, the margin of stall speed between the model and the actual setting may cause large inaccuracies (in the case of B737-800). A more detailed discussion for these biases is addressed in section 9.6.3.

When looking at five distributions of the measurements and the final MAP estimates, it is evident that the MAP estimates have a smaller variance than the

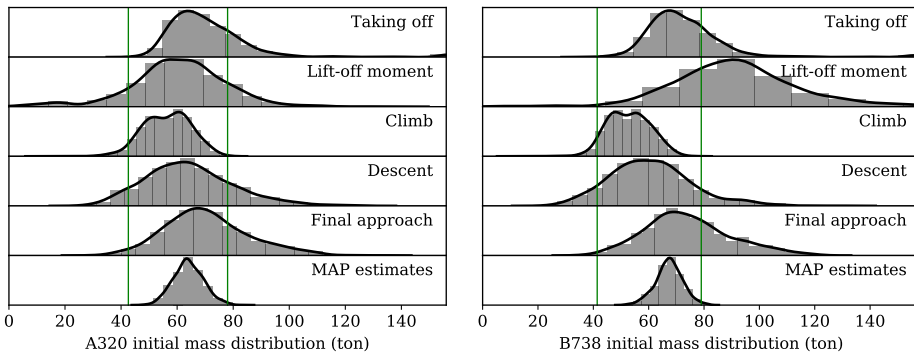


Figure 9.5: Comparison of estimations at different flight phases. Solid vertical lines refer to the operational empty mass and maximum takeoff mass.

individual estimates, and most of the initial estimates are within the boundaries of m_{oeW} and m_{mtw} .

9.4.3. Sensitivity analysis

In this section, the goal is to study how variations of different factors lead to changes in the estimation results, as well as the degree of these differences. Follow-on experiments are performed on the A320 dataset that was previously used.

Prior

When applying Bayesian inference, the prior stands as an important factor when observations are few. A stronger belief in the prior knowledge will increase the confidence of the final estimation. However, these estimates can be biased towards the prior belief. To study this influence, the same A320 dataset is treated with a set of different priors to produce different distributions of estimates. In Figure 9.6, six combinations of (μ_0, σ_0) are chosen to study this effect. The probability distribution functions that are biased towards the low, medium, and high mass are shown in different shards of gray. Priors with high and low confidence are shown in solid and dashed lines, respectively.

In Figure 9.7, six different results from the same estimation configuration are displayed. Results are grouped by μ_0 , corresponding to the bias of the prior. On the left-hand side, the groups are distributions from low confidence priors (higher σ_0), On the right-hand side, the groups are distributions from high confidence priors (lower σ_0).

From these plots, it is possible to see that with a weak prior, the estimates are less biased. They depend more on the observations (i.e.: different initial masses calculated based on data from different flight phases). They also, in turn, produce less confident results. For instance, all three experiments end up with distributions with larger spreads. A good choice of a prior often requires some empirical knowledge with regard to the flights.

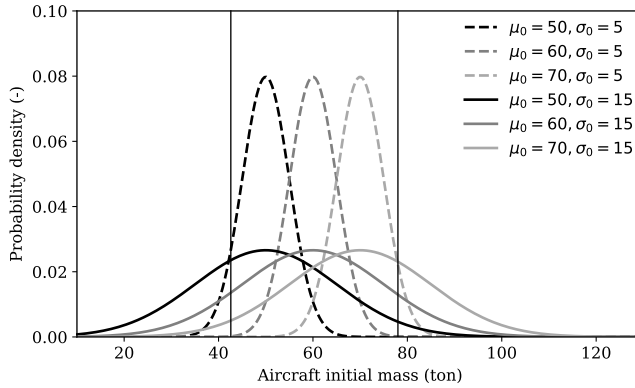


Figure 9.6: Different initial mass prior probability density functions (for A320). Vertical lines indicate the minimum mass and maximum mass of the aircraft.

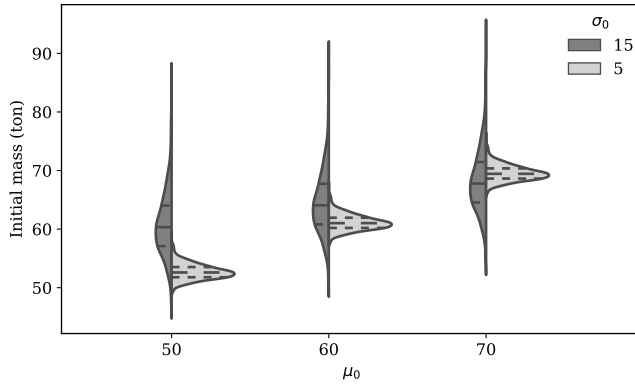


Figure 9.7: Variation in distributions of estimates due to different priors (A320 dataset)

Thrust

The level of correctness of aircraft mass estimations based on observed performance parameters is essentially dependent on knowledge of thrust settings. As shown in section 9.2, most of the mass computation methods along the flight path need to take into account the thrust profile while estimating aircraft mass. In these methods, an optimization using the least squares method is implemented to find the best thrust setting and aircraft mass, which provides a minimal squared error. This approach has also been proposed previously [2].

In order to study the influence of different thrust settings, based on the same dataset, two sets of fixed minimum and maximum thrust profiles are used to produce two different sets of mass estimates. For takeoff and climb, maximum thrust and a 30% reduced thrust profile are used. While, for the descent, two thrust settings at 8% and 20% of maximum climbing thrust setting are used. In Figure 9.8, results from different settings are shown, where the distributions from all flights

Table 9.1: Median of estimated mass ($\times 10^3$ kg) under two thrust settings

Phase	T_{\min}	T_{\max}	Δm
TO	60.7	83.9	23.2
LOF	56.1	56.1	–
CL	42.1	64.4	22.3
DE	56.4	45.0	-11.4
FA	68.7	68.7	–
MAP	58.1	63.7	14.6

at different flight phases are compared.

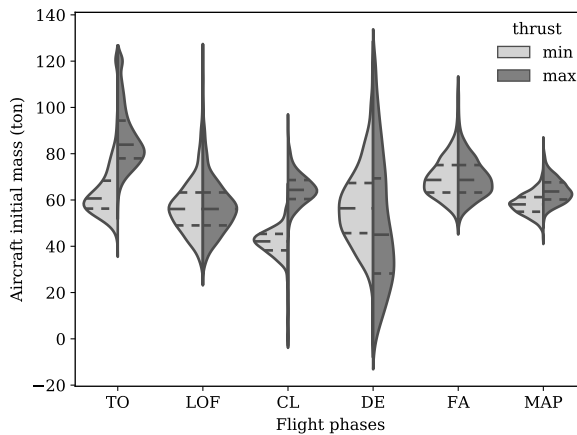


Figure 9.8: Variation in distributions of estimates due to thrust settings (A320 dataset)

During the takeoff (TO) and climb (CL), higher thrust settings lead to higher estimations of initial masses, with the difference being quite significant. During the descent (DE) phase, such a tendency is not visible. However, the uncertainty of mass estimated under high powered descent is much larger. Due to the fact that thrust is not involved for estimates from liftoff (LOF) and final approach (FA), these results remain the same. When comparing these two sets of distributions, the final MAP distributions display considerable differences.

It is worth emphasizing that the mass observations obtained in these experiments are not the optimal solution in comparison with the results from Figure 9.5. All values are computed under a fixed thrust setting. For the latter case, mass from the best mass/thrust pair was selected. From this test, it is possible to conclude that thrust setting is a major factor that affects the mass estimation. The link between mass and thrust should, therefore, not be neglected. Numerical differences (median of the distributions) are shown in Table 9.1.

A comparison of the distributions under two thrust settings shows that the difference in MAP is smaller than the ones seen in the previous mass observation methods. This is because, under the prior assumption, measurements that are far away from a believable value are considered with lower probability. The large

fluctuation caused by distinct thrust settings is reduced. Therefore, more convincing distributions are produced. It is positive to see that the bias still persists in the MAP results, which confirms that the final result maintains the knowledge about thrust settings.

As it is common practice nowadays to apply de-rated thrust takeoff/climb to reduce engine wear, as well as minimum thrust descent to save fuel, the minimum thrust profile used in this experiment is closer to reality. This can be reflected in the resulting distributions, where maximum thrust profile produced much less realistic results.

Airspeed

All estimation methods introduced in section 9.2 require correct measurements of aircraft airspeed. However, data collected from ground measurements (ADS-B or radar) only reveal the ground speed. Estimations can either assume the ground speed as airspeed or integrate wind data to approximate the airspeed. Intuitively, this uncertainty in wind can affect the estimation results.

In the experiment dataset, meteorological data are integrated with ground speed to approximate the true airspeed of aircraft. The Global Forecast System (GFS) re-analysis dataset is used to construct the wind field globally. Airspeeds of all trajectories are computed as the difference between the ground speed and wind speed.

For each flight, two sets of initial masses are computed with ground speed condition and airspeed condition. Finally, the difference in estimated mass is found and illustrated in Figure 9.9. It can be seen that the estimation can vary up to five metric tons of difference for the Airbus A320, which is around 6% of the maximum takeoff mass.

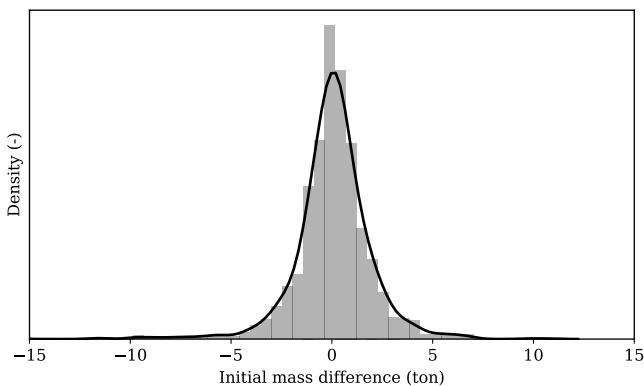


Figure 9.9: Difference in estimation caused by wind

9.5. Demonstration with Cessna Citation flight data

We use data from a set of student demonstration flights with a Cessna Citation II laboratory aircraft to test the proposed method. A total of 50 flights were carried out in a period of three weeks in March 2017. For each flight, the initial mass is computed as the sum of measured passenger weight, fuel weight, and aircraft basic empty weight. During the flight, a large number of flight parameters (such as fuel flow) are recorded. The average duration of the flights is approximately 80 minutes.

Figure 9.10 shows how the initial mass varied from flight to flight. As the number of passengers is almost always the same, most of this initial mass variation is caused by the refueling schedule. Figure 9.10 also shows that all flights are taking off at a weight close to the maximum takeoff weight (indicated by the top solid line). It should be noted that for some of the flights (numbers 2, 6, 14, 27, 37, and 49), the initial fuel mass was not recorded in the flight data sheet. The amount of fuel on these flights is interpolated for these flights. These data points are marked distinctly in the figure.

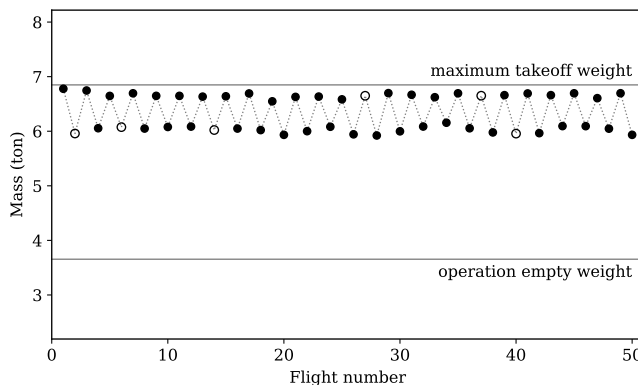


Figure 9.10: Aircraft initial mass for all flights

Figure 9.11 illustrates the result of Bayesian estimation applied on all flights. The difference between true mass and estimated mass is shown for each individual flight. The mean absolute error (MAE) is 4.3% of the total true mass, while all estimates are within 12% of MAE, with prior parameters of $\mu_0=6000$ kg and $\sigma_0=500$ kg.

To assess the effect of the choice of prior, six different priors are tested with the same data, as shown in Figure 9.12. Here, the plots on the right-hand side represent the distribution from different priors. Within the distribution, the scattered points are all estimated mass values. It can be seen that as expected, with the true mass as a comparison, the prior that yields the best result is the one biased towards the true mass and with a high level of belief. The prior yielding the worst result is the one with a high level of belief but biased towards the wrong values. On the other hand, selecting a weak prior (larger σ_0 , lower level of belief), reduces the influence of bias but increases the estimation uncertainty.

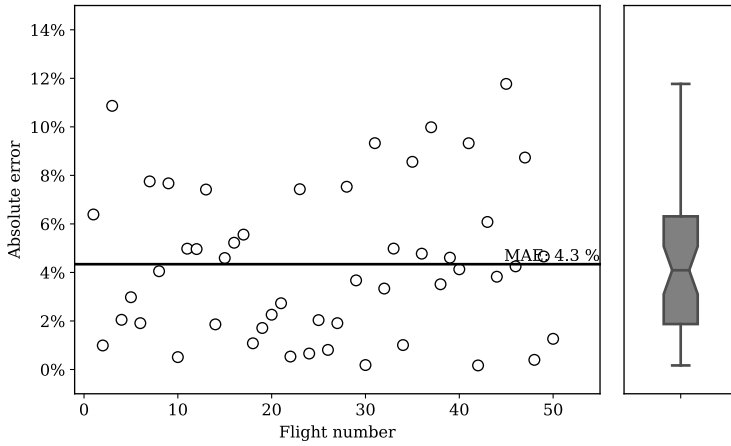


Figure 9.11: Bayesian estimation for all flights (prior parameters: $\mu_0=6000$ kg, $\sigma_0=500$ kg)

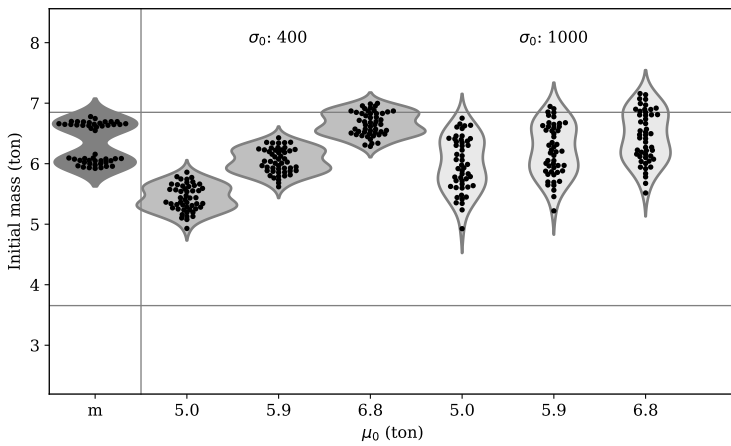


Figure 9.12: Prior test in relation to the true mass data. Horizontal lines refer to m_{oeW} and m_{mtow} .

9.6. Discussion

This chapter presents a set of principles, methods, and validations for Bayesian inference of aircraft mass. From the Bayesian inference point of view, the choice of prior distribution is very important. On the other hand, related factors such as wind and thrust can influence the accuracy of *mass observations*, which can propagate to the final estimations. These factors and their limitations are addressed in this section.

9.6.1. Priors in Bayesian inference

The Bayesian inference approach is based on the conditional probability theory where a certain prior (or “belief”) of the parameter is known. This leads the estimate to be biased to what the “belief” is. There are two ways to improve the final estimate: more measurements or, when fewer measurements are available, a better prior.

Because the number of *mass observations* is low, constructing a better prior is important. In practice, other than the minimum and maximum allowed weight, one can construct the prior based on direct or indirect indicators such as historical data, occupancy, and fuel reserves. For example, for short-haul flights from low-cost airlines, a highly-biased prior towards the maximum takeoff mass can be used. One can also look at traveler statistics to construct the priors according to seasons. In the cases where the types of flight are uncertain, based on our experiments, we select a μ_0 around 80% of the maximum takeoff mass, and a σ_0 that is around 25% of the mass difference between the maximum and empty mass.

Normally distributed priors are assumed in this chapter. The Normal distribution assumption is based on the results of [2, 3]. Furthermore, it serves the purpose of simplifying the computation. Nonetheless, a more informative prior with a skewed distribution can be used, depending on the level of confidence. The fixed variance reduces the computation and simplifies the closed-form solution. However, this is also a minor limitation that future studies would overcome. In other cases of the uncommon prior distribution, the posterior can still be computed using numerical methods.

9.6.2. The mass observations

For Bayesian inference, the more *mass observations* available, the better the quality of the result. In order to obtain more observations, the long duration of climb and descent can be better utilized. Most previous studies tend to consider the climb phase as an entire section for their estimation process. However, the climb and descent phases can be segmented further to create multiple measurements at different flight levels or at several continuous climb/descent segments. The method used in Equation 9.21 can be applied to obtain more observations to include in the Bayesian inference process.

It is worth noting that each flight in the dataset of this chapter contains the complete flight trajectory data that starts before takeoff and lasts until landing is completed. It may include some missed segments during trans-oceanic cruise where no data is available. The completeness of trajectory data is required to be able to use all different mass calculation methods at each flight phase. However, such a condition is not required in order to arrive at a set of estimates. A partial trajectory can be used in the same way, where the number of measurements is simply reduced.

9.6.3. Uncertainties

Since the mass is tightly linked to the thrust in the total energy model, the uncertainty in thrust always exists. This can result in a difference between the actual

mass and mass computed under maximum thrust profile assumption. On the other hand, using the total energy model, with observed aircraft kinematic states and calculated drag, there are multiple possible thrust and mass combinations. That is, both higher and lower thrust-mass combinations may satisfy the equation at the same time (with different levels of error).

It is common that an aircraft does not climb at the maximum thrust setting. The computation of several observations in this chapter requires knowledge of the reduced thrust ratio. This approach is similar to the study from [2]. The effect of thrust settings was studied in this chapter where the difference in mass estimation is found under fixed maximum and minimum thrust profile.

Compared to takeoff and climb, the descent situation is more complicated. Aircraft trajectories are subject to more procedures from air traffic controllers. Additionally, different types of continuous descent approaches may be used. Thus, the thrust settings can vary substantially among flights. Optimization according to Equation 9.21 may not yield the best thrust-mass setting to resemble reality. This effect can be seen in Figure 9.8 in the descent phase.

9.7. Conclusion

In this chapter, a Bayesian inference process for estimating aircraft initial mass was proposed. The maximum a posteriori probability (MAP) estimation was derived by combining mass calculations using data from different flight phases, in addition to the prior that is based on the empirical knowledge of the aircraft mass. The calculation of multiple initial masses was another major focus of this chapter. We tested different hypotheses of prior selections and their influence on the estimation results. It was also found that thrust setting and wind can act as important factors that influence mass estimation.

In this chapter, the Bayesian inference was applied to the multiple masses computed using the entire trajectory. It requires long observation of the flight trajectory and is best suited for the post-processing of the flight data. However, the requirement of extended observation of flight trajectory also poses the major limitation for this method being used in real-time cases.

To overcome this limitation, an improvement would be integrating the Bayesian estimation at flight dynamics level. This challenge will be addressed in the next chapter.

10

Mass estimation using Bayesian particle filtering

In the last chapter, we relied on Bayesian statistics to combine empirical and professional knowledge of mass computed at different flight phases to derive initial takeoff mass. However, Bayesian estimator become more powerful when applied to system dynamics. This chapter focuses on estimating aircraft mass and thrust settings using a recursive Bayesian method called particle filtering. The method is based on a non-linear state-space system derived from aircraft point-mass performance models. Using only aircraft surveillance data, flight states such as position, velocity, wind speed, and air temperature are collected and used for the estimation. Using the regularized Sample Importance Re-sampling particle filter, we are able to estimate the aircraft mass within 30 seconds once an aircraft is airborne. Using this short flight segment allows the assumption of constant mass and thrust settings in estimation. The segment at the start of the climb also represents the time when maximum thrust setting is most likely to occur. This study emphasizes an important aspect of the estimation problem, the observation noise modeling. Four observation noise models are proposed, which are all based on the native navigation accuracy parameters that have been obtained automatically from the surveillance data. Simulations and experiments are conducted to test the theoretical model. The results show that the particle filter is able to quantify uncertainties, as well as determine the noise limit for an accurate estimation.

This chapter is based on the following publications:

- 1) Sun, J., Blom, H.A., Ellerbroek, J. and Hoekstra, J.M., 2018. Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method. 8th International Conference on Research in Air Transportation [124].
- 2) Sun, J., Blom, H.A., Ellerbroek, J. and Hoekstra, J.M., 2018. Particle filter for aircraft mass estimation and uncertainty modeling. Transportation Research Part C: Emerging Technologies (Under review).

10.1. Introduction

Estimating aircraft mass using flight trajectory data has long been a topic of interest in ATM research. Aircraft mass is not only an important parameter for aircraft performance-related studies but is also a desired piece of knowledge for air traffic controllers in practice. Airlines, however, treat this data as confidential, and access is rarely granted to either researchers or air traffic controllers. In practice, mass information is, thus, not accessible nor actively used by the research community. Having an accurate estimate of aircraft mass at takeoff (or during the initial climb) can, for example, be beneficial for obtaining accurate trajectory and fuel consumption predictions during the flight.

Earlier studies that addressed this problem commonly involved deterministic methods based on the aircraft total energy model. A least-squares solution to this problem is proposed in [2], and an adaptive estimation method is presented in [118]. Using radar track data, these two approaches are compared in [3]. All these methods employ active radar data for estimation. With advancements in machine learning studies, supervised machine learning methods that require a well-established training set were also introduced to address the estimation problem in [4] and [19, 17]. With the growing adoption of Automatic Dependent Surveillance-Broadcast (ADS-B), estimation studies have been conducted around this open data source, such as [129].

The approaches by [2] and [17] have in common that they make use of regression analysis. [17] focuses on estimating aircraft mass during the takeoff roll, whereas [2] focuses on the climb phase. The advantage of regression analysis is that it avoids the need for an accurate model of the various noise levels. However, regression analysis requires the exploitation of model linearity. In Chapter 9, we have shown that ADS-B data availability makes it worthwhile to use Bayesian inference methods for the estimation of aircraft mass. The goal of this study is to extend this Bayesian inference approach such that it can incorporate non-linear aircraft evolution equations for the nonlinear Bayesian filtering of aircraft mass, based on ADS-B and Enhanced Mode-S surveillance data.

From most of the aforementioned studies, a strong link between aircraft mass and thrust setting is evident, and it is not possible to estimate one of the two parameters without some knowledge of the other. Several of these studies essentially addressed the estimation process as a minimization problem. Solutions were obtained using a form of least-squares fitting. Aircraft mass estimated under these conditions could be unrealistic and even outside plausible physical boundaries. Often, the main cause is related to uncertainty in the trajectory data, as well as uncertainty and non-linearity in the system model. Although the effect of noise is an important aspect in all inference processes, previous studies have not yet comprehensively studied its relation to mass estimation. Therefore, alongside estimating aircraft mass, observation noise modeling for the nonlinear aircraft equations is one of the main focuses of this study.

