

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**DIGITAL-TWIN FLIGHT MODELLING THROUGH
MACHINE LEARNING
FOR TRAJECTORY ERROR ESTIMATION AND RECOVERY**



Ph.D. THESIS

Mevlüt UZUN

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Engineering Programme

MARCH 2021

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**DIGITAL-TWIN FLIGHT MODELLING THROUGH
MACHINE LEARNING
FOR TRAJECTORY ERROR ESTIMATION AND RECOVERY**

Ph.D. THESIS

**Mevlüt UZUN
(511152115)**

Department of Aeronautics and Astronautics Engineering

Aeronautics and Astronautics Engineering Programme

**Thesis Advisor: Prof. Dr. Gökhan İNALHAN
Thesis Co-Advisor: Dr. Mustafa Umut DEMİREZEN**

MARCH 2021

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**ROTA HESAPLAMALARINDA MAKİNE ÖĞRENMESİ TABANLI
DİJİTAL-İKİZ UÇUŞ MODELİ**

DOKTORA TEZİ

**Mevlüt UZUN
(511152115)**

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

**Tez Danışmanı: Prof. Dr. Gökhan İNALHAN
Eş Danışman: Dr. Mustafa Umut DEMİREZEN**

MART 2021

Mevlüt UZUN, a Ph.D. student of ITU Graduate School student ID 511152115, successfully defended the dissertation entitled “ DIGITAL-TWIN FLIGHT MODELLING THROUGH MACHINE LEARNING FOR TRAJECTORY ERROR ESTIMATION AND RECOVERY”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Gökhan İNALHAN**
Istanbul Technical University

Co-advisor : **Dr. Mustafa Umut DEMİREZEN**
Roketsan A.Ş.

Jury Members : **Prof. Dr. İbrahim ÖZKOL**
Istanbul Technical University

Asst. Prof. Dr. Hayri ACAR
Istanbul Technical University

Dr. Miguel VILAPLANA
Airbus UTM

Prof. Dr. Mehmet Turan SÖYLEMEZ
Istanbul Technical University

Asst. Prof. Dr. Uraz YAVANOĞLU
Gazi University

Date of Submission : 19 February 2021

Date of Defense : 17 March 2021





To my family,



FOREWORD

I consider this thesis as a marathon's sprint phase. The manuscript's content is mostly of my work of the last one year. The time before consists of identifying problems, doing lots of mistakes, and missing a myriad of critical points. Indeed, I am still doing those. However, it took me hours of meetings, countless readings, and sleepless nights to come up with an idea. It reminds me of a story about Pablo Picasso that I love very much: A woman approaches Picasso in a restaurant, asks him to scribble something on a napkin, and says she would be happy to pay whatever he felt it was worth. Picasso complies and then says, "That will be \$10,000.". "But you did that in thirty seconds," the astonished woman replies. "No," Picasso says. "It has taken me forty years to do that."

First and foremost, I would like to thank my research advisor Prof. Gokhan İNALHAN for his support and guidance throughout this study. He has always spared time to me regardless of the time of a day. He has sat with me not only to supervise my research, but also to get his hands dirty with equations. I have done the technical work, but he has been the one that gives me the opportunity in a golden plate. I would also like to thank Prof. Umut DEMİREZEN for his guidance. As a highly experienced machine learning researcher, he accelerated me to level up from zero to state of art in this domain, in a very short time. Along with his positive, encouraging, and dedicated character, he has been more of a close friend than an advisor to me. I also thank to my dear friends in Aerospace Research Center, who created a wonderful research ambience with lots of fun. Finally, I would like to express my gratitude to Dr. Javier LOPEZ LEONES and our team for hosting and guiding me during my visiting research at Boeing Research and Technology.

I thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for honoring and supporting me with its prestigious international doctoral research fellowship.

Lastly and most importantly, I thank my family for their endless love and support. I would not have gotten to where I am now without it.

March 2021

Mevlüt UZUN
(Aeronautical & Control Engineer)

TABLE OF CONTENTS

	Page
FOREWORD.....	ix
TABLE OF CONTENTS.....	xi
ABBREVIATIONS	xiii
LIST OF TABLES	xv
LIST OF FIGURES.....	xvii
SUMMARY	xxi
ÖZET	xxiii
1. INTRODUCTION	1
2. AIRCRAFT PERFORMANCE FUNDAMENTALS	7
2.1 Flight Dynamics	7
2.2 Cruise Fuel Consumption Dynamics.....	13
3. DEEP LEARNING PRELIMINARIES	19
3.1 Neural Networks with Multi-Layer Perceptrons (MLP)	19
3.2 Optimization Algorithms.....	21
3.2.1 ADAM	21
3.2.2 Yogi	22
3.3 Data Preprocessing for Training.....	23
3.4 Data Used in the Thesis.....	24
3.4.1 Operational flight data.....	24
3.4.2 Flight plan data	27
4. DATA-DRIVEN FUEL CONSUMPTION MODELING.....	29
4.1 Digital Twin Performance Modeling via Machine Learning.....	31
4.1.1 Black-box fuel flow estimation with QAR throttle.....	33
4.1.2 Fuel flow estimation with dynamically consistent thrust regeneration.....	36
4.1.3 Fuel flow estimation with throttle/auto throttle estimation.....	38
4.2 Comparison of ML Models and Sensitivity Analysis of the Correction Factor.....	40
5. UPDATING BASELINE PERFORMANCE MODELS	47
5.1 Problem Definition	47
5.2 Characterizing the Correction Factor	50
5.2.1 Types of discrepancies	50
5.2.2 Instance-based comparison	52
5.2.3 Distributions of the correction factor	56
5.3 Machine Learning for Correction Factor Estimation	56
5.3.1 Feature extraction.....	57
5.3.1.1 Statistical analysis on fuel flow	57
5.3.1.2 Statistical analysis on the correction factor	59
5.3.1.3 Machine learning based feature extraction and selection	59
5.3.2 Proposed custom MLP architecture for estimation of the correction factor	60

5.4 Experiment	62
5.4.1 Single tail number	63
5.4.2 Multiple tail numbers	67
6. PHYSICS-GUIDED MACHINE LEARNING FOR AIRCRAFT PERFOR-	
MANCE MODELING.....	69
6.1 Problem Formulation	72
6.2 Fuel Flow Dynamics	75
6.3 Methodology	75
6.3.1 Physics-based loss function design for fuel consumption	75
6.3.2 Implementation of the physics guided loss function.....	78
6.4 Experiment	80
7. WIND MODEL CALIBRATION.....	89
7.1 Problem Definition	91
7.2 Meteorological Data Used in the Study.....	95
7.3 Machine Learning Based Wind Estimation.....	98
7.4 Conclusion	102
8. FLIGHT PLANNING APPLICATIONS.....	103
9. CONCLUSION	109
REFERENCES.....	113
CURRICULUM VITAE.....	123

ABBREVIATIONS

QAR:	Quick Access Recorder
FDR:	Flight Data Recorder
MLP:	Multi-layer Perceptron
OEM:	Original Equipment Manufacturer
BADA:	Base of Aircraft Data
ML:	Machine Learning
DL:	Deep Learning
NN:	Neural Networks
DNN:	Deep Neural Networks
MAE:	Mean Absolute Error
MAPE:	Mean Absolute Percentage Error
TAS:	True Airspeed
CAS:	Calibrated Airspeed
MTWO:	Maximum Take-Off Weight
ZFW:	Zero Fuel Weight
OEW:	Operating Empty Weight
FL:	Flight Level



LIST OF TABLES

	<u>Page</u>
Table 2.1 : Summary of flight and atmospheric parameters that affect fuel consumption.....	16
Table 3.1 : Summary of parameters directly used from QAR dataset.....	25
Table 3.2 : Summary of parameters directly used from flight plan dataset.....	26
Table 4.1 : Summary of data sizes used in training.....	32
Table 4.2 : Input features of the black box model for climb and descent.	34
Table 4.3 : Input features of the black box model for cruise.....	34
Table 4.4 : Input features of the model with δ_T parameter for climb and descent..	37
Table 4.5 : Input features of the model with δ_T parameter for cruise.	37
Table 4.6 : Input features of the throttle estimation network.	39
Table 4.7 : Mean absolute errors on the test sets, in $[kg/s]$	42
Table 4.8 : The noted figures are percentage errors (i.e. % error), in terms of mean and standard deviation with respect to actual fuel flow rates.....	42
Table 5.1 : Features extracted by the Random Forest based methodologies.....	60
Table 5.2 : Summary of the input sets of the models.	63
Table 5.3 : Size of the training and testing sets for the single tail number dataset.	63
Table 5.4 : Estimation results for single tail-number case.	64
Table 5.5 : Estimation results for multiple tail-number case.....	64
Table 6.1 : Summary of parameters directly used from QAR dataset.....	81
Table 6.2 : Operational altitude, ISA deviation, and aircraft mass limits.	82
Table 6.3 : Comparison of prediction results on the test set.	85
Table 7.1 : Summary of parameters to be utilized for machine learning process. ..	97
Table 8.1 : Flight plan analysis showing the mean and max fuel loading differences for two cases of under usage of fuel and extra fuel usage.	105



LIST OF FIGURES

	<u>Page</u>
Figure 1.1 : A brief comparison of the parametric and the operational discrepancies between the model and the data for climb, cruise and descent.....	2
Figure 1.2 : Architecture of the automated system for developing corrected aircraft performance models.	5
Figure 2.1 : Flight phases and modes of an aircraft trajectory.	7
Figure 2.2 : Operational and parametric discrepancies between baseline model and actual trajectory.	10
Figure 2.3 : Scatter plot of data with reduced dimension.	11
Figure 2.4 : Extraction of flight modes using deep auto encoders.	12
Figure 2.5 : Thrust required with increasing Mach number and mass.	14
Figure 2.6 : Mach number vs fuel consumption for selected atmospheric conditions, and gross weight.	15
Figure 2.7 : Distributions of parameters from QAR dataset.	16
Figure 2.8 : The effect of gross weight on throttle setting and fuel consumption.	16
Figure 3.1 : Generic architecture of a Multi-Layer Perceptron.	19
Figure 3.2 : An example of filed flight plan.	26
Figure 4.1 : Fuel flow estimation with throttle information from QAR.	35
Figure 4.2 : Distributions of regenerated thrust utilizing QAR and BADA.	36
Figure 4.3 : Estimation results of training with δ_T for climb, cruise and descent.	38
Figure 4.4 : Custom architecture of throttle estimation for climb, cruise and descent.	39
Figure 4.5 : Throttle estimation for climb, cruise and descent.	40
Figure 4.6 : Fuel flow estimation with estimated throttle positions: climb, descent and cruise.	41
Figure 4.7 : Comparison of estimation from all models with BADA4 and actual fuel flow.	43
Figure 4.8 : Comparison of correction factors of all models.	44
Figure 4.9 : Sensitivity of models to Mach.	44
Figure 4.10 : Sensitivity of models to aircraft mass.	45
Figure 4.11 : Sensitivity of models to altitude.	45
Figure 4.12 : Sensitivity of models to δ_T	45
Figure 5.1 : Comparison between the model and the data in similar flight conditions.	52
Figure 5.2 : Kernel-density plots of the fuel flow profiles in BADA and QAR for given approximately the same climb conditions.	53
Figure 5.3 : Kernel-density plots of the fuel flow profiles in BADA and QAR for given approximately the same cruise conditions.	54
Figure 5.4 : Kernel-density plots of the fuel flow profiles in BADA and QAR for given approximately the same descent conditions.	54

Figure 5.5	: Histogram plots of fuel flow calculated by BADA, fuel flow recorder in QAR and the correction factor for climb.	55
Figure 5.6	: Histogram plots of fuel flow calculated by BADA, fuel flow recorder in QAR and the correction factor for cruise.	55
Figure 5.7	: Histogram plots of fuel flow calculated by BADA, fuel flow recorder in QAR and the correction factor for descent.	56
Figure 5.8	: Spearman rankings of the parameters for fuel flow correlation.	58
Figure 5.9	: Spearman rankings of the parameters for correction factor correlation.	59
Figure 5.10	: Proposed neural network architecture.	61
Figure 5.11	: Some snapshots from the test set illustrating the estimation results of the proposed model.	65
Figure 5.12	: Evolutions of the loss functions of the intermediate throttle and the correction factor.	66
Figure 6.1	: The hybrid system consisting of neural networks, and physics-guidance.	73
Figure 6.2	: Thrust times Mach versus fuel consumption for various flight regimes.	76
Figure 6.3	: Lateral profiles of the flights in the dataset.	80
Figure 6.4	: Full flight envelope compared to the regimes represented in the data.	82
Figure 6.5	: Mach sensitivity of the model without including physics.	83
Figure 6.6	: Mach sensitivity of the model with including physics.	85
Figure 6.7	: Fuel consumption prediction with the trained physics-guided NN model.	86
Figure 6.8	: Model sensitivity to pressure ratio, temperature ratio, and mass.	86
Figure 7.1	: The summary of 4D trajectory generation.	91
Figure 7.2	: Digital twin performance model through machine learning based estimations.	92
Figure 7.3	: Example of a comparison between GFS and QAR wind speed components for long haul case.	93
Figure 7.4	: Example of a comparison between GFS and QAR wind speed components for short haul case.	94
Figure 7.5	: Timeline chart for wind information during flight planning phase. ..	95
Figure 7.6	: An example of a wind map depicting wind speed and direction for different pressure levels.	96
Figure 7.7	: Trajectories of short haul flights used in the study.	98
Figure 7.8	: Trajectories of long haul flights used in the study.	99
Figure 7.9	: u component of wind speed after update.	99
Figure 7.10	: v component of wind speed after update.	100
Figure 7.11	: Reduction of wind uncertainty for short haul flights.	100
Figure 7.12	: Reduction of wind uncertainty for long haul flights.	101
Figure 8.1	: An example of trajectory in a dispatched flight plan.	103
Figure 8.2	: Performance factor (left) and planned waypoints in a Flight Management Computer (right).	104
Figure 8.3	: Comparison between the planned and the actual trajectories. Existing ground based flight planning tools estimate that the fuel burn is 104,782 kg, where the actual flight reveals a fuel burn of 105,872 kg.	105

Figure 8.4 : Distribution of fuel load differences.....106





DIGITAL-TWIN FLIGHT MODELLING THROUGH MACHINE LEARNING FOR TRAJECTORY ERROR ESTIMATION AND RECOVERY

SUMMARY

High fidelity aircraft performance models are essential for the precise estimations of 4D trajectories and fuel consumption. A correct fuel flow model is critical for flight planning purposes, especially in calculating the fuel load required for a trip. Airlines use manufacturers' performance models to calculate fuel consumption for flight planning rather than at the integrated operations center or at the cockpit level. As such, within the scope of Air Traffic Management (ATM), for problems such as trajectory planning/estimation for separation assurance and capacity and demand balancing, performance models such as the Base of Aircraft Data (BADA) by Eurocontrol are utilized. These baseline models are structured around polynomial approximations of Original Equipment Manufacturer (OEM) performance tables within generic aircraft models. However, flight data strongly suggest that the aircraft tend to deviate from these baseline models as they age and as, are exposed to maintenance, and are operated in diverse regions. Current ground-based flight planning systems utilize aircraft type-specific performance tables to determine fuel flows for given flight conditions and parameters such as altitude, mass, and speed. A performance factor corrects these tables as the aircraft ages. Despite this update, planned fuel consumption may indeed not overlap with the actual one. In addition to the aircraft performance model, wind uncertainty affects fuel consumption by its impact on ground speed. Because ground speed changes trip time, it determines how much fuel is to be consumed for a travel distance.

This thesis focuses on the flight planning at the ground level and propose a data-driven machine learning approach to develop a data-driven aircraft digital-twin model using the most fundamental and airborne recorded flight trajectory and flight parameters data set. In particular, this thesis considers on two distinct problems: uncertainties in aircraft performance models and uncertainties in the wind. In this sense, this thesis proposes methodologies to improve baseline models for fuel flow, and wind estimations via operational data. In particular, this study has investigated introducing neural network architectures to estimate tail-number specific updates to the aircraft baseline fuel flow model through deep learning for climb, cruise, and descent phases of flight. For training our neural network models, this study has utilized a QAR data set from 30 different tail-numbers with more than 2 million flight segments in total from a fleet of a wide-body aircraft operated by a major European flag carrier airline. The proposed architecture is designed to retain the compatibility to the flight planning applications by using the same features that the baseline aircraft models do for computing flight trajectory and the fuel flow. Throughout an extensive feature

engineering, the thesis reveals the parameters with a significant impact on fuel flow. The analysis shows that the discrepancy between the modeled and the actual fuel flows stems from a) parametric biases in fuel flows and b) operational variations in the actual implementation of the flight modes. As such, the estimation errors in throttle and the corresponding thrust and excess thrust levels, play a key role in error dynamics. This fact is utilized to further refine the neural network architecture into a novel cascaded structure, which embeds estimation of critic engine related parameters that are not available within baseline models for flight planning. Furthermore, the study has implemented a physics-guided deep neural network framework to improve data-driven models' consistency in flight regimes that are not covered by data. In particular, we guide the neural network with the equations that represent fuel flow dynamics. In addition to the empirical error, we embed this physical knowledge as several extra loss terms. In wind uncertainty part, historical Global Forecast System (GFS) predictions are utilized as baseline estimations. Wind measurements in QAR footprints are considered as ground-truth. State-of-the-art deep learning algorithms are deployed to map baseline estimations for fuel flow and wind to their ground truths.

A comparison of the models with real flight data shows that precise estimation of fuel flow with mean absolute errors lower than %0.7 can be achieved at all the flight modes. We further tested these models on different tail numbered aircraft to show the generalization capability of the algorithm. Results on the wind side also show that we can achieve a considerable reduction in wind uncertainty both from a mean error and variance sense. Finally, our fuel flow models are compared with the Airline ground-based planning fuel flow models on the actual flight plans generated by the ground-based system. Total trip fuel comparisons show discrepancies up to %3.5 total fuel loading weight, which will result in potential fuel savings by decreasing the fuel load during take-off for flights with unnecessary excess fuel load. An example of a specific tail-number suggests that for typical operation of 200 long-haul flights per day, yearly savings on the order of 17 million USD can be achieved at current jet fuel prices. This "tail-number specific" performance modeling approach is projected to open considerable frontiers, including further reduction of fuel load safety margins and in-flight update of performance models through machine learning methods.

ROTA HESAPLAMALARINDA MAKİNE ÖĞRENMESİ TABANLI DİJİTAL-İKİZ UÇUŞ MODELİ

ÖZET

Hava trafik yönetiminde öncelikli amaç artan trafiğin akışını mevcut güvenlik seviyesini ihlal etmeden sağlayabilmektir. Mevut sistem anlık trafiği taktiksel müdahaleler ile yönetmek üzerine kuruludur. Ancak, sistemin proaktif olması amaçlanmaktadır. Bu bağlamda verimlilik planlama seviyesindeki doğruluğun artırılması ile sağlanabilecektir. Her bir uçuş planlama esnasında ne kadar doğru hesaplanabilirse, sistem içerisindeki tüm oyuncular için durum o kadar iyi olacaktır. Hava trafik sistemini oluşturan bu oyuncular; seyrüsefer hizmeti veren Devlet Hava Meydanları İşletmesi gibi kuruluşlar, havayolları, havaalanları, hava aracı üreticileri, ve yolculardır. Her kesmin kendi açısından verimlilik tanımı farklıdır. Ortak kriter hava ulaşımına erişimdeki veya hava ulaşımının sağlanmasındaki maliyeti düşürmek olsa da, bunun nasıl yapılacağı ve diğer oyuncuları nasıl etkileyeceği farklılaşmaktadır.

Mevcut hava trafik sistemi operasyonel regülasyonlara dayanmaktadır. Güvenlik açısından birçok bölgede köklü değişiklik yapmak uzun bir süre gerektirmektedir. Örneğin, alternatif enerji kullanan uçak motorları doğal olarak maliyet ve kirlilik iyileştirilmesinde büyük etki yaratacaktır. Başka bir örnek ise tamamen otonom uçuşlar olarak verilebilir. Kalkıştan inişe kadar tüm fazların otopilot tarafından yapıldığı, trafikte çarpışma önleyici ve kaçınma sistemlerinin algoritmalar tarafından yönetildiği bir hava trafik sisteminde aynı anda daha fazla uçuş gerçekleşebilir. Ancak bu tür teknolojilerin halihazırda yeterince kompleks bir sisteme entegrasyonu büyük ölçüde ar-ge çalışmaları, insan kaynağı, bütçe ve zaman gerektirmektedir. Bu tür teknolojiler hava trafik yönetimi sistemi içerisinde bulunan kurumların uzun vadeli planında yer almaktadır. Öte yandan, kısa vadeli olarak ise mevcut sistemde neyin geliştirilebileceği üzerine çalışılmaktadır. Havacılık yavaş işleyen bir endüstri olduğu için mevcut bileşenlerin birçoğu yirmi ila otuz yıllık teknolojilere dayanmaktadır. Bunda doğal olarak bir yolcu uçağının ortalama ömrünün de bu civarda olmasının etkisi vardır. Tüm bunlar düşünüldüğünde, kısa vadeli iyileştirmelerde daha pratik ve mevcut sisteme daha kolay entegre edilebilir teknolojilere yoğunlaşılmaktadır.

Bu tezde rota planlama sürecinde uçağa yüklenen yakıtın doğru hesaplanması teması işlenmiştir. Her havayolu, uçuştan önce o uçuşa dair bir plan oluşturmakla ve bunu ilgili regülatörler ile paylaşmakla yükümlüdür. Bir uçuş planı temel olarak enlem, boylam, irtifa gibi konum bilgileri ile birlikte uçağın hangi coğrafi konumlardan ne zaman geçeceği, hangi hava sahasında ne kadar süre geçireceği, hedef varış noktasına ne zaman varacağı, hangi hızları takip edeceği ve bu rotayı tamamlamak için ne kadar yakıtı ihtiyaç duyacağını içermektedir. Uçuş rotasının enlem ve boylam olarak nasıl

bir yörünge izleyeceği bölgedeki operasyonel faktörler ile belirlenir. Dikey profilde ise seyir irtifası, tırmanma ve alçalma profilleri ile birlikte uçağın hangi hızlarda uçağı ise üretici tarafından belirtilmiş performans modeli tarafından hesaplanır. Bir uçuş yörüngesini etkileyen önemli bir diğer faktör ise meteorolojik bilgidir. Hangi bölgelerde şiddetli türbülans, veya tehlikeli hava koşulları beklendiğine göre yatay yörünge ayarlanır. Ancak uçuş süresini etkileyen en kritik meteorolojik nicelik rüzgardır. İrtifa yükseldikçe rüzgar hızı da artmaktadır. Uçuşların seyir fazları da yüksek irtifada bulunmaktadır. Bunun sebebi turbofan motorlarının en verimli olduğu irtifa ve hızlarda uçuşma gayesidir. Günümüz yolcu uçaklarının büyük bir çoğunluğu turbofan jet motoru kullanmaktadır. Bu tür uçaklar için en yakıt verimli uçuş profili olabildiğince yüksekte ve olabildiğince hızlı uçmaktır. Rüzgarın tanımının da havanın yere göre bağıl hızı olduğu düşünüldüğünde hava içerisinde seyir halindeki bir uçağın yere göre hızı ise rüzgar hızı ile vektörel toplamıdır. Dolayısıyla rüzgar hızı uçuş süresini etkilemektedir. Uçak performans modelleri ise uçak üreticileri tarafından sağlanan, aerodinamik ve itki kuvvetleri ile birlikte yakıt tüketimi, hız profilleri, manevra zarfı gibi uçak spesifik bilgileri kapsar. Bu tezde özellikle yakıt tüketim modelleri üzerine yoğunlaşmaktadır.

Standart bir yer tabanlı rota hesaplama algoritması birtakım giriş parametreleri almaktadır. Bu parametreler ile birlikte uçak hareket denklemleri integre edilmekte ve zamana bağıl uçuş rotası hesaplanmaktadır. Bulunan rota uçuş planı olarak gerekli regülatörlere iletilmektedir. Ancak, hesaplanan bu rotaların doğruluğunun giriş parametrelerinin belirsizliğine karşı hassasiyeti bulunmaktadır. Birçok faktör sebebiyle planlanan ve gerçekleşen uçuşlar bir olmamaktadır. Başlıca etken rota üzerinde trafiğe bağıl olarak hava trafik kontrolörlerinin komutlarıdır. Hava sahasının yoğunluğuna bağıl olarak hava trafik kontrolörü uçağı planlanan irtifasından daha aşağıda veya yukarıda uçurmak için komut verebilir. Benzer şekilde planlanandan farklı bir hızda uçuşmasını isteyebilir. Haliyle planlama aşamasında herhangi bir hız veya irtifa optimizasyonu yapıldı ise bu durum geçerliliğini yitirmektedir. Örneğin, geniş gövdeli yolcu uçakları için en verimli hız 0.83 Mach süratı civarındır. Ancak okyanus ötesi uçuşlarda, güvenliliğin sağlanması açısından bazı bölgelerde uçaklardan 0.82 Mach süratında uçuşması istenebilir. Veyahut planlanan seyir irtifası 39,000 feet iken belirli hava sahalarında bu uçuşun 38,000 feet irtifada devam etmesi istenebilir. Başka bir örnek ise yatay yörüngeyi değiştirilmesi olarak verilebilir. Trafik yoğunluğunun azalmasını beklemek üzere uçuşlardan rotalarını uzatmaları istenebilir, veya açık bir hava sahasında en kısa yoldan gitmeleri komutu verilebilir. Bu tür durumlar gerçekte tüketilen yakıtın planlanandan farklı olmasına sebebiyet verir. Ancak bunlar öngörülemeyen veya modellenemeyen faktörlerdir. Tamamen operasyonel faktörlere dayanmaktadır. Bu tezde ise halihazırda modellenen ve uçuş yakıt tüketiminde etkisi olan parametrelerin iyileştirilmesi amaçlanmaktadır.

Özetle, bu tez uçuş planlamada giriş olarak sağlanan uçuş performans modeli ve rüzgar bilgisi faktörlerini ele almaktadır. Bir önceki paragrafta uçuş performans modellerinin uçak üreticileri tarafından sağlandığı belirtilmişti. Bu modeller test uçuşu verilerinin yoğun mühendislik tecrübesi ile yorumlanması sonucu oluşturulan performans tabloları ve denklemleri kapsamaktadır. Ancak tablo değerleri ve denklem katsayıları uçağın ilk üretilmiş versiyonuna ait test uçuş sonuçları ile oluşturulmuştur. Her ne kadar uçuş dinamikleri aynı olsa da kuyruk numarası bazında uçuş performansları değişiklik göstermektedir. İki uçak birebir aynı yapıda olsa dahi farklı bakım ve uçuş operasyonları sonucunda aynı koşullarda

aynı yakıt tüketimini vermeyebilir. Bu durumda ağırlıklı olarak motorların bakım periyodu etkilidir. Model parametrelerinin yanı sıra modellemenin kendisinde hesaba katılmayan faktörler ve belirsizlikler de bulunmaktadır. Bu tür bileşenler, gerçekleşen yakıt tüketiminin planlanandan farklı olmasına sebep olmaktadır. Dolayısıyla uçak üreticisi tarafından sağlanan modelin kuyruk numarası bazında düzeltilmesi ve bir şekilde güncellenmesi gereklidir. Mevcut uygulamalarda yüzdesel performans düşüşü hesaplanmakta ve bu birim yakıt tüketimine lineer bir dönüşüm ile yansıtılmaktadır. Ancak gerçek uçuş verileri yakıt tüketim dinamiklerinin daha kompleks olduğunu göstermektedir. Bu bağlamda nonlinear fonksiyonlar ile uçak performans modellerinin geliştirilmesi gereklidir. Uçuş planlama algoritmalarına rüzgar bilgisi ise meteoroloji merkezlerinden gelmektedir. Rüzgar bilgisi belirli çözünürlükte tüm enlem, boylam ve irtifaları kapsayacak şekilde üretilir. Belirli bir süre boyunca hava gözlem istasyonları, hava balonları, uçak gibi bileşenlerin sensörlerinden veri toplanır. Dünya üzerindeki tüm noktalardan veri toplamak mümkün olmadığından, toplanan bu veri iklim modellerinden geçirilerek global hava tahminleri elde edilir. Bu tahminler Avrupa’da altı saatte bir, Amerika’da ise saatte bir güncellenir. Ancak, her modelde olduğu gibi bu modellerin çıktılarında da belirsizlik ve hata bulunmaktadır. Hava gözleminin yapılabildiği bölgelerde rüzgar hatası genellikle düşük iken, diğer bölgelerde yüksek tahmin hataları gözlemlenebilmektedir.

Bu tezde, uçuş planlama sırasında gerekli yakıtın daha doğru hesaplanabilmesi için uçak performans modeli ve rüzgar bilgisindeki hataları azaltmak amaçlanmaktadır. Bu kapsamda operasyonel uçuş verisi olarak Quick Access Recorder adı verilen, bir uçağın tüm sensörlerinden aldığı saniyelik ölçümleri kaydettiği veritabanı kullanılmıştır. Bu veri havayolları merkezlerinde bulunmaktadır. Her uçak uçuş sonrası topladığı ve kaydettiği bilgileri kablosuz veya kablolu olarak bu merkezlere aktarmaktadır. QAR verisinin operasyonel olarak çeşitli kullanım amaçları bulunmaktadır. Bunlardan başlıcaları uçak performansını gözlemlemek, bileşenlerin durumunu kontrol etmek, anomalileri tespit etmek ve verimliliği artırmaktır. QAR verisinde kaydedilen parametrelerden bu çalışma kapsamında en önemli ikisi birim yakıt tüketimi ve rüzgar hız/yön bilgileridir. QAR içerisinde kaydedilen bu değerler, gerçek değerler olarak nitelendirilmiştir. Aynı parametrelerin uçak performans modelleri ve rüzgar modelleri tarafından hesaplanan versiyonları ise planlanan olarak ele alınmıştır. Her iki nitelik de, yakıt tüketimi ve rüzgar hızı, birtakım parametrelere bağlıdır. Turbofan motorlarda birim zamanda tüketilen yakıt sürat, irtifa, uçak ağırlığı, motor devri, ve gaz kolu seviyelerinin bir fonksiyonudur. Rüzgar hızı daha periyodik özellikler taşımakla birlikte coğrafi konum, yükseklik, ve sıcaklık gibi parametrelere bağlıdır. Bu tez kapsamında, yakıt tüketimi ve rüzgar hızı değerlerine, yine gerek QAR içerisinde direk olarak bulunan, veya QAR içerisindeki parametrelerden türetilen nitelikler cinsinden nonlinear fonksiyon üretilmiştir. Bu fonksiyonları oluştururken derin öğrenme tabanlı yapay sinir ağları kullanılmıştır. Yapay sinir ağlarının teorisi yirmi ile otuz yıl öncesine dayanmaktadır. Son birkaç sene içerisinde özellikle büyük veri ile birlikte yüksek işlem kapasiteli sistemlerin geliştirilmesi ile çok katmanlı yapay sinir ağları modelleri geliştirilebilmiştir. Benzer şekilde alandaki matematiksel ilerlemeler, özellikle optimizasyon tarafında yenilikçi çözümler ortaya çıkarmıştır. Bu tez kapsamında da probleme yönelik özel yapay sinir ağları mimarileri geliştirilerek yakıt tüketimi ve rüzgar hızı modelleri oluşturulmuştur.

Tezin büyük bir kısmı uçak performans modeli üzerine yoğunlaşmaktadır. Rüzgar modeli güncellemesi, tezin son bölümlerinde yer almaktadır. Ağırlıklı olarak uçak

performans modellerinin nasıl geliştirildiği, mevcut modellerin neyi eksik yaptığı, ve derin öğrenme tabanlı yapay sinir ağları ile bunların nasıl geliştirileceği işlenmiştir. Uçak üreticileri tarafından geliştirilen performans modellerini temsilen Base of Aircraft Data (BADA) uçak performans veri tabanı kullanılmıştır. Analizlerde gerçek uçuş verileri ile kıyaslama yapılmış, bunlara bağlı olarak hangi parametrelerin model girişi olarak kullanılacağına karar verilmiştir. Gerçek uçuş verisi ile kıyaslama temelde iki farklılık olduğunu göstermektedir: a) basitleştirilmiş dinamikler b) parametrik hatalar. Buna bağlı olarak uygun yapay sinir ağları mimarileri geliştirilmiştir. Tezin bir kısmında ise makine öğrenmesi tabanlı sistemlerin açıklanabilirliğini ve genelleştirilebilirliğini artırmak için fizik tabanlı yönlendirmelerin, fizik kurallarının, modellere nasıl entegre edilebileceği üzerine odaklanılmıştır. Çünkü makine öğrenmesi algoritmaları, öğrenmede kullanılan verinin kapsamı ile sınırlıdır. Bu modellerin hiç görmediği bölgelerde yaptıkları hesaplamalar tutarlı ve anlamlı olmayabilir. Uçuş yakıt tüketimi için de bu geçerlidir. Uçuş verisi operasyonel rejimleri içerdiği için tüm manevra zarfını kapsamaz. Dolayısıyla modeli geliştirmek için hiç bilmediği alanlarda en azından nasıl bir çıktı vereceği sisteme entegre edilebilir. Sonuç olarak, tez kapsamında üretilen yakıt tüketim modelleri tırmanma, seyir ve alçalma uçuş modlarının hepsi için yüzde bir altında hata ile kestirim yapabilmektedir. Fizik tabanlı öğrenme metodolojileri ile de modellerin veride bulunmayan uçuş rejimleri için tutarlı sonuçlar üretebilmesi sağlanmıştır. Rüzgar modeli tarafında ise, rüzgar belirsizliği varyansın düşürülmesi şeklinde azaltılmıştır. Son olarak, geliştirilen bu modeller uçuş planlama şeması içerisine yerleştirilmiş, gerçek uçuş planları bu modeller ile tekrar üretilmiş, ve toplam yakıt tüketimi farkına bakılmıştır. Sonuçlar yüzde üçe kadar yakıt tasarrufu olabileceğini göstermektedir. Bu, uçağı kalkış sırasında gereksiz yakıt taşımaktan kurtarmak anlamına gelir. Dahası, uçaklar bu fazla yakıtı taşımadığı için de birim yakıt tüketimini azaltacaklardır. Geniş gövde uçakların uçuş planlarında yapılan analiz iki tona kadar yakıt tasarrufu göstermektedir. Ortalama olarak, geniş gövdeli bir uçak uzun menzil bir uçuşta toplam ağırlığının üçte biri kadar yakıt taşımaktadır. Bu, taşınmayan iki ton için yaklaşık altı yüz kilogram tasarruf edileceğini de gösterir. Bunun üzerinden hesap yapıldığında, geniş filoya sahip bir havayolu yılda milyonlarca lira kazanç sağlayabilir.

1. INTRODUCTION

Aviation Management is a complex process considering the distinct interests of several stakeholders such as airlines, air navigation service providers, aircraft manufacturers, and airports. In addition to being safe and resilient, an ideal air transportation system should also be efficient. Even though the definitions of efficiency vary with stakeholders, the essential point to achieve the maximum efficiency goes through accurate prediction of the actual flight operations. At the microscopic level, this could be enabled by improving the accuracy of individual aircraft trajectories. In this context, two terms are essential as parts of a trajectory: trip time and trip fuel. Proper estimation of trip time is necessary for demand and capacity balancing [1], whereas the correct estimation of trip fuel is crucial for airline economics and environmental impacts [2] [3]. The core element of calculating the trip time and fuel of a trajectory is an aircraft performance model, which is in the form of analytic equations or look-up tables that compute lift, drag, thrust, and fuel flow at a specified flight regime or aircraft configuration. Additionally, it covers take-off and landing calculations as well as structural and operational limitations. These models are results of a combination of wind tunnel experiments, data from test flights, or empirical approximations. They are generic, i.e. they are tailored to a unique aircraft type. However, even the same type of aircraft may have different performances. This is a direct consequence of how the airlines operate them. The diversity in operations could be at maintenance operations, flight regimes, travel distances and flight frequencies. Moreover, performance of an aircraft tends to deviate from its initial condition as the aircraft ages. This fact can also be seen directly from flight data as illustrated in Figure 1.1. Here there is a clear indication of performance off-sets beyond variations to operational procedures in distinct flight modes. Essentially, all the above factors lead to a need for a tail-specific update of the baseline performance models. In this thesis, we address this need through a novel data-driven and machine-learning based methodology to generate tail-number specific updates to OEM base flight performance models. Specifically, we use the most

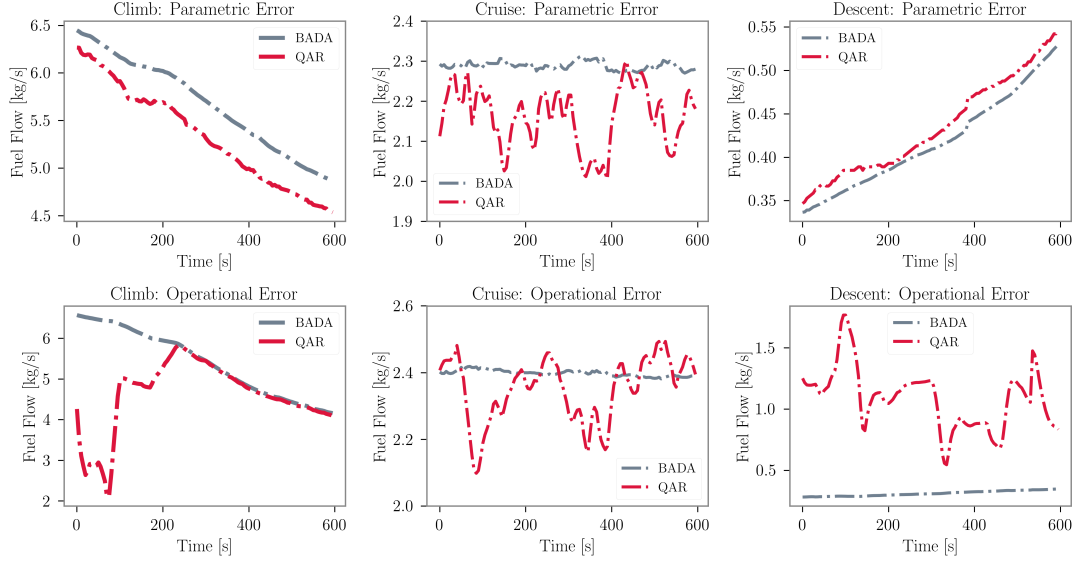


Figure 1.1 : A brief comparison of the parametric and the operational discrepancies between the model and the data for climb, cruise and descent.

common in-flight recorded data, namely Quick Access Recorder (QAR) data, to train neural network structures that capture or estimate the main input-output relationships essential for precise flight performance and fuel usage modelling.

In the literature and real-world applications, there exist many approaches to model aircraft performance with differences on the reference data utilized and the scope of flight envelope they cover. For example, ICAO Aircraft Engine Emission Databank [4] provides fuel flow information for a various range of engines, but only for landing and take-off cycle. On the other hand, aircraft manufacturers have more sophisticated models taking advantage of wind tunnel experiments and test flights. They design these models to be used in especially airline operations such as flight dispatching, fleet management and performance engineering. The lead examples of OEM models are Boeing's Performance Software (BPS) and Airbus' Performance Engineering Package (PEP). These tools enable the generation of high-quality reference data for the full flight envelope. The Base of Aircraft Data 4 (BADA4) [5, 6] is an advanced tool and pioneer in leveraging this capability and consists of polynomial regressions fitted to synthetic data generated by BPS and PEP. However, it inherits the uncertainties and modeling errors of the reference models. Unlike the manufacturers' software, it is available to the development and research community under licensing conditions.

Performance models are very common for all aspects of flight management and trajectory based operations for both jet aircraft and UAVs. Especially in flight

optimization, they are the key elements because the fuel consumption is always a part of the cost function. Hence, realistic models are essential to have the correct solutions. A problem scheme of multiobjective trajectory optimization proposed by Gardi et al. [7] used BADA3, which models the fuel flow with crude empirical equations and is further surpassed by BADA4 in modelling enhancements [8]. Similarly, Malaek et al. introduced an optimal maneuvering planning for ATM purposes, in which the aircraft mass is a critical parameter but its evolution is modelled with BADA3, as well. Rodrigues [9] developed an optimal control framework for the calculation of the best fuel mileage and the best endurance speeds, but assumed a constant specific fuel consumption in exchange for simplicity. Additionally, Brown and Anderson [10] utilized the same approach for the design of UAV missions. However, even though simple performance models are favorable for real-time flight management applications, the omitted dynamics including the fuel usage make both the precision of the results and also the validity of conclusions questionable [11]. In that respects, access to manufacturer based models or inference of underlying dynamics through real flight data becomes crucial.

Since access to manufacturer data is highly limited, researchers have focused on developing performance models via alternative data sources. An engine performance model developed by Ghazi [12, 13] is based on trajectory data generated by a level-D research flight simulator. Sun et al. [14, 15] proposed using open-source data, and applied probabilistic methods on ADS-B and Mode-S records to extract parameters to form a performance model. Another open-source model by Metz et al. [16] used publicly available aircraft data from the Jane's all the World's Aircraft database and empirical approaches. Hratsovec and Solina [17] proposed a machine learning process to train a dynamic performance model based on a combination of radar recordings, flight plan and weather data. However, these studies either validated their modeled performance with BADA3 [18], the predecessor of BADA4, or revealed high deviations from actual performances.