The problem of mass estimation by a (ground-based) observer can be considered as having to solve an inverse non-linear multi-state system using noisy observation data. To approach this complex system, in this study, we construct a detailed point-mass flight performance model with 11 system states and 8 ob-

servable states. A regularized Sample Importance Re-sampling (SIR) particle filter is introduced to estimate the aircraft states from noisy observations. In addition, four models of different levels of observation noise are constructed, which are based on ADS-B navigation accuracy standards. These models are used in particle filtering.

The remainder of this chapter is structured as follows. Section two describes the nonlinear point-mass flight dynamic system equations. Section three explains the theory of recursive Bayesian estimation and particle filtering. Section four develops models for the process and observation noise. Section five specifies the particle filter for aircraft mass estimation during initial takeoff. Section six presents the experiments conducted with simulated and real flight trajectories. Discussion and conclusion are presented in sections seven and eight.

10.2. The point-mass aircraft dynamic model

10.2.1. Aircraft state

This section describes the system equations for aircraft locomotion, implemented as a point-mass model. To illustrate, Figure 10.1 shows the observable states of an aircraft, primarily based on aircraft ADS-B broadcasts. The wind can be obtained from a weather prediction model or estimated based on Enhanced Mode-S data, which is described in Chapter 4. The left figure shows a horizontal projection of a trajectory, and the right figure a vertical projection. The heading and track are relative to the true north. For ease of computation, latitudes, longitudes, and altitudes are converted to a Cartesian coordinate system, with its origin at the first observed position. The point-mass model used in this study ignores the angle of attack and assumes no vertical wind.

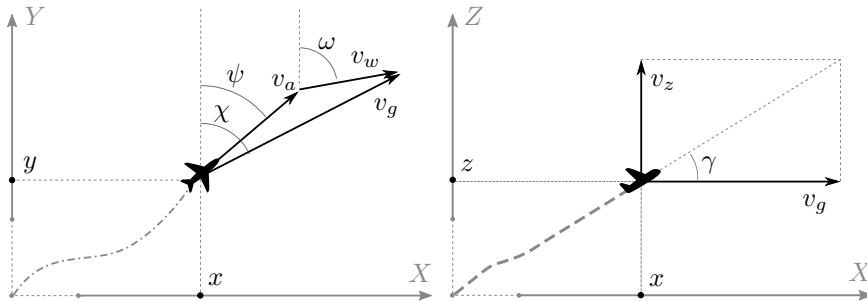


Figure 10.1: Observed states in the flight dynamic system

Denoting aircraft mass as m , thrust setting coefficient as δ_T , distance flown as \vec{s} , altitude as z , ground speed as \vec{v}_g , true airspeed as \vec{v}_a (in horizontal projection), vertical rate as v_z , wind speed as \vec{v}_w , and air temperature as τ , the system state \mathbf{x} vector can be written as:

$$\mathbf{x} = (m, \delta_T, \vec{s}, z, \vec{v}_a, v_z, \vec{v}_w, \tau) \tag{10.1}$$

The vector variables corresponding to the x and y axes are:

$$\begin{aligned}\vec{s} &= (x, y) \\ \vec{v}_a &= (v_{ax}, v_{ay}) \\ \vec{v}_w &= (v_{wx}, v_{wy})\end{aligned}\tag{10.2}$$

Additional angular parameters in Figure 10.1 are flight path angle γ , ground track χ , aircraft heading ψ , and wind direction ϕ . In addition, the measurement vector \mathbf{y} is represented as:

$$\mathbf{y} = (\tilde{s}, \tilde{z}, \tilde{v}_g, \tilde{v}_z, \tilde{v}_w, \tilde{\tau})\tag{10.3}$$

where \tilde{s} , \tilde{z} , \tilde{v}_g , and \tilde{v}_z are computed from ADS-B data. Wind \tilde{v}_w and temperature $\tilde{\tau}$ are obtained using Enhanced Mode-S data, which is described in Chapter 4.

10.2.2. State evolution

We want to model the system accurately in this point-mass performance model. The process noise is considered to be zero for states where exact process equations are known. These states are m_t , $\delta_{T,t}$, \vec{s}_t , z_t , and $\vec{v}_{a,t}$. For states in which a perfect process equation cannot be established accurately or are unknown, we use an auto-regressive model to describe their state evolution. These states are the vertical rate v_z , wind \vec{v}_w , and temperature τ . The complete state process equations can be described as follows:

$$m_t = m_{t-1}\tag{10.4}$$

$$\delta_{T,t} = \delta_{T,t-1}\tag{10.5}$$

$$\vec{s}_t = \vec{s}_{t-1} + (\vec{v}_{a,t-1} + \vec{v}_{w,t-1}) dt\tag{10.6}$$

$$z_t = z_{t-1} + v_{z,t-1} dt\tag{10.7}$$

$$\vec{v}_{a,t} = \vec{v}_{a,t-1} + \vec{a}_{t-1} dt\tag{10.8}$$

$$v_{z,t} = \alpha_{vz} v_{z,t-1} + \varepsilon_{vz}\tag{10.9}$$

$$\vec{v}_{w,t} = \vec{\alpha}_w \vec{v}_{w,t-1} + \vec{\varepsilon}_w\tag{10.10}$$

$$\tau_t = \alpha_\tau \tau_{t-1} + \varepsilon_\tau\tag{10.11}$$

where \vec{a} is the horizontal acceleration. State updates for v_z , \vec{v}_w , and τ are expressed using autoregressive (AR) models with lag $p = 1$. The coefficients of the AR models are constructed using historical data, which is described in Section 10.4.2. It is worth noting that since we are working with a short segment of the flight (within 30 seconds), constant mass (m) and thrust setting (δ_T) are assumed. Obviously, such an assumption cannot be applied to a longer period of the flight.

To compute the acceleration (\vec{a}), we need to consider the forces acting on the aircraft, which are thrust, drag, lift, and gravity. The equations used at each time step to compute the acceleration are listed as follows:

$$\vec{a}_t = (a_t, \psi_t) \quad (10.12)$$

$$\psi_t = \arctan_2(v_{ax,t}, v_{ay,t}) \quad (10.13)$$

$$a_t = \frac{1}{m}(\delta_{T,t}T_t - D_t) - g \frac{v_{z,t}}{v_{a,t}} \quad (10.14)$$

$$T_t = f_{\text{thr}}(z_t, v_{a,t}, v_{z,t}) \quad (10.15)$$

$$D_t = f_{\text{drag}}(m_t, v_t, z_t, C_{D0}, k, S) \quad (10.16)$$

where T and D are maximum thrust and drag forces expressed as functions of other state parameters. S is the aircraft wing surface. The thrust is calculated based on the empirical model proposed by [8]. The drag is calculated based on the drag polar of the aircraft modeled in Chapter 7.

In general, only the maximum thrust provided by the engines can be calculated from aircraft speed and altitude. During operation, aircraft engines only stay close to the maximum thrust for a short period of time at takeoff and at the start of the climb. Sometimes, a reduced thrust is applied, where the reduction of thrust is constrained by the actual aircraft mass.

In the particle filter, such a constraint is reflected at the initialization of the particles. The relation of mass and thrust setting is expressed using the following equation adapted from the Base of Aircraft Data (BADA) reduced climb power model from [103]¹:

$$\delta_{T_{\min}} = 1 - \Delta\delta_T \frac{m_{\max} - m}{m_{\max} - m_{\min}} \quad (10.17)$$

$$0 \leq \Delta\delta_T \leq 0.2 \quad (10.18)$$

Here, $\Delta\delta_T$ is the minimum derated thrust ratio, which is aircraft type specific and commonly small. m_{\max} and m_{\min} are considered as the maximum takeoff mass and minimum operational mass respectively.

Equation 10.17 shows the relationship between mass and the possibly reduced thrust ratio. It indicates that the higher the aircraft mass, the smaller the range in which the thrust can be reduced. Aircraft mass is assumed to be uniformly distributed between the operational empty mass and maximum takeoff mass. The thrust setting is assumed to be uniformly distributed between this minimum possible derated thrust ratio and one:

$$m \sim \mathcal{U}[m_{\text{ow}}, m_{\text{mtow}}] \quad (10.19)$$

$$\delta_T \sim \mathcal{U}[\delta_{T_{\min}}, 1] \quad (10.20)$$

¹Note that a similar equation in BADA is used to determine power setting. In this study, the form of the equation is adopted for the thrust reduction ratio.

10.3. Recursive Bayesian estimation

In recursive Bayesian estimation, an estimate of the system state is obtained using all previous observations. To this end, a recursive system with additive noise can be generalized by a discrete-time state model and observation model:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{f}(\mathbf{x}_{t-1}) + \boldsymbol{\nu}_{t-1} \\ \mathbf{y}_t &= \mathbf{h}(\mathbf{x}_t) + \mathbf{n}_t \end{aligned} \quad (10.21)$$

where \mathbf{x}_t represents the set of system states and \mathbf{y}_t the set of observations at time t ($t \in \mathbb{N}$), \mathbf{f} and \mathbf{h} represent the state transition function and observation function, $\boldsymbol{\nu}_t$ is the process noise, and \mathbf{n}_t is the observation noise. $\boldsymbol{\nu}_t$ and \mathbf{n}_t are assumed to be mutually independent sequences of independently and identically distributed variables. In the particular case of this study, we assume an additive Gaussian model for both process noise and observation noise as a common practice.

The goal of filtering is to compute the probability of the system state at a given time t , based on the observations from time 1 to time t , denoted as $p(\mathbf{x}_t|\mathbf{y}_{1:t})$:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \quad (10.22)$$

where the first part of the numerator, $p(\mathbf{y}_t|\mathbf{x}_t)$, is the observation probability that follows the observation function and observation noise \mathbf{n} from Equation 10.21:

$$p(\mathbf{y}_t|\mathbf{x}_t) = \mathcal{N}\{\mathbf{y}_t; \mathbf{h}(\mathbf{x}_t), \boldsymbol{\Sigma}_n\} \quad (10.23)$$

where $\boldsymbol{\Sigma}_n$ is the covariance of the observation noise. Because the system described in Equation 10.21 is a first-order Markov process, the second part $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ becomes:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \quad (10.24)$$

where the first term $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ on the right-hand side of the equation is the state transition probability. It follows the state transition function and the process noise model $\boldsymbol{\nu}$ in Equation 10.21:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}\{\mathbf{x}_t; \mathbf{f}(\mathbf{x}_{t-1}), \boldsymbol{\Sigma}_v\} \quad (10.25)$$

where $\boldsymbol{\Sigma}_v$ is the covariance of the process noise. Combining Equation 10.22 and Equation 10.24, the recursive form becomes:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \quad (10.26)$$

where the denominator of the fraction is a normalizing factor which does not need to be computed explicitly. The challenge is to resolve Equation 10.26 in a

recursive way using input data $\mathbf{y}_{1:t}$. This is where Sequential Monte Carlo (SMC) simulation-based particle filtering can be used.

10.3.1. Particle filtering

The specific Sequential Monte Carlo simulation method considered in this study is known as particle filtering [33]. A particle filter is a recursive Bayesian estimator based on importance sampling that computes the posterior density in a Monte Carlo fashion. It approximates the target distribution (denoted as $p(x)$) using a large number of samples (particles), drawn from a proposal distribution (denoted as $q(x)$) that updates it recursively.

To describe the SMC process, at time t , let $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$ be a set of particles that can represent the posterior density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, where \mathbf{x}_t^i is the i^{th} possible state with weight w_t^i . Henceforth, the posterior density is approximated with the empirical probability density:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^N w_t^i \delta(\mathbf{x}_t^i) \quad (10.27)$$

where $\delta(\cdot)$ is a Dirac delta function centered at \mathbf{x}_t^i and w^i is the normalized weight of a particle which satisfies $w^i = p(x^i)/q(x^i)$. The most important part of particle filtering is the weight updating. Sequentially, the particle weight w_t^i is updated in a recursive form. The solution is presented in Equation 10.28, as derived by [6] :

$$\begin{aligned} \mathbf{x}_t^i &\sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_{1:t}) \\ \tilde{w}_t^i &\propto w_{t-1}^i \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_{1:t})} \\ w_t^i &= \frac{\tilde{w}_t^i}{\sum_{i=1}^N \tilde{w}_t^i} \end{aligned} \quad (10.28)$$

At each iteration, the sum of all weights is normalized to one, as shown in the last part of equation 10.28. The posterior filtered density is approximated using Equation 10.27. We can compute the expected state values at each time step using the obtained particle weights:

$$\mathbb{E}[\mathbf{x}_t] = \sum_{i=1}^N \mathbf{x}_t^i w_t^i \quad (10.29)$$

There are different ways to choose the proposal distribution $q(x)$. A specific particle filter - *Sample Importance Re-sampling* (SIR) - is used for solving the problem of this study. The SIR particle filter uses the state transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$ as the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}^i, \mathbf{y}_k)$. Therefore, the particle update equations in Equation 10.28 can be simplified to:

$$\begin{aligned} \mathbf{x}_t^i &\sim p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i) \\ w_t^i &\propto w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_t^i) \end{aligned} \quad (10.30)$$

For a SIR particle filter, an optional additional re-sampling process at each iteration is included. The re-sampling is useful for systems which involve a large number of states (commonly more than 3) and prevents the impoverishment of particles. The re-sampling step generates a new set of particles based on the approximated $p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)$. Once particles are re-sampled, all weights are assigned to $1/N$.

The re-sampling step is essentially a redistribution of particles, which replaces low-weight particles with high-weight particles. The standard re-sampling algorithm is called *residual re-sampling*, which was proposed by [83]. Other forms of re-sampling such as *systematic re-sampling* and *stratified re-sampling* were summarized by [32]. In this study, the commonly used residual re-sampling is applied.

In summary, the process of flight state estimation using the particle filter is illustrated in Figure 10.2. In this figure, we can see that ADS-B and Enhanced Mode-S data are decoded and used as state observations that support the update of the particle filter. The system dynamic equations define the particle state evolution. In section 10.5, different elements of the SIR particle filtering algorithm are detailed. Before this can be implemented, we first need to introduce the system equations for describing the aircraft performance model.

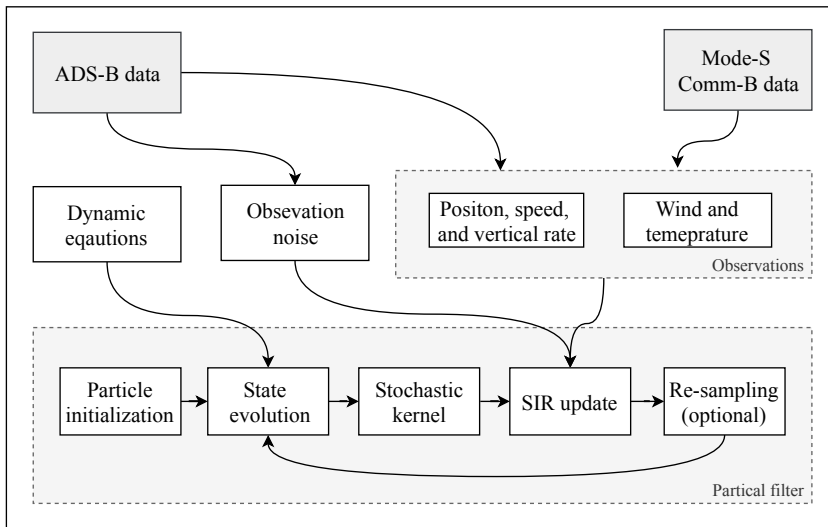


Figure 10.2: Estimation diagram including the source of observations and process of SIR particle filter. Further details are given in section 10.5.

10.3.2. Application for the aircraft mass and thrust estimation

The observable states are distance flown and ground speed in the horizontal and vertical direction, as well as the atmospheric conditions. The measurement vector is denoted as \mathbf{y} . Recall the measurement function from Equation 10.21:

$$\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{n}_t; \quad \mathbf{n}_t \sim \mathcal{N}(0, \Sigma_n) \quad (10.31)$$

where the state and measurement vectors at time t are:

$$\begin{aligned} \mathbf{x}_t &= [m_t, \delta_{T,t}, \vec{s}_t, z_t, \vec{v}_{a,t}, v_{z,t}, \vec{v}_{w,t}, \tau_t] \\ &= [m_t, \delta_{T,t}, (x_t, y_t), z_t, (v_{ax,t}, v_{ay,t}), v_{z,t}, (v_{wx,t}, v_{wy,t}), \tau_t] \\ \mathbf{y}_t &= [\tilde{s}_t, \tilde{z}_t, \vec{v}_{g,t}, \tilde{v}_{z,t}, \vec{v}_{w,t}, \tilde{\tau}_t] \\ &= [(\tilde{x}_t, \tilde{y}_t), \tilde{z}_t, (\tilde{v}_{gx,t}, \tilde{v}_{gy,t}), \tilde{v}_{z,t}, (\tilde{v}_{wx,t}, \tilde{v}_{wy,t}), \tilde{\tau}_t] \end{aligned} \quad (10.32)$$

The observation functions for all states can be written in detail as follows:

$$\tilde{s}_t = \vec{s}_t + \vec{n}_{s,t} \quad (10.33)$$

$$\tilde{z}_t = z_t + n_{z,t} \quad (10.34)$$

$$\vec{v}_{g,t} = \vec{v}_{a,t} + \vec{v}_{w,t} + \vec{n}_{vg,t} \quad (10.35)$$

$$\tilde{v}_{z,t} = v_{z,t} + n_{vz,t} \quad (10.36)$$

$$\vec{v}_{w,t} = \vec{v}_{w,t} + \vec{n}_{vw,t} \quad (10.37)$$

$$\tilde{\tau}_t = \tau_t + n_{\tau,t} \quad (10.38)$$

As previously mentioned, the observation noise is assumed to be uncorrelated additive Gaussian noise. The noise \mathbf{n} from Equation 10.21 can then be considered as a multi-variate Gaussian distribution. The diagonal covariance matrix Σ_n for these noise models is expressed as:

$$\Sigma_n = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_{vg}^2, \sigma_{vg}^2, \sigma_{vz}^2, \sigma_{vw}^2, \sigma_{vw}^2, \sigma_\tau^2) \quad (10.39)$$

where σ_{vg}^2 and σ_{vw}^2 apply to both x and y directions. Since the noise follows a multivariate normal distribution, the associated weight of the particles, $p(\mathbf{y}_t | \mathbf{x}_t^i)$, can be conveniently computed as follows:

$$\begin{aligned} w_t^i &\propto w_{t-1}^i p(\mathbf{y}_t | \mathbf{x}_t^i) \\ &\propto w_{t-1}^i \exp\left(-\frac{1}{2} [\mathbf{y}_t - \mathbf{h}(\mathbf{x}_t^i)]^T \Sigma^{-1} [\mathbf{y}_t - \mathbf{h}(\mathbf{x}_t^i)]\right) \end{aligned} \quad (10.40)$$

In this study, the particles are re-sampled at each time step and normalized to $1/N$. Hence, the weight calculation is simplified to:

$$w_t^i \propto \exp\left(-\frac{1}{2} [\mathbf{y}_t - \mathbf{h}(\mathbf{x}_t^i)]^T \Sigma^{-1} [\mathbf{y}_t - \mathbf{h}(\mathbf{x}_t^i)]\right) \quad (10.41)$$

10.4. Observation and process noise models

10.4.1. The observation noise models

Observation noise is closely related to sensor errors. For example, GPS errors affect position measurements (related to the state \hat{x}) and altimeter errors affect altitude measurements (\hat{z}). ADS-B transponders operate under regulations that define the minimum accuracy of sensors [111]. Different categories of uncertainty indicators are transmitted through ADS-B. In this study, these *Navigation Accuracy Categories* (NAC) are considered for the construction of observation noise models.

Table 10.1 illustrates the *Navigation Accuracy Category - velocity* (NACv), which defines the level of accuracy in terms of horizontal and vertical speed. The NACv indicator is broadcast along with the airborne velocity (message Type Code 18). HFOM and VFOM, short for the horizontal and vertical figure of merit, indicate the 95% confidence interval which corresponds to twice the standard deviation in the observation noise model. The HFOM and VFOM can therefore be used to obtain values for σ_{vg}^2 , σ_{va}^2 , and σ_{vz}^2 .

Table 10.1: Navigation Accuracy Category - velocity (NACv)

NACv	HFOM	VFOM
4	<0.3 m/s	<0.46 m/s
3	<1 m/s	<1.52 m/s
2	<3 m/s	<4.57 m/s
1	<10 m/s	<15.24 m/s
0	>10 m/s or unkown	>15.24 m/s or unkown

Similarly, Table 10.2 shows the *Navigation Accuracy Category - position* (NACp), which defines the level of accuracy in terms of horizontal and vertical position. For each NACp level, an Estimated Position Error (EPU) and a Vertical Estimated Position Error (VEPU) are defined. Similarly, they indicate a 95% confidence interval for the horizontal and vertical bounds. From these values, we can obtain σ_x^2 , σ_y^2 , and σ_z^2 for the observation noise model.

Table 10.2: Navigation Integrity Category - position (NACp)

NACp	EPU	VEPU
11	<3 m	<4 m
10	<10 m	<15 m
9	<30 m	<45 m
8	<0.05 NM	n/a
7 - 0	not used in this study	

Four sets of observation noise models are proposed based on the ADS-B specifications, each corresponding to a different combination of NACp and NACv (see

Table 10.3). These four models are also the foundation for the experiments carried out later in this study.

Table 10.3: Noise models

Noise model	NACp	NACv
Σ_{n1}	11	4
Σ_{n2}	10	3
Σ_{n3}	9	2
Σ_{n4}	8	1

The uncertainties in the wind and temperature measurements depend on the Enhanced Mode-S data that was used to generate weather information. The variances for wind and temperature are set to be 2.5^2 and 1^2 for the noise model Σ_{n3} . These two values are based on the variance of the model results obtained in Chapter 4. For other noise models, they are adjusted with respect to the scale of the velocity uncertainty. The exact values of all four noise models for Equation 10.39 used in the experiments are listed as follows:

$$\begin{aligned}
 \Sigma_{n1} &= \text{diag}(1.5^2, 1.5^2, 2^2, 0.15^2, 0.15^2, 0.23^2, 0.2^2, 0.2^2, 0.1^2) \\
 \Sigma_{n2} &= \text{diag}(5^2, 5^2, 7.5^2, 0.5^2, 0.5^2, 0.76^2, 0.8^2, 0.8^2, 0.3^2) \\
 \Sigma_{n3} &= \text{diag}(15^2, 15^2, 22.5^2, 1.5^2, 1.5^2, 2.28^2, 2.5^2, 2.5^2, 1^2) \\
 \Sigma_{n4} &= \text{diag}(48^2, 48^2, 68^2, 5^2, 5^2, 7.62^2, 7.5^2, 7.5^2, 3^2)
 \end{aligned} \tag{10.42}$$

10.4.2. Process equations for vertical rate, wind, and temperature

As described in Equations 10.9 and 10.10, vertical rate, wind, and temperature along the climb path are modeled as autoregressive (AR) models. This is because the underlying process model is unknown, and these time series exhibit a strong correlation between consecutive data points. For simplification, we will use a first-order AR model (AR1) to describe state evolution. In general, the AR1 model (without the bias term) can be expressed in the following form:

$$\begin{aligned}
 x_t &= \alpha x_{t-1} + \varepsilon \\
 \varepsilon &\sim \mathcal{N}(0, \sigma^2)
 \end{aligned} \tag{10.43}$$

where α is the model parameter and ε is the white noise with variance of σ^2 . Using real flight data, these parameters can be estimated. For a given flight, α and σ can be estimated in Equation 10.44 using the least-squares regression :

$$\begin{aligned}
 \alpha &= \frac{\sum_{t=1}^n \tilde{x}_{t-1} \tilde{x}_t}{\sum_{t=1}^n \tilde{x}_{t-1}^2} \\
 \sigma^2 &= \text{Var}(x_t - \alpha \tilde{x}_{t-1})
 \end{aligned} \tag{10.44}$$

where \tilde{x} represent the wind, temperature, or vertical rate data extracted from real flights. This real flight data source should be different from the data source $y_{1:t}$ that is used for mass estimation by the particle filter.