In the last decade, there has been an increase in utilizing operational flight data, namely Quick Access Recorder (QAR) or Flight Data Recorder (FDR) [19] for many applications such as performance monitoring, anomaly detection or weather forecasting [20] [21] [22] [23]. These data consist of historical logs of all parameters

that are measurable or observable through on-board sensors and systems. Even though they do not have information of thrust, drag or lift, they record critical performance indicators such as vertical speed, gross weight and fuel flow. This capability makes them highly suitable for supervised learning of aircraft performance. Chati and Balakrishnan used FDR data in tree-based learning and Gaussian process algorithms to model fuel flow [24], [25], [26]. Baklacioglu [27] combined genetic algorithms and neural networks for the same purpose. Baumann and Klingauf [28] proposed another scheme for supervised learning of fuel flow with FDR. The works of Huang et al. [29] and Khadilkar [30] are other examples that have used operational data. These studies proved that the aircraft performance could be represented through machine learning techniques, however, they did not investigate tail-number specific calibration and its implementation on trajectory planning. As such, the applicability of the proposed methods to real ground-based planning remains infeasible given the fact that they require availability of input features of models in advance. In that respect, the proper integration into flight planning systems aspect of the problem still remains open.

Regardless of what kind of data or mathematical model the performance models are built on, they need to be continuously monitored, as the actual performance deteriorates over time. Aircraft manufacturers do this task by aircraft health monitoring systems [31] [32]. The update is usually done through the performance factor [33]. This variable is represented by only a single scalar denoting the percentage of degradation in performance compared to default setting. Even though it is an update, it only provides a linear correction and does not capture all the flight dynamics. Several studies also investigated the calibration of performance model parameters. Ghazi et al. [34] updated drag-polar tables of the Flight Management Systems to increase fuel flow accuracy. However, they tested the results on simulator data and for cruise phase only. Puranik et al. [35] proposed calibrating empirical models with Pilot Operating Handbooks and flight data, but only studied a general aircraft, which does not have a complex engine system like turbofans. Operational updates to the Aviation Environmental Design Tool [36] for ground-based operations are suggested by Clemons et al. [37] through an analysis of the operational discrepancies. None of the previous studies investigated tail-number specific calibrations or considered applications on flight planning.

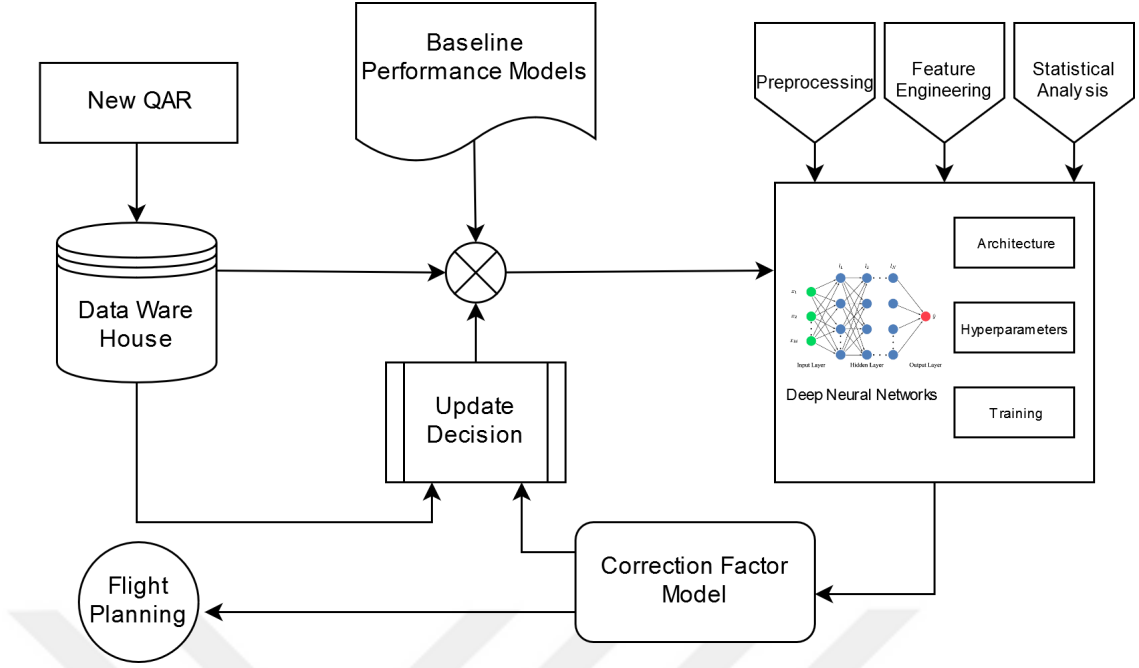


Figure 1.2 : Architecture of the automated system for developing corrected aircraft performance models.

This thesis proposes the data-driven performance modelling by introducing a custom neural network architecture and a comprehensive feature engineering that leverages the state-of-the-art algorithms. As such, this thesis embeds a calibrated wind model feature by utilizing the baseline forecast similar to the performance update process.

The proposed system is illustrated in Figure 1.2 as an automated process, Specifically, as the system gathers new operational flight data, it calibrates the performance models on a predetermined frequency. The first functionality of the system is to transfer either in real-time or by post-flight data processing of flight data records to flight data warehouse. The data can be transferred through wireless links, ACARS messages, in-person collection of data from the aircraft through memory devices. Comparing the actual fuel flow from QAR with the fuel flow computed by the baseline performance models is the first step of the algorithm. Based on the significance level of the discrepancy, the system decides whether an update on the correction factor is required or not. The determination and pre-processing of the specific data involves the selection of the type of data, the flight segments, etc. An extensive feature engineering determines which parameters to be used for training the models. The essential part of this process is to select features and model architectures such that we estimate realistic fuel flow characteristics for trajectories created during flight planning.

This thesis is a continuation of the author's study in trajectory prediction uncertainties and mainly consists of key publications dedicated to aircraft performance and wind modelling, in particular. These papers are either published or being prepared for submission, and listed as below:

- **Uzun, M.**, Demirezen, U., & Inalhan, G. (2020). Physics Guided Neural Network for Data-Driven Aircraft Performance Modeling, *Aerospace* 2021, 8, 44..
- **Uzun, M.**, Demirezen, U., & Inalhan, G. (2020). Deep Learning Based Digital Twin for Updating Baseline Aircraft Performance Models, *IEEE Transactions on Aerospace and Electronic Systems* (Under Review).
- **Uzun, M.**, Demirezen, M. U., Koyuncu, E., Inalhan, G., Lopez, J., & Vilaplana, M. (2019, September). Deep Learning Techniques for Improving Estimations of Key Parameters for Efficient Flight Planning. In *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)* (pp. 1-8). IEEE, San Diego, CA, US.
- **Uzun, M.**, Demirezen, M. U., Koyuncu, E., & Inalhan, G. (2019, March). Design of a hybrid digital-twin flight performance model through machine learning. In *2019 IEEE Aerospace Conference* (pp. 1-14). IEEE, Big Sky, MT, US.

Rest of the thesis is organized as follows: Chapter 2 is dedicated to a technical background of the scope of baseline aircraft performance models. Chapter 3 describes the operational QAR data and filed flight plan data used in this study. In Chapter 4, deep learning techniques used in this thesis are explained. Data-driven fuel consumption is introduced in Chapter 5, and continued with updating baseline performance models in Chapter 6. In Chapter 7, physics-guided modelling on digital-twin aircraft performance model is introduced. Chapter 8 applies the similar machine learning techniques for wind model calibration. Applications on flight planning systems to illustrate the impact of high accuracy aircraft performance models are provided in Chapter 9. Finally, Chapter 10 concludes the thesis discussing open issues and future work.

2. AIRCRAFT PERFORMANCE FUNDAMENTALS

This chapter describes the aircraft flight mechanics and performance fundamentals.

2.1 Flight Dynamics

A typical gate-to-gate flight consists of several flight modes. In brief, an aircraft climbs up to its scheduled cruise altitude after take-off, followed by an initial climb. Flight continues with a descending trajectory starting at a top-of-descent location. Finally, it is terminated by landing after final approach. Figure 2.1 depicts a generalized picture of flight modes observed in a typical commercial jet flight. Each mode represented in this figure is defined in terms of speed and altitude intent. For instance, an aircraft climbs at a constant calibrated airspeed in CAS climb and maintains its target Mach number during Mach climb. The same process goes for descending trajectories. However, initial climbs and final approaches depend on departure and arrival procedures of airports, respectively. Hence speed and altitude schedules of these modes cannot be generalized. A similar argument can also be made for accelerations and decelerations during cruise as they are usually due to varying operational discrepancies (i.e., air traffic controller interventions, speed adjustments).

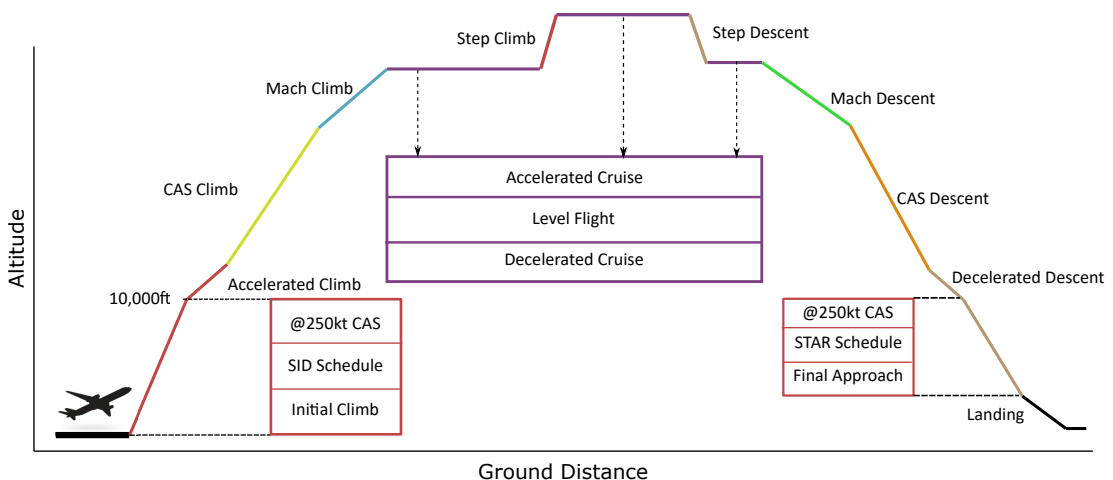


Figure 2.1 : Flight phases and modes of an aircraft trajectory.

Regardless of flight modes, general form of the time-invariant dynamics of the aircraft is considered as:

$$\dot{x}(t) = f(x(t), u(t)) \quad x(0) = x_0 \quad (2.1)$$

where the $x(t) \in X \subseteq \mathbb{R}^n$, $u(t) \in U \subseteq \mathbb{R}^m$ such that $n, m \in \mathbb{N}$ and the state $x_0 \in X$ is called the initial state of the ownship. Aircraft states are defined as $x = \{p_x, p_y, h, V_{TAS}, \chi, m\}$ and input space is $u = \{\chi, \gamma, Th\}$. The equations of aircraft motion are:

$$\dot{p}_x = V_{TAS} \cos \chi \cos \gamma_{Q_j} + w_x \quad (2.2)$$

$$\dot{p}_y = V_{TAS} \sin \chi \cos \gamma_{Q_j} + w_y \quad (2.3)$$

$$\dot{h} = V_{TAS} \sin \gamma + w_z \quad (2.4)$$

$$\dot{V}_{TAS} = -\frac{C_D S \rho V_{TAS}^2}{2m} - g \sin \gamma_{Q_j} + \frac{Th_{Q_j}}{m} \quad (2.5)$$

$$\dot{\chi} = \frac{C_L S \rho V_{TAS}}{2m} \frac{\sin \mu}{\cos \gamma_{Q_j}} \quad (2.6)$$

$$\dot{m} = -F_{Q_j} \quad (2.7)$$

where p_x and p_y are the aircraft coordinates, h is the geopotential pressure altitude, V_{TAS} is the true airspeed, χ is the heading and m is the aircraft mass. C_L and C_D are the coefficients of lift and drag, respectively. The air density is denoted by ρ and the wing area is represented by S . As for the control inputs, Th is the thrust force and γ is the flight path angle. $\{w_x, w_y, w_z\} \in V_{wind}$ are x , y and z components of wind speed $\vec{V}_{wind}$, where

$$||W_{wind}|| = \sqrt{w_x^2 + w_y^2 + w_z^2} \quad (2.8)$$

$$\angle V_{wind} = \text{atan2}(w_y, w_x) + \pi \quad (2.9)$$

and $\mathbb{E}[w_x], \mathbb{E}[w_y] \in \mathbb{R}$. Throughout the study, we assume that $\mathbb{E}[w_z] = 0$ and $\mathbb{E}[w_z w_z^T] = 0$. $\vec{V}_{wind}$ is a stochastic component to the system and its corresponding covariance matrix is represented by $\mathbb{E}[\vec{V}_{wind} \vec{V}_{wind}^T] = Q_w \in \mathbb{R}^{3 \times 3}$.

Q_j stands for the flight mode aircraft is operating and $Q = \{Q_j | j \in \mathbb{N}\}$ is the countable set consists of these modes. The parameters with index Q_j indicates that their formulations differ with flight mode. In this paper, we are going to follow the similar modeling fashion as described in BADA [38]. BADA have two kinds of modeling: First one is generalized model for drag, engine thrust and fuel consumption. Second is distinct formulations and assumptions for engine thrust and fuel consumption.

Drag computation is independent from flight modes and its generalized formulation is:

$$D = 0.5 \delta p_0 \kappa S M^2 C_D \quad (2.10)$$

$$C_L = \frac{2mg}{\rho V_{TAS}^2 S \cos(\phi)} \quad (2.11)$$

where δ is the pressure ratio, κ is the adiabatic index of air, p_0 is the air pressure at the sea level, ϕ is the bank angle and M is the Mach number. When any of flaps δ_{fp} , landing gear δ_{lg} and speed brakes δ_{sb} are not deployed, following equation is used to calculate C_D .

$$C_D = C_{D_0} + (C_{D_1} C_L^2) + (C_{D_2} C_L^6) \quad (2.12)$$

$$C_{D_j} = \sum_i \frac{d_i}{f(1-M^2)} \quad (2.13)$$

where $d_{(.)}$ represent the coefficients from the aircraft performance model. Generalized thrust is calculated by the formula below:

$$Th = \delta W_{mref} C_T \quad (2.14)$$

$$C_T = \sum_{j=1}^6 \sum_{i=1}^6 a_{6(i-1)+j} M^{i-1} \delta_T^{j-1} \quad (2.15)$$

where W_{mref} is the reference weight and $a_{(.)}$ are the polynomial coefficients from BADA. In this general form, thrust is controlled through a throttle parameter δ_T . Modeling δ_T as a single equation is a bit challenging as throttle setting naturally depends on aircraft and pilot intent. BADA assumes specific thrust settings for climb, cruise and descent. In cruise, thrust equals drag, $Th = D$. The engine operates at the maximum rating in climb and corresponding throttle parameter δ_T is computed as:

$$\delta_T = \sum_{j=1}^6 \sum_{i=1}^6 b_{6(i-1)+j} M^{i-1} \delta_T^{j-1} \quad (2.16)$$

where $b_{(.)}$ denotes the polynomial coefficient of the model. In a similar approach, the engine operates at low idle rating in descent and thrust coefficient C_T is computed through the following equation:

$$C_T = \sum_{j=1}^3 \sum_{i=1}^4 t_{3(i-1)+j} \delta_T^{i-2} M^{j-1} \quad (2.17)$$

As for the fuel flow, we use generalized fuel consumption equation from BADA, which is defined as:

$$F = \delta \theta^{1/2} W_{mref} a_0 L_{HV}^{-1} C_F \quad (2.18)$$

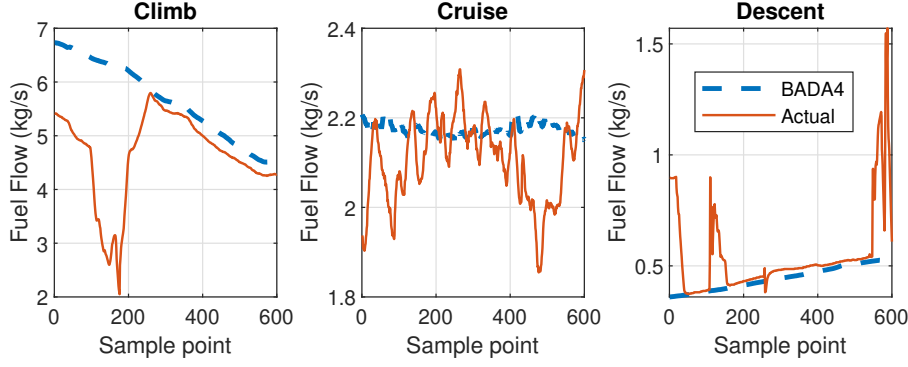


Figure 2.2 : Operational and parametric discrepancies between baseline model and actual trajectory.

where L_{HV} is the fuel lower heating value and θ is the temperature ratio. C_F is the coefficient of fuel consumption and defined as:

$$C_F = \sum_{j=1}^5 \sum_{i=1}^5 f_{5(i-1)+j} C_T^{i-1} M^{j-1} \quad (2.19)$$

These assumptions lead to a hybrid model approach for aircraft performance modeling. This description can be further streamlined into a hybrid system model which can serve as the formal modal description through which trajectory planning is based on [39]. Figure 2.2 illustrates a comparison between fuel consumptions recorded in QAR and computed via BADA with the inputs from QAR.

This real flight data example reveals some key points at performance modeling. We observe two types of discrepancies: Operational: In BADA based trajectory predictions, a single type of thrust setting is assumed, maximum climb for climb mode, low-idle for descent mode. On the other hand, thrust settings may differ at actual flights. Moreover, accelerations during cruise also cause differences. Parametric: Projected as bias from the actual fuel flow. The trend in reduction of aircraft mass is approximately the same but there is a slight bias error between the values. In this paper, we are focusing on a similar approach like BADA, which is separating the digital-twin model into sub-models based on flight modes. This classification can be achieved by unsupervised machine learning methods. Deep auto encoders can be utilized for this purpose. The idea behind is reducing the dimension of the data (encoding) and expanding it back to its original dimension with minimum information loss (decoding). Afterwards, clustering the data with reduced dimension yields segregated sets, which represent system modes. An example of this approach is illustrated with results in Figures 2.3 and 2.4.

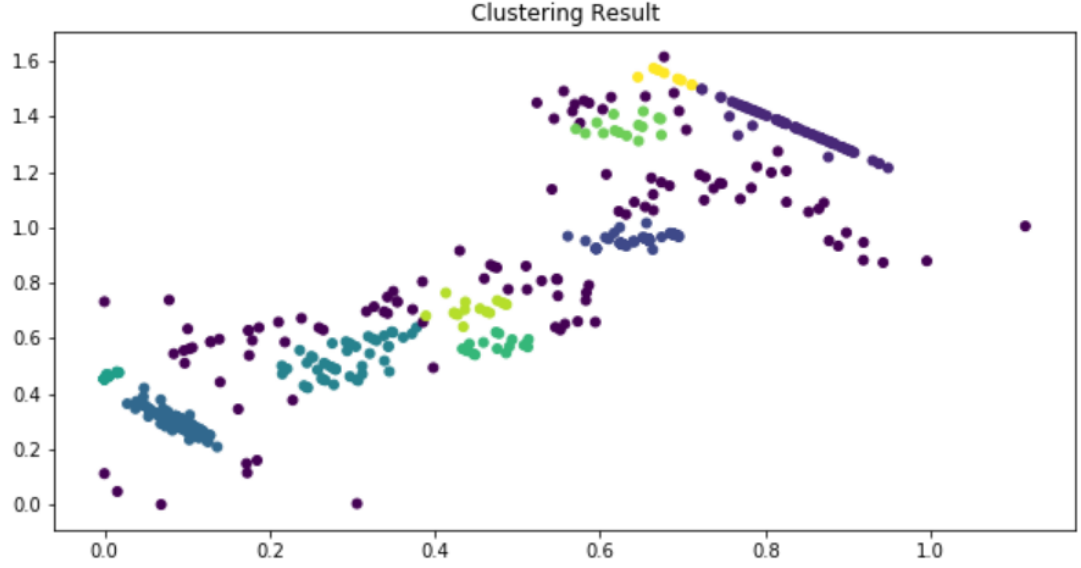


Figure 2.3 : Scatter plot of data with reduced dimension.

The feature set to this deep auto encoder network model is $S_{ae} = \{h, \dot{V}_{TAS}, ESF, AF, Tr_1, Tr_2\}$, where $Tr_{1,2}$ are throttle positions from QAR. ESF and AF stand for energy share factor and acceleration factor, respectively. These parameters are calculated as:

$$ESF = \left[1 + \frac{V_{TAS}}{g} \frac{dV_{TAS}}{dh} \right]^{-1} \quad (2.20)$$

$$AF = \frac{V_{TAS}}{g} \frac{dV_{TAS}}{dh} \quad (2.21)$$

Figure 2.3 represents the scatter plot of the data reduced to 2-dimension form. Separated sets can easily be observed and these are clustered through density-based clustering. When substituted into the actual data, this process yields plausible flight modes, as depicted in Figure 2.4. For instance, take-off and landing are extracted. Moreover, CAS/Mach climb and descent modes are observed. However, this process should be validated on other types of aircraft, too. It may also require much more data to generalize these modes.

In this paper, we propose adding flight mode based correction factors to BADA fuel consumption model. We chose to follow the modeling approach in BADA and selected *climb*, *cruise* and *descent* to be our main flight modes as all of them definitely exist in a typical flight.

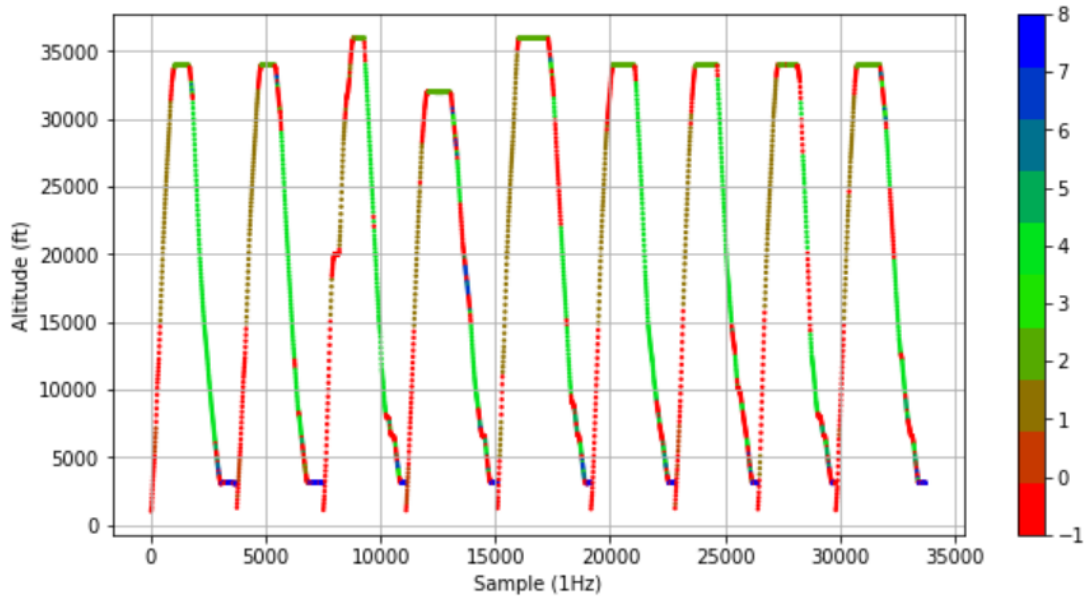


Figure 2.4 : Extraction of flight modes using deep auto encoders.

The objective is to find functions that map mode based input space to fuel flow from all engines:

$$f_{Q_j} : \mathbb{R}^{\rho(\cdot)} \rightarrow \mathbb{R} \quad (2.22)$$

where $Q_j \in Q$ and $\rho(\cdot)$ assigns the dimension of the input set for a given flight mode. The functions f_{Q_j} are represented by multilayer perceptrons and composed of several functions:

$$f_{Q_j} = f_{Q_j}^{(1)} \circ f_{Q_j}^{(2)} \dots \circ f_{Q_j}^{(L)} \quad (2.23)$$

where L is the number of layers in the neural network. Let x_{Q_j} and y_{Q_j} be input and output sets in mode Q_j and $x_{Q_j}, y_{Q_j} \in X$. A multilayer perceptron described in Eq. 7.8 defines the following mapping:

$$y_{Q_j} = f_{Q_j}(x_{Q_j}; \xi) \quad (2.24)$$

During training process, the model learns the value of ξ such that it results in the best function approximation. This supervised learning utilizes maximum likelihood estimation and optimizes the following cost function:

$$J_{Q_j}(\xi) = -\mathbb{E}_{x_{Q_j}, x_{Q_j} \sim \hat{p}_{data}} \log p_{model}(y_{Q_j} | x_{Q_j}) \quad (2.25)$$

where $p_{model}(x_{Q_j}; \xi)$ maps any configuration x to a real number estimating the true probability $p_{data}(x)$. The flight mode set is $Q = \{climb, cruise, descent\}$. Finally, the

correction factor τ corresponds to:

$$\tau = \frac{f_{Q_j}(x_{Q_j}; \xi)}{C_F(x_{Q_j})} \quad (2.26)$$

2.2 Cruise Fuel Consumption Dynamics

To understand the parameters affecting fuel consumption, we first need a thrust analysis. The analysis in this section are based on model-based approaches in BADA 4 [5], which leverages OEM performance models of Airbus [40] and Boeing [33]. Developers of BADA 4 showed the good approximation to these reference models [6]. The cruise phase of a flight is considered to be the equilibrium of both lateral and vertical forces:

$$L = W \quad (2.27)$$

$$Th = D \quad (2.28)$$

where L is the lift, D is the drag, W is the aircraft weight, and Th is the thrust. Equation 2.28 denotes that thrust required is the drag, and equals to the following aerodynamic equation:

$$Th_{req} = D = \frac{1}{2} \delta p_0 \kappa S M^2 C_D \quad (2.29)$$

where δ is the pressure ratio, $p_0 = 101,325 [N/m^2]$ is the air pressure at the sea level, $\kappa = 1.4$ is the adiabatic index, M is the Mach number, and C_D is the drag coefficient. The drag coefficient's dependencies are the lift L , and Mach number M . Thrust produced by the engines is a function of pressure ratio δ , Mach number M , and throttle setting δ_T :

$$Th = f(\delta, M, \delta_T) \quad (2.30)$$

Combining Equations 2.27, 2.29, we can formulate the thrust required in cruise as:

$$Th_{req} = f(\delta, M, \delta_T, W) \quad (2.31)$$

The throttle setting δ_T is also referred to RPM or N1. They all determine low-pressure spool speed, which proportionally affects how much fuel is injected into the combustion chamber. In cruise, autopilot systems adjust this parameter to maintain the equilibrium in Equation 2.29. As seen in Equation 2.29, pressure ratio dependency is considered to be proportional. Therefore, the nonlinearities are due to Mach number

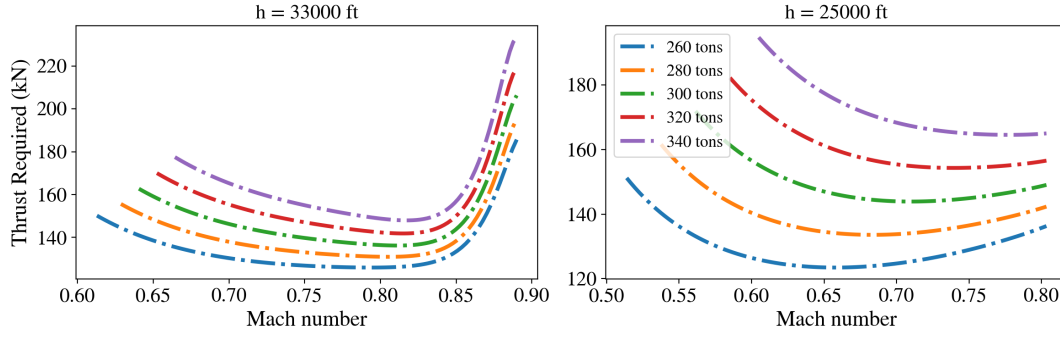


Figure 2.5 : Thrust required with increasing Mach number and mass.

and aircraft mass. Figure 2.5 illustrates Mach sensitivity of thrust required for selected altitude and mass values. The graphs show higher amounts of thrust required at lower altitudes because higher density air causes more friction with aircraft body-frame. Additionally, greater aircraft weight results in more drag due to drag-polar. Transition from thrust to fuel consumption is called *thrust specific fuel consumption*. Simplified analyses assume it to be constant at each flight level. However, a correction for the Mach number should also be considered because airspeed at the engine inlets affects the engine dynamics, as well. In summary, fuel consumption denoted as F is a function of thrust, Mach number, and altitude. Dependence on altitude is considered to be proportional to the pressure ratio and the square root of temperature ratio. This is shown through dimensional analysis and verified by experimental results [41]. We can formulate fuel consumption F as:

$$F = \delta \sqrt{\theta} T h (M, \delta_T) \quad (2.32)$$

In the predecessor of BADA, namely BADA3 [42], this is modeled as:

$$F = C_{f1} \left(1 + \frac{M \sqrt{\theta}}{C_{f2}} \right) T h \quad (2.33)$$

where the coefficients $C_{f1,2}$ are customized to aircraft family. BADA 4 fits high-order polynomials to the synthetic data generated from OEM performance models:

$$F = \delta \sqrt{\theta} \sum_p \sum_q a_{pq} \left(\frac{T h}{\delta} \right)^q M^p \quad (2.34)$$

where a_{pq} are the polynomial coefficients, similarly tailored to aircraft types. Equations 2.32, 2.33, and 2.34 are widely utilized real world applications. Figure 2.6 illustrates fuel flow sensitivity to Mach number for the same conditions presented

for thrust analysis. It is not clear to have a generalized statement about the Mach number effect. At high Mach numbers, fuel flow tends to increase. However, it is the opposite at low Mach cruises. As the altitude increases, the minimum Mach number would increase to compensate for lift loss due to decreased air density. This stands for high Mach number cruises at higher altitudes. However, it cannot be generalized to all conditions.

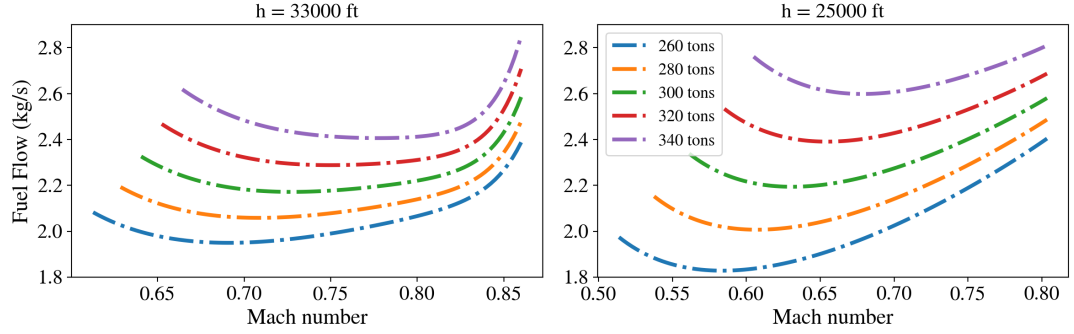


Figure 2.6 : Mach number vs fuel consumption for selected atmospheric conditions, and gross weight.

In summary, model-based approaches acknowledge five parameters that alter fuel consumption: (i) *pressure ratio* δ : appears as a proportional component for constant temperature, mass, and Mach number. (ii) *temperature ratio* θ : in analytical expressions, fuel consumption is proportional to the square root of the temperature ratio. As side effects, it determines the speed of sound and Mach number, and lower ambient temperatures allow the engine to operate at higher throttle settings [43]. Additionally, a higher atmospheric temperature causes an increase in drag by elevating the Reynolds number. (iii) *throttle setting* δ_T : an increasing throttle consistently escalates fuel flow injected to the fuel chamber. Note that, autopilot system computes the throttle position. Furthermore, how much it should be set depends on how much thrust to be produced. In cruise, this is also a function of aircraft mass because thrust required is drag. (iv) *Mach number* M : the Mach number effect on fuel consumption is a combination of its impact on drag, and thrust. Equation 2.29 reveals a proportional increase in drag with M^2 . Coupled with the impact on thrust, fuel consumption for given pressure ratio δ , temperature ratio θ , and aircraft mass m with increasing Mach number M results in a convex curve.

The dynamics described until here were model-based approaches utilized in real-world operations. Figure 2.7 illustrates how the data present these correlations. They are the

results of the combination of all factors. The monotonic relationships with pressure ratio, temperature ratio, and throttle setting are noticeable. However, the Mach number dependency is not clear. Moreover, it is not possible to filter the data for unique aircraft masses and have measurements for every possible Mach number like illustrated in Figure 2.6. As expected, aircraft operationally had flown only a small subset of the whole flight envelope. The Mach number distribution is skewed, and most of the population is around 0.83, which is usually the nominal cruise speed regime for this aircraft.

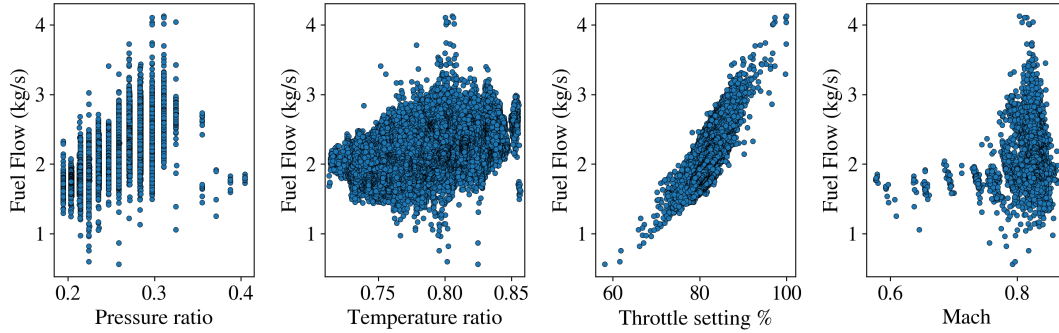


Figure 2.7 : Distributions of parameters from QAR dataset.

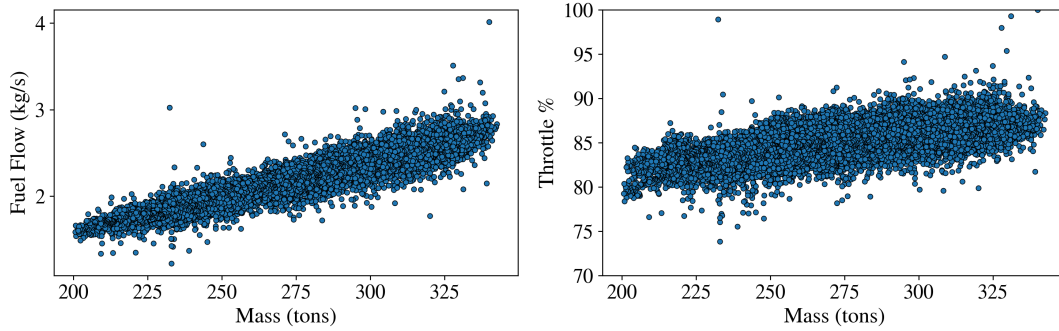


Figure 2.8 : The effect of gross weight on throttle setting and fuel consumption.

For a greater gross weight, an autopilot system sets throttle to a higher position to maintain Equation 2.28. Figure 2.8 depicts gross weight correlation with the throttle δ_T , and fuel consumption from data. Higher aircraft masses result in higher throttle settings and higher fuel flows.

Table 2.1 : Summary of flight and atmospheric parameters that affect fuel consumption.

Approach	Parameters that affect fuel consumption
Model-based	$\delta, \theta, \delta_T, M$
Data-driven	$\delta, \theta, \delta_T, m_{ac}, \mathcal{V}$

Finally, Table 2.1 summarizes parameters in the model-based approaches and QAR data that significantly affect fuel consumption. Statistical significance in the data is computed by Spearman rank, which measures the monotonic relations. Mach number is not in the list for the data because it does not have an acceptable variance. Instead, true airspeed appears, but it is the Mach number corrected with temperature ratio:

$$\mathcal{V} = 340.26M\sqrt{\theta} \quad (2.35)$$

Hence, the true airspeed's statistical significance is mostly due to the variation of altitude. This is the main issue with the operational data because even though the Mach number is an important parameter, it does not appear like one. The next section describes how we use fuel consumption dynamics to define a physics-based loss function and implement it into the ML framework.



3. DEEP LEARNING PRELIMINARIES

3.1 Neural Networks with Multi-Layer Perceptrons (MLP)

We adopt a fundamental multi-layer perceptron architecture to regress the fuel consumption, Y , using X . For a fully-connected network with L hidden layers, this amounts to the following modeling equations relating the input features x , to its target prediction y .

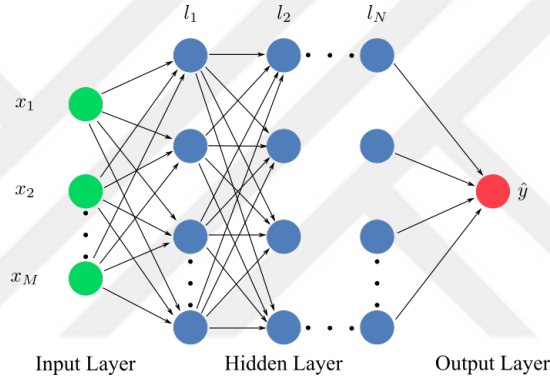


Figure 3.1 : Generic architecture of a Multi-Layer Perceptron.

To find the nonlinear mapping for the correction factor τ , we utilize deep neural networks, which have been proven to capture complex input-output relationships through gradient-based optimization and can be used to model any continuous function [44], [45]. A generic structure of a deep neural network consisting of a multilayer perceptron with M input features and N layers is illustrated in Figure 3.1. It is composed of sequentially connected layers, which comprise sets of neurons that are combinations of mathematical operations followed by nonlinear activation functions. The model parameters ξ are defined as $\xi = \{W, b\}$, where $W = \{w_i\}_{i=1}^N$, and $b = \{b_i\}_{i=1}^N$. The output of the l^{th} layer is:

$$f_{FF}(x, \xi_l) = f_{w_l, b_l}(x) = z_l \left(\sum_{j=1}^{N_l} w_{lj} x_j + b_l \right) \quad (3.1)$$

$$= Z_l (w_l^T x_l + b_l) \quad (l = 1, \dots, N) \quad (3.2)$$

where N_l is the neuron number and Z_l is a nonlinear activation function of a specific l^{th} layer, and x_l is the input of the l^{th} layer and also the output of the $(l-1)^{th}$ layer. In

addition to this, w_l and b_l are learnable parameters and called weight and bias terms of that layer respectively. The output of the last layer is a result of the composite and complex mapping defined as

$$\hat{y}(x) := (f_{w_n, b_n} \circ \dots \circ f_{w_1, b_1})(x) \quad (3.3)$$

Back-propagation algorithm [46] is used to train the neural network by utilizing gradient based optimization methods using the loss function given in Equation 3.4:

$$f_{FF} = \arg \min_{\xi} J_E(Y, \hat{Y}) + \lambda_s J_s(\xi) \quad (3.4)$$

To prevent overfitting and improve generalization capability of the neural network, an additional regularization term can be applied to the total loss function in Equation 3.4. Either L_1 , or L_2 norm penalties, which is also the model complexity loss are given in Equations 3.5 and 3.6 :

$$J_s(\xi) = J_{s_{L1}}(w) = \|\mathbf{w}\|_1 = |w_1| + |w_2| + \dots + |w_N| \quad (3.5)$$

$$J_s(\xi) = J_{s_{L2}}(w) = \|\mathbf{w}\|_2^2 = w_1^2 + w_2^2 + \dots + w_n^2 \quad (3.6)$$

and can be used to regularize the model. The L_1 regularization applies an L_1 norm penalty equal to the absolute value of the coefficient scale. It restricts the scale of the coefficients. L_1 may generate sparse models with few parameters, and specific coefficients may become zero and be discarded. L_2 regularization applies an L_2 penalty equal to the scale square of the coefficients. L_2 can not generate sparse models, and all coefficients are reduced by the same factor. Other regularization methods such as Dropout [47] may help preventing the overfitting for neural networks.