The reason that wind and temperature can be modeled in this way is that locally they tend to be homogeneously distributed with some degree of variability. There is often a gradual increase in wind magnitude and a lapse of temperature with increasing altitude. On the other hand, the vertical rate is often a controlled variable in point mass flight models. Without more information from the aircraft, an AR model can capture the change in vertical rate from the observer's point of view.

To determine representative values for α and σ^2 , ADS-B and Enhanced Mode-S data was collected for climbing flights in a period of one month (around 10,000 flights) using our receiver. α and σ^2 were computed for each state parameter (see Figure 10.3).

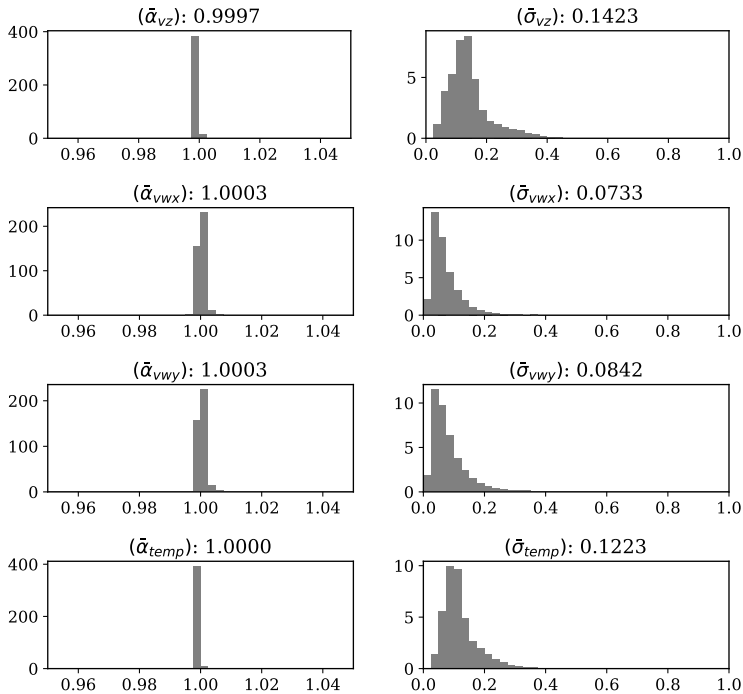


Figure 10.3: First order AR model parameters for v_z , v_{wx} , and v_{wy}

The mean value of each parameter is then used for the AR models. Table 10.4 summarizes the obtained parameters for v_z , \vec{v}_w , and τ (used in Equation 10.9 to 10.11).

Table 10.4: Parameter summary for the first order AR models

	α	σ
v_z	0.9997	0.1423
v_{wx}	1.0003	0.0733
v_{wy}	1.0003	0.0842
τ	1.0000	0.1223

10.5. The algorithm

The detailed algorithm is specified in Algorithm 1, which is adapted from [12]. The most essential part of the algorithm is the *SIR update*, where the weights of the particles are updated based on the observations. This step reflects Equation 10.40 that is described in the previous section. In addition, the *state evolution* is defined by the aircraft performance that is described by Equations 10.4 to 10.11.

Algorithm 1 The SIR particle filter: $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \rightarrow p(\mathbf{x}_t|\mathbf{y}_{1:t})$

- 1: **for** $i=1$ to N **do** ▷ initialize particles
 - 2: draw $m_0^i \sim \mathcal{U}(m_{oew}, m_{mtow})$
 - 3: draw $\delta_{T_0}^i \sim \mathcal{U}\left(1 - \delta_{T_{\min}}(m_{mtow} - m_0^i)/(m_{mtow} - m_{oew}), 1\right)$
 - 4: draw $(\vec{s}, z_0^i, \vec{v}_{a0}^i, v_{z0}^i, \vec{v}_{w0}^i, \tau_0^i) \sim \mathcal{N}(\vec{x}_0, \Sigma_n)$
 - 5: $\mathbf{x}_0^i := (m_0^i, \delta_{T_0}^i, \vec{s}_0^i, z_0^i, \vec{v}_{a0}^i, v_{z0}^i, \vec{v}_{w0}^i, \tau_0^i)$
 - 6: $w_0^i := 1/N$
 - 7: let $\vec{x}_0 := h^{-1}(\mathbf{y}_0)$
 - 8: let Σ_n be observation noise covariance
 - 9:
 - 10: **for** $t=t_0$ to t_N **do**
 - 11: **for** $i=1$ to N **do** ▷ SIR update
 - 12: $w_t^i := w_{t-1}^i p(\mathbf{y}_t|\mathbf{x}_t^i)$
 - 13: $\bar{w}_t^i := w_t^i / \sum_{i=1}^N w_t^i$
 - 14: **for** $i=1$ to N **do** ▷ re-sampling
 - 15: draw $\vec{x}_t^i \sim \sum_{i=1}^N \bar{w}_t^i \delta(\mathbf{x} - \mathbf{x}_t^i)$
 - 16: $w_t^i := 1/N$
 - 17: **for** $i=1$ to N **do** ▷ state evolution
 - 18: draw $\nu_t \sim \mathcal{N}(0, \Sigma_\varepsilon)$
 - 19: $\mathbf{x}_{t+1}^i := f(\vec{x}_t^i) + \nu_t$
 - 20: **for** $i=1$ to N **do** ▷ apply kernel
 - 21: draw $dm^i \sim \mathcal{N}(0, \sigma_{k,m}^2)$ as kernel noise for m
 - 22: $m_t^i := m_{t-1}^i + dm^i$
 - 23: draw $d\delta_{T^i} \sim \mathcal{N}(0, \sigma_{k,\delta_T}^2)$ as kernel noise for δ_T
 - 24: $\delta_{T,t}^i := \delta_{T,t-1}^i + d\delta_{T^i}$
 - 25: draw $d\psi^i \sim \mathcal{N}\{0, \sigma_{k,\psi}^2\}$ as kernel noise for heading
 - 26: compute $d\vec{v}_a^i$ from $d\psi^i$
 - 27: $\vec{v}_{a,t}^i := \vec{v}_{a,t-1}^i + d\vec{v}_a^i$
-

In this study, we use the regularized SIR particle filter proposed in [98]. Specifically, a stochastic kernel filter (KF) is used as an effective technique to prevent degeneration and impoverishment of the set of particles that are maintained by

the particle filter. The stochastic kernel is applied to the states of interest (m and δ_T), as well as a hidden state, aircraft heading (ψ). It adds a small random Gaussian noise to the state variables of all particles after re-sampling. For m and δ_T , the kernel helps to prevent the degeneration of particles and maintain a local diversity of values. In the case of aircraft heading ψ , the kernel is essential for the functioning of the particle filter to track small changes in heading. These kernels are defined as:

$$\begin{aligned} k_m &\sim \mathcal{N}(0, \sigma_{k,m}^2) \\ k_{\delta_T} &\sim \mathcal{N}(0, \sigma_{k,\delta_T}^2) \\ k_\psi &\sim \mathcal{N}(0, \sigma_{k,\psi}^2) \end{aligned} \quad (10.45)$$

In this study, the $\sigma_{k,m}$ is chosen to be 0.5% of the maximum m range, which is $0.005 \times (m_{\text{mtow}} - m_{\text{ow}})$. The σ_{k,δ_T} is chosen to be 0.5% of the maximum δ_T range, which is less than 0.005×0.2 . The choice of $\sigma_{k,\psi}$ is 2 degrees.

From a nonlinear system point of view, this noise has the same effect as the process noise, since they both introduce uncertainty to the state evolution (see Equation 10.21). However, in the context of particle filtering, this approach should be considered as a form of regularization [98]. This is a simple but effective technique to prevent the degeneration and impoverishment of particles which would lead to run-time execution errors in practice. To accomplish this, the kernel produces only a tiny variation of the states. Overall, it also compensates, to some extent, the limitation of the constant mass and thrust assumption.

10.6. Experiments

This section describes three different experiments that are designed to test the proposed method. In the first experiment, a simulated flight is generated with a known mass and thrust setting. Four rounds of estimations are carried out under four different noise models. The simulation is undertaken to ensure the validity of the particle filter based on the proposed system equations. In the second experiment, a real flight is chosen, and the estimation is undertaken with the same four noise models. This is to ensure that the results from a real flight are in line with the simulation results. The number of particles is set to one million. This choice reflects a balance between accuracy and computational speed. The third experiment is based on a number of flights with a Cessna Citation II laboratory aircraft with known mass.

10.6.1. Experiment I: The simulation of a Boeing B737 climb

In this experiment, the aircraft takeoff mass m is set to be 60,000 kg, and the thrust setting δ_T is set to be 0.96. The actual observation noise for the simulated trajectory is Σ_{n2} . The wind is absent in the simulation, and the 20-second initial climb is performed with a constant climb rate. Using the simulated trajectory, particle filters with four distinct noise levels are applied to estimate the mass and thrust settings.

In Figure 10.4, the convergence of particles under assumed observation noise $\Sigma_{n,2}$ (the same as in the simulation) is illustrated. In each convergence plot, the solid line represents the true state value, the black dots are simulated observations ², and the gray area is bounded by the minimum and maximum state values at each iteration. Under this condition, it can be seen that the mass and thrust settings nicely converge to their true values. On the other hand, convergence is less important for other states. The main goal is to be able to track the changes in these states during the filtering process.

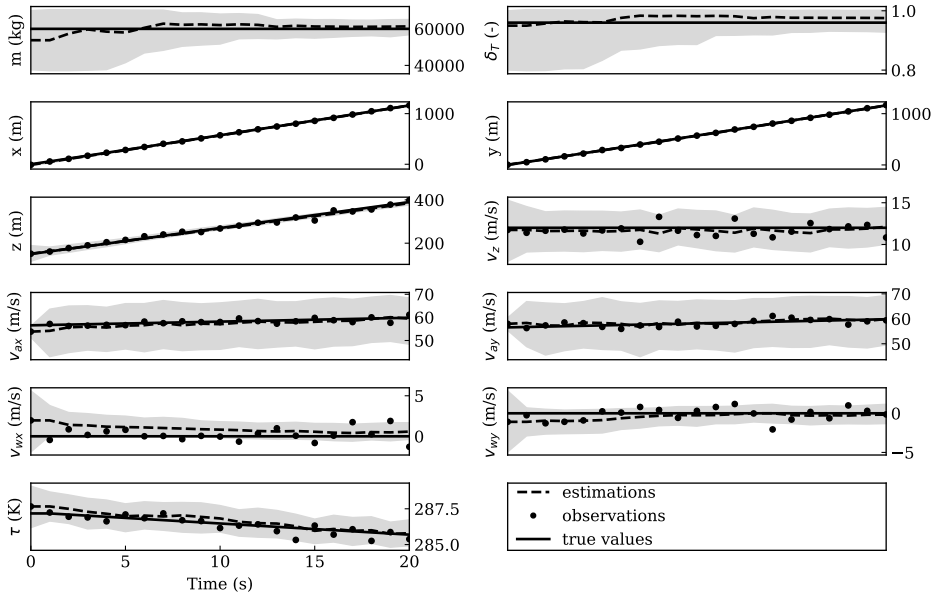


Figure 10.4: Convergence of the particle filter states (noise level $\Sigma_{n,2}$)

In Figure 10.5, the convergence of m and δ_T under all four observation noise levels is shown. In each plot, the left-hand side distribution corresponds to mass, and the right-hand side distribution represents the thrust setting. It is apparent that with increasing assumed observation noise, the uncertainty of the final results increases.

The Figure also shows that the estimates can become trapped in incorrect states when the assumed noise is lower than actual noise (in the first two plots). When the assumed noise level is much higher than the actual noise, the uncertainty in the final estimate becomes large (as demonstrated in the last two plots). These observations are consistent with the characteristics of a particle filter.

10.6.2. Experiment II: A real Boeing B737 flight

In this second experiment, an actual climb trajectory of a Boeing 737-700 is used to demonstrate the application of the SIR particle filter to real flights. The trajectory

²Note that the airspeed is calculated as the difference between ground speed and wind speed.

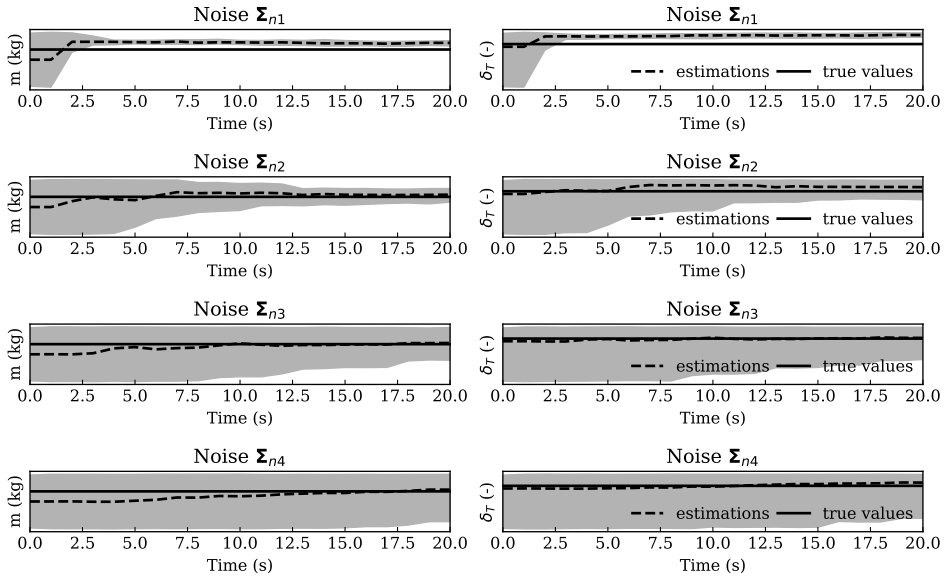


Figure 10.5: Final distribution of m and δ_T under different observation noise levels

data is gathered from ADS-B and decoded. The wind data are computed as the combination of ADS-B and Enhanced Mode-S data.

The trajectory and convergence of the particles under noise level Σ_{n2} are shown in Figure 10.6. Similar to the previous simulated case, the black dots represent actual observed data, while the solid lines represent weighted average state values of particles at each iteration. In these plots, not only the mass and thrust setting exhibit convergence, but we can also see that the changes in airspeed, vertical rate, wind, and temperature are tracked by the particle filter. The particle filter can cope with missing data points, as shown in the plots.

Figure 10.7 illustrates the convergence of m and δ_T under different noise levels, showing uncertainties with similar characteristics to the previous simulated case.

The use of these simulated and real flights demonstrates the applicability of the SIR particle filter for mass and thrust setting estimations. However, these experiments are based on a single sequence of filtering under each noise level. We can observe that the estimation uncertainty increases with increased observation noise in both experiments, but this is not yet sufficient to quantify the estimation uncertainty.

10.6.3. Experiment III: Demonstration with Cessna C550 flights

A set of 50 real flights with known mass is used to test the proposed system model and the particle filter. These flights were carried out by a Cessna Citation II (C550) aircraft that is operated by TU Delft for student practicals. The mass of the aircraft is obtained accurately by weighing all passengers and measuring the exact amount

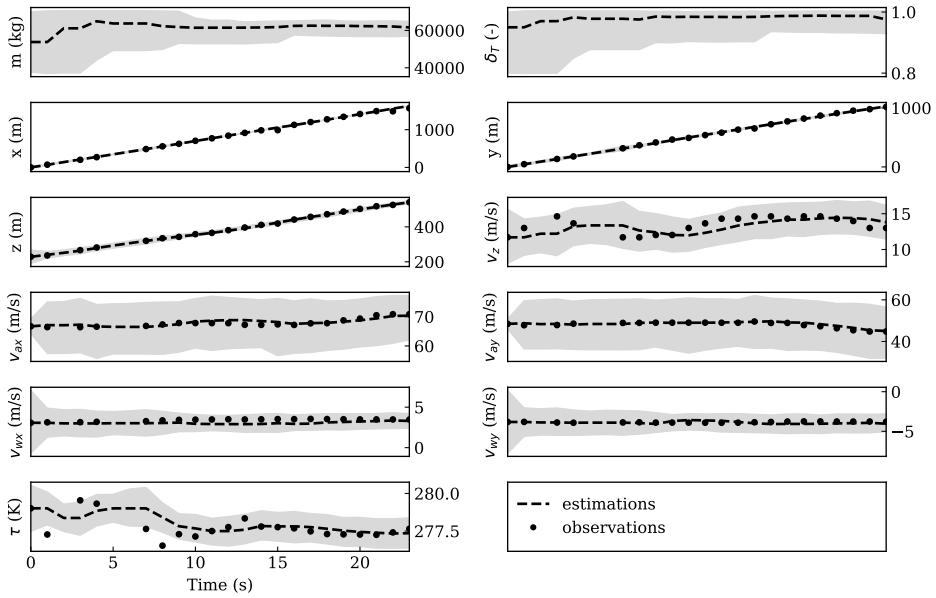


Figure 10.6: Convergence of the particle filter states (noise level Σ_{n2})

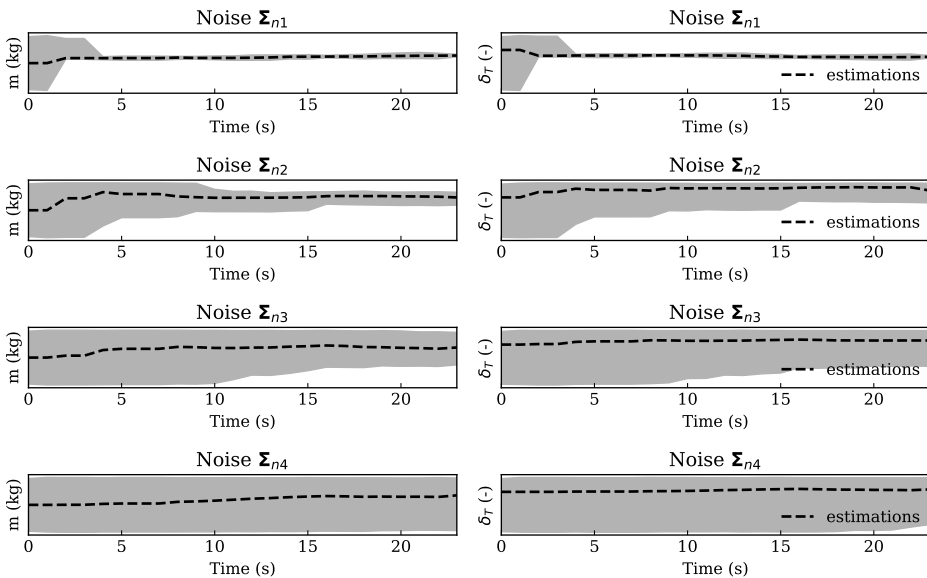


Figure 10.7: Final distribution of m and δ_r under different observation noise levels

of fuel on board prior to the start of each flight.

Although accurate FMS trajectory data is available, we used the collected

Mode-S data for the experiment. In this way, the validation is closer to real cases. In addition, NACp and NACv values are decoded from raw ADS-B data to automatically select the observation noise model. In the validation, the $\Delta\delta_T$ is assumed to be 0.1 for this aircraft.

To obtain a stable estimation, a 20-second straight climb at the start of the climb is extracted. Based on the same short segment of each trajectory, the particle filter is executed 30 times independently.³ The averaged final results are shown in Figure 10.8. Estimated mass (indicated with crosses) is plotted against the real mass of the aircraft (indicated with circles) in the top plot of the figure. The estimated thrust setting is plotted in the bottom plot.

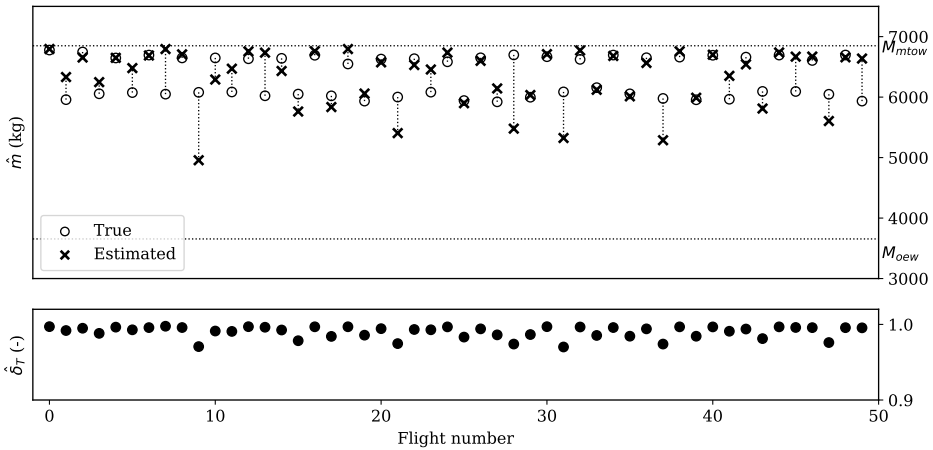


Figure 10.8: Estimated results for all validation flights

The estimation errors are plotted in Figure 10.9. The plots on the left-hand side show the estimation errors ($\Delta m/m_{\text{true}}$), where the mean and median error are both around -0.5% of the true mass. There is a very small but not significant negative bias in the results. On the right hand side, the absolute estimation errors ($|\Delta m|/m_{\text{true}}$) are computed and shown. As a result, the mean absolute error (MAE) is found to be 4.3%, while the median of the absolute error is 2.6%.

10.7. Discussion

This section presents sensitivity analyses of the fundamental aspects of aircraft mass estimation. This yields insight into the recommendations on choosing a noise model and practical particle filtering.

10.7.1. Estimation uncertainty

A large number of additional simulations are performed to study the uncertainty of the estimation. 500 rounds of estimations under each defined noise model

³This is similar to using 30 million particles in one round. The implication of this will be discussed later on.

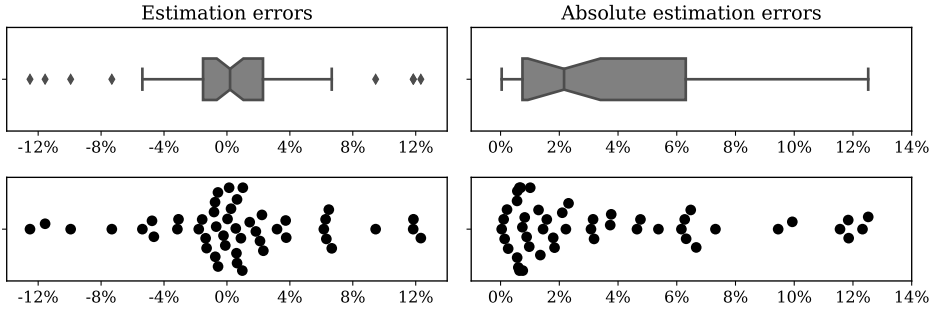


Figure 10.9: Distribution of estimation errors. Circles represent the actual mass, while the crosses represent the estimated mass.

are performed for the simulated flight. The simulated Boeing B737-700 flights are generated with a mass of 60,000 kg, a thrust setting of 0.96, and a small simulation noise of $\Sigma_{n1}/4$.

First, we want to illustrate the estimation accuracy using the particle filter. In Figure 10.10, the distributions of estimates at the final time step of all rounds are shown. The results are grouped by observation noise. On the left-hand side of the figure, the results of m are indicated, with the Y-axis ranging from m_{ow} to m_{mtow} . The horizontal black line indicates the actual mass (60,000 kg) used in the simulations. On the right-hand side, the thrust settings are plotted in the same fashion. We can see that the particle filter yields a high level of accuracy with the simulated trajectory.

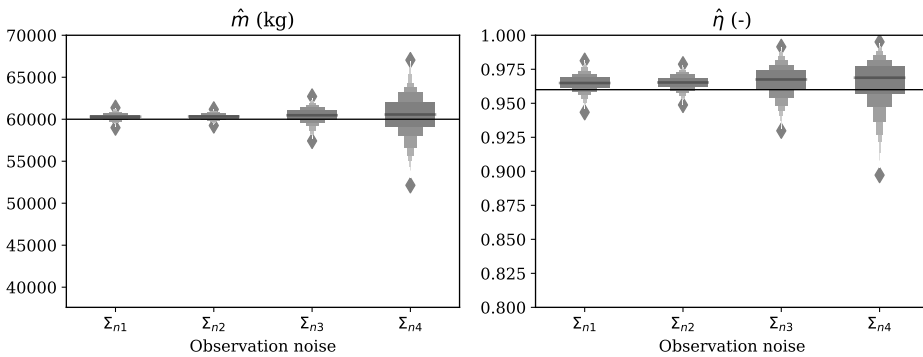


Figure 10.10: The distributions of estimated mass and thrust setting based on 500 rounds of simulated trajectory under different observation noise models

Next, we want to study the uncertainty of the estimation. By computing the standard deviations of the mass and thrust values from all particles at the end of each run, we are able to visualize estimation uncertainty under different noise models. On the left-hand side of Figure 10.11, distributions of particle mass standard deviation are shown. It can be observed that with increasing observation

noise, uncertainty also increases. The right-hand side of the plot indicates the same trends for the thrust setting.

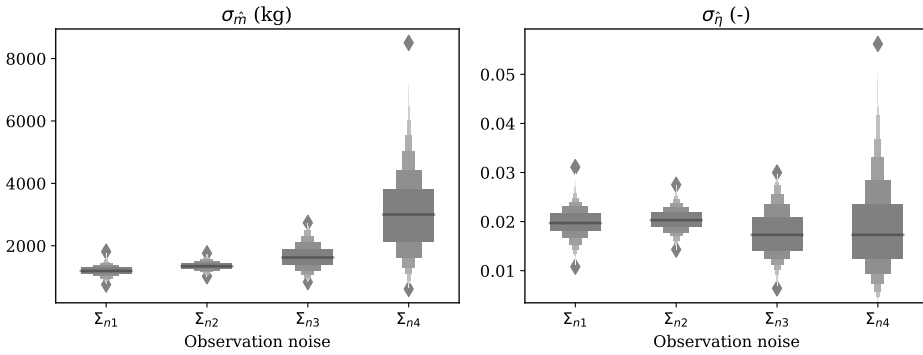


Figure 10.11: Standard deviation of particles states on mass and thrust setting among 500 rounds of simulations under different observation noise models

Figure 10.10 and Figure 10.11 display similarities, but show different aspects of the estimation result. The earlier figure shows the final estimates by using the mean state values from particles at last time steps of all runs. In the latter figure, the spread of particles by the measurement of standard deviation is shown.

To quantify the uncertainty under different noise models, twice the standard deviation is used as an indicator. This is then translated into the percentage of the aircraft maximum takeoff mass and thrust settings, which are shown in Table 10.5.

Table 10.5: Uncertainties related to observation noise obtained using 500 round of estimations with simulated trajectories under different observation noise levels

Noise	$2\bar{\sigma}_{\hat{m}}$	% of m_{mtow}	$2\bar{\sigma}_{\hat{\delta}_T}$	%
Σ_{n1}	2403 kg	3.0%	0.040	4.0%
Σ_{n2}	2694 kg	3.4%	0.041	4.1%
Σ_{n3}	3307 kg	4.2%	0.035	3.5%
Σ_{n4}	6079 kg	7.7%	0.037	3.7%

Similarly, the accuracy of the estimation can also be obtained by comparing the estimation results with the input mass of these simulated trajectories. The results are listed in Table 10.6.

Table 10.6: Mean accuracy of the estimation based on 500 round of estimations with simulated trajectories under different observation noise levels

Noise	$2\bar{\sigma}_{\hat{m}}$	% of m_{mtow}	$2\bar{\sigma}_{\hat{\delta}_T}$	%
Σ_{n1}	304 kg	0.4%	0.005	0.5%
Σ_{n2}	308 kg	0.4%	0.005	0.5%
Σ_{n3}	437 kg	0.6%	0.006	0.6%
Σ_{n4}	509 kg	0.6%	0.006	0.6%

However, we have to be cautious when extrapolating the accuracy listed in Table 10.6. It is based on simulated trajectories using the exact model created for estimation. In practice, the result from Section 10.6.3 better reflects the accuracy of the estimations. In addition, we also see that the uncertainty results are obtained based on one aircraft type. However, the percentage values from this table can still be an important indicator to quantify the uncertainties of estimations. It can be used as an indicator for the missing uncertainty component for other estimation methods that are based on ADS-B surveillance data.

In order to ensure the functioning of the SIR particle filter, we are required to have an accurate understanding of the observation noise. If the assumed noise is much larger than the actual observation noise level, the states of particles would have large variances. This, in turn, would produce estimations with greater uncertainties. Examples of this condition can be seen in the last plots of Figure 10.5 and 10.7. On the other hand, if the assumed noise is much smaller than the actual noise, the particles would converge to a set of incorrect states. With the assumed noise model, all other possible states have extremely low probabilities. As such, the convergence to the real states becomes difficult. Under such a noise level assumption, the estimator would result in high confidence (low variance) but with possibly incorrect (biased) estimates. Examples can be seen in the first plots of Figure 10.5 and 10.7.

10.7.2. Choice of observation noise model

As a general rule, the observation noise model (the covariance matrix Σ_{n*}) should reflect the actual noise in the measurement data. As such, the particles would be able to track the changes in states accurately. However, in real life, actual noise from the data is not often known directly.

One way to find out the noise level in ADS-B data is to decode the NACp and NACv values as described in Section 10.4.1, and, then, use the corresponding noise models proposed in Table 10.3 to identify the proper noise model. This is only possible, however, when access to raw ADS-B data is available. When using other data sources, the proper noise model can be constructed using the vector autoregression (VAR) method [85, p15].

As a rule of thumb, when raw NACp and NACv values are not available, the noise models of Σ_{n2} or Σ_{n3} are generally good starting points. This conclusion is based on our observation that the majority of the ADS-B messages meet the accuracy levels of NACp 9/10 and NACv 3/2.

10.7.3. Influence of variation in mass and thrust setting

Based on previous experiments, the thrust setting (δ_T) does not always converge. The same trend is not encountered with mass estimations unless a larger noise model has been introduced. To understand the reason for this, the variation of these two parameters and their influence on the (simulated) flight trajectory is studied. Two sets of simple simulations are conducted.

In the first set of simulations, the aircraft mass is fixed at $(m_{\text{ow}} + m_{\text{mtow}})/2$, and the thrust setting varies from 0.85 to 1. Results are shown on the left two plots

of Figure 10.12. In the second set of simulations, the thrust setting is kept at 0.9, and aircraft mass varies from m_{ow} to m_{tow} . Results are shown on the right-hand side of Figure 10.12. Here, only the horizontal flight distances and speed profiles are illustrated.

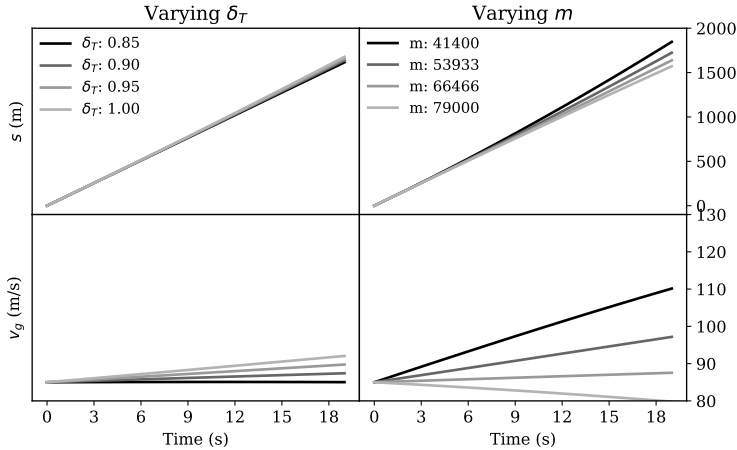


Figure 10.12: Varying δ_T versus varying m

The influence of different thrust settings on the flight trajectory is much smaller than the influence of differences in mass. This is shown in both distance traveled and the speed profile. Next, noise is added to the simulation. For simplification, only the resulting speed profiles are shown in Figure 10.13.

Here, we can see that the trajectories become hard to distinguish for thrust settings when noise is larger than Σ_{n2} . In the case of mass, the trajectories only become hard to distinguish when noise reaches Σ_{n3} . This offers an intuitive demonstration of the noise effects on trajectories.

From a statistical point of view, we can analyze the relationships between mass and thrust settings through particle distributions when convergence is reached. In Figure 10.14, the resulting distributions of m and their δ_T are shown. The particles are grouped by the final mass distribution. Here, m values are separated in small bins, and corresponding δ_T values are shown in the right-hand plots. m and δ_T from the same groups are plotted with the same shades of gray. The diversity of the particles disappears with noise level Σ_{n1} . With noise level Σ_{n4} , a large range of estimates is considered possible. With noise level Σ_{n2} and Σ_{n3} , the true values of the states are represented. The corresponding mass and thrust setting values are clear in this visual representation.

The results from Figure 10.5 can be confirmed with the distributions of the particles. The estimation of mass can only be possible when the noise is smaller than or equal to Σ_{n3} . The equivalent NACp and NACv values are 9 and 2 respectively. With the first two plots of Figure 10.5, we can also observe that when the assumed observation noise (Σ_{n1}) is smaller than the actual observation noise (Σ_{n2}), the estimation results are biased.

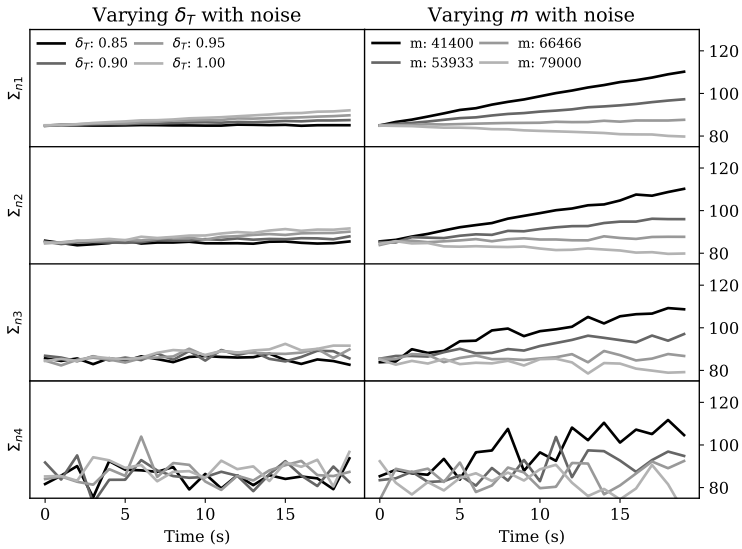


Figure 10.13: Simulated speeds with different noise models

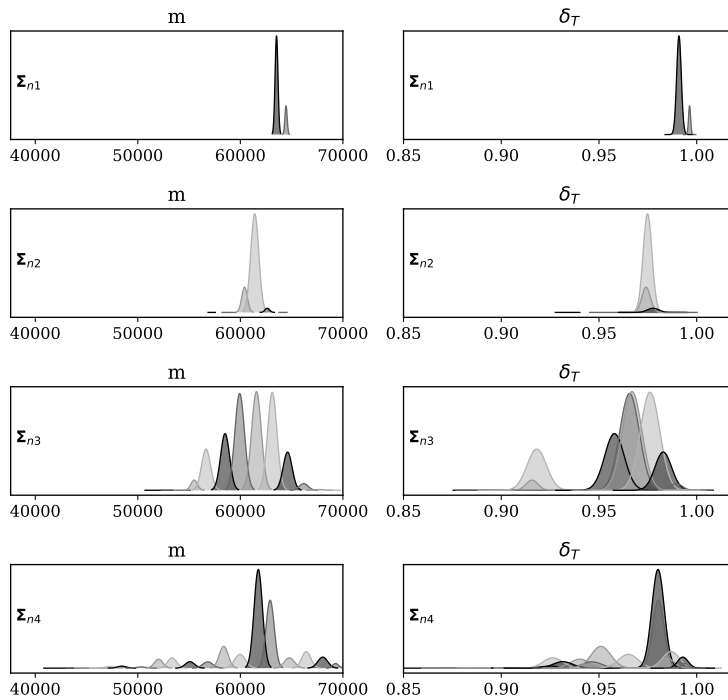


Figure 10.14: Relationship between deviations in estimated thrust setting and mass

10.8. Conclusion

In this study, steps were taken to construct a comprehensive model and estimation method to derive aircraft mass and thrust setting solely using ADS-B and Enhanced Mode-S observations. The complexity of such a problem was discussed at the start of this study. In summary, the difficulty of the estimation lied in solving an inverse non-linear system that consists of eleven states.

In order to address this challenging nonlinear estimation problem, this study developed a stochastic recursive Bayesian approach using a particle filter. The recursive Bayesian approach not only provided estimates but also allowed for quantifying uncertainties under different noise levels. If realistic observation noise levels could be identified, the proposed particle filter was able to estimate aircraft mass and thrust setting when noise constraints were satisfied. Simulated, real, and validation flights were used to test the method proposed in this study, specifically to test the method, define the uncertainty, and validate the accuracy of the method. For validated datasets, this approach yielded a mean absolute error of around 4.3% of the true mass on the validation dataset. Finally, the convergence speeds were swift. In most cases, estimation could be obtained using around 30 seconds of the flight data at the start of the climb.

In this study, we have selected the flight segment where the thrust setting is most likely to be at the maximum, which was in the initial climb phase. The maximum thrust was computed using the model proposed by [8]. Even with the reduction setting (δ_T) modeled, we still could not completely mitigate the uncertainty in the underlying thrust model. However, for the errors of the particle filter, estimated mass and thrust setting during initial takeoff a strong positive correlation is known, i.e., a too high assumed thrust setting leads to a too high estimated aircraft mass. When testing the particle filter with reference flights from a Cessna Citation II, we were able to increase the accuracy of the estimation with better knowledge of the thrust setting. Changing from 0.85 to 0.95 allowed the particle filter to converge closer to the real mass. However, this assumption was specific to our test flight. For common commercial airlines, we still recommend using the value of 0.85 unless there is evidence indicating otherwise.

By linking the ADS-B native uncertainty reporting system with the observation noise, we were able to construct realistic observation noise models. At the same time, we were also able to automatically select the appropriate noise model for each flight using raw ADS-B data when available. Knowing the effect of these uncertainties by using a particle filter can be beneficial for future ADS-B-related studies. Better understanding the noise model in surveillance data is the key to improving the accuracy of the dynamic model used for estimation. This can be an interesting area for follow-on studies.

In the current model, the bank angle was left out of the equations, which was, of course, a simplification. The consequence was that we were only able to use non-turning flight segments for the estimations. Introducing an additional parameter for the bank angle could enable the estimation to use data from turning trajectories. However, this may increase the complexity of the model, as well as the estimation uncertainty.

A simplified form of the observation noise was assumed in this study, where the

noise for different observable states was assumed to be uncorrelated. However, this may not be true in real situations. Auto-correlation of errors may exist as suggested in [95]. This can be considered in the model to improve the estimation if such knowledge can be discovered. The noise models that were defined in this study correspond to Version 1 and Version 2 of the ADS-B implementation. For older ADS-B Version 0 transponders (equipped on older aircraft types), the noise should be defined based on its Navigation uncertainty Categories (NUC). The NUC levels are less refined than NAC in Version 1 and 2. Future implementations of our method should take this into consideration.

The proposed Bayesian filtering framework has shown its potential for parameter estimation. The designed particle filter can also be a good candidate for aircraft trajectory state filtering. When mass can be estimated at the start of a flight, other state parameters along the entire flight can be better estimated or predicted. This will bring more insight to, for example, the estimate of fuel consumption and trajectory prediction during the rest of the flight. Now that the estimation of mass using a particle filter has been demonstrated, one of the logical follow-on steps is to investigate applications of other non-linear estimation approaches.

Finally, to allow open access to the method developed in this study, the source code of the particle filter is published alongside this study.⁴

⁴Available at: <https://github.com/junzis/acsmc>

11

Conclusions and recommendations

This dissertation started with the discussion of an important missing element in air transportation studies, namely the lack of an open performance model. Throughout the dissertation, a research path has been developed in order to use open data to model and estimate aircraft performance parameters. This last chapter of the dissertation revisits the research questions proposed at the beginning of the dissertation and the associated conclusions reached based on our studies. Limitations of the overall research and recommendations for future studies are also indicated in this chapter.

11.1. Overview

The first part of the dissertation discussed how to extract useful information for aircraft performance studies from surveillance data. Next, different components of the performance model have been studied. These components include aviation weather, a kinematic model, and a thrust and fuel model, as well as an aerodynamic model. In the third part of the dissertation, the estimations of performance parameters of individual flights were addressed. Open data and models proposed in the earlier chapters were used for these estimations. In general, each chapter addressed a research question proposed in the introduction chapter.

11.2. Research questions and conclusions

As defined in the introduction, the main goal of this dissertation is to determine:

How can we use aircraft surveillance and other open data to improve aircraft performance modeling and parameter estimation?

To better respond to this inquiry, it was useful to break this main question down into multiple sub-questions. At the beginning of the dissertation, seven research questions were proposed regarding the open aircraft performance modeling.

Q1. How to extract valuable information from open aircraft surveillance signals?

Chapter 2 addressed this question by introducing two data sources from the Mode-S surveillance communications: ADS-B and Comm-B. Combined, these two data sources provided valuable information on aircraft performance, such as position, ground speed, vertical rate, and different types of airspeed. The extraction of information from Comm-B messages was the most challenging part of this phase of the study because the crucial information required to identify the message types is missing for third-party observers. A combination of heuristic and probabilistic approaches were designed to infer Comm-B message types and decode their content. In addition to the decoding of the Comm-B message types, a procedure to detect corrupted messages was also presented. A ground truth validation dataset showed that the model proposed in this chapter yielded a success rate of 97.7% and a false negative rate of 0.008%.

Conclusion 1

In addition to the large quantities of broadcast surveillance messages, aircraft responses to secondary radar interrogations can be used to enhance the accuracy of aircraft performance modeling. Inferring and decoding of these messages with high accuracy are made possible by a two-fold heuristic and probabilistic approach.

Q2. How to turn scattered broadcast data into continuous and segmented flights?

This question responded to the challenge that followed the successful decoding of information from the Mode-S messages. In order to make data beneficial for

aircraft performance studies, flight trajectories should be constructed based on these messages. In many studies, flights needed to be segmented according to flight phases. Since there was little information available regarding flights and no direct information on flight phases, Chapter 3 proposed a two-step method to address this challenge. The first step used a clustering algorithm to group data belonging to the same flight. Once a flight trajectory was extracted, a second step employed a fuzzy logic based process to identify different flight phases in the trajectory.

Conclusion 2

Unsupervised machine learning techniques can be used to extract flights from scattered surveillance data and to identify specific flight phases in these flights.

Q3. How to use surveillance data to improve aviation weather models?

Weather is one of the external factors that significantly affects flights. Earlier studies have indicated the influence of wind on the design, optimization, and prediction of flights. A common approach for air traffic management research has been to use wind information from the numerical weather forecast dataset. However, the low temporal-spatial resolution in these datasets often fails to meet accuracy requirements, especially in performance modeling studies. Chapter 4 of this dissertation proposed a weather model, the Meteo-Particle model, which made use of aircraft surveillance data to construct real-time wind and temperature fields. This approach showed that weather accuracy can be increased by using aircraft surveillance data. The weather models in this dissertation demonstrated a reduction of wind and temperature errors of 67% and 26%, respectively, compared to the numerical weather forecast dataset. With accurate wind field information, aircraft airspeed was accurately computed based on ground speed obtained from ADS-B.

Conclusion 3

A weather field, including wind and temperature, can be accurately constructed in real-time using aircraft surveillance data and the Meteo-Particle model.

Q4. How to construct the kinematic model using surveillance data?

A kinematic performance model deals with the speed, acceleration, and travel distance of the flight without considering the forces acting on the aircraft. This class of models enables a description of a flight without involving, for example, thrust, drag, or mass of the aircraft. In Chapter 5, kinematic models for common commercial aircraft types were constructed. Using the flight phase identification method proposed in Chapter 3, kinematic performances of the aircraft at different flight phases were studied. The flights of each aircraft type were first segmented according to flight phase, including takeoff, initial climb, climb, cruise, descent,

final approach, and landing. Then, kinematic parameters for each flight phase were computed. Based on these computed values, a parametric statistical model was constructed that best described each performance parameter. The kinematic model - WRAP - proposed in this dissertation included the optimal, minimum, and maximum values of each performance parameter. As a result, a set of more than 30 parameters describing the complete flight profile were constructed. The parametric statistical model also made it possible to construct distributions of performance parameters in large flight simulations, which could be used in representative fast-time airspace simulations, for example.

Conclusion 4

Accurately sorted and labeled surveillance data can be used to effectively model the kinematic performance of different aircraft types at distinctive flight phases.