3.2 Optimization Algorithms

3.2.1 ADAM

Adam optimization algorithm [48] is used to update weights W and biases b . At each iteration t , the derivatives $\frac{dJ}{dW}$ and $\frac{dJ}{db}$ are computed for the current mini-batch. As the iteration grows, exponentially weighted averages of previous gradients are reflected on V_{dW} and V_{db} :

$$V_{dW}^{[l]} = \beta_1 V_{dW}^{[l]} + (1 - \beta_1) \frac{dJ}{dW}^{[l]} \quad (3.7)$$

$$V_{db}^{[l]} = \beta_1 V_{db}^{[l]} + (1 - \beta_1) \frac{dJ}{db}^{[l]} \quad (3.8)$$

where $\beta_1 = 0.9$. In a similar trend, S_{dW} and S_{db} are introduced to store exponentially weighted average of the squares of the previous gradients:

$$S_{dW}^{[l]} = \beta_2 S_{dW}^{[l]} + (1 - \beta_2) \left(\frac{dJ}{dW}^{[l]} \right)^2 \quad (3.9)$$

$$S_{db}^{[l]} = \beta_2 S_{db}^{[l]} + (1 - \beta_2) \left(\frac{dJ}{db}^{[l]} \right)^2 \quad (3.10)$$

where $\beta_2 = 0.999$. Note that the initial values of S_{db} , S_{dW} , V_{db} and V_{dW} are set to zero. After the process above for a single mini-batch, these values are corrected for bias as the iteration t grows. The prime symbol indicates the corrected value:

$$V'_{dW} = \frac{V_{dW}^{[l]}}{1 - \beta_1^t} \quad (3.11)$$

$$S'_{dW} = \frac{S_{dW}^{[l]}}{1 - \beta_2^t} \quad (3.12)$$

$$V'_{db} = \frac{V_{db}^{[l]}}{1 - \beta_1^t} \quad (3.13)$$

$$S'_{db} = \frac{S_{db}^{[l]}}{1 - \beta_2^t} \quad (3.14)$$

Finally, the weight matrices and the bias vectors of each layer, $W^{[l]}$ and $b^{[l]}$ respectively, are updated through the following equations:

$$W^{[l]} = W^{[l]} - \alpha \frac{V'_{dW}}{\sqrt{S'_{dW}} + \epsilon} \quad (3.15)$$

$$b^{[l]} = b^{[l]} - \alpha \frac{V'_{db}}{\sqrt{S'_{db}} + \epsilon} \quad (3.16)$$

In this process, α is the learning rate, β_1 , β_2 and $\epsilon = 10^{-8}$ are the hyper-parameters to adjust the decay rates of $\frac{dJ}{dW}$ and $\left(\frac{dJ}{dW}\right)^2$ (also for b).

3.2.2 Yogi

Yogi optimization algorithm [49] is used to update weights W and biases b . At each iteration t , the derivatives $\frac{dJ}{dW}$ and $\frac{dJ}{db}$ are computed for the current mini-batch. As the iteration grows, exponentially weighted averages of previous gradients are reflected on V_{dW} and V_{db} :

$$V_{dW}^{[l]} = \beta_1 V_{dW}^{[l]} + (1 - \beta_1) \frac{dJ}{dW}^{[l]} \quad (3.17)$$

$$V_{db}^{[l]} = \beta_1 V_{db}^{[l]} + (1 - \beta_1) \frac{dJ}{db}^{[l]} \quad (3.18)$$

where $\beta_1 = 0.9$. In a similar trend, S_{dW} and S_{db} are introduced to store exponentially weighted average of the squares of the previous gradients:

$$S_{dW}^{[l]} = S_{dW}^{[l]} - (1 - \beta_2) \text{sign} \left(S_{dW}^{[l]} - \left(\frac{dJ}{dW}^{[l]} \right)^2 \right) \left(\frac{dJ}{dW}^{[l]} \right)^2 \quad (3.19)$$

$$S_{db}^{[l]} = S_{db}^{[l]} - (1 - \beta_2) \text{sign} \left(S_{db}^{[l]} - \left(\frac{dJ}{db}^{[l]} \right)^2 \right) \left(\frac{dJ}{db}^{[l]} \right)^2 \quad (3.20)$$

where $\beta_2 = 0.999$. Note that the initial values of S_{db} , S_{dW} , V_{db} and V_{dW} are set to zero. After the process above for a single mini-batch, these values are corrected for bias as the iteration t grows. The prime symbol indicates the corrected value:

$$V'_{dW} = \frac{V_{dW}^{[l]}}{1 - \beta_1^t} \quad (3.21)$$

$$S'_{dW} = \frac{S_{dW}^{[l]}}{1 - \beta_2^t} \quad (3.22)$$

$$V'_{db} = \frac{V_{db}^{[l]}}{1 - \beta_1^t} \quad (3.23)$$

$$S'_{db} = \frac{S_{db}^{[l]}}{1 - \beta_2^t} \quad (3.24)$$

Finally, the weight matrices and the bias vectors of each layer, $W^{[l]}$ and $b^{[l]}$ respectively, are updated through the following equations:

$$W^{[l]} = W^{[l]} - \alpha \frac{V'_{dw}}{\sqrt{S'_{dw}} + \epsilon} \quad (3.25)$$

$$b^{[l]} = b^{[l]} - \alpha \frac{V'_{db}}{\sqrt{S'_{db}} + \epsilon} \quad (3.26)$$

3.3 Data Preprocessing for Training

In all these three models, we only consider climb, cruise and descent modes. In order to separate all QAR trajectory data with respect to these flight phases, we utilized a heuristic approach. Let $\mathbb{X} = \{x^{(1)}, \dots, x^{(m)}\} \in \mathbb{R}^{m \times n}$ denote the historical trajectory set of a single aircraft, consisting several concatenated flights. Even though the detailed modal description of an aircraft trajectory consists of more discrete modes as described in Section 2.1, we considered Q^{cmb} , Q^{des} and Q^{crz} as they cover almost all the flight modes. Altitude gradient is selected as the proper metric to extract these 3 modes from the data:

$$\mathbb{X}_{cmb} = \left\{ x^{(i)} \mid \dot{h}^{(i)} \geq 3ft/s, \quad h^{(i)} \in x^{(i)} \in \mathbb{R}^{p_{cmb}(\cdot)} \right\} \quad (3.27)$$

$$\mathbb{X}_{crz} = \left\{ x^{(i)} \mid |\dot{h}^{(i)}| \leq 1ft/s, \quad h^{(i)} \in x^{(i)} \in \mathbb{R}^{p_{crz}(\cdot)} \right\} \quad (3.28)$$

$$\mathbb{X}_{des} = \left\{ x^{(i)} \mid \dot{h}^{(i)} \leq -3ft/s, \quad h^{(i)} \in x^{(i)} \in \mathbb{R}^{p_{des}(\cdot)} \right\} \quad (3.29)$$

where $x^{(i)}$ refers to the i^{th} row of the data and columns are the selected features for the model.

After training sets and test sets are generated, two types of scaling are applied to map all values between $[0, 1]$ or to a distribution with a zero mean and one variance. This is crucial as the range of selected parameters differ from each other and causes convergence problems during training. For example, Mach number and altitude take values at distinct intervals: $M \in [0.15, 0.84]$ and $h \in [0, 43000]$. Training without such scaling causes large scaled variables to outbalance the cost function and model's sensitivity to other parameters could be faded. Let $\mathbb{X}_{(\cdot)}^{sc}$ denote the scaled data. The first scaling is min-max scaling:

$$\mathbb{X}_{(\cdot),trn}^{sc} = \frac{\mathbb{X}_{(\cdot),trn} - \min \mathbb{X}_{(\cdot),trn}}{\max \mathbb{X}_{(\cdot),trn} - \min \mathbb{X}_{(\cdot),trn}} \quad (3.30)$$

Note that the min-max scaling is done through column by column. $\min \mathbb{X}_{(\cdot),trn}$ and $\max \mathbb{X}_{(\cdot),trn}$ are the countable sets of the minimum and the maximum values of the features in the training set. The same scaling with the same parameters is then applied to the test set:

$$\mathbb{X}_{(\cdot),tst}^{sc} = \frac{\mathbb{X}_{(\cdot),tst} - \min \mathbb{X}_{(\cdot),trn}}{\max \mathbb{X}_{(\cdot),trn} - \min \mathbb{X}_{(\cdot),trn}} \quad (3.31)$$

The second scaling is standard scaling and defined as:

$$\mathbb{X}_{trn}^{sc} = \frac{\mathbb{X}_{trn} - \mu(\mathbb{X}_{trn})}{\sigma(\mathbb{X}_{trn})} \quad (3.32)$$

The same parameters, namely the mean and standard deviation indicated in the equation above are applied to the test set as:

$$\mathbb{X}_{tst}^{sc} = \frac{\mathbb{X}_{tst} - \mu(\mathbb{X}_{trn})}{\sigma(\mathbb{X}_{trn})} \quad (3.33)$$

3.4 Data Used in the Thesis

3.4.1 Operational flight data

In this thesis, Quick Access Recorder (QAR) data from a major European flag carrier airline is utilized for the design of deep learning neural network architectures. The dataset comprises of a myriad of features sampled at 1Hz. The dataset of interest for this study includes information about aircraft's dynamic states (airspeed, altitude, position, vertical speed, heading), body axis states (angle of attack, longitudinal and lateral acceleration), performance states (throttle settings, fuel flow from engines, mass), configuration states (high lift devices, landing gear, speed brake) and environmental states (wind speed/direction, static and total temperature of air, air pressure). Even though these states are the most critical variables used in this kind of research, a QAR device usually records more than 1000 features. Table 1 summarizes the variables selected from the QAR dataset used.

The QAR data used for the analysis presented in this study is 88 long-haul flights of a single tail number Boeing 777-300ER aircraft. This data is extracted from a set of more than 1000 long-haul flights. Even though the QAR data is highly reliable, some parameters may indeed be not consistent to the actual values. These inconsistencies

Table 3.1 : Summary of parameters directly used from QAR dataset.

h	Barometric altitude [ft]
T_c	Static air temperature [C]
V_{CAS}	Calibrated airspeed [kt]
M	Mach speed
m	Aircraft mass [kg]
F_{eng}	Fuel flow from engines [lb/hr]
$Tr_{1,2}$	Throttle positions of engines 1 and 2
V_{wind}	Wind speed [kt]
χ_{Wind}	Wind direction [deg]
χ_{ac}	True heading [deg]
δ_{flap}	Flap deflection [deg]
δ_{lg}	Landing gear status $\{0, 1\}$
δ_{sb}	Speed brake deflection [deg]
F_{apu}	Apu fuel consumption [lb/hr]

are omitted or modified through domain expertise provided from the associated airline. For instance, although the maximum take-off mass of a Boeing 777-300ER is 353 tons, the gross weight column in the QAR data indicated aircraft mass more than this value. Additionally, the change rate of gross weight and fuel flow from engines and Auxiliary Power Unit do not overlap each other. Aircraft mass is an essential parameter for performance modeling and training with this corrupted information may lead to improper results. Hence, the following modification is applied:

$$m^{(i)} = m_{EZF} - 0.4359 \frac{F_{eng}^{(i)} + F_{APU}^{(i)}}{3600} \quad (3.34)$$

where m_{EZF} is the zero fuel mass. This value represents aircraft mass without fuel on-board and varies with flight because payload changes. In addition to aircraft mass, true airspeed is also derived from the available parameters:

$$V_{TAS}^{(i)} = M^{(i)} \sqrt{\kappa R (T_c^{(i)} + 273)} \quad (3.35)$$

Table 3.2 : Summary of parameters directly used from flight plan dataset.

h	Altitude [FL]
ISA	ISA temperature deviation [C]
V_{CAS}	Calibrated airspeed [kt]
V_{TAS}	True airspeed [kt]
M	Mach speed
m	Aircraft mass [kg]
V_{wind}	Wind speed [kt]
χ_{Wind}	Wind direction [deg]
χ_{ac}	True heading [deg]

Finally, derivatives of true airspeed and altitude are calculated by a simple process defined as follows:

$$\dot{h}^{(i)} = \lim_{\delta_t \rightarrow 1} \frac{h^{(i+\delta_t)} - h^{(i)}}{\delta_t} \quad (3.36)$$

$$\dot{V}_{TAS}^{(i)} = \lim_{\delta_t \rightarrow 1} \frac{V_{TAS}^{(i+\delta_t)} - V_{TAS}^{(i)}}{\delta_t} \quad (3.37)$$

where $\delta_t \in \mathbb{N}$.

AWY MOCA	WPT NAME/FIR LAT/LONG	FRQ	FL TRO SHR	MT TT VAR	WIND SAT TDV	TAS MN G/S	DIST REMD ACCD	TIME ACCT REMT	ETA ATA REV	RQRD ACCF FOB
VADEN1Y 044	TOC VADEN/-15NM N41557E027291		CLB	299 305 06E			58 4442 87	9 0016 0921		80573 6223
VADEN1Y 044	VADEN LBSR SOFIA FIR N42039E027130		300 407 01	299 305 06E	314/027 M42 P2	491 .829 465	15 4427 102	2 0018 0919		80282 6514
L610 052	UTEKA LYBAF BELGRADE FIR N43547E022235		300 414 01	294 299 05E	252/028 M43 P1	490 .829 471	240 4187 342	31 0049 0848		75685 11111
DCT 053	S/C(320) ESIRI/-33NM N44026E022058		CLB	297 302 05E			15 4172 357	1 0050 0847		75242 11554
DCT 058	ESIRI ESIRI N44195E021278		320 404 01	297 302 05E	240/032 M48 P0	486 .831 471	33 4139 390	5 0055 0842		74636 12160
DCT 025	PARAK LHCC BUDAPEST FIR N46098E020057		320 401 01	327 332 05E	228/040 M48 P0	486 .831 494	125 4014 515	15 0110 0827		72452 14344

Figure 3.2 : An example of filed flight plan.

3.4.2 Flight plan data

The second dataset utilized within this study is actual filed flight plans from Airline Operation Centers (AOC) following the estimates of take-off weight, fuel on-board, wind and the flight trajectory. However, this data of the filed flight plan is much less detailed and coarse in comparison to QAR. Figure 3.2 reveals an example of a filed flight plan. Since these documents are aimed to present the flight operation in an understandable language, the frequency of trajectory data is based on the waypoints that are entered to Flight Management Computers. Hence, trajectory information is kind of an abstraction of the one that exists in a QAR dataset. A flight with a distance of 5900nm corresponds to approximately 43000 trajectory points in QAR where as it corresponds to approximately 130 trajectory points in a filed flight plan. We utilized these flight plans as a baseline for comparison between default and digital-twin models. In total, we have 480 filed flight plans of the investigated aircraft with specific tail number. Unlike its QAR trajectories, flight plan trajectories consist of both short and long-haul routes.

Flight plan represented in Figure 3.2 corresponds to the final publication version and the corresponding flight plan data is stored in XML files (with potential data omissions and format changes). Table 2 summarizes the features which are used from the filed flight plans. Even though these parameters are similar to the ones within QAR, couple of modifications are done as to be concise. First, altitude values in flight plans are represented as flight levels. The following correction is applied:

$$h^{(i)} = 100h^{(i)} \quad (3.38)$$

to obtain the corresponding altitude. Second, static air temperature is calculated by correcting ISA temperature with predicted deviation:

$$T_c^{(i)} = 15 - 1.9812 \frac{h^{(i)}}{1000} - ISA^{(i)} \quad (3.39)$$



4. DATA-DRIVEN FUEL CONSUMPTION MODELING

An aircraft performance model is one of the essential tools for flight planners, which are used by airline operation centers. These models considering aircraft types, have generic specifications employ universal model tables by sacrificing the precision. However, individual aircraft have their unique flight performance due to aging, variation in flight and maintenance, user-configurable specifications, etc. Flight planning with high precision modeling tailored to each tail number provides an enabling solution towards fuel-optimal operations as they allow estimation of take-off weight and fuel on-board.

In this context, proceeding aerospace operations with digital twin models is vital because even aircraft of the same family may differ from each other in terms of flight performance. This problem is usually handled at airline operation centers by setting a "performance factor" as a correction rate assigned to each tail number. The ground-based flight planning tools of airlines utilize performance charts from aircraft manufacturers, which are generated through wind tunnel experiments and flight tests. The performance factor is applied to update the models based on these databases.

The role of high-fidelity models for aircraft fuel consumption on air traffic operations has been widely studied. Some studies investigated its effects on airline costs [50], [51], [52], whereas some analyzed the problem from environmental aspects [2], [53], [54]. As an application of fuel flow models, [55] and [56] viewed the problem from the airline's aspect and proposed speed and altitude optimization techniques. These kinds of research topics require a good estimation of fuel burn. In this context, Aviation Environmental Design Tool (AEDT) of FAA [36], Base of Aircraft Data (BADA) of Eurocontrol [38] and ICAO Aircraft Engine Emissions Databank [57], are broadly used by the community [58], [59]. However, these are generalized models for each aircraft type and use several assumptions to cover all phases of a flight as much as possible. In this chapter, we provide a generic correction factor that can be tailored to each tail number and captures all operational and parametric discrepancies. There

are also previous works in the literature that focus on to overcome such problems with similar approaches to ours. Some researchers worked on improving the baseline fuel burn estimators utilizing QAR data. The study in [30] used flight data to estimate the fuel burn of a taxiing aircraft by linear function approximation. Machine learning techniques are applied to the QAR data in [24], [26], and [25] for the same purpose, and the scope is extended to climb, cruise, and descent phases. Another study in [60] deployed artificial neural networks for fuel burn estimation but did not work with QAR trajectories. On the other hand, none of those, as mentioned studies incorporated the throttle information, which has a substantial correlation with fuel flow. Since this parameter is only available in a QAR dataset, it brings the question of how it is going to be selected considering the flight conditions. The baseline performance models assume specific thrust settings; however, this may substantially differ in reality considering the variety of flight operations.

In this study, we propose a new approach to applying the correction process for flight performance models. We use historical aircraft trajectories, namely Quick Access Recorder (QAR) data and state-of-the-art deep learning algorithms, to acquire information about "tail number specific" performance. Recent advances in deep learning algorithms and high computation systems enable us to process this data and pave the way to built digital-twin models for fuel consumption of aircraft. We aim to have sustainable and live models updated as new trajectory data are provided. Specifically, we offer a solution to the problem of building an accurate fuel flow model covering the major part of a flight operation. We propose two-level methodology: first, the model utilizes BADA equations and physically consistent structure; then, it generates a new feature to recover the information loss from the unavailability of throttle positions. In the second level, we build a cascaded system, where throttle positions are estimated first in advance to fuel flow estimation. Moreover, we demonstrated how successful the training is given the throttle positions, and the developed digital-twin models are also applied to actual filed flight plans to illustrate potential fuel savings.

The rest of the thesis is organized as follows: In Section 2, we provide the hybrid system model for aircraft, which serves as the basis of the digital twin performance model. The system description is based on the BADA Formulation, where correction

factors are integrated through Machine Learning. In Section 3, the data used in this Project is given in detail. Section 4 explains how the data is prepared, introduces the proposed models for fuel burn estimation, and provides a mathematical background of the deep learning algorithms. In Section 5, we compare the estimation results of the models, illustrate sensitivity analysis and discuss the behavior of the correction factor. Section 6 shows an example usage of these models on actual filed flight plans and provides a comparison of fuel consumption with respect to the baseline model. Finally, we comment on the results and discuss the encountered issues in Section 7.

4.1 Digital Twin Performance Modeling via Machine Learning

This section describes three distinct digital twin performance models obtained through machine learning techniques. In the first one, we build a black-box model utilizing only QAR dataset itself. Using domain expertise and correlation tables, we select the features that correlate with fuel flow the most. Then we build the black-box model via artificial neural networks. This specific model corresponds to the in-flight fuel flow estimation case where throttle information from all the engines are available. However, this is not the case for the ground planning use case. To address this problem, we have designed two other architectures. The second model considered within this thesis incorporates artificially generated δ_T data (using dynamically consistent data regeneration following the BADA equations) into the QAR data and adds new features to create a physically consistent system. Through this methodology, we can obtain a fuel flow model consistent with data that will be available at the ground planning phase. In the third and the last model, we use a cascade neural network system, where throttle positions are estimated through a neural network architectures at first and substituted into the fuel flow estimator obtained in the very first model. In this model, we again get a fuel flow model consistent with data that will be available at the ground planning phase. Following this generalistic perspective, dividing the generated results by the nominal BADA fuel flow model indeed correspond to correction factors to BADA based fuel flow model specific to the tail number of the associated aircraft. The resulting correction factors are a function of the flight conditions and the features as captured by the associated neural network architectures.

Table 4.1 : Summary of data sizes used in training.