Q5. *How to model thrust and fuel flow with open data?*

Chapter 6 addressed the dynamic performance of the aircraft and one of the core elements in the dynamic performance model, thrust. In addition, Chapter 6 discussed the modeling of engine fuel consumption. Aircraft thrust was computed based on the combination of the simplified two-shaft engine thrust model proposed in [8] and the empirical model proposed in [136]. As a result, with knowledge of altitude, speed, and vertical rate, the maximum thrust of the engines could be computed. In addition to these dynamic variables, the engine bypass ratio was considered a key element in determining the parameters for the thrust model. The use of bypass ratio as the key variable allowed the maximum thrust of different engine types to be consistently described. Another key parameter used to compute the thrust was the maximum engine thrust at cruise conditions. When this data was not available, the empirical model proposed in [136] could be used to derive an approximation based on the maximum thrust at sea-level.

The second part of Chapter 6 investigated an engine fuel flow model based on public information from the ICAO engine emission data-bank [63]. In this database, fuel flow is measured at four different test conditions, namely 100%, 85%, 30%, and 7% of engine maximum power. In order to approximate the fuel flow at any given power setting for each engine type, Chapter 6 constructed a polynomial model to best describe these conditions. When applying the model to actual flights, the fuel flow could be computed based on the ratio of actual thrust over the maximum thrust of the engine.

Conclusion 5

- 1) Models from the literature provide new ways to model aircraft thrust using open data.
- 2) ICAO engine emission data can be used to provide a satisfactory fuel consumption model for a wide range of aircraft types.

Q6. How to use open data to construct aircraft drag polar models?

Modeling the aircraft drag polar was addressed in Chapter 7. The chapter started by explaining the rationale for using drag polars in a point-mass aircraft performance model. The drag polar describes the quadratic relationship between the lift coefficient and the drag coefficient. It differs between aircraft types, as well as between aerodynamic configurations of an aircraft. The coefficients that describe a drag polar are the zero-lift drag coefficient and the lift-induced drag coefficient. In order to estimate these parameters, Chapter 7 introduced a hierarchical model named the stochastic total energy model. This model considered the parameters from the total energy model as random variables. Based on the trajectory data from a climbing flight, this model employed Bayesian computing to approximate the probability density of these random variables. Using multiple flights from the same aircraft type, the zero-lift and lift-induced drag coefficients were derived for different aircraft types.

Thanks to the long duration of aircraft climb, the stochastic model was able to derive the drag polar under clean configuration. However, for other flap configurations, there existed a lack of data to construct the drag polars. Chapter 7, therefore, also relied on several empirical models to estimate the increase of drag due to the deployment of flaps and landing gear. As a result, different drag polar models were computed corresponding to the taxi, takeoff, initial climb, en-route, approach, and landing phases of the flight.

Conclusion 6

- 1) By combining accurate flight data with a hierarchical stochastic total energy model, the drag polar of an aircraft can be constructed.
- 2) Uncertainty in the thrust model has a significant influence on the drag polar estimation.

Q7. How to estimate performance parameters using open surveillance data?

Chapters 8, 9, and 10 constituted the last part of the dissertation. In earlier chapters, performance models were categorized by aircraft type. The last three chapters studied the possibility of estimating the performance parameters of individual flights. Specifically, the estimation of turn performance and aircraft mass were studied.

When dealing with turning flights, one important performance parameter is the bank angle. With a known bank angle, it is possible to infer related parameters such as load factor, turn rate, lift, and drag, as well as the thrust of the aircraft. Chapter 8 proposed a solution for estimating bank angles during turns using ADS-B data. The bank angle was calculated based on the turning speed and the turn radius. In order to calculate the turn radius, the three-dimensional arc of the turn was first transformed into a two-dimensional plane. Then, using the least-squares regression, a circle was found that best fit the positions, where the radius of the circle was obtained as the turn radius.

Sometimes, the aircraft roll angle is interrogated by the secondary surveillance

radar. Such data is downlinked using the Mode-S track and turn report. Though they are not frequently updated, when available, they were used to validate the estimated values. For each of the 17 aircraft types, around 800 turns were studied. The median error for the bank angle calculated using the estimation method proposed in this chapter was found to be around 2 degrees.

Aircraft mass is another crucial parameter in many air transportation studies. For example, with a known mass, the drag, thrust, and fuel flow during a flight can be computed accurately. Knowing the mass of aircraft also increases the accuracy of flight trajectory predictions. However, because aircraft mass is a commercially sensitive piece of information, it is not shared, and aircraft mass remains a missing piece of flight information in many studies. It is even an unknown parameter for air traffic controllers. Chapter 9 and Chapter 10 focused on the estimation of aircraft mass based on two different approaches.

Chapter 9 proposed the use of Bayesian inference so that empirical and professional knowledge of the aircraft in the estimation could be considered. At first, different masses were calculated at different stages of the flight. Combining fuel consumption, multiple initial masses were computed. Then, viewing the empirical information as a prior, a maximum a posteriori estimation was produced.

Unlike the earlier chapter where a complete trajectory was required, Chapter 10 focused on a short segment of the flight at the start of the climb. The chapter proposed the use of a particle filter as the estimator for multiple aircraft states, which included the mass of the aircraft. The particle filter, a Bayesian state estimator, can be used to estimate nonlinear system states. The particles representing the flight states were initiated based on the prior probability distribution functions of the states and updated at each time step based upon new observations. As a result, the mass could be estimated within 30 seconds. The uncertainties in the Mode-S data were studied specifically alongside the estimation. In this manner, the method also identified the uncertainty conditions where estimations became impossible.

Conclusion 7

- 1) Using aircraft surveillance data, it is possible to estimate hidden performance parameters of individual flights, such as bank angle and load factor during the turn, as well as initial mass.
- 2) Uncertainties in ADS-B data that affect the mass estimation accuracy can be quantified.

11.3. Limitations and recommendations

11.3.1. Mode-S

Studies in this dissertation were supported by open aircraft surveillance data, and data availability was one of the major prerequisites to enable these studies. ADS-B data is constantly broadcast by aircraft. However, not all data can be intercepted given the limitations of receiver coverage. Large crowd-sourced networks such as *FlightRadar24*, *ADS-B Exchange*, and the *OpenSky Network* rely on the research and

hobby community to share the data collected by individual receivers. There is a lack of ground-based coverage for aircraft that are flying over oceans and over less populated areas (for example in Siberia, deserts or mountainous areas).¹ On the other hand, there are multiple receivers providing redundant coverage in regions like Europe, North America, South-East Asia, and Australia. Recent developments in space-based solutions have shown the possibility of using satellites to intercept surveillance signals [101, 13, 142]. The space-based system has the potential to extend coverage to the entire Earth. However, whether any of the space-based platforms will provide open access to their collected ADS-B data remains uncertain.

The Comm-B signals used in this dissertation to derive the weather field and airspeed of aircraft were also limited geographically. They were originated by the interrogations of secondary surveillance radars, which only exist in the controlled airspace.

All ADS-B and Comm-B signals are transmitted using the 1090MHz Mode-S transponder.² The congestion of this frequency is becoming another serious constraint for communication as discussed in Chapter 2. The direct impact of the congestion was found to be the reduced update rate of ADS-B and Comm-B messages in busy airspace.

Recommendation 1

Using a distributed local network of ground receivers could potentially increase the total number of correct messages received, as compared to the current situation where messages are potentially dropped due to frequency congestion.

11.3.2. Citation flights

To validate the mass estimation methods, as well as the thrust and fuel flow model, some chapters in this dissertation used the onboard data from demonstration flights carried out by the TU Delft Cessna Citation II aircraft, which was fitted with an advanced data gathering system. These flights were different from common passenger flights. During the cruise, multiple maneuvers were performed to demonstrate the dynamic behavior of the aircraft. These maneuvers included excitations of the phugoid, short period, a-periodic roll, and Dutch roll, as well as parabolic flights. In order to have sufficient excess engine power, the flights were cruising at low altitude (around 10,000 ft) instead of at optimal cruise altitude (above 30,000 ft). However, during the climb, the aircraft still followed a common flight procedure. Thus, the studies could still use the flights to validate the results of mass estimations. Nevertheless, a single aircraft type was one of the limitations in validation experiments.

¹In rare cases, regional regulations may actively take down existing ground receiver installations by ADS-B hobbyists.

²Other Mode-S messages, such as All-call replies, ACAS, and Comm-D, are also sharing the same frequency.

Recommendation 2

Thrust and fuel flow model, as well as the mass estimation, should be validated with more flights from different aircraft types by future studies.

11.3.3. Thrust model

One of the main performance components in the dynamic model is the thrust model. The uncertainty in thrust affects the estimation of other components of the performance model. One example of such inherited uncertainty was shown in Chapter 7, where the drag polar was estimated. Due to the limited information available on aircraft engine configuration, it was not possible to identify the exact engine type for each flight. Hence, a default engine type was assumed for each aircraft type. Related experiments observed that aircraft with more engine options displayed larger uncertainty in the final drag polar coefficients.

Another example could be seen in Chapter 10, where thrust again acted as a source of uncertainty in the mass estimation. This chapter studied the influence of varying thrust settings on the mass estimation. Even though the change in trajectory due to different thrust settings was smaller than the change in takeoff mass, such uncertainty could still affect the final estimation of the aircraft mass. At the start of the climb, a derated maximum thrust profile was assumed. The ratio of maximum thrust (between 0.8 and 1) was assumed to be constant. This, however, was not entirely true in reality. For example, when an aircraft leveled during the climb, the ratio could suddenly decrease and remain at a lower value during the leveling.

Recommendation 3

When thrust is involved in the estimation, the segment where maximum thrust profile is most likely to be applied should be chosen.

11.3.4. Non-linear system

The point-mass model has simplified the flight dynamics. However, there is still a high order non-linearity in the system. For example, since the drag polar is used, the mass has a quadratic relationship with drag, and velocity has a quartic relationship with drag. At the same time, thrust also has a piece-wise higher order relationship with velocity and altitude. Thus, state estimations and trajectory predictions cannot be easily solved with linear filters.

This dissertation introduced a nonlinear *particle filter* to solve the challenge of estimating aircraft mass. The particle filter required a very large number of particles to approximate systems and contained many states variables. A significant amount of computing resources was required to update these particles. In order to apply the method to a large number of flights at the same time, a cluster of computers may be required in the future. The particle filter was designed only for mass estimation in this dissertation. However, this filter had also demonstrated the potential for other applications, such as short-term trajectory predictions.

Recommendation 4

Future studies should explore the possibility of using the particle filter for short-term trajectory prediction in air traffic management studies. The efficiency of a particle filter can be improved by the implementation of parallel computing.

11.4. Final conclusions

A large amount of open surveillance data has become accessible for researchers thanks to the deployment of ADS-B transponders in airliners. This open data offers a new path to study and model aircraft performances in the domain of air transportation science. The chapters of this dissertation addressed many topics related to the decoding and processing of this data, using the data to model aircraft performance, as well as combining the data and model to perform parameter estimations.

Although designed with the goal of creating an open performance model for the BlueSky simulator, the models and methods proposed in this dissertation can also be used for other air transportation studies, such as trajectory prediction and optimization. In Appendix A, an independent open aircraft performance library, OpenAP, is described in detail. As a result of this dissertation, the OpenAP model is currently one of the most complete alternatives to the commonly used closed-source models. Hopefully, this open data and open model approach will bring more comparable and reproducible studies for future studies in air transportation science.

Appendix

A

The OpenAP Python Library

Throughout the dissertation, different modules related to aircraft performance are studied. In this appendix, the corresponding open-source implementation of the performance model, OpenAP, is described. The OpenAP is a code library written in Python programming language. It allows anyone quick access to all the performance modules covered in this dissertation. For example, it makes the computation of the thrust, drag, and fuel flow with a few lines of Python code. This appendix first covers the general structure of the OpenAP. Then, each of the components of the library is explained, and examples are also given.

A.1. Basics of the OpenAP

The OpenAP ¹ library consists of four main components, which are *aircraft properties*, *kinematic performances*, *dynamic performances*, and *utilities*. In each main component, there can be multiple sub-components. Figure A.1 shows the structure of the OpenAP model as of March 2019.

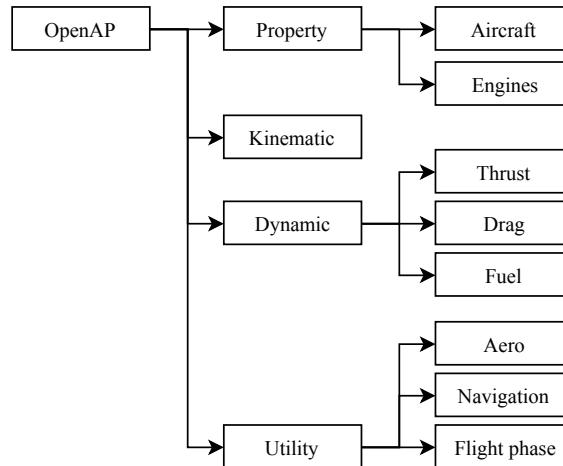


Figure A.1: The structure of the OpenAP Python library

In OpenAP, multiple file formats exist to store different types of data effectively. The main file format is the *YAML* format [10], which is a human friendly data serialization standard. It can be easily interpolated by both human and programming languages. For example, aircraft properties and drag polar models are presented using the *YAML* file format. *YAML* is effective for storing key-value like data structure but not efficient for larger tabular data. In these cases, the *fixed width text* files are used. They can be processed by computers easily, while still maintaining readability for users. For example, the engine and kinematic performance databases are stored using fixed width text files.

OpenAP aims to be a practical model with easy applicability. The toolkit interacts with the model data and implements the equations for different calculations. It gives other high-level applications easy access to the OpenAP model.

A.2. Aircraft properties

The property module includes characteristics and engine configurations for different aircraft types. It also provides performance data for a large number of engines. Currently, the most common aircraft types are included in this module. For each aircraft type, parameters such as dimensions, limits, nominal cruise condition, and engine options are given. For example, the Airbus A320 aircraft (`/openap/openap/data/aircraft/a320.yml`) parameters are shown as follows:

¹Available at: <https://github.com/junzis/openap>

```

aircraft: Airbus A320

fuselage:
  length: 37.57
  height: 4.14
  width: 3.95

wing:
  area: 124
  span: 35.8
  mac: 4.29
  sweep: 25
  t/c: null

flaps:
  type: single-slotted
  area: 21.1
  bf/b: 0.780

limits:
  MTOW: 78000
  MLW: 66000
  OEW: 42600
  MFC: 24210
  MMO: 0.82
  ceiling: 12500

cruise:
  height: 11000
  mach: 0.78

engine:
  type: turbofan
  mount: wing
  number: 2
  default: CFM56-5A3
  options:
    A320-111: CFM56-5-A1
    A320-211: CFM56-5-A1
    A320-212: CFM56-5A3
    A320-214: CFM56-5B4
    A320-215: CFM56-5B5
    A320-216: CFM56-5B6
    A320-231: V2500-A1
    A320-232: V2527-A5
    A320-233: V2527E-A5

```

The unit for dimension and height in the configuration files is the meter, while the unit for weight is the kilogram. The aircraft data are gathered based on public data from aircraft manufacturers, as well as the literature, for example in [70, 105].

Performance data on around 400 different engines are presented in the OpenAP engine databases (`/openap/openap/data/engine/engines.txt`). Each engine has 11 parameters that are related to its performance. A select list of engines are supplied in Appendix C.

The retrieval of any aircraft and engine data can be performed with a few lines of Python code. For example, using the following commands, the aircraft configurations and parameters of a specific engine for the Airbus A320 aircraft can be obtained:


```

from openap import prop

aircraft = prop.aircraft('A320')
engine = prop.engine('CFM56-5B4')

print(aircraft['wing']['span'])
print(engine['max_thrust'])

```

A

A.3. Kinematic performance

The kinematic model studies the motion of the aircraft without considering the force acting on the aircraft. In OpenAP, the kinematic performance database is based on the results of a data-driven statistical model (WRAP), which is proposed in Chapter 5.

By combining all the parameters, the WRAP model (`/openap/openap/data/wrap`) is able to describe the trajectory from takeoff until landing for most of the common aircraft types. Using the OpenAP kinematic model, the trajectory of the aircraft can be constructed using a set of ordinary differential equations. Assuming the International Standard Atmosphere (ISA) condition, no wind², and a small flight path angle, these equations are expressed as:

$$\frac{dx}{dt} = V \sin \psi \quad (\text{A.1})$$

$$\frac{dy}{dt} = V \cos \psi \quad (\text{A.2})$$

$$\frac{dh}{dt} = VS \quad (\text{A.3})$$

$$\frac{dV}{dt} = a \quad (\text{A.4})$$

where the true airspeed (V) can be computed from either calibrated airspeed or Mach number depending on the flight phase. For example, the true airspeed can be computed from calibrated airspeed using the following equations:

$$V = \sqrt{\frac{7p}{\rho} \left[\left(1 + \frac{q_c}{\rho} \right)^{\frac{2}{7}} - 1 \right]} \quad (\text{A.5})$$

$$q_c = p_0 \left[\left(1 + \frac{\rho_0}{7p_0} V_{\text{cas}}^2 \right)^{\frac{7}{2}} - 1 \right] \quad (\text{A.6})$$

In cases where Mach number is given, the true airspeed can be calculated using temperature (τ) and the speed of sound (a) as:

²For non-standard atmospheric conditions, temperature and wind corrections need to be applied.

$$V = a_0 M \sqrt{\frac{\tau}{\tau_0}} \quad (\text{A.7})$$

where ρ , p , τ can be obtained from the ISA model based on the altitude of the aircraft.

Each parameter from the WRAP kinematic model is given with a default value, minimum value, maximum value, and a parametric statistical model. The minimum and maximum values provide the boundaries of the parameter, while the parametric statistical model can be used to sample values from a given probability density function (Normal, Gamma, or Beta). With the OpenAP toolkit, all kinematic performance parameters can be queried using the functions defined as follows:

```

from openap import WRAP

wrap = WRAP(ac='A320')

param = takeoff_speed()
param = takeoff_distance()
param = takeoff_acceleration()
param = initclimb_vcas()
param = initclimb_vs()
param = climb_range()
param = climb_const_vcas()
param = climb_const_mach()
param = climb_cross_alt_concas()
param = climb_cross_alt_conmach()
param = climb_vs_pre_concas()
param = climb_vs_concas()
param = climb_vs_conmach()
param = cruise_range()
param = cruise_alt()
param = cruise_init_alt()
param = cruise_mach()
param = descent_range()
param = descent_const_mach()
param = descent_const_vcas()
param = descent_cross_alt_conmach()
param = descent_cross_alt_concas()
param = descent_vs_conmach()
param = descent_vs_concas()
param = descent_vs_post_concas()
param = finalapp_vcas()
param = finalapp_vs()
param = landing_speed()
param = landing_distance()
param = landing_acceleration()

```

Each of the functions returns the optimal, minimum, maximum, and the parameters to construct the probability density function that reflects the performance indicator.

A.4. Dynamic performance

Similarly, the dynamic module is included. In addition to speed, vertical rate, and distance, the dynamic module deals with more performance parameters, specifically the mass and forces.

A.4.1. Aircraft dynamic model

The aircraft motion in the OpenAP is defined using a four degree of freedom *roll-point-mass* model, which describes the aircraft translations in three perpendicular axes, as well as the roll rotation. In this roll-point-mass model, the forces are assumed to be applied at the aircraft's center of gravity. The model can be expressed with a set of ordinary differential equations similarly. Assuming the ISA and zero wind, these equations can be written as:

$$\frac{dx}{dt} = V \sin \psi \cos \gamma \quad (\text{A.8})$$

$$\frac{dy}{dt} = V \cos \psi \cos \gamma \quad (\text{A.9})$$

$$\frac{dh}{dt} = V \sin \gamma \quad (\text{A.10})$$

$$\frac{d\psi}{dt} = \frac{g \tan \phi}{V \cos \gamma} \quad (\text{A.11})$$

$$\frac{dV}{dt} = \frac{T - D}{m} - g \sin \gamma \quad (\text{A.12})$$

$$\frac{dm}{dt} = f_{\text{fuel}}(T) \quad (\text{A.13})$$

The dynamic components introduced in these differential equations are mass, thrust, drag, and fuel flow. The net thrust is expressed as the product throttle setting (δ_T) and the maximum thrust. Based on an empirical model proposed by [8], the maximum thrust of the aircraft (f_{thr}) is expressed as a function that is dependent on the aircraft altitude, speed, and vertical rate:

$$T = \delta_T f_{\text{thr}}(h, V, VS) \quad (\text{A.14})$$

The drag force of the aircraft can be calculated knowing the dynamic pressure (q), the wing surface (S), and the drag coefficient (C_D):

$$D = C_D q S \quad (\text{A.15})$$

$$q = \frac{1}{2} \rho V^2 \quad (\text{A.16})$$

The fuel flow model is a function that is dependent on the net thrust of the aircraft engines. It is constructed using the fuel flow coefficients proposed in the engine performance database that is shown in Appendix C. The details of the thrust, drag, and fuel flow models are explained in the rest of this appendix.

A.4.2. Thrust

The thrust model included in OpenAP is able to calculate maximum takeoff thrust and maximum thrust during the flight separately. The key parameters that determine the max thrust of the turbofan engine are the maximum static thrust at the sea level, the bypass ratio, and the rated engine thrust at cruise. These parameters are included in the engine database (*/openap/openap/data/engine/engines.txt*).

The OpenAP library implements the thrust model that is described in Chapter 6. The maximum thrust of the aircraft during the flight can be conveniently computed as shown in the following examples:

```
from openap import Thrust

thrust = Thrust(ac='A320', eng='CFM56-5B4')

T = thrust.takeoff(tas=100, alt=0)
T = thrust.climb(tas=200, alt=20000, roc=1000)
T = thrust.cruise(tas=230, alt=32000)
```

where the inputs of the function are true airspeed (in knots), altitude (in feet), and rate of climb (in feet/minute). The outputs are maximum thrust forces (in Newton).

A.4.3. Drag

In order to compute the drag of an aircraft, the drag coefficient needs to be determined first. The drag coefficient is dependent on the drag polar and the lift. In Chapter 7, drag polar models are derived from common airliners. In the OpenAP toolkit, these drag polar models are included. The following is an example drag polar model for Airbus A320 (*/openap/openap/data/dragpolar/a320.yml*):

```
aircraft: Airbus A320

cd0:
  clean:      0.018
  initclimb:  0.020
  finalapp:   0.024

k:
  clean:      0.039
  initclimb:  0.036
  finalapp:   0.034

e:
  clean:      0.798
  initclimb:  0.850
  finalapp:   0.902

gears:      0.017
mach_crit:  0.632
```

The OpenAP toolkit provides the calculation of drag directly under different flaps configurations. Given the aircraft mass, airspeed, altitude, and flight path angle, the drag of the aircraft can be computed using the following examples:

```

from openap import Drag

drag = Drag(ac='A320')

D = drag.clean(mass=60000, tas=200, alt=20000)

# higher drag due to compressibility
D = drag.clean(mass=60000, tas=250, alt=20000)

# increase of drag due to flaps
D = drag.nonclean(mass=60000, tas=150, alt=1000,
                 flap_angle=20, path_angle=10, landing_gear=False)

# increase of drag due to landing gear
D = drag.nonclean(mass=60000, tas=150, alt=200,
                 flap_angle=20, path_angle=10, landing_gear=True)

```

where the inputs of these functions are mass (in kilogram), true airspeed (in knots), and altitude (in feet). The outputs are drag force (in Newton).