	Train	Validate	Test
Climb	109k	5.4k	11k
Cruise	2.1M	105k	235k
Descent	133k	6.7k	13k

All three models have common processes. At first, we split the data into training and test sets. %10 of the data sets for each mode are separated as test set; $\mathbb{X}_{(\cdot),trn}$ and $\mathbb{X}_{(\cdot),tst}$, respectively. During the training %5 the training data for each mode are used as validation set $\mathbb{X}_{(\cdot),val}$. The essential point is, the data is shuffled and split by using a stratified sampling method with respect to the target variable. Data is divided into small bins by using its histogram, and then stratified sampling algorithm has applied. As a result of this, training and test sets have been included approximately the same amount of binned values and make training and test distribution as similar as possible. This provides that validation and test performance illustrates the expected performance of the model with less bias and variance. Generally, well-known performance metrics for regression are mean squared error and mean absolute error. By using either metric, the error generated from an individual sample typically increases as its target value increases, in that a fixed percentage error gives a larger absolute difference for greater target values. As a result of this, inadequate distribution concinnity will increase the variance in the result. Let l represent l^{th} layer in the network. The output of the layer is calculated as:

$$z^{[l](i)} = g(W^{[l]}x_{(\cdot),trn}^{(i),sc} + b^{[l]}) \quad (4.1)$$

where $x_{(\cdot),trn}^{(i),sc}$ is the i^{th} training sample from $\mathbb{X}_{(\cdot),trn}^{sc}$, the weight matrix is denoted by $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, $b^{[l]} \in \mathbb{R}^{n_l}$ is the bias vector, g is the activation function and n_l is the number of neurons in the l^{th} layer. In the last layer, which is the output layer, final value $\hat{y}^{(i)}$ is calculated. The other common equations employed in all models and training related hyper parameters are summarized below.

RELU is selected as the activation function for all layer except the output layer:

$$g(z) = \max(0, z) \quad (4.2)$$

In order to assign initial values to weights $W^{[l]}$ at each layer l , He initialization [61] is selected.

$$W_0^{[l]} \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_l}}\right) \quad (4.3)$$

Mini-batch training [62] is used to train the deep neural networks. $\mathbb{X}_{(\cdot),trn}^{sc}$ is processed by being divided into n subsets with equal length.

$$z^{[l](k)} = g(W^{[l]}x_{(\cdot),trn}^{(k),sc} + b^{[l]}) \quad (4.4)$$

where $k = \{\frac{m}{n}(t-1) + 1, \frac{m}{n}(t-1) + 2, \dots, \frac{m}{n}t\}$ and t is the iteration number. Batch size is selected as $\frac{m}{n} = 512$. The loss function is selected as the mean absolute error:

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m |y^{(i)} - \hat{y}^{(i)}| \quad (4.5)$$

where $y^{(i)}$ is the actual output value in the data for the given input $x^{(i)}$. We tuned the learning rate α by checking the status of loss function defined in 7.26. If the loss function is not improved 10 times consecutively, α is reduced by %90. Its initial value is chosen as 0.001. All models are trained for 2000 epochs. Size of train, validation and test datasets are illustrated in Table 3. All Deep Neural Network models are implemented and trained with Apache MXNET framework 1.2.0 [63]. Hidden layer size hyper-parameter is selected to 5. Dropout ($keepprob. = 0.4$) and L2 ($\lambda = 0.001$) regularization methods [64] are also implemented and used separately.

4.1.1 Black-box fuel flow estimation with QAR throttle

In the first model, a black-box fuel estimator is developed. By the phrase "black-box", we mean utilizing only the features that are available in QAR. Distinct fuel flow estimation networks are built for climb, cruise and descent. In all networks, the output is total fuel flow from all engines, in $[kg/s]$. Input features are selected with the similar modeling fashion in BADA. Additionally, throttle positions are added to the feature set as they represent very high correlation with fuel flow.

The input feature set for *climb* and *descent* is given in Table 4. The neural network consists of 5 hidden layers, which have 512, 256, 256, 128 and 128 neurons, respectively. Note that the elements of this input set are not completely orthogonal to each other. For instance, static air temperature decreases with increasing altitude. True airspeed and Mach can be converted to each other via equation 3.35. However,

Table 4.2 : Input features of the black box model for climb and descent.

h	Altitude [ft]
T	Static air temperature [C]
m	Aircraft mass [kg]
V_{TAS}	True airspeed [m/s]
M	Mach speed
Tr_1	Throttle 1 relay angle [deg]
Tr_2	Throttle 2 relay angle [deg]

static air temperature also differs with geographic locations and the flights used in this study passes from varying locations with distinct climates. As for using Mach and true airspeed at the same time, the sensitivity of aircraft performance is higher at higher altitudes. Hence, true airspeed is also utilized in climb and descent models to prevent information loss at especially altitudes lower than 10,000 ft.

Table 4.3 : Input features of the black box model for cruise.

h	Altitude [ft]
T	Static air temperature [C]
m	Aircraft mass [kg]
M	Mach speed
Tr_1	Throttle 1 relay angle [deg]
Tr_2	Throttle 2 relay angle [deg]

The input feature set for *cruise* is given in Table 5. The neural network consists of 5 hidden layers, which have 2048, 1024, 512, 256 and 128 neurons, respectively. The number of neurons is increased in *cruise* model, because it reflects approximately %90 of the flight dataset and comprises more hidden features than climb and descent.

The estimation results on test sets for climb, cruise and descent are illustrated in Figure 4.1. The plot illustrates the result for a randomly selected flight.

Mean absolute errors on the test sets are $0.0195[kg/s]$ for climb, $0.0143[kg/s]$ for cruise and $0.0236[kg/s]$ for descent. The results show a good approximation to fuel flow for all three modes. All operational and parametric discrepancies are covered. However,

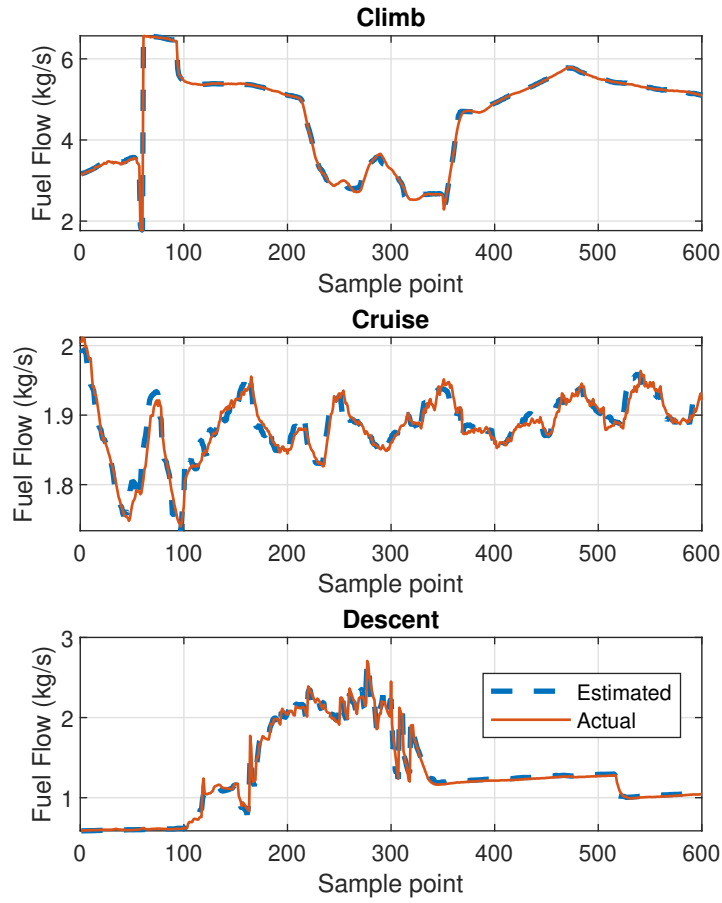


Figure 4.1 : Fuel flow estimation with throttle information from QAR.

the major contributor to this achievement is throttle information. The same training is done with the same model and inputs but throttle relay angles. Mean absolute errors on the test sets are $0.1247[kg/s]$ for climb, $0.0430[kg/s]$ for cruise and $0.0751[kg/s]$ for descent. It can be seen that information loss is huge. This makes throttle information a crucial parameter. However, neither itself nor its function approximation is available in ground based trajectory generation tools. The next two models present approaches to deal this problem.

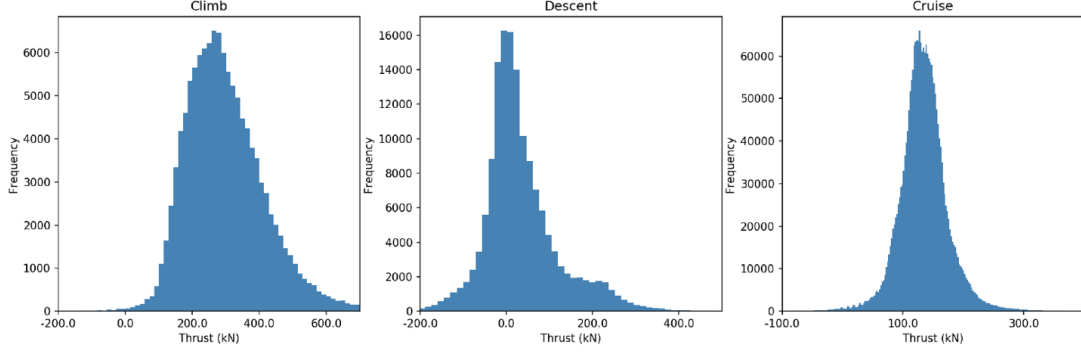


Figure 4.2 : Distributions of regenerated thrust utilizing QAR and BADA.

4.1.2 Fuel flow estimation with dynamically consistent thrust regeneration

In this model, the aim is to recover the information loss for not utilizing QAR throttle directly. We proposed dynamically consistent thrust regeneration using BADA intuition for this purpose. Thrust can be derived from the modified version of equation 5:

$$Th = m(\dot{V}_{TAS} + g \sin \gamma) + D \quad (4.6)$$

Flight path angle γ can be computed by equation 4:

$$\gamma \approx \frac{\dot{h}}{V_{TAS}} \quad (4.7)$$

Note that V_{TAS} , $\dot{V}_{TAS}$, $\dot{h}$, γ and m are already available or can be computed from QAR. The only remaining term is the Drag, D . We compute drag by equation 12. Additionally, we use the following equation from BADA to calculate drag when high lift devices, landing gear or speed brakes are deployed. This is necessary for climb and descent modes.

$$C_D = dn_{i^*,1} + dn_{i^*,2}C_L + dn_{i^*,3}C_L^2 + k\delta_{sb} \quad (4.8)$$

where matrix $dn \in \mathbb{R}^{N \times 3}$ comprises the coefficients for corresponding configuration. Let $A \in \mathbb{R}^{N \times 2}$ be the matrix representing flap and landing gear values defined in BADA. One can find the correct polynomial coefficients by $i^* = \arg \min_{i \leq N} ||\{\delta_{fp}, \delta_{lg}\} - A_{i,1-2}||$. Contribution of speed break to drag coefficient is computed by multiplying its level by a constant k . Figure 4.2 illustrates histogram plots of thrust force regenerated by equation 4.6. The scales and the expected values show consistency with pre-computed look-up tables at BADA.

Throttle information is then reflected on δ_T by calculating it through C_T and using equations 14 and 15 and the real root of δ_T is selected. Finally, recently derived throttle

parameter δ_T is given to the network as a new feature. Note that there could also be error in drag and thrust. We reflect all these errors on fuel flow.

Table 4.4 : Input features of the model with δ_T parameter for climb and descent.

h	Altitude [ft]
T	Static air temperature [C]
m	Aircraft mass [kg]
V_{TAS}	True airspeed [m/s]
M	Mach speed
δ_T	Throttle parameter [$-$]
γ	Flight path angle [rad]
γ/δ_T	[rad]

The input set of the neural network for *climb* and *descent* is given in Table 6. In order to reflect how much of the excess power is distributed to acceleration or climbing, the ratio between γ and δ_T is also added to the feature set. It is also placed in descent mode. The neural network consists of 5 hidden layers, which have 1024, 512, 256, 256 and 128 neurons, respectively.

As for cruise, flight path angle γ is discarded from the feature set. It should be zero during cruise but γ derived from QAR is noisy and training with this information could lead over-fitting. In result, feature set for *cruise* is provided in Table 7.

Table 4.5 : Input features of the model with δ_T parameter for cruise.

h	Altitude [ft]
T	Static air temperature [C]
$1/m$	Aircraft mass [kg^{-1}]
M	Mach speed
δ_T	Throttle parameter [$-$]

Distinctively, we chose $1/m$ for aircraft mass information. The algorithm gave better results compared to the training without taking inverse of m .

Mean absolute errors on the test sets are $0.0293[kg/s]$ for climb, $0.0402[kg/s]$ for cruise and $0.0352[kg/s]$ for descent. As can be seen in Figure 4.3 even though utilizing throttle information from QAR gives better results, training with δ_T is also satisfactory.

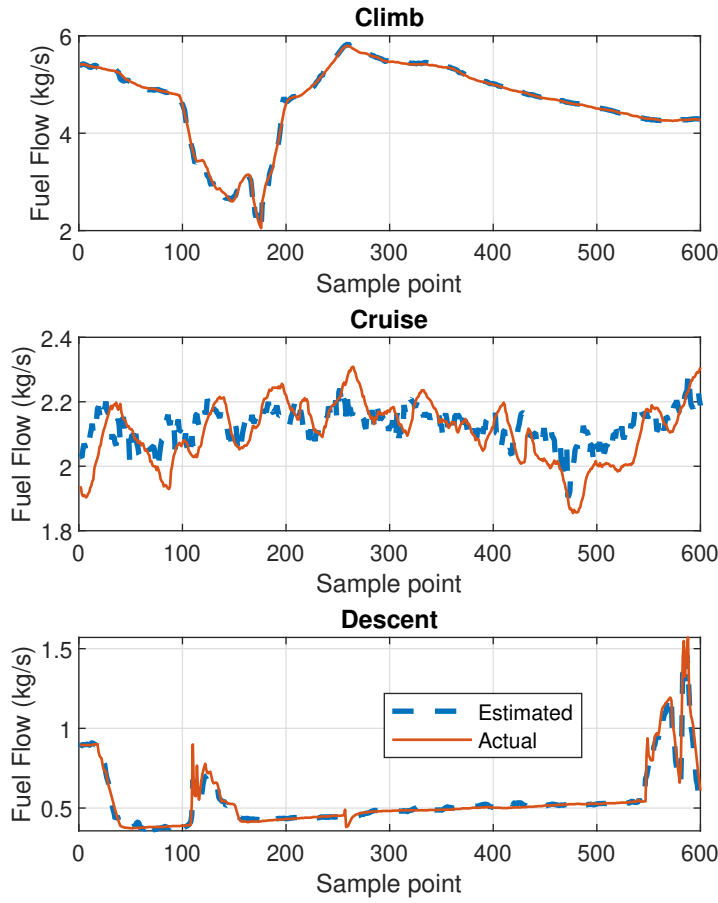


Figure 4.3 : Estimation results of training with δ_T for climb, cruise and descent.

Especially errors in climb and descent phases are close to the first model. However, fuel flow error in cruise is a bit larger. The reason behind this could that regenerated throttle parameter δ_T may not reflect all actual throttle movements.

4.1.3 Fuel flow estimation with throttle/auto throttle estimation

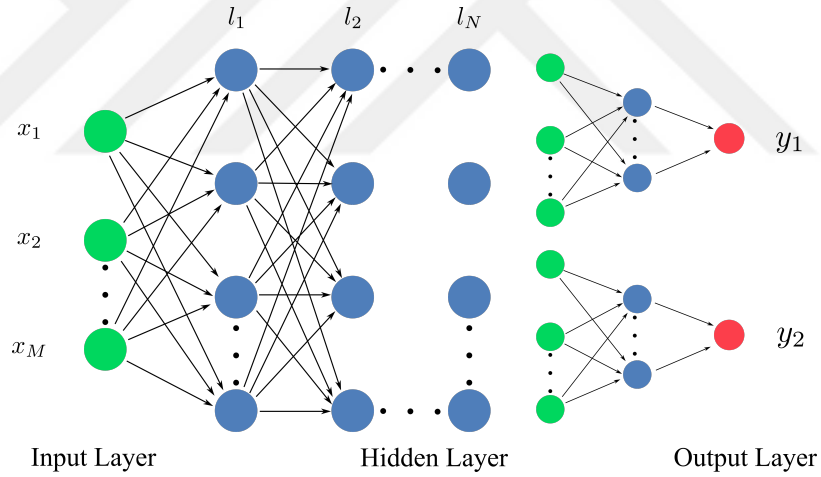
In the last model, we propose a cascaded system: First, throttle positions are estimated. Then, estimated throttle positions are substituted into the first fuel flow estimator. All modes share the same feature set for throttle estimation. Inputs are illustrated in Table 8. The outputs are throttle positions, Tr_1 and Tr_2 . In this feature set, we also included wind information in order to capture auto-throttle functionalities. This includes adjusting throttle positions with respect to head or tail winds to reach target speed. In a similar concern, we added calibrated airspeed as speed intent of an aircraft is defined in terms of calibrated airspeed and Mach in Flight Management Computers.

Table 4.6 : Input features of the throttle estimation network.

h	Altitude $[ft]$
T	Static air temperature $[C]$
$1/m$	Aircraft mass $[kg^{-1}]$
M	Mach speed
Th	Thrust $[N]$
$Th \times M$	$[N]$
$V_{wind,avg}$	Component of wind speed on true airspeed $[kt]$
V_{CAS}	Calibrated airspeed $[kt]$
γ	Flight path angle $[rad]$

Component of wind speed on true airspeed is calculated as:

$$V_{wind,avg}^2 = V_{wind}^2 + V_{TAS}^2 - 2V_{wind}V_{TAS}\cos(\chi - \chi_{wind}) \quad (4.9)$$

**Figure 4.4 :** Custom architecture of throttle estimation for climb, cruise and descent.

In this model, a multi task learning based custom deep neural network architecture is built. This structure is depicted in Figure 4.4. First part of the network consists of 5 hidden layers with 512, 256, 256, 128 and 128 neurons. The network then splits into two other networks, which are comprised of 3 hidden layers with 32, 16 and 8 neurons.

Figure 4.5 illustrate estimation results for throttle positions. Mean absolute errors are $0.271[deg]$ for climb, $0.304[deg]$ for cruise and $0.492[deg]$ for descent. After substituting estimated throttles into the first model, fuel flow is obtained as illustrated in Figure 4.6.

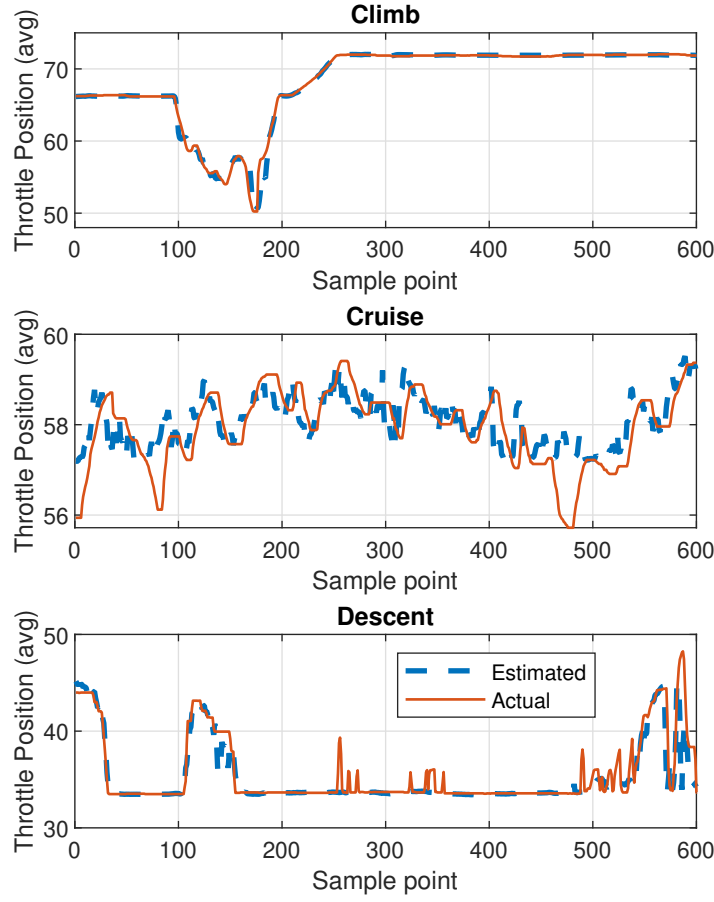


Figure 4.5 : Throttle estimation for climb, cruise and descent.

Mean absolute errors in fuel flow estimation with this approach are $0.046[kg/s]$ for climb, $0.042[kg/s]$ for cruise and $0.049[kg/s]$ for descent. This model also give satisfactory results although there are some cases where throttle positions are could not estimated correctly. The reason behind this could be the events during flight which are challenging to be approximated through functions. For instance, when auto-throttle is disabled, throttles are controlled directly with pilot intent. Moreover, instantaneous level changes may not be modeled through analytic approximations.

4.2 Comparison of ML Models and Sensitivity Analysis of the Correction Factor

In this section, we illustrate comparisons between proposed models and some intuitions about correction factors based on flight modes.

As can be seen in Figure 4.7, providing the actual throttle information from QAR gives the better approximation to the actual fuel flow. Proposed approaches with the cascade network and model with δ_T parameter also yield satisfactory results as they

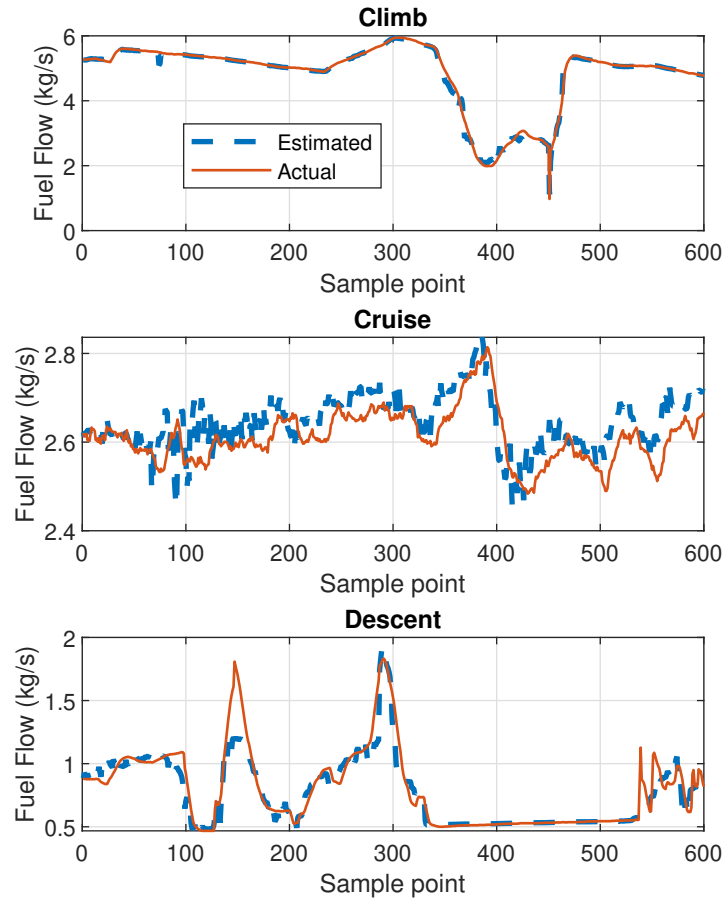


Figure 4.6 : Fuel flow estimation with estimated throttle positions: climb, descent and cruise.

capture both operational and parametric trends. Compared to BADA, both models are successful at estimating the acceleration factors. This is explicitly seen in cruise results. The mean absolute errors of all models are given in Table 9. As illustrated before, providing throttle position information from QAR to the neural network yields the best estimation result. However, the model with δ_T and the cascade model still have satisfactory results. In general, the mean absolute error is below $0.05[kg/s]$ for all modes. The statistical information of the percentage errors is given in Table 10. They represent normal distributions and their expected values are generally below %0.1 and standard deviations are not more than %2.5. Initial tests reflect the fact that even better accuracy can be achieved for all models as the data set size increases.

Figure 4.8 demonstrates correction factors of all models for climb, cruise and descent. The results show that in all modes, correction factor is generally close to 1. However, it reduces down to 0.4 and increases up to 3 for climb and descent, respectively. As

Table 4.7 : Mean absolute errors on the test sets, in $[kg/s]$.

	Black-box	δ_T model	Cascade Model
Climb	0.0195	0.0247	0.0389
Cruise	0.0143	0.0411	0.0420
Descent	0.0236	0.0250	0.0465

Table 4.8 : The noted figures are percentage errors (i.e. % error), in terms of mean and standard deviation with respect to actual fuel flow rates.

	Black-box	δ_T model	Cascade Model
Climb	0.149 - 0.938	0.0569 - 1.066	0.212 - 1.527
Cruise	0.0634 - 0.844	0.089 - 2.521	0.154 - 2.608
Descent	0.069 - 2.872	0.610 - 2.931	0.137 - 2.634

mentioned before, this is mostly due to the operational discrepancies between actual and BADA throttle setting. Correction factors around 0.8 stand for parametric errors. In cruise, the variance of correction factor is slight compared to other modes. However, it still reveals parametric error and oscillations due to acceleration factors.

Figures 4.9, 4.10, 4.11 and 4.12 show sensitivity of correction factor to Mach, aircraft mass, altitude and δ_T , respectively. These plots indicate that fuel flow estimations based on flight modes is a proper technique for digital-twin performance modeling. Mach, aircraft mass and altitude are key parameters in this aspect and neural networks of climb, cruise and descent yield different sensitivities to these variables. Even though both models are good at fuel flow estimation, their partial dependence on the inputs do not always follow the same trend. For instance, they have similar sensitivities to aircraft mass in climb and cruise, whereas it is not the case in altitude in climb and descent. This is also in-line with the fact that the models show different accuracy with respect to each of the flight modes.

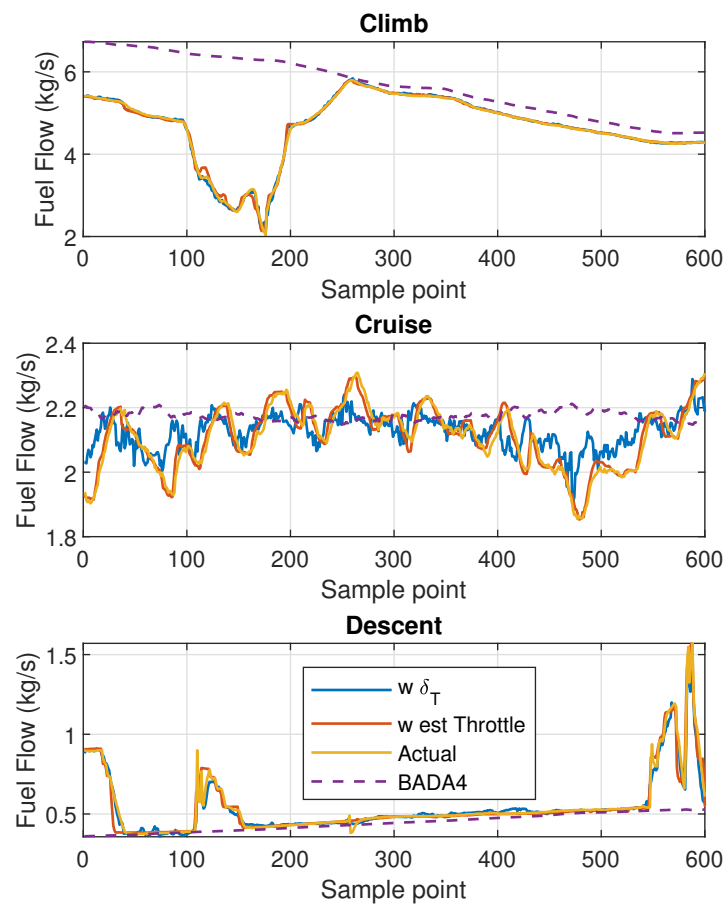


Figure 4.7 : Comparison of estimation from all models with BADA4 and actual fuel flow.

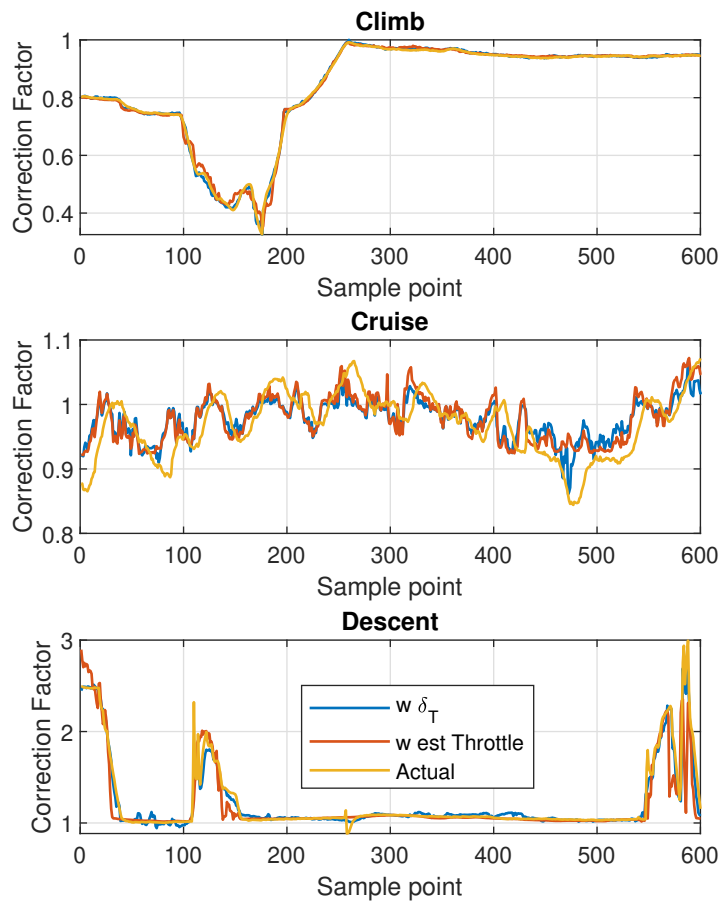


Figure 4.8 : Comparison of correction factors of all models.

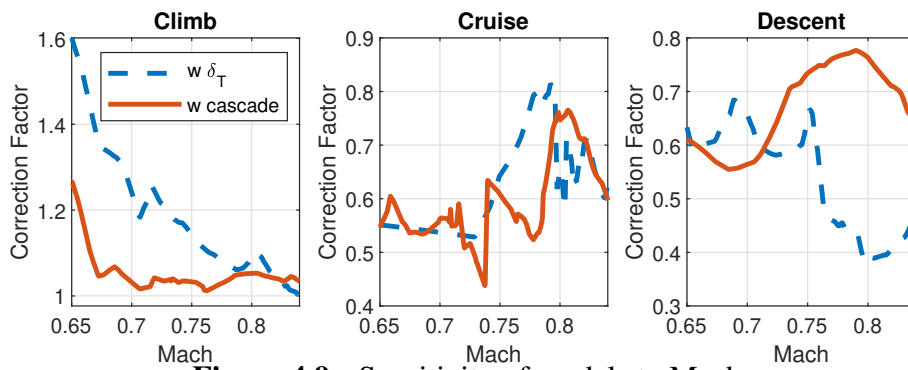


Figure 4.9 : Sensitivity of models to Mach.

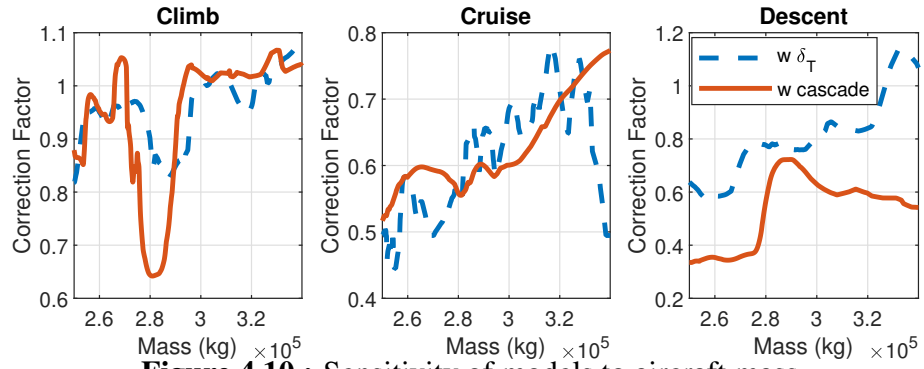


Figure 4.10 : Sensitivity of models to aircraft mass.

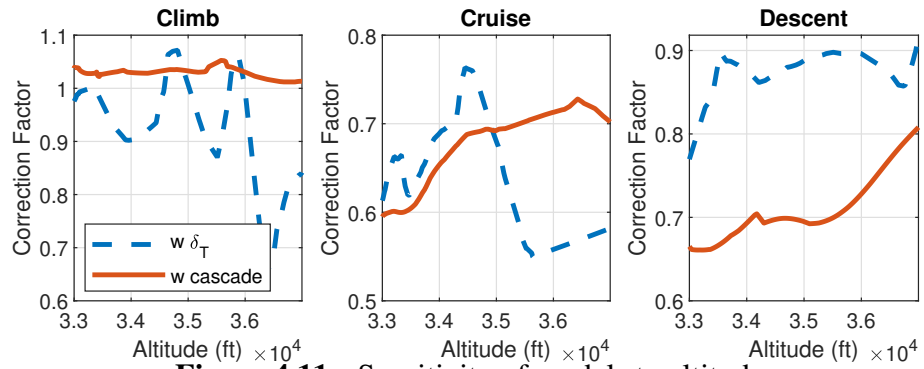


Figure 4.11 : Sensitivity of models to altitude.

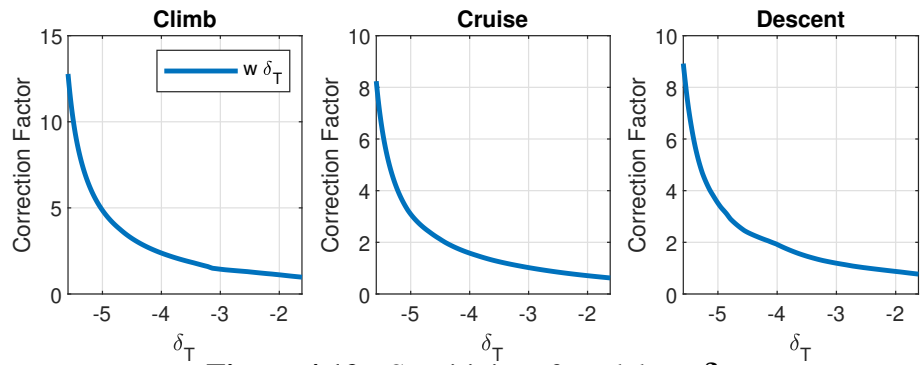


Figure 4.12 : Sensitivity of models to δ_T .



5. UPDATING BASELINE PERFORMANCE MODELS

In this section we provide a summary of the underlying mathematical model of the flight dynamics and the specifics associated with the BADA formulation. In addition, we introduce the correction factor formulation and provide insight on the resulting problem statement following this specific formulation. In comparison to the previous methodology, this part also trains a much simplified neural network in which we define a correction factor to BADA4 fuel flow baseline model. The idea behind learning the correction factor instead of learning the fuel flow model itself is that we capture the complex and nonlinear input-output relationship and factors under a unified aerodynamic coefficient like element which can be easily embedded as a coefficient to existing baseline models. In addition, the training results also indicate a considerable reduction in model variation via this approach.

5.1 Problem Definition

Let $\mathcal{B}$ be the baseline aircraft performance model, which is a set of functions:

$$\mathcal{B} = \{b_1, b_2, \dots, b_n\} \quad (5.1)$$

Each function $b_j : A_j \rightarrow B_j$ maps a set of parameters consisting aircraft and atmosphere states to the set of performance parameters, denoted by $\Upsilon^{\mathcal{B}} = \{v_1^{\mathcal{B}}, v_2^{\mathcal{B}}, \dots, v_k^{\mathcal{B}}\}$. The domain A_n is a vector space of the set of source variables s_j and $|s_j| = m_j$, $A_n \in \mathbb{R}^{m_j}$. Similarly, the target domain B_j is the vector space for the target performance parameters, v_j , where $|v_j| = n_j$ and $B_j \in \mathbb{R}^{n_j}$. Note that the source domains A_j may not be unique, i.e., it is possible that $s_j \cap s_k \neq \emptyset$. For instance, the pressure ratio and the Mach number are inputs to calculate both drag and thrust. In this case, let b_1 and b_2 be the functions of $\mathcal{B}$ that calculate the drag $v_1^{\mathcal{B}} = \{D\} \in \Upsilon$ and the thrust $v_2^{\mathcal{B}} = \{T\} \in \Upsilon$, respectively. Then the input sets are $s_1 = \{M, \delta, \dots\} \in \mathbb{R}^{m_1}$ and $s_1 \cap s_2 = \{M, \delta\}$. In summary, we can write $b_1 : \{\delta, M, \dots\} \rightarrow \{D\}$ and $b_2 : s_2 \rightarrow v_2$. The functions in $\mathcal{B}$ are represented by polynomials or look-up tables:

$$b_j = \sum_k a_k \phi(s_j) \quad (5.2)$$

$$b_j = f(s_j, \zeta) : A_j \rightarrow B_j \quad (5.3)$$

where $\zeta \in \mathbb{R}^{M \times N}$.

The operational flight data is denoted by $\mathcal{Q}$. Another common name is Quick Access Recorder (QAR) data, which logs the history of trajectories of flight parameters. Indeed, each trajectory belongs to a particular tail number. Hence, we can formally define the operational flight data as:

$$\mathcal{Q} = \{\mathbf{X}_k^{q_j} : j, k, q \in \mathbb{N}\} \quad (5.4)$$

Here, q_j represents an aircraft of type q with a unique tail number j . The instance of the flight is denoted by k . Note that the parameters available can differ with varying aircraft type q . Moreover, flight times would be different since the whole dataset has both short and long haul trajectories. Hence, the dimension of a particular QAR data is $\mathbf{X}_k^{q_j} \subset \mathbb{R}^{M_k^{q_j} \times N_k^{q_j}}$, where $M_k^{q_j}$ is the flight duration of the flight k of the aircraft q_j and $N_k^{q_j}$ is the number of the parameters logged in a QAR device of the aircraft q_j . The aircraft performance that are directly available in a QAR set is denoted by $\Upsilon^{\mathcal{Q}} = \{v_n^{\mathcal{Q}} : n \in \mathbb{N}\}$. Note that the list of performance parameters in $\mathcal{Q}$ are limited by the availability of sensors. Hence, all elements of $\Upsilon^{\mathcal{B}}$ may not be in $\Upsilon^{\mathcal{Q}}$, and vice versa. For example, even though $\mathcal{B}$ has functions to model drag or thrust, these forces are not available in $\mathcal{Q}$ since they are not observable. We specifically focus on the parameters that both exist in $\Upsilon^{\mathcal{B}}$ and $\Upsilon^{\mathcal{Q}}$. We define this set as:

$$\Upsilon_* = \{v_n^* : n \in \mathbb{N}\} = \Upsilon^{\mathcal{B}} \cap \Upsilon^{\mathcal{Q}} \quad (5.5)$$

The proposed system first quantifies the discrepancy between the aircraft performance recorded $\Upsilon_*^{\mathcal{Q}}$, and the one that the baseline model $\mathcal{B}$ utilizes the functions b_j to compute with the input parameters s_j acquired from the operational data, $\mathcal{Q}$.

$$\Upsilon_*^{\mathcal{B}} = \bigcup b_j(s_j^{\mathcal{Q}}, \zeta) \quad (5.6)$$

where $s_j^{\mathcal{Q}} \subset \mathbf{X}_k^{q_j}$. We can select a proper statistical method to quantify how similar the distributions of $\Upsilon_*^{\mathcal{B}}$ and $\Upsilon_*^{\mathcal{Q}}$ are. Based on some criteria, the proposed system would automatically update the baseline model $\mathcal{B}$. In this sense, we represent this update

with the term correction factor, where we aim to calculate how to correct Υ_*^B with a multiplier:

$$\tau_{q_j} = f\left(\frac{\Upsilon_*^{\mathcal{Q}}}{\Upsilon_*^{\mathcal{B}}}\right) \quad (5.7)$$

where τ_{q_j} is the correction factor tailored to a particular tail number q_j . First, the fuel flow from BADA equations is calculated by providing the same trajectory history of QAR:

$$F_{\mathcal{B}} = \mathcal{B}(s_j^{\mathcal{Q}}) \quad (5.8)$$

Thus, the fuel flow is formulated as

$$F_{\mathcal{B}} = b_n(M_{\mathcal{Q}}, m_{\mathcal{Q}}, \delta_{\mathcal{Q}}, \theta_{\mathcal{Q}}), \quad b_n \in \mathcal{B}, \quad F_{\mathcal{B}} \in \Upsilon^* \quad (5.9)$$

Following this line of formulation, we update $\mathcal{B}$ as

$$b_{n,*}(s_*) = \begin{cases} \tau_{q_j} \Upsilon_*^{\mathcal{B}} & g^s(\Upsilon_*^{\mathcal{B}}, \Upsilon_*^{\mathcal{Q}}) > \varepsilon \\ \Upsilon_*^{\mathcal{B}} & o.w. \end{cases} \quad (5.10)$$

where $b_{n,*} \in \mathcal{B}$ are the functions that compute the performance parameters $\Upsilon_*^{\mathcal{B}}$ and g^s is a statistical function that quantifies the discrepancy level between $\Upsilon_*^{\mathcal{B}}$ and $\Upsilon_*^{\mathcal{Q}}$.

$$\tau_{q_j} : \mathbb{R}^{\rho(\cdot)} \rightarrow \mathbb{R} \quad (5.11)$$

where $\rho(\cdot)$ assigns the input set for the model. Note that it is not always $\rho(\cdot) \neq s_j$, since there could be additional features from the data $\mathcal{Q}$, which provide new information on why the baseline performance $\Upsilon_*^{\mathcal{B}}$ differs from the real flight data $\Upsilon_*^{\mathcal{Q}}$.