A.4.4. Fuel flow

The fuel flow model is also discussed in Chapter 7. Using the OpenAP toolkit, the fuel flow can be calculated conveniently as follows:

```

from openap import FuelFlow

fuel = FuelFlow(ac='A320', eng='CFM56-5B4')

FF = fuel.at_thrust(acthr=50000, alt=0)
FF = fuel.at_thrust(acthr=50000, alt=20000)
FF = fuel.enroute(mass=60000, tas=200, alt=20000, path_angle=3)
FF = fuel.enroute(mass=60000, tas=230, alt=32000, path_angle=0)
FF = fuel.takeoff(tas=100, alt=0, throttle=1)

```

where the inputs of this functions are thrust (in Newton), mass (in kilogram), true airspeed (in knots), altitude (in feet), flight path angle (in degrees), and throttle setting (between 0 and 1).

In summary, the fuel flow can be computed under three conditions in OpenAP: 1) with a known net thrust of all engines; 2) during the takeoff with a known thrust setting and unknown aircraft mass; 3) during the flight, knowing the mass, speed, altitude, and flight path angle.

A.5. Other databases and utility libraries

In addition to the core components explained in the earlier sections, OpenAP also provides several utility libraries to facilitate calculations related to aircraft performance.

A.5.1. Flight phase

Based on the method proposed in Chapter 3, the flight phase library enables the easy use of the fuzzy logic flight phase identification algorithm. Providing the trajectory (with time, altitude, speed, and vertical rate), one can obtain corresponding flight phases for the entire trajectory. The flight data are labeled with the ground, climb, cruise, descent, and leveling phases.

Functions in the flight phase library also enable the extraction of a specific flight segment among all available flight phases, which are takeoff, initial climb, climb, cruise, descent, final approach, and landing. For example:

```
from openap import FlightPhase

FP = FlightPhase()
FP.set_trajectory(ts, alt, spd, roc)

# produce the phase label for each data point
labels = FP.phaselabel()

# return the set of indices each phase starts
indices = FP.flight_phase_indices()
```

In the third code line of the example, the trajectory data is set. The last two functions produce the flight phase labels and the indices of the data point when a different flight phase starts.

A.5.2. Aeronautical calculations

Derived from the open-source BlueSky simulator [54], the aeronautical library provides common calculations related to aircraft performance. For example, this library provides the functions to compute distance and bearing. The standard atmospheric conditions at different flight levels can also be calculated. In addition, speed conversions among CAS, TAS, and Mach number are easily made with this library. There are many functions included in this model, and the details can be found in the Python package (*/openap/openap/aero.py*).

A.5.3. Navigation database

The navigation database provides information on airport, navigation data, and fixes (*/openap/openap/data/nav*). The data in the navigation database are based on the open data published by X-Plane [107]. The navigation library is designed to give easy access to these datasets. For example, the airport location can be queried using its ICAO code. This library can also be used to determine the closest airport or fixes of a specific location. Examples are shown as follows:

```
from openap import nav

print(nav.airport('EHAM'))
print(nav.closest_airport(52.011, 4.357))

print(nav.fix('EH155'))
print(nav.closest_fix(52.011, 4.357))
```

A.6. Discussion

As mentioned in the earlier sections of this appendix, the thrust model included in the OpenAP is based on the studies of [8] and [136], where only turbofan engines are modeled. Hence, the OpenAP does not cover aircraft that are equipped with different engine types. Currently, only the most common civil aircraft types are covered in the OpenAP. This is due to the availability of flight data. Though the lack of more diverse aircraft types may be a limiting factor for some applications,

the data-driven approach used to construct the OpenAP allows more aircraft to be included with the growing available volume of flight data in the future.

The OpenAP model data and source code for the toolkit are published under the GNU GPL v3 license. This allows anyone to freely use (including commercial and private use), distribute, and change the content.³ The advantage of having such an open license is to allow easy access and encourage contributions from the ATM research community. The source is published on GitHub, where anyone can take part in the future development of the model.

A

A.7. Conclusion

The details of the open aircraft performance programming library are explained in this appendix of the dissertation. The OpenAP model is aimed at providing the ATM research community an open alternative to current closed-source aircraft performance models. Accompanied by an off-the-shelf user library written in Python programming language, the performance model consists of four major components, which are aircraft and engine properties, kinematic performance, dynamic performance, and utility libraries.

The characteristics of 27 common aircraft and 400 turbofan engines are included in the OpenAP library. The aircraft database provides basic information on aircraft dimensions, limits, and engine configurations. The engine database provides information such as maximum thrust, bypass ratio, cruise performance, and fuel flow coefficients for each engine type.

As a final note, the OpenAP model already serves as one of the performance engines that powers the BlueSky simulator, which is a tool designed for large scale air traffic simulations. All OpenAP model data and the toolkit source code are published on GitHub under the GNU GPL license.

³License details at: <https://www.gnu.org/licenses/gpl-3.0.en.html>

B

Drag polar calculation using CFD simulations

B.1. Introduction

One cost-effective way to study the aerodynamic properties of an aircraft is to perform numerical simulations. Computational fluid dynamic (CFD) simulations are often designed to evaluate the new designs of aircraft wings and structures. On the other hand, we can also use CFD to study existing aircraft aerodynamic properties. This dissertation also studied the possibility of using open 3D models and open CFD tools to construct drag polar of the aircraft.

The CFD approach computes the air flow velocity and pressure around a 3D aircraft model in a steady flow. Then, it can calculate the drag and lift forces along and perpendicular to the body axis of the aircraft. Knowing the drag and lift, the drag and lift coefficient can be calculated. We can control the angle of attack for the 3D model for the simulation. Hence lift and drag coefficient can be estimated for different configurations of the angle of attack.

For this study, the open-source CFD software OpenFOAM (Open source Field Operation And Manipulation) is used to construct the simulation. As part of the OpenFOAM, the SimpleFOAM solver is used to solve the Navier-Stokes equations, which describe the motion of the fluid. Once the solver converges, the forces in each axis can be computed based on the pressure difference. Knowing the reference airflow velocity and aircraft wing surface, the lift and drag coefficients can be computed.

The open-source community has produced many open 3D models of aircraft, but the biggest challenge is identifying accurate 3D models. In order to derive a drag polar that corresponding to the actual aircraft type, the airfoil profiles of the wing have to be consistent with the actual design. The large variety of 3D model formats also makes the utilization of these models difficult. This chapter focuses on these challenges and illustrates a path that combines open tools and open data to derive drag polar based on CFD simulations.

B.2. 3D model

Accurate 3D models of aircraft are rarely published by aircraft manufacturers. However, online 3D communities, such as *GrabCAD* and *3D Warehouse*, provide easy access to several aircraft models that were made by hobbyists. The flight simulation community, such as *FlightGear*, also shares their 3D models of aircraft. However, the quality of these 3D models is not always guaranteed. Some of the authors try to be as accurate as possible to real aircraft, while others may focus on the rendering of the 3D models. In Figure B.1, two designs of a same aircraft model are illustrated. The structural differences and resolutions of these two designs are clearly visible.

Other than the accuracy of the aircraft shape, the most important factor is the airfoil profile, which is also one of the least accurate elements of these open 3D models. 3D wing models constructed by the open source community are often based on public airfoil data, for example, the open airfoil database [119]. Two reasons account for this inaccuracy. First, the detail of the airfoil profile is insignificant for general visualization purposes. Second, the inaccuracy in wing design is also due to the fact that modern aircraft manufacturers rarely disclose the aerody-

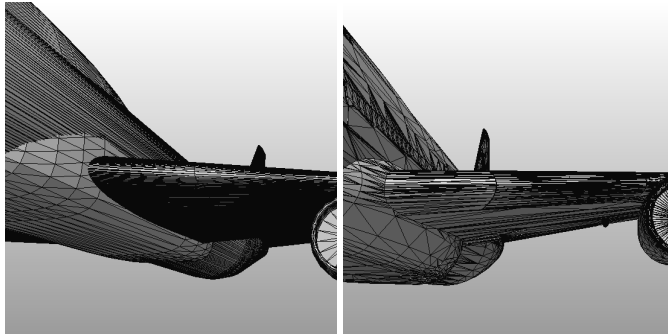


Figure B.1: Different levels of design detail

namic design of wings. A lot of open 3D models use generic NACA profiles, which significantly affects the results of CFD simulation. According to a study conducted by [79], for the two major passenger aircraft manufacturers, Boeing and Airbus, only a few airfoils used on very old aircraft types are available to the public. It is understandable that aircraft manufacturers desire to treat these data as private in order to maintain their competing edge. However, this limitation poses a great challenge to this study.

Research conducted by Kagkaras [73] shows an alternative possibility using laser scanning to construct 3D models of aircraft. However, such a method requires the complicated setup of scanners and intensive post-processing to acquire a single aircraft model. It also requires a certain level of cooperation from aircraft owners.

Even with a 3D model of an aircraft, there are still many detailed processes to obtain a workable solution. Firstly, there are many varieties of 3D data format generated by different software. More than 20 different formats are commonly used. Secondly, not all 3D designs are ready for CFD simulation. After the 3D file is converted to the desired data format, a series of clean-ups and reparations are often required before CFD can be applied. In the following diagram (shown in Figure B.2), a detailed working flow is designed to illustrate the entire process of the open CFD simulation. In the remainder of this appendix, a Boeing 747-400 aircraft model [108] is used to illustrate the major steps of CFD simulation.

B.3. Pre-processing of 3D models

Since the CFD simulations in this study are completely based on public domain data, the source of aircraft 3D models come in varying levels of quality and formats. Quite often, these models contain geometric errors. Thus, the pre-process step covers selecting, evaluating, repairing, and cleaning the 3D aircraft model prior to CFD simulation.

To evaluate a 3D model, its format needs to be checked first. With the large variety of 3D model definitions, choosing the right software for pre-processing is essential. The two most used open-source tools for the preparation of this study are *Blender* [14] and *FreeCAD* [38]. Once the model is rendered, the general

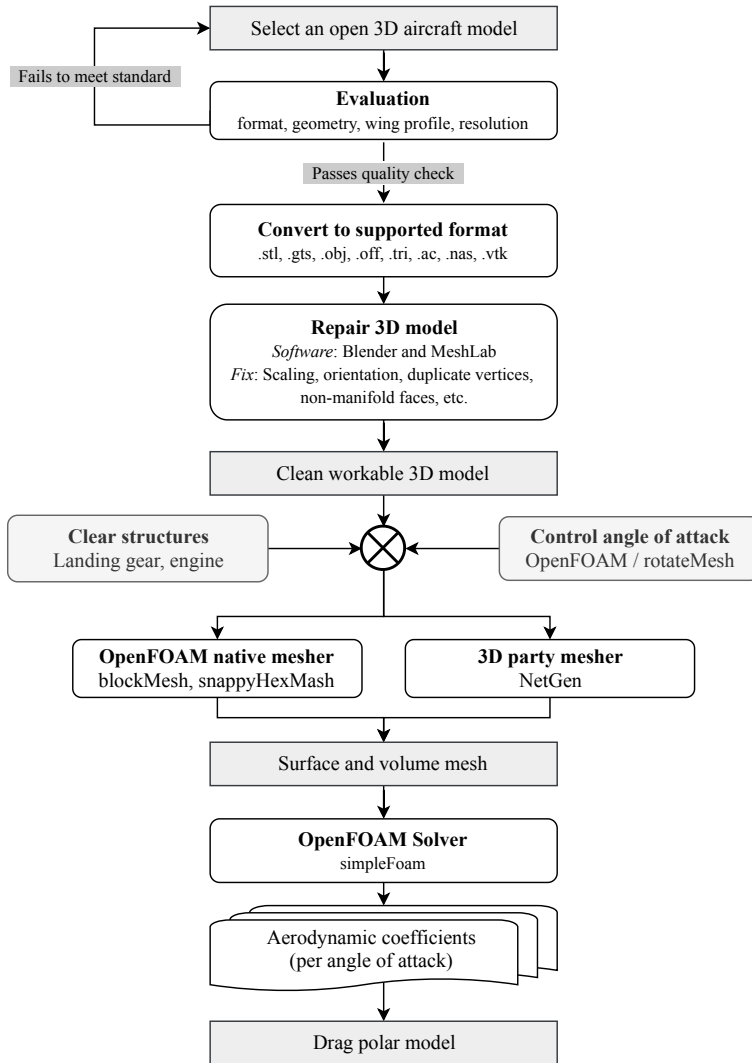


Figure B.2: Flow for open CFD simulation using OpenFOAM

geometry of an aircraft needs to be checked. Airfoil at the cross sections of the wings needs to be carefully examined. The scaling and swapping angle of the wings should also be compared with the aircraft design specification. If the model does not resemble the real structure of the aircraft, it cannot be used for the CFD simulation. Upon successful initial quality control, using *Blender* or *FreeCAD*, most of the 3D formats can be converted to a common interchange format, for example, STL (STereoLithography) format. Sometimes, specific plug-ins must be installed to read corresponding formats. However, in the example of Boeing 747-400 for this study, *OpenVSP* is used for the initial conversion from the less common VSP

(Vehicle Sketch Pad) format to STL format.

The next step is to fix errors in the 3D model, so it can be used for CFD simulation. With the correct scaling ratio, it is very common that the model may not come with proper size and unit. The 3D-orientation for the model is not always fixed. To ease the CFD configuration, a standard scaling and orientation of the aircraft are set in order to standardize the simulation. Other design problems need to be fixed at the same time. Extra or duplicated vertices (edges or faces) in the 3D design are removed, and non-Manifold faces are corrected. Most of these tasks can be executed automatically with software (such as *Blender* or *MeshLab* [22]). However, some models (especially when no alternatives are available) do also require an extensive amount of manual 3D design work.

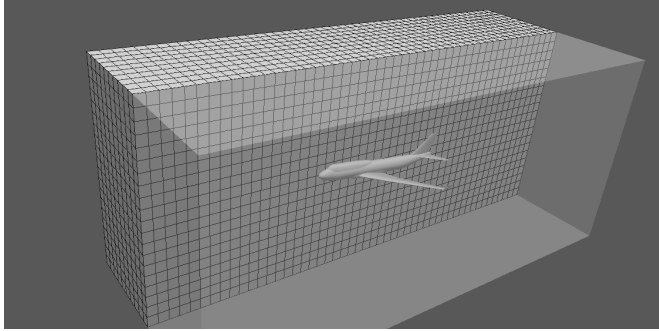
In addition to corrections, there is also a need to remove some of the body structures (or add them in cases of comparison studies). For example, landing gears, which produce an enormous amount of the drag, are to be removed when present. Engine and parts engines are also removed to study their effects on drag polar models.

Once a clean 3D model is produced and before it is used for CFD, a rotation procedure is used to control the angle of attack of the aircraft. This is achieved using the *rotationMesh* method from the *OpenFOAM*. Angles of attack are set from -5 to 15 degrees with a 1-degree increment, as well as from 16 to 30 degrees with a 2-degree increment. At each pitch angle configuration, this study applies the meshing and solving steps. In total, around 30 rounds of CFD simulation, with 500 iterations each round, are performed for an aircraft model to construct the final drag polar.

B.4. Meshing

In Figure B.2, two different meshing tools are proposed. The first one is the native meshing tool (*snappyHexMesh*) from *OpenFOAM*, while the other is a 3D-party open-source meshing software, *NetGen* [117]. A variety of other open-source mesh generation tools also exist. For CFD simulations in *openFOAM*, the aircraft and a volume surrounding the aircraft need to first be defined. Then, the general meshing process consists of three steps, which are described as follows:

1. **Creating the surface mesh and extracting feature:** A mesh of all the surfaces of aircraft is first created. This mainly consists of refining and fixing faces from the existing 3D model. After that, edges (conjunct faces with an angle larger than a threshold) of the aircraft are extracted. Later on, the volume mesh surrounding the edge will be refined to a much more detailed level.
2. **Generating the block mesh:** The entire defined volume surrounding the aircraft is divided into tiny 3D blocks defined by the user. At the same time, predefined boundaries are also extracted, which consist of an inlet, an outlet, and walls. In Figure B.3, such background mesh is illustrated with respect to the aircraft model.
3. **Generating the volume mesh:** Based on previously generated background



B

Figure B.3: Background block meshes

mesh, the *snappyHexMesh* program will create the 3D meshes of hexahedra and split-hexahedra from the aircraft model. Iteratively, the mesh conforms to the aircraft surface. A different level of details may be defined for refinement. In general, the areas around the edges are refined in more details than a flat surface. This is achieved by splitting cells around the edges that are extracted in the first step of the meshing. The volume mesh will be snapped to the surface of the aircraft. Finally, as a common practice, an additional layer is added to the area where irregular cells exist. All these procedures are handled by *snappyHexMesh* automatically. However, setting up the proper parameters do require some empirical knowledge and many trial-and-error corrections. As the result of *snappyHexMesh*, the surface and volume meshes are illustrated in Figure B.4 respectively.

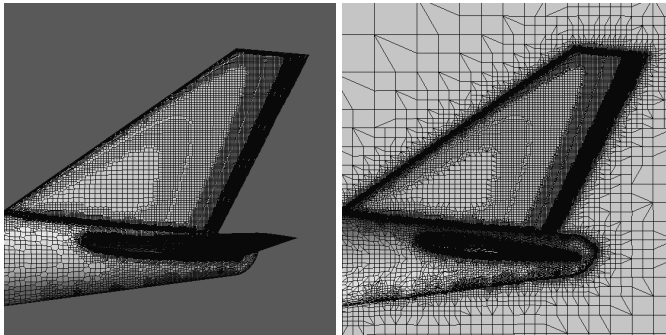


Figure B.4: Final meshes around the aircraft tail

- Examining the mesh result:** The final mesh should always be examined to ensure the quality of the mesh. This can be performed with visual inspection, such as *paraFoam*, a wrapper for 3D visualization tool *paraView* or *CheckMesh* tool. Both are part of the *OpenFOAM* package.

B.5. Navier-Stoke solver

Once the correct meshes are generated for the CFD simulation, the solution of Navier-Stoke equations is straightforward using solvers provided by *OpenFOAM*. In this experiment, the *simpleFoam* solver is used. It is a steady-state solver for incompressible flows with turbulence models. Depending on the scale and resolution of the mesh, it can take hours to days to complete a simulation run.

Once the simulation is complete, we can obtain the velocity and pressure of the flow around the aircraft body. In Figure B.5, the air velocity and pressure surrounding the wing at the end of the simulation are illustrated using streamlined tubes. The scales of the velocity and pressure are marked in each plot respectively. In this experiment, the reference free stream velocity is 100m/s . In the pressure plot, the values are shown as relative values to the static air pressure.

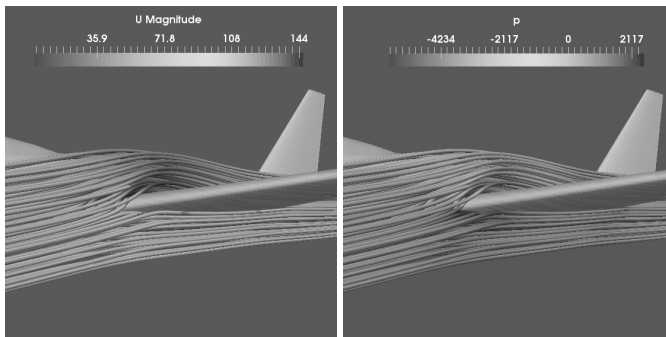


Figure B.5: Visualization of flow velocity and air pressure

At each CFD iteration, the total lift and drag forces are calculated. Knowing the stream velocity, air density, and reference area, lift and drag coefficient can then be computed. In Figure B.6, computed lift and drag coefficients are plotted. The convergences of both coefficients are found at the end of the simulation.

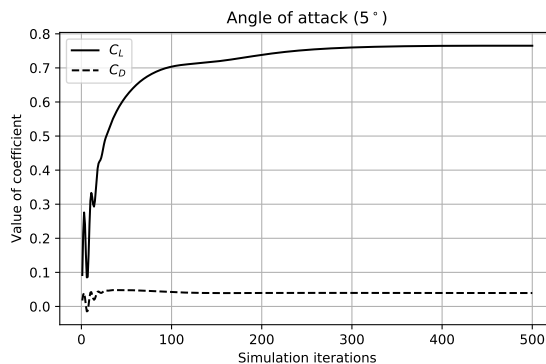


Figure B.6: Convergence of C_L and C_D (B747-400)

B.6. Computation of the drag polar

The aircraft is configured with different angles of attack at each simulation. Drag and lift coefficients for all simulations are gathered to generate the profiles of lift and drag coefficient. For example, a second degree or third-degree polynomial can be used to approximate these two coefficients, and they are illustrated in Figure B.7.

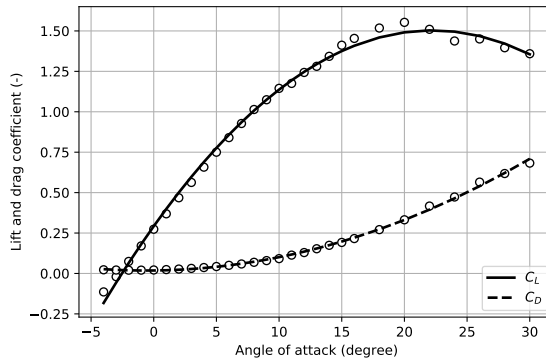


Figure B.7: Polynomial spline models of lift and drag coefficients (B747-400 simulated, clean configuration)

Knowing the relation of drag and lift coefficients under each condition of the angle of attack, the drag polar can be computed using the least-squares regression. Figure B.8 illustrates an example of drag polar by using CFD simulation for an aircraft with a clean (no flaps or extended landing gear) configuration. It is constructed based on the previous example of B.7.