To find a nonlinear mapping for the correction factor τ , we utilize deep neural networks, which have been proven to capture complex input-output relationships through gradient-based optimization and can be used to model any continuous function [44], [45].

Let x_j represent the feature set defined by $\rho_j(\cdot)$ and y_j be the target, the correction factor. Then, $y_j = \tau_{q_j}(x_j, \xi)$, where MLP model learns a configuration $\xi = \{w_l, b_l | l \in \mathbb{N}\}$ such that it yields the best approximation.

This supervised learning task is based on minimization of a predefined loss function, which is generally calculated as difference of the data label and the model output (prediction). By using this loss function, output error information used to update all

weight and bias terms by using backpropagation algorithm [65]. In our case, the loss function is selected as mean absolute value error:

$$\tau_{q_j} = g^* = \arg \min_g \mathbb{E}_{x,y \sim p_{data}} ||y - g(x)|| \quad (5.12)$$

Details on calculation of the nonlinear function of the correction factor τ_{q_j} is given further. The following section describes the data used for training.

5.2 Characterizing the Correction Factor

In this section, we characterize the correction factor across the different dynamic models associated with climb, cruise, and descent flight modes. Specifically, the baseline BADA 4 model for each flight mode is directly tied to the identification process performed on outputs of the performance modelling software of the aircraft manufacturers. More specifically, aircraft manufacturers utilize tools that integrate test flight data with generic techniques of performance modelling and flight segment-specific modules [43]. Based on this context, these tools generate performance databases that are comprised of configurations with different thrust settings, flap/landing gear combinations and flight regimes. The performance tables and the models tailored to the unique aircraft are then utilized to calculate the aircraft performance for given sets of flight conditions or reference trajectories. The reference data for BADA 4 is the output of this process ran for many flight cases [5], [6]. In that sense, BADA 4 is a pioneer at utilizing sophisticated techniques to build generic aircraft performance models and covers most of the aircraft types that are active in the current air transportation system. However, as the aircraft operate at different airlines and in various conditions, there is an inevitable shift from the generic baseline models to tail-number specific performances.

5.2.1 Types of discrepancies

As illustrated in Section 1, Figure 1.1 shows some random snapshots of 10 minutes of trajectories illustrating the comparisons between the fuel flow profiles of the model and the data for climb, cruise, and descent. We categorize the types of these discrepancies as the operational and the parametric ones. The operational errors are highly likely due to violations of the assumptions that the model does. However, even though all the assumptions are valid, there still might be some bias errors. We call these as the

parametric errors, since simple calibrations on the polynomial coefficients could be satisfactory to correct them.

Parametric errors in climb result in both profiles to have similar tendencies since the aircraft is possibly climbing with the thrust setting defined in the model. There is only a slight bias error, which can be corrected by calibrating the model coefficients. The magnitudes of these errors are usually less than 0.3 [kg/s]. On the other hand, the profiles on the operational error part of climb have different tendencies in addition to the bias error. The aircraft is assumed to climb at a specific thrust rating, namely maximum climb. A unique equation is defined to set the throttle parameter δ_T for this configuration:

$$\delta_T = \sum_j \sum_i b_{6(i-1)+j} M^{i-1} \delta^{j-1} \quad (5.13)$$

where $b_{(.)}$ are the model coefficients. Even though Equation 2.17 represents a generalized formula for calculating the thrust, utilizing Equation 5.13 to define the throttle parameter δ_T reduces the envelope to a sub-mode of the system. Indeed, this causes computed fuel flow to be different from actual. The effect of the trajectory of the engine variables might result in fuel flow differences up to 5.0 [kg/s], which is not a negligible magnitude. In summary, the error in fuel flow in climb mode is not only due to utilizing a specific thrust setting, but also a parametric bias in where the maximum climb is not violated.

The tendency in cruise is a bit different, since the autopilot mandates the aircraft to fly at a constant altitude for a long period of time. The governing equations for modelling cruise are:

$$T = D \quad (5.14)$$

$$L = W \quad (5.15)$$

where the primary assumption is assuming thrust is equal to drag and lift is equal to weight. Hence, there is no particular thrust setting under this assumption. It is not so apparent as in climb to distinguish the type of error due to the lack of variance of fuel flow in cruise. The fuel flow calculated by the model is like a linear curve. We can

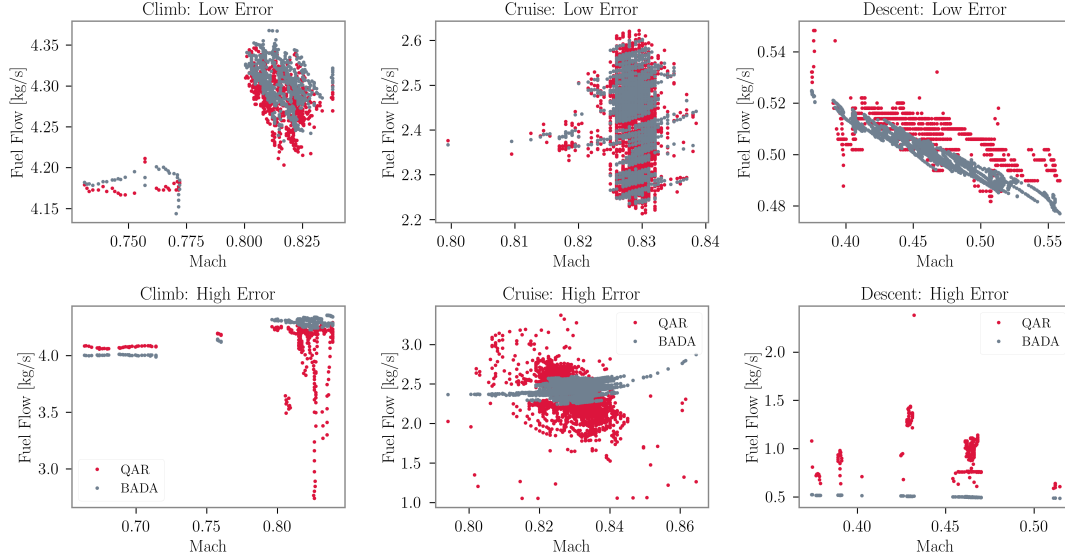


Figure 5.1 : Comparison between the model and the data in similar flight conditions.

observe parametric errors as slight shifts from the average line of the fuel flow from the data. In some cases, the output of the model is like the mean of the real flight data. These trajectories reveal small fluctuations around this mean line, which are highly likely to be due to the evolution of the engine states.

Similar to the climb phase, the model assumes a specific thrust setting for descent, as well. The thrust coefficient C_T is calculated as below:

$$C_T = \sum_j \sum_i t_{3(i-1)+j} \delta^{i-2} M^{j-1} \quad (5.16)$$

where $t_{(.)}$ are the polynomial coefficients from the model. Apart from these coefficients, the descent fuel flow is represented as a function of Mach number M , pressure ratio δ and temperature ratio θ . The behavior of the discrepancies are similar to climb; i.e., we observe slight shifts from the data when the assumption for the thrust setting is valid. The error rises up to 1,5 [kg/s] when the difference is not only due to the coefficients but also not including operational information.

5.2.2 Instance-based comparison

This section goes in detail to further understand the discrepancies between the model and the real flight data. To comprehend the physics behind it, we first consider the same flight and environmental conditions in the model and the data. When we look at the baseline modelling, the fuel flow for climb is a function of the Mach number M , the pressure ratio δ and the temperature ratio θ . In order to understand the dynamics

behind the correction factor, we select specific regimes of these values and compare fuel flows computed by BADA and recorded in the QAR data set. First, we extract the data points in QAR where the ISA conditions are met in a tolerance. ISA pressure and temperature ratios are calculated as $\delta_{ISA} = (1 - \alpha_1 h_{QAR})^{\beta_1}$ and $\theta_{ISA} = 1 - \alpha_2 \beta_2 h_{QAR}$, where $\alpha_1 = 2.256 \times 10^{-5}$, $\beta_1 = 5.256$, $\alpha_2 = 0.0065$ and $\beta_2 = 0.0035$. Note that h_{QAR} is the altitude history in the QAR with the unit of $[m]$. Air pressure is directly recorded in the QAR data of the aircraft of interest and denoted as p_{QAR} . As for the temperature, the outside air temperature is available in the QAR and denoted as T_{QAR} . The corresponding pressure and temperature ratios are δ_{QAR} and θ_{QAR} . The points in the data satisfying the following conditions are considered to be at ISA conditions:

$$\mathbb{X}_{ISA} = \left\{ x^{(i)} \left| \|\sigma_{ISA} - \sigma_{QAR}\| < \varepsilon, \|\theta_{ISA} - \theta_{QAR}\| < \varepsilon, i \in \mathbb{N} \right. \right\} \quad (5.17)$$

where $\varepsilon = 0.01$ is the tolerance value. After extracting $\mathbb{X}_{ISA}$, we select reference values for the pressure ratio δ_{ref} and the temperature ratio θ_{ref} such that we have the largest amount of data points at these ISA conditions. A filtering similar to the one in Equation 5.17 is performed with the tolerance value of $\varepsilon = 0.05$. So far, in terms of the baseline model, we have the data at ISA conditions and at specific temperature and pressure ratios. The idea is to observe what happens after this point. We specifically observe the distribution of the fuel flow over the Mach number, to see how the data reflects the modelled dynamics. Indeed, aircraft mass is also the remaining parameter, but it would be easy to see the clusters of low and high mass values.

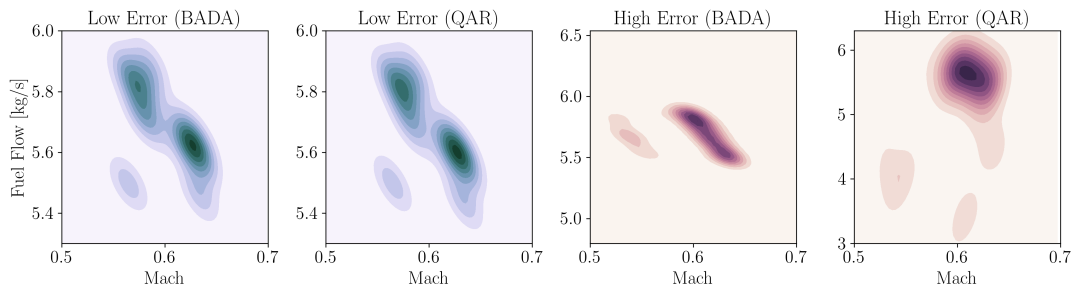


Figure 5.2 : Kernel-density plots of the fuel flow profiles in BADA and QAR for given approximately the same climb conditions.

In climb, the reference values of the environmental states are selected as $\theta_{ref} = 0.9$ and $\delta_{ref} = 0.6$. Figure 5.2 illustrates the distribution of fuel flow with respect to Mach number. The low error points have fuel flow discrepancies less than $0.02[kg/s]$,

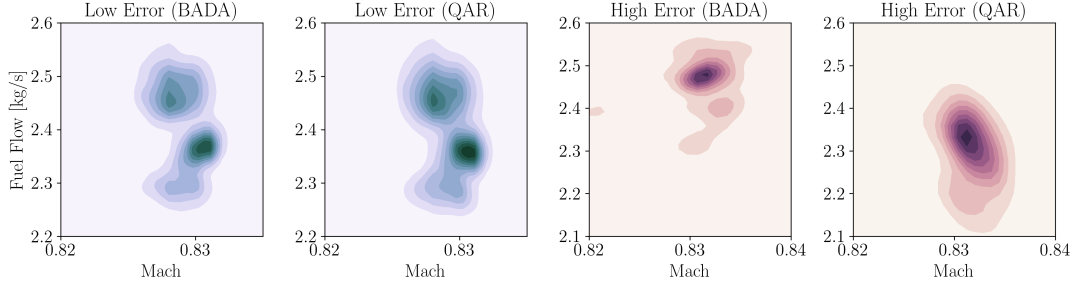


Figure 5.3 : Kernel-density plots of the fuel flow profiles in BADA and QAR for given approximately the same cruise conditions.

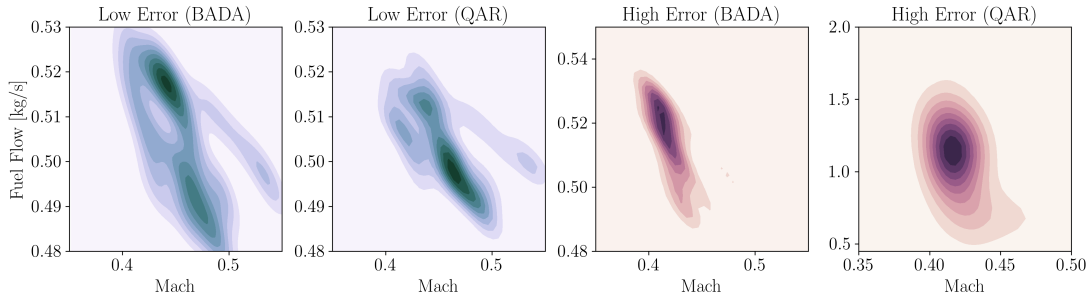


Figure 5.4 : Kernel-density plots of the fuel flow profiles in BADA and QAR for given approximately the same descent conditions.

whereas this value is higher than $0.05[kg/s]$ for the high error points. The idea behind analyzing the distribution of fuel flow with respect to Mach is to understand whether the data shows the same tendency as the model does with polynomials. It is expected to see similar distributions in the low error case. However, they are at different regions at the high error case. The model limits the fuel flow between 5.5 and $6.0[kg/s]$, whereas the data shows three separate kernels ranging from $3.0 - 6.0[kg/s]$ for the same flight conditions. The climb part in Figure 5.1 also shows how the points with low and high errors are distributed over Mach. We can say that after filtering the data to a particular flight condition, some points in the data actually overlaps with the dynamics described in the model. There are two clusters as well representing the low and high gross weights. The points lying between the range $[4.15, 4.22] [kg/s]$ have the mean value of $230,000$ kg as the aircraft mass. The mean value of the mass of the cluster on the top-right is $310,000$ kg. In summary, if all the assumptions in the model are valid, the data reveals increasing fuel flow with increasing Mach number and mass.

A similar data filtering to the one in climb is applied to cruise to see how fuel flow is distributed over Mach number for low and high error cases, with $\delta_{ref} = 0.275$ and

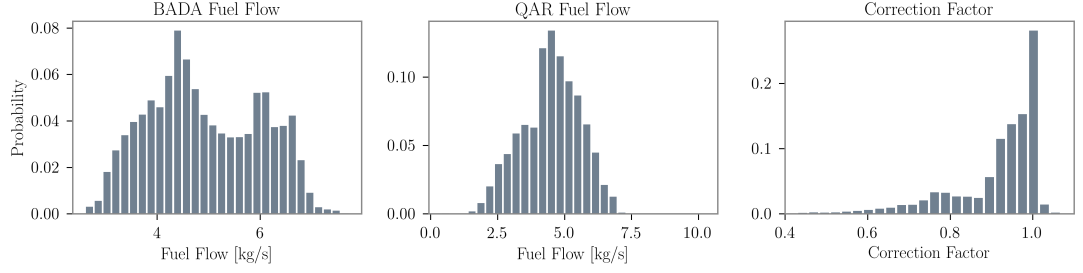


Figure 5.5 : Histogram plots of fuel flow calculated by BADA, fuel flow recorder in QAR and the correction factor for climb.

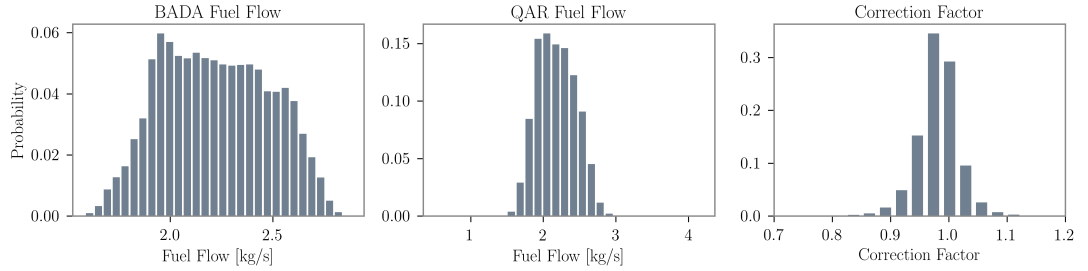


Figure 5.6 : Histogram plots of fuel flow calculated by BADA, fuel flow recorder in QAR and the correction factor for cruise.

$\theta_{ref} = 0.78$. In Figure 5.3, we can observe that the data has a subset that reflects the same tendency in the model for some flight conditions. However, even though it is the same flight conditions, some points show differences from the model, in terms of fuel flow. A possible cause for this discrepancy could be the impact of other parameters that the model does not fully cover. We can see the instance-based plots for cruise in Figure 5.1, representing the ranges of the fuel flow within the model and the data. When the model error is low, we can see several curves, which are parts of a convex function representing the behavior of the fuel flow with increasing Mach. It is like a vertical stack of several curves, where each of them represents different aircraft mass. These curves go up with increasing mass, which ranges from 276,000 kg to 320,000 kg in this plot. The high error case for cruise reveals a high variance on the actual fuel flow compared to the modeled one. The model computes fuel flow around 2.5 [kg/s]. However, the data shows fuel flow profiles bounded between 1.0 [kg/s] and 3.0 [kg/s]. The same analysis on descent is given in Figure 5.4. Similar to the other flight modes, we can see high variances of fuel flow in the data compared to the output of the model. Since the model assumes a low-idle thrust setting, the resulting fuel flow is around 0.51 [kg/s]. However, it goes up to more than 1.5 [kg/s] in the flight. We can see the

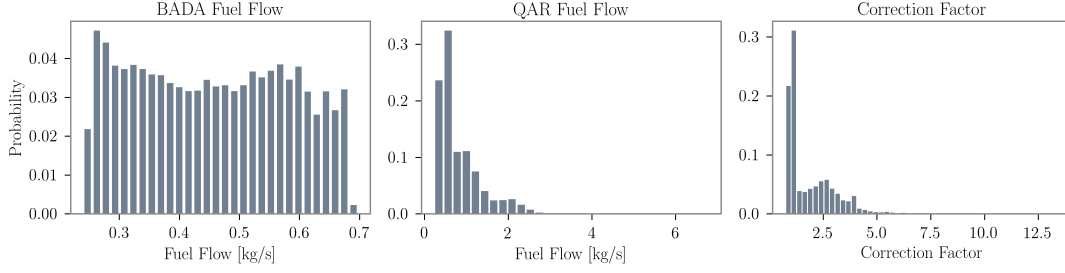


Figure 5.7 : Histogram plots of fuel flow calculated by BADA, fuel flow recorder in QAR and the correction factor for descent.

corresponding trajectory points of descent in Figure 5.1. The low error case reveals the similar tendencies of the fuel flow with increasing Mach. However, the figure with high error shows varying fuel flow magnitudes even for the same Mach numbers. Indeed, such a difference could be due to aircraft mass, but since these are long-haul flights, they usually have similar landing weights. Hence, having a fuel flow error up to 1.0 [kg/s] for $M = 0.45$ is possibly due to engine variables.

5.2.3 Distributions of the correction factor

The distribution of the correction factor in Figure 5.5 implies that there exist two kernels; one with an expected value of 1 is possibly the points with parametric error, and the second one ranges from 0.6 to 0.9 and represents the operational error. Figure 5.6 demonstrates the distributions of the modeled fuel flow, the actual flow and the correction factor for the cruise phase. The correction factor has kind of normal distribution, with a mean value of slightly lower than 1. This is expected because the correction factor consists of this slight bias of fuel profile of the model from the actual, and fluctuations are possibly due to engine dynamics. Figure 5.7 demonstrates the distributions of the modeled fuel flow, the actual flow and the correction factor for descent phase. The correction factor has a normal distribution around one, but kind of uniform distribution up to five. This implies small bias errors when the assumed thrust settings is valid, and several different thrust settings exist uniformly in the data. Considering the all descent points that the data covers, the model calculates a fuel flow profile in the range $[0.21, 0.75]$ [kg/s]. On the other hand, the measurements from the data show fuel flow up to 3.0 [kg/s].

5.3 Machine Learning for Correction Factor Estimation

This section provides the details of the deep neural networks used in the study. Each flight mode has its distinct feature engineering, and thus, the corresponding architectures tend to be different. As such, since each flight mode has a unique distribution of data, feature importance, tendency of loss functions, and hyperparameters are also different. For instance, even though the Mach number is physically significant for all modes, its lack of variance in the cruise mode may cause the machine learning algorithm to not capture its actual effect on aircraft performance. Hence, we applied small adjustments to each model to obtain the best configuration