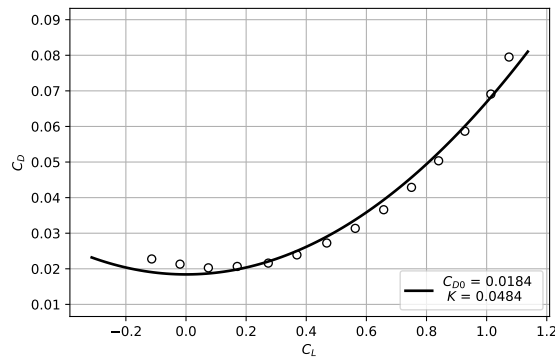


Figure B.8: Drag polar (B747-400 simulated, clean configuration, $\alpha < 10^\circ$)

C

Aircraft engine performance

AC	Engine	T_0	BPR	PR	C_{ff3}	C_{ff2}	C_{ff1}	h_{cr}	M_{cr}	T_{cr}	SFC_{cr}
A319	CFM56-5A4	97.89	6.00	23.80	0.448	-0.591	1.043	35	0.8	22.24	
A319	CFM56-5A5	104.53	6.00	25.10	0.451	-0.559	1.083	35	0.8	22.24	
A319	V2522-A5	102.50	4.90	25.70	0.638	-0.998	1.336				
A319	V2524-A5	108.90	4.80	27.00	0.685	-1.035	1.394				
A319	V2527M-A5	111.20	4.82	27.20	0.558	-0.817	1.316				
A320	CFM56-5-A1	111.20	6.00	26.60	0.438	-0.502	1.118				
A320	CFM56-5-A1	111.20	6.00	26.60	0.438	-0.502	1.118				
A320	CFM56-5A3	117.88	6.00	27.90	0.441	-0.468	1.161	35	0.8	22.24	0.0169
A320	CFM56-5B4	117.90	5.90	27.10	0.411	-0.466	1.224	35	0.8	22.24	0.0154
A320	V2500-A1	111.20	5.30	29.80	0.579	-0.816	1.355				
A320	V2527-A5	110.30	4.80	27.10	0.705	-1.057	1.406				
A320	V2527E-A5	111.20	4.82	27.20	0.558	-0.817	1.316				
A321	CFM56-5B1	133.45	5.70	30.20	0.427	-0.397	1.331	35	0.8	25.98	0.0169
A321	CFM56-5B2	137.90	5.60	31.30	0.517	-0.472	1.384	35	0.8	25.98	0.0154
A321	V2530-A5	133.00	4.60	32.00	0.894	-1.184	1.619				
A321	CFM56-5B1	133.45	5.70	30.20	0.427	-0.397	1.331	35	0.8	25.98	0.0169
A321	CFM56-5B2	137.90	5.60	31.30	0.517	-0.472	1.384	35	0.8	25.98	0.0154
A321	V2533-A5	140.60	4.50	33.70	1.133	-1.418	1.722				
A321	V2530-A5	133.00	4.60	32.00	0.894	-1.184	1.619				
A332	CF6-80E1A2	287.04	5.10	33.10	0.977	-0.854	2.650				
A332	CF6-80E1A4	297.44	5.10	34.50	1.101	-0.905	2.713				
A332	CF6-80E1A3	304.80	5.10	35.70	1.266	-1.043	2.771				
A332	PW4168A	302.50	5.10	31.84	1.118	-1.269	3.040				
A332	PW4170	311.40	4.86	33.80	1.351	-1.488	3.161				
A332	Trent 772	320.30	5.03	36.33	1.396	-1.353	3.103				
A333	CF6-80E1A2	287.04	5.10	33.10	0.977	-0.854	2.650				
A333	CF6-80E1A4	297.44	5.10	34.50	1.101	-0.905	2.713				
A333	CF6-80E1A3	304.80	5.10	35.70	1.266	-1.043	2.771				
A333	PW4164	284.68	5.20	30.10	1.013	-1.235	2.896				
A333	PW4168	302.50	5.10	31.84	1.118	-1.269	3.040				
A333	PW4168A	302.50	5.10	31.84	1.118	-1.269	3.040				
A333	Trent 768	304.20	5.15	34.48	1.058	-1.057	2.913				
A333	Trent 772	320.30	5.03	36.33	1.396	-1.353	3.103				
A333	Trent 772	320.30	5.03	36.33	1.396	-1.353	3.103				
A343	CFM56-5C2	138.78	6.80	28.80	0.413	-0.415	1.314	35	0.8	30.76	0.0154
A343	CFM56-5C3	144.57	6.70	29.90	0.374	-0.340	1.343	35	0.8	30.74	0.0154
A343	CFM56-5C4	151.25	6.60	31.15	0.393	-0.323	1.389	35	0.8	31.58	0.0154
A359	Trent XWB-84	379.00	9.01	41.10	1.498	-1.879	3.209				
A388	Trent 970-84	338.70	8.45	38.00	0.842	-0.935	2.707				
A388	Trent 972-84	345.90	8.41	38.68	0.852	-0.924	2.752				
A388	GP7270	332.39	8.71	36.62	0.781	-0.739	2.602	35	0.85	56.19	
B734	CFM56-3B-2	98.30	5.10	24.10	0.524	-0.732	1.268				
B734	CFM56-3C-1	82.29	5.10	21.30	0.504	-0.775	1.148				
B737	CFM56-7B20	91.63	5.40	22.61	0.425	-0.603	1.094	35	0.8	24.24	0.0179
continued on next part											

AC	Engine	T_0	BPR	PR	C_{ff3}	C_{ff2}	C_{ff1}	h_{cr}	M_{cr}	T_{cr}	SFC_{cr}
B737	CFM56-7B22	100.97	5.30	24.41	0.473	-0.624	1.176	35	0.8	24.24	
B737	CFM56-7B24	107.65	5.20	25.78	0.471	-0.591	1.226	35	0.8	24.38	0.0178
B737	CFM56-7B26	116.99	5.10	27.61	0.511	-0.578	1.291	35	0.8	24.38	
B737	CFM56-7B27	121.44	5.00	28.63	0.579	-0.629	1.337				
B738	CFM56-7B24	107.65	5.20	25.78	0.471	-0.591	1.226	35	0.8	24.38	0.0178
B738	CFM56-7B26	116.99	5.10	27.61	0.511	-0.578	1.291	35	0.8	24.38	
B738	CFM56-7B27	121.44	5.00	28.63	0.579	-0.629	1.337				
B739	CFM56-7B24	107.65	5.20	25.78	0.471	-0.591	1.226	35	0.8	24.38	0.0178
B739	CFM56-7B26	116.99	5.10	27.61	0.511	-0.578	1.291	35	0.8	24.38	
B739	CFM56-7B27	121.44	5.00	28.63	0.579	-0.629	1.337				
B744	PW4062	275.80	4.60	31.00	2.007	-2.169	2.889				
B744	CF6-80C2B1F	254.26	5.10	30.13	0.716	-0.638	2.349	35	0.8	57.03	0.016
B744	RB211-524G	253.00	4.25	32.10	1.881	-2.163	2.910				
B748	GEnx-2B67	299.80	8.00	42.40	1.003	-1.208	2.661				
B752	PW2037	166.35	5.71	26.70	0.556	-0.723	1.743	35	0.85	28.91	0.0165
B752	RB211-535E4	178.40	4.10	26.00	1.052	-1.279	2.081				
B763	PW4056	249.10	4.70	29.30	0.803	-0.909	2.453				
B763	CF6-80C2B2	231.08	5.10	27.53	0.595	-0.607	2.141	35	0.8	53.38	0.0163
B772	GE90-94B	430.92	8.33	40.82	1.201	-0.842	3.162				
B772	PW4077	343.00	6.70	33.20	1.079	-1.030	2.974				
B772	Trent 895	413.05	5.70	41.52	2.412	-2.482	4.107				
B773	GE90-110B1	492.60	7.34	39.73	1.772	-1.458	4.018				
B773	PW4090	408.30	6.00	38.90	3.639	-3.903	4.198				
B773	Trent 892	411.48	5.70	41.38	2.038	-1.909	3.787				
B77W	GE90-115B	513.90	7.08	42.24	2.790	-2.471	4.381				
B788	GEnx-1B70	321.60	8.80	43.50	0.707	-0.542	2.335				
B788	GEnx-1B67	308.70	8.90	41.90	0.676	-0.564	2.261				
B788	GEnx-1B64	298.00	9.00	40.60	0.645	-0.570	2.197				
B788	Trent 1000-E2	268.00	9.62	36.32	0.982	-1.298	2.237				
B788	Trent 1000-C2	334.70	9.16	44.10	0.922	-0.938	2.527				
B789	Trent 1000-K2	350.90	9.04	46.08	0.996	-0.949	2.624				
B789	Trent 1000-J2	350.90	9.04	46.08	0.996	-0.949	2.624				
B789	Trent 1000-A2	310.90	9.32	41.30	0.951	-1.097	2.440				
C550	JT15D-4	11.12	2.68	10.10	0.154	-0.255	0.271				
E145	AE3007A1	34.91	4.77	17.97	0.206	-0.287	0.466				
E170	CF34-8E5	59.68	5.13	23.18	0.299	-0.346	0.701				
E170	CF34-8E6	59.68	5.13	23.18	0.299	-0.346	0.701				
E190	CF34-10E5	77.40	5.09	25.60	0.310	-0.400	0.885				
E195	CF34-10E5	77.40	5.09	25.60	0.310	-0.400	0.885				
E75L	CF34-8E5	59.68	5.13	23.18	0.299	-0.346	0.701				
E75L	CF34-8E6	59.68	5.13	23.18	0.299	-0.346	0.701				

Table C.1: Engine performance for common aircraft types.

Unit for thrust: kN

Unit for SFC: kg/s

Unit for altitude: k ft

References

- [1] Aldrich, J. “R.A. Fisher and the making of maximum likelihood 1912-1922”. In: *Statistical Science* 12.3 (1997), pp. 162–176. DOI: 10.1214/ss/1030037906.
- [2] Alligier, R., Gianazza, D., and Durand, N. “Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights”. In: *Transportation Research Part C: Emerging Technologies* 36 (2013), pp. 45–60. DOI: 10.1016/j.trc.2013.08.006.
- [3] Alligier, R., Gianazza, D., Ghasemi Hamed, M., and Durand, N. “Comparison of Two Ground-based Mass Estimation Methods on Real Data”. In: *6th International Conference on Research in Air Transportation (ICRAT)*. 2014, pp. 1–8.
- [4] Alligier, R., Gianazza, D., and Durand, N. “Machine Learning and Mass Estimation Methods for Ground-Based Aircraft Climb Prediction”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015), pp. 3138–3149. DOI: 10.1109/TITS.2015.2437452.
- [5] Anderson, J. D. *Aircraft performance and design*. McGraw-Hill Science/Engineering/Math, 1999.
- [6] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. In: *Signal Processing, IEEE Transactions on* 50.2 (2002), pp. 174–188. DOI: 10.1109/78.978374.
- [7] Barker, D. M., Huang, W., Guo, Y.-R., Bourgeois, A. J., and Xiao, Q. N. “A Three-Dimensional Variational Data Assimilation System for MM5: Implementation and Initial Results”. In: *Monthly Weather Review* 132.4 (2004), pp. 897–914.
- [8] Bartel, M. and Young, T. M. “Simplified Thrust and Fuel Consumption Models for Modern Two-Shaft Turbofan Engines”. In: *Journal of Aircraft* 45.4 (2008), pp. 1450–1456. DOI: 10.2514/1.35589.
- [9] Bayes, T., Price, R., and Canton, J. *An essay towards solving a problem in the doctrine of chances*. C. Davis, Printer to the Royal Society of London, 1763.
- [10] Ben-Kiki, O., Evans, C., and Ingerson, B. “Yaml ain’t markup language (YAML) version 1.1”. In: *yaml.org, Tech. Rep* (2005), p. 23.
- [11] Blank, D. and Shinar, J. “Efficient combinations of numerical techniques applied to aircraft turning performance optimization”. In: *Journal of Guidance, Control, and Dynamics* 5.2 (1982), pp. 124–130.
- [12] Blom, H. A. and Bloem, E. A. “Exact Bayesian and particle filtering of stochastic hybrid systems”. In: *IEEE Transactions on Aerospace and Electronic Systems* 43.1 (2007), pp. 55–70. DOI: 10.1109/TAES.2007.357154.
- [13] Blomenhofer, H., Pawlitzki, A., Rosenthal, P., and Escudero, L. “Space-based automatic dependent surveillance broadcast (ads-b) payload for in-orbit demonstration”. In: *Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012 6th*. IEEE, 2012, pp. 160–165.

- [14] Brito, A. *Blender 3D: architecture, buildings, and scenery: create photorealistic 3D architectural visualizations of buildings, interiors, and environmental scenery*. Packt Publishing Ltd, 2008.
- [15] Calders, P. "G.A.M.E. Aircraft Performance Model Description". In: *Eurocontrol, DIS/ATD Unit, DOC.CoE-TP-02002* (2002).
- [16] Chati, Y. S. and Balakrishnan, H. "Data-Driven Modeling of Aircraft Engine Fuel Burn in Climb Out and Approach". In: *Transportation Research Record* (2018), p. 0361198118780876.
- [17] Chati, Y. S. and Balakrishnan, H. "Modeling of Aircraft Takeoff Weight Using Gaussian Processes". In: *Journal of Air Transportation* (2018), pp. 1–10.
- [18] Chati, Y. S. and Balakrishnan, H. "Statistical Modeling of Aircraft Engine Fuel Flow Rate". In: *30th Congress of the International Council of the Aeronautical Science*. 2016.
- [19] Chati, Y. S. and Balakrishnan, H. "Statistical Modeling of Aircraft Takeoff Weight". In: *Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar*. FAA/EUROCONTROL. 2017.
- [20] Chatterji, G. "Fuel Burn Estimation Using Real Track Data". In: *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference* (2011). DOI: 10.2514/6.2011-6881.
- [21] Choi, S. C. and Wette, R. "Maximum likelihood estimation of the parameters of the gamma distribution and their bias". In: *Technometrics* 11.4 (1969), pp. 683–690.
- [22] Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. "Meshlab: an open-source mesh processing tool." In: *Eurographics Italian Chapter Conference*. Vol. 2008. 2008, pp. 129–136.
- [23] Clements, J. C. "Real-time optimal control of aircraft turn trajectories". In: *Journal of guidance, control, and dynamics* 16.2 (1993), pp. 385–387.
- [24] Commercial Aviation Safety Team / Common Taxonomy Team. *Phase of Flight: Definitions and Usage Notes*. 2013, pp. 1–11.
- [25] Coppenbarger, R. "Climb trajectory prediction enhancement using airline flight-planning information". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 1999, pp. 1–11. DOI: 10.2514/6.1999-4147.
- [26] Dalmau, R., Pérez-Batlle, M., and Prats, X. "Estimation and prediction of weather variables from surveillance data using spatio-temporal Kriging". In: *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*. Vol. 2017-Septe. IEEE. 2017, pp. 1–8. DOI: 10.1109/DASC.2017.8102132.
- [27] Dalmau, R. and Prats, X. "Fuel and time savings by flying continuous cruise climbs. Estimating the benefit pools for maximum range operations". In: *Transportation Research Part D: Transport and Environment* 35 (2015), pp. 62–71. DOI: 10.1016/j.trd.2014.11.019.
- [28] Dalmau, R., Prats, X., Ramonjoan, A., and Soley, S. "Estimating fuel consumption from radar tracks: a validation exercise using FDR data from descent trajectories". In: *8th International Conference on Research in Air Transportation*. 2018.
- [29] Delahaye, D., Puechmorel, S., and Vacher, P. "Windfield estimation by radar track Kalman filtering and vector spline extrapolation". In: *Digital Avionics Systems Conference 2003*. Vol. 1. IEEE. 2003. DOI: 10.1109/DASC.2003.1245869.

- [30] Delahaye, D. and Puechmorel, S. "TAS and wind estimation from radar data". In: *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*. IEEE. 2009, 2-B. DOI: 10.1109/DASC.2009.5347547.
- [31] Delahaye, D. and Puechmorel, S. "Aircraft local wind estimation from radar tracker data". In: *2008 10th International Conference on Control, Automation, Robotics and Vision, ICARCV 2008*. IEEE. 2008, pp. 1033–1038. DOI: 10.1109/ICARCV.2008.4795661.
- [32] Douc, R., Cappé, O., and Moulines, E. "Comparison of Resampling Schemes for Particle Filtering". In: *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. IEEE. 2005, pp. 64–69. DOI: 10.1109/ISPA.2005.195385. eprint: 0507025 (cs).
- [33] Doucet, A., Freitas, N. de, and Gordon, N. "An Introduction to Sequential Monte Carlo Methods". In: *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14. DOI: 10.1007/978-1-4757-3437-9_1.
- [34] Ellerbroek, J., Inaad, M., and Hoekstra, J. M. "Fuel and Emission Benefits for Continuous Descent Approaches at Schiphol". In: *8th International Conference on Research in Air Transportation*. 2018.
- [35] Es, G. W. H. van. "Rapid estimation of the zero-lift drag coefficient of transport aircraft". In: *Journal of aircraft* 39.4 (2002), pp. 597–599.
- [36] Ester, M., Kriegel, H. P., Sander, J., and Xu, X. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Second International Conference on Knowledge Discovery and Data Mining* (1996), pp. 226–231.
- [37] EuroControl. *EuroControl Aircraft Performance Database*. 2017. URL: <https://contentzone.eurocontrol.int/aircraftperformance>.
- [38] Falck, B., Falck, D., and Collette, B. *Freecad [How-To]*. Packt Publishing Ltd, 2012.
- [39] FlightRadar24 AB. *FlightRadar24, Terms and Conditions, Item 9 (accessed on August 2017)*. 2017. URL: <http://www.flightradar24.com/terms-and-conditions>.
- [40] Fricke, H., Seiß, C., and Herrmann, R. "Fuel and Energy Benchmark Analysis of Continuous Descent Operations". In: *Air Traffic Control Quarterly* 23.1 (2015), pp. 83–108.
- [41] Fu, T. C. "A review on time series data mining". In: *Engineering Applications of Artificial Intelligence* 24.1 (2011), pp. 164–181. DOI: 10.1016/j.engappai.2010.09.007.
- [42] Gallo, E., Navarro, F. A., Nuic, A., and Iagaru, M. "Advanced aircraft performance modeling for ATM: BADA 4.0 results". In: *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*. IEEE. 2006, pp. 1–12.
- [43] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian data analysis*. Vol. 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.
- [44] Gertz, J. L. *Fundamentals of Mode S Parity Coding*. Tech. rep. Massachusetts Institute of Technology, Lincoln Laboratory, 1984.
- [45] Gnanadesikan, R., Pinkham, R. S., and Hughes, L. P. "Maximum likelihood estimation of the parameters of the beta distribution from smallest order statistics". In: *Technometrics* 9.4 (1967), pp. 607–620.

- [46] Grappel, R. D., Harris, G. S., Kozar, M. J., and Wiken, R. T. "Elementary Surveillance (ELS) and Enhanced Surveillance (EHS) Validation via Mode S Secondary Radar Surveillance". In: *Project Report ATC-337, Lincoln Lab., MIT* April (2008).
- [47] Gur, O., Mason, W. H., and Schetz, J. A. "Full-configuration drag estimation". In: *Journal of Aircraft* 47.4 (2010), pp. 1356–1367.
- [48] Haan, S. de. *Availability and quality of Mode-S MRAR (BDS4. 4) in the MUAC area: a first study*. Tech. rep. IR 2014-01. KNMI, the Netherlands, 2014.
- [49] Haan, S. de. "High-resolution wind and temperature observations from aircraft tracked by Mode-S air traffic control radar". In: *Journal of Geophysical Research: Atmospheres* 116.D10 (2011).
- [50] Haan, S. de, Haij, M. de, and Sondij, J. *The use of a commercial ADS-B receiver to derive upper air wind and temperature observations from Mode-S EHS information in The Netherlands*. Tech. rep. TR-336. KNMI, the Netherlands, 2013.
- [51] Hempe, D. W. "Airworthiness Approval of Automatic Dependent Surveillance-Broadcast (ADS-B) Out Systems". In: *Advisory Circular 20-165* (2010).
- [52] Herndon, A. A., Cramer, M., and Sprong, K. "Analysis of advanced flight management systems (fms), flight management computer (fmc) field observations trials, radius-to-fix path terminators". In: *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*. IEEE, 2008, 2–A.
- [53] Hoberman, S. *Data Modeling for MongoDB: Building Well-Designed and Supportable MongoDB Databases*. Technics Publications, 2014, p. 226.
- [54] Hoekstra, J. M. and Ellerbroek, J. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *7th International Conference on Research in Air Transportation*. 2016, pp. 1–8.
- [55] Hoerner, S. F. *Aerodynamic Drag: Practical Data on Aerodynamic Drag Evaluated and Presented*. Otterbein Press, 1951, p. 259.
- [56] Hoerner, S. F. *Fluid-dynamic Drag: practical information on aerodynamic drag and hydrodynamic resistance*. Hoerner Fluid Dynamics, 1958.
- [57] Hoffman, M. D. and Gelman, A. "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.
- [58] Hollister, W. M., Bradford, E. R., and Welch, J. D. "Using aircraft radar tracks to estimate wind aloft". In: *The Lincoln Laboratory Journal* 2.3 (1989), pp. 555–565.
- [59] Houtekamer, P. L. and Mitchell, H. "Data Assimilation Using an Ensemble Kalman Filter Technique". In: *Monthly Weather Review* 126.3 (1998), pp. 796–811.
- [60] Hrastovec, M. and Solina, F. "Prediction of aircraft performances based on data collected by air traffic control centers". In: *Transportation Research Part C: Emerging Technologies* 73 (2016), pp. 167–182. DOI: 10.1016/j.trc.2016.10.018.
- [61] Hrastovec, M. and Solina, F. "Obtaining meteorological data from aircraft with mode-S radars". In: *IEEE Aerospace and Electronic Systems Magazine* 28.12 (2013), pp. 12–24. DOI: 10.1109/MAES.2013.6693664.
- [62] Hurter, C., Alligier, R., Gianazza, D., Puechmorel, S., Andrienko, G., and Andrienko, N. "Wind parameters extraction from aircraft trajectories". In: *Computers, Environment and Urban Systems* 47 (2014), pp. 28–43. DOI: 10.1016/j.compenvurbsys.2014.01.005.

- [63] ICAO. "Aircraft Engine Emissions Databank". In: *International Civil Aviation Organization* (2016).
- [64] ICAO. *Annex 10 to the Convention on International Civil Aviation, Aeronautical Telecommunications*. International Civil Aviation Organization, 2002.
- [65] ICAO. "DOC 8643 - Aircraft Type Designators". In: *International Civil Aviation Organization* (2018).
- [66] ICAO. *Technical Provisions for Mode S Services and Extended Squitter*. International Civil Aviation Organization, 2008.
- [67] ICAO. *Technical Provisions for Mode S Services and Extended Squitter, 2nd Edition*. International Civil Aviation Organization, 2012.
- [68] Inger, G. "Application of Oswatitsch's theorem to supercritical airfoil drag calculation". In: *Journal of aircraft* 30.3 (1993), pp. 415–416.
- [69] Jackson, M. R., Zhao, Y. J., and Slattery, R. A. "Sensitivity of Trajectory Prediction in Air Traffic Management". In: *Journal of Guidance, Control, and Dynamics* 22.2 (1999), pp. 219–228. DOI: 10.2514/2.4398.
- [70] Jenkinson, L. R., Simpkin, P., Rhodes, D., Jenkison, L. R., and Royce, R. *Civil jet aircraft design*. Vol. 338. Arnold London, 1999.
- [71] Jong, P. M. A. de, Laan, J. J. van der, Veld, A. C. i. ', Paassen, M. M. van, and Mulder, M. "Wind-Profile Estimation Using Airborne Sensors". In: *Journal of Aircraft* 51.6 (2014), pp. 1852–1863. DOI: 10.2514/1.C032550.
- [72] Justusson, B. "Median filtering: Statistical properties". In: *Two-Dimensional Digital Signal Processing II*. Springer, 1981, pp. 161–196.
- [73] Kagkaras, A. "Laser scanning modelling of a Cessna citation: For Computational Fluid Dynamics (CFD) Studies". In: *Repository of TU Delft* (2006).
- [74] Kanamitsu, M. "Description of the NMC Global Data Assimilation and Forecast System". In: *Weather and Forecasting* 4.3 (1989), pp. 335–342.
- [75] Kapoor, A., Horvitz, Z., Laube, S., and Horvitz, E. "Airplanes aloft as a sensor network for wind forecasting". In: *IPSN 2014 - Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (Part of CPS Week)* (2014), pp. 25–33. DOI: 10.1109/IPSN.2014.6846738.
- [76] Kaune, R., Steffes, C., Rau, S., Konle, W., and Pagel, J. "Wide area multilateration using ADS-B transponder signals". In: *Proc. of FUSION'12 - 15th IEEE International Conference on Information Fusion*. IEEE, 2012, pp. 727–734.
- [77] Koelle, R. "Open source software and crowd sourced data for operational performance analysis". In: *Proceedings of the 12th ATM Seminar*. 2017.
- [78] Kroo, I. and Shevell, R. "Aircraft design: Synthesis and analysis". In: *Desktop Aeronautics Inc., Textbook Version 0.99* (2001).
- [79] Lednicer, D. *The incomplete guide to airfoil usage [Webpage, accessed on May, 2018]*. 2010. URL: <http://m-selig.ae.illinois.edu/ads/aircraft.html>.
- [80] Lee, H. T. and Chatterji, G. "Closed-Form Takeoff Weight Estimation Model for Air Transportation Simulation". In: *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. 2010, p. 9156. DOI: 10.2514/6.2010-9156.
- [81] Leege, A. M. P. de, Van Paassen, M. M., and Mulder, M. "Using automatic dependent surveillance-broadcast for meteorological monitoring". In: *Journal of Aircraft* 50.1 (2012), pp. 249–261.

- [82] Lin, J., Brunner, D., Gerbig, C., Stohl, A., Luhar, A., and Webley, P. *Lagrangian modeling of the atmosphere*. Vol. 200. John Wiley & Sons, 2013.
- [83] Liu, J. S. and Chen, R. “Sequential Monte Carlo methods for dynamic systems”. In: *Journal of the American statistical association* 93.443 (1998), pp. 1032–1044.
- [84] Loftin, L. K. *Quest for performance: The evolution of modern aircraft*. 468. Scientific, Technical Information Branch, National Aeronautics, and Space Administration, 1985.
- [85] Lütkepohl, H. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005. DOI: 10.1007/3-540-27752-8. eprint: arXiv:1011.1669v3.
- [86] Mair, W. A. and Birdsall, D. L. *Aircraft performance*. Vol. 5. Cambridge University Press, 1996.
- [87] Mair, W. A. and Edwards, B. J. (Engineering Department, C. U. “A Parametric Study of Take-Off and Landing Distances for High-Lift Aircraft”. In: *Aeronautical Research Council C. P. No. 823* (1965).
- [88] Mason, W. “Analytic models for technology integration in aircraft design”. In: *Aircraft Design, Systems and Operations Conference*. 1990, p. 3262.
- [89] Massey Jr, F. J. “The Kolmogorov-Smirnov test for goodness of fit”. In: *Journal of the American statistical Association* 46.253 (1951), pp. 68–78.
- [90] McCormick, B. W. *Aerodynamics, aeronautics, and flight mechanics*. Vol. 2. Wiley New York, 1995.
- [91] McGee, T., Spry, S., and Hedrick, K. “Optimal path planning in a constant wind with a bounded turning rate”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2005, p. 6186.
- [92] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.
- [93] Metz, I., Hoekstra, J., Ellerbroek, J., and Kügler, D. “Aircraft Performance for Open Air Traffic Simulations”. In: *AIAA Modeling and Simulation Technologies Conference*. 2016, p. 3522.
- [94] Milligan, G. and Cooper, M. “A study of standardization of variables in cluster analysis”. In: *Journal of Classification* 5.2 (1988), pp. 181–204.
- [95] Mohleji, S. and Wang, G. “Modeling ADS-B Position and Velocity Errors for Airborne Merging and Spacing in Interval Management Application”. In: *Center for Advanced Systems Development (CASD), ...* (2010), pp. 10–3026.
- [96] Moninger, W. R., Mamrosh, R. D., and Pauley, P. M. “Automated meteorological reports from commercial aircraft”. In: *Bulletin of the American Meteorological Society* 84.2 (2003), pp. 203–216+149. DOI: 10.1175/BAMS-84-2-203.
- [97] Moradkhani, H., Hsu, K.-L., Gupta, H., and Sorooshian, S. “Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter”. In: *Water Resources Research* 41.5 (2005). DOI: 10.1029/2004WR003604.
- [98] Musso, C., Oudjane, N., and Gland, F. “Improving Regularised Particle Filters”. In: *Sequential Monte Carlo Methods in Practice*. Springer, 2001, pp. 247–271. DOI: 10.1007/978-1-4757-3437-9_12.

- [99] Nagel, J. B. and Sudret, B. “Bayesian Multilevel Model Calibration for Inverse Problems Under Uncertainty with Perfect Data”. In: *Journal of Aerospace Information Systems* 12.1 (2015), pp. 97–113. DOI: 10.2514/1.I010264.
- [100] Niță, M. and Scholz, D. *Estimating the oswald factor from basic aircraft geometrical parameters*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2012.
- [101] Noschese, P., Porfili, S., and Di Girolamo, S. “Ads-b via iridium next satellites”. In: *Digital Communications-Enhanced Surveillance of Aircraft and Vehicles (TIWDC/E-SAV), 2011 Tyrrhenian International Workshop on*. IEEE, 2011, pp. 213–218.
- [102] Nuic, A. *Base of Aircraft DATA (BADA) Product Management Document*. Tech. rep. EEC Technical/Scientific Report No. 2009-008. Tech. Rep. EEC Technical/Scientific Report, 2009.
- [103] Nuic, A. “User manual for the Base of Aircraft Data (BADA) revision 3.12”. In: *EUROCONTROL* (2014).
- [104] Obert, E. and Slingerland, R. *Aerodynamic design of transport aircraft*. Ios Press, 2009, pp. 527–529.
- [105] Paul, J. “Jane’s All The World’s Aircraft. 2004-2005”. In: *Jane’s Information Group Inc, Alexandria* (2005).
- [106] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [107] Peel, R. A. *Data Development Documentation*. URL: https://developer.x-plane.com/docs/data-development-documentation/#Terms_of_use.
- [108] Purpura, S. *747-400 Desktop Aero Demo [Data file]*. 2013. URL: <http://hangar.openvsp.org/vspfiles/184>.
- [109] Rasmussen, C. E. “Gaussian Processes in Machine Learning”. In: *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71. DOI: 10.1007/978-3-540-28650-9_4. eprint: 026218253X.
- [110] Rawlins, F., Ballard, S. P., Bovis, K. J., Clayton, A. M., Li, D., Inverarity, G. W., Lorenc, A. C., and Payne, T. J. “The Met Office global four-dimensional variational data assimilation scheme”. In: *Quarterly Journal of the Royal Meteorological Society* 133.623 (2007), pp. 347–362.
- [111] RTCA. “Minimum operational performance standards for 1090MHz extended squitter automatic dependent surveillance-broadcast (ADS-B) and traffic information services-broadcast (TIS-B)[J]”. In: *RTCA DO-260B with Corrigendum 1.1* (2011), pp. 1365–1372.
- [112] Ruijgrok, G. *Elements of airplane performance*. Vol. 2. Delft Academic Press, 2009.
- [113] Saha, S., Moorthi, S., Pan, H.-L., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., et al. “The NCEP climate forecast system reanalysis”. In: *Bulletin of the American Meteorological Society* 91.8 (2010), pp. 1015–1057.
- [114] Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. “Probabilistic programming in Python using PyMC3”. In: *PeerJ Computer Science* 2 (2016), e55.
- [115] Savitzky, A. and Golay, M. J. E. “Smoothing and differentiation of data by simplified least squares procedures.” In: *Analytical chemistry* 36.8 (1964), pp. 1627–1639.

- [116] Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M. "Bringing up OpenSky: A large-scale ADS-B sensor network for research". In: *Proceedings of the 13th international symposium on Information processing in sensor networks*. IEEE Press. 2014, pp. 83–94.
- [117] Schöberl, J. "NETGEN An advancing front 2D/3D-mesh generator based on abstract rules". In: *Computing and visualization in science* 1.1 (1997), pp. 41–52.
- [118] Schultz, C. a., Thippavong, D., and Erzberger, H. "Adaptive Trajectory Prediction Algorithm for Climbing Flights". In: *AIAA Guidance, Navigation, and Control Conference*. August. 2012, p. 2. DOI: 10.2514/6.2012-4931.
- [119] Selig, M. *UIUC airfoil coordinates database [accessed on May 17, 2018]*. 2018. URL: <http://m-selig.ae.illinois.edu/ads.html>.
- [120] Singh, T. and Darshan, V. S. "A Modern Data Architecture with Apache Hadoop". In: *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*. IEEE. 2015, pp. 574–579.
- [121] Sobolewski, J. S. "Cyclic Redundancy Check". In: *Encyclopedia of Computer Science*. John Wiley and Sons Ltd., pp. 476–479. ISBN: 0-470-86412-5.
- [122] Stone, E. K. and Kitchen, M. "Introducing an approach for extracting temperature from aircraft GNSS and pressure altitude reports in ADS-B messages". In: *Journal of Atmospheric and Oceanic Technology* 32.4 (2015), pp. 736–743. DOI: 10.1175/JTECH-D-14-00192.1.
- [123] Sun, J., Ellerbroek, J., and Hoekstra, J. "Aircraft initial mass estimation using Bayesian inference method". In: *Transportation Research Part C: Emerging Technologies* 90 (2018), pp. 59–73. DOI: 10.1016/j.trc.2018.02.022.
- [124] Sun, J., Blom, H. A. P., Ellerbroek, J., and Hoekstra, J. M. "Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method". In: *8th International Conference on Research in Air Transportation*. 2018.
- [125] Sun, J., Ellerbroek, J., and Hoekstra, J. "Bayesian Inference of Aircraft Initial Mass". In: *Europe Air Traffic Management Research and Development Seminar*. 2017.
- [126] Sun, J., Ellerbroek, J., and Hoekstra, J. "Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets". In: *Journal of Aerospace Information Systems* 14.10 (2017), pp. 566–571. DOI: 10.2514/1.I010520.
- [127] Sun, J., Ellerbroek, J., and Hoekstra, J. "Large-scale flight phase identification from ads-b data using machine learning methods". In: *7th International Conference on Research in Air Transportation*. 2016.
- [128] Sun, J., Ellerbroek, J., and Hoekstra, J. "Modeling Aircraft Performance Parameters with Open ADS-B Data". In: *12th USA/Europe ATM R&D Seminar* (2017).
- [129] Sun, J., Ellerbroek, J., and Hoekstra, J. "Modeling and Inferring Aircraft Takeoff Mass from Runway ADS-B Data". In: *7th International Conference on Research in Air Transportation*. 2016.
- [130] Sun, J., Ellerbroek, J., and Hoekstra, J. M. "WRAP: An open-source kinematic aircraft performance model". In: *Transportation Research Part C: Emerging Technologies* 98 (2019), pp. 118–138.
- [131] Sun, J., Hoekstra, J. M., and Ellerbroek, J. "Aircraft Drag Polar Estimation Based on a Stochastic Hierarchical Model". In: *8th International Conference on Research in Air Transportation*. 2018.

- [132] Sun, J., Vũ, H., Ellerbroek, J., and Hoekstra, J. “Ground-based Wind Field Construction from Mode-S and ADS-B Data with a Novel Gas Particle Model”. In: *Seventh SESAR Innovation Days 28* (2017), 30th.
- [133] Sun, J., Vũ, H., Ellerbroek, J., and Hoekstra, J. “PyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019). DOI: 10.1109/TITS.2019.2914770.
- [134] Sun, J., Vũ, H., Ellerbroek, J., and Hoekstra, J. M. “Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model”. In: *PloS one* 13.10 (2018), e0205029.
- [135] Sun, J., Vũ, H., JoseAndresMR, Orta, F., Hornbostel, K. von, Robitaille, T., Nobrega, R., and Watterson, J. *junzis/pyModeS: The Python Decoder for ADS-B and Mode-S Comm-B*. 2018. DOI: 10.5281/zenodo.894419. URL: <https://doi.org/10.5281/zenodo.1319559>.
- [136] Svoboda, C. “Turbofan engine database as a preliminary design tool”. In: *Aircraft design* 3.1 (2000), pp. 17–31.
- [137] Thippavong, D. P., Schultz, C. A., Lee, A. G., and Chan, S. H. “Adaptive algorithm to improve trajectory prediction accuracy of climbing aircraft”. In: *Journal of Guidance, Control, and Dynamics* 36.1 (2012), pp. 15–24.
- [138] Torenbeek, E. *Synthesis of subsonic airplane design*. Delft University Press, 1976.
- [139] Trim, R. M. “Mode S: an introduction and overview (secondary surveillance radar)”. In: *Electronics & Communication Engineering Journal* 2.2 (1990), pp. 53–59.
- [140] Wang, Z., Liang, M., and Delahaye, D. “Short-term 4D Trajectory Prediction Using Machine Learning Methods”. In: *Sesar innovation days 2017*. November. 2017, pp. 1–9.
- [141] Weiser, A. and Zarantonello, S. E. “A Note on Piecewise Linear and Multilinear Table Interpolation in Many Dimensions”. In: *Mathematics of Computation* 50.181 (1988), pp. 189–196. DOI: 10.1090/S0025-5718-1988-0917826-0.
- [142] Werner, K., Bredemeyer, J., and Delovski, T. “ADS-B over satellite: Global air traffic surveillance from space”. In: *Digital Communications-Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), 2014 Tyrrhenian International Workshop on*. IEEE, 2014, pp. 47–52.
- [143] Wetmore, J. W. “The Rolling Friction of Several Airplane Wheels and Tires and the Effect of Rolling Friction on Take-Off”. In: 1937.
- [144] Williams, C. K. I. “Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond”. In: *Learning in Graphical Models*. Springer, 1998, pp. 599–621. DOI: 10.1007/978-94-011-5014-9_23.
- [145] Young, T. M. *Performance of the Jet Transport Airplane: Analysis Methods, Flight Operations, and Regulations*. John Wiley & Sons, 2017.
- [146] Zadeh, L. A. “Fuzzy sets”. In: *Information and Control* 8.3 (1965), pp. 338–353.

Nomenclature

Symbols

α	angle of attack	deg
ν	process noise	
f	state transition function	
h	observation function	
n	observation noise	
χ	track angle	deg
δ	Dirac delta function	
δ_T	thrust setting	-
γ	flight path angle	deg
λ	bypass ratio	-
\mathbf{x}	state vector	
\mathbf{y}	observation vector	
VS	vertical speed	m/s
ϕ	bank angle	deg
ψ	heading angle	deg
ρ	air density	kg/m ³
ρ_0	air density at sea level	kg/m ³
τ	air temperature	K
τ_0	air temperature at sea level	K
A	aspect ratio	-
a	speed of sound	m/s
C_{ff}	fuel flow coefficient	-
C_D	drag coefficient	-
C_L	lift coefficient	-
C_{D0}	zero drag coefficient	-
$C_{L,max}$	maximum lift coefficient	-

D	total drag	N
e	Oswald efficiency factor	-
F_c	centrifugal force	N
g	gravitational acceleration	m/s ²
h	altitude	m
h_{cr}	reference cruise altitude	alt
k	lift induced drag coefficient	-
L	total lift	N
M	Mach number	-
m	aircraft mass	kg
M_{cr}	reference cruise Mach number	-
n	load factor	-
p	pressure	Pa
p_0	pressure at sea level	Pa
p_{cr}	pressure at reference cruise altitude	Pa
p_{10}	pressure at altitude of 10,000 feet	Pa
q	dynamic pressure	Pa
R	universal gas constant	-
S	wing area	m ²
T	net thrust	N
T_0	maximum static thrust, sea level	N
T_{cr}	thrust at the top of climb	N
T_{10}	thrust at altitude of 10,000 feet	N
u	U-component of wind	m/s
V	aircraft speed	m/s
v	V-component of wind	m/s
v	aircraft speed	m/s
V_a	aircraft airspeed	m/s
v_a	aircraft airspeed	m/s
V_g	aircraft ground speed	m/s
v_g	aircraft ground speed	m/s
V_w	wind speed	m/s

v_w	wind speed	m/s
w	particle weight	
x	downrange position	m
y	cross-range position	m
z	vertical position	m

Abbreviations

ACAS	Airborne Collision Avoidance System
ADS-B	Automatic Dependent Surveillance-Broadcast
AMDR	Aircraft Meteorological Data Relay
ATC	Air Traffic Control
ATM	Air Traffic Management
BADA	Base of Aircraft Data
BDS	Comm-B Data Selector
CAS	Calibrated Airspeed
CFD	Computational Fluid Dynamics
CL	Climb
CR	Cruise
CRC	Cyclic Redundancy Check
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DE	Descent
ECMWF	European Centre for Medium-Range Weather Forecasts
EHS	Enhanced Surveillance
ELS	Elementary Surveillance
FA	Final Approach
FAA	Federal Aviation Administration
FMS	Flight Management System
GFS	Global Forecast System
GPS	Global Positions System
GS	Ground Speed
HFOM	Horizontal Figure of Merit

HMC	Hamiltonian Monte Carlo
IAS	Indicated Airspeed
IC	Initial Climb
ICAO	International Civil Aviation Organization
LD	Landing
LOF	Liftoff
MAE	Mean Absolute Error
MAP	Maximum A Posteriori estimation
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimation
MLW	Maximum Landing Weight
Mode-S	Mode Selective
MRAR	Meteorological Routine Air Report
MTOW	Maximum Takeoff Weight
NAC	Navigation Accuracy Category
NIC	Navigation Integrity Category
NUTS	No-U-Turn Sampler
NWP	Numerical Weather Prediction
OEW	Operational Empty Weight
RMSE	Root Mean Squared Error
SIR	Sample Importance Re-sampling
SMC	Sequential Monte Carlo
SSR	Secondary Surveillance Radar
STE	Stochastic Total Energy
TAS	True Airspeed
TCAS	Traffic Collision Avoidance System
TO	Takeoff
TSFC	Thrust Specific Fuel Consumption
VFOM	Vertical Figure of Merit

Curriculum Vitae

Junzi Sun

05/Dec/1985 Born in Chaohu, Anhui, China.

Education

- 2003–2007 B.Sc., Electronic Information Engineering
Beijing University of Posts and Telecommunications
- 2007–2010 M.Sc., Aerospace Engineering
Polytechnic University of Catalonia
Thesis: Ground Stations Scheduling with Genetic Algorithm
- 2015–2019 Ph.D., Aerospace Engineering
Delft University of Technology
Thesis: Open Aircraft Performance Modeling
Promotor: Prof.dr.ir. J.M. Hoekstra
Co-promotor: Dr.ir. J. Ellerbroek

Work Experience

- 2007–2011 Research Engineer
Aerospace Research and Technology Center
Barcelona, Spain
- 2011–2012 Research Engineer
ASCAMM Foundation
Barcelona, Spain
- 2012–2014 Academic Coordinator
International Space University
Strasbourg, France

List of research outputs

from 2016 to 2019, as of May 2019

Journal

1. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2017. *Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets*. Journal of Aerospace Information Systems, pp.566-572.
2. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2018. *Aircraft initial mass estimation using Bayesian inference method*. Transportation Research Part C: Emerging Technologies, 90, pp.59-73.
3. Sun, J., Vũ, H., Ellerbroek, J. and Hoekstra, J.M., 2018. *Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model*. PloS one, 13(10), p.e0205029.
4. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2019. *WRAP: An open-source kinematic aircraft performance model*. Transportation Research Part C: Emerging Technologies, 98, pp.118-138.
5. Sun, J., Vũ, H., Ellerbroek, J. and Hoekstra, J.M., 2019. *PyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research*. IEEE Transactions on Intelligent Transportation Systems, accepted for publication.
6. Sun, J., Blom, H.A., Ellerbroek, J. and Hoekstra, J.M., 2019. *Particle filter for aircraft mass estimation and uncertainty modeling*. Transportation Research Part C: Emerging Technologies, under review.
7. Sun, J., Hoekstra, J.M., and Ellerbroek, J., 2019. *Estimating aircraft drag polar using open flight surveillance data and a novel stochastic total energy model*. Transportation Research Part C: Emerging Technologies, to be submitted.
8. Sun, J., Hoekstra, J.M., and Ellerbroek, J., 2019. *OpenAP: An open-source aircraft performance model for air traffic management and simulations*. Simulation Modelling Practice and Theory, to be submitted.

Conference

1. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2016, June. *Large-Scale Flight Phase Identification from ADS-B Data Using Machine Learning Methods*. In Proceedings of the 7th International Conference on Research in Air Transportation.
2. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2016, June. *Modeling and inferring aircraft takeoff mass from runway ADS-B data*. In Proceedings of the 7th International Conference on Research in Air Transportation.

3. Verbraak, T.L., Ellerbroek, J., Sun, J. and Hoekstra, J.M., 2017, June. *Large-Scale ADS-B Data and Signal Quality Analysis*. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar.
4. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2017, June. *Modeling aircraft performance parameters with open ads-b data*. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar.
5. Sun, J., Ellerbroek, J. and Hoekstra, J.M., 2017, June. *Bayesian inference of aircraft initial mass*. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar. FAA/EUROCONTROL.
6. Sun, J., Vũ, H., Ellerbroek, J. and Hoekstra, J.M., 2017. *Ground-based Wind Field Construction from Mode-S and ADS-B Data with a Novel Gas Particle Model*. In Proceedings of the 7th SESAR Innovation Days, 28, p.30th.
7. Sun, J., Blom, H.A., Ellerbroek, J. and Hoekstra, J.M., 2018, June. *Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method*. In Proceedings of the 8th International Conference on Research in Air Transportation.
8. Sun, J., Hoekstra, J.M. and Ellerbroek, J., 2018, December. *Aircraft Drag Polar Estimation Based on a Stochastic Hierarchical Model*. In Proceedings of the 8th SESAR Innovation Days.

Open-access ebook

1. Sun, J. et al. *The 1090MHz Riddle: A book on decoding ADS-B and Enhanced Mode-S data*. URL: <https://mode-s.org/decode/>.

Open-source software

1. Sun, J., Vũ, H., Andres J., Orta, F., Hornbostel, K., Robitaille, T., Nobrega, R., Waterson, J., et al. *junzis/pyModeS: The Python Decoder for ADS-B and Enhance Mode-S*. URL: <https://github.com/junzis/pyModeS>.
2. Sun, J., *junzis/flight-data-processor: Aircraft flight data processing tools and libraries in Python using machine leaning and fuzzy logic*. URL: <https://github.com/junzis/flight-data-processor>.
3. Sun, J., Vũ, H., *junzis/meteo-particle-model: The Meteo-Particle model in Python*. URL: <https://github.com/junzis/meteo-particle-model>.
4. Sun, J., *junzis/acsmc: Aircraft mass estimator based on sequential Monte Carlo method*. URL: <https://github.com/junzis/acsmc>.
5. Sun, J., *junzis/openap: Open Aircraft Performance Model and Python Toolkit*. URL: <https://github.com/junzis/openap>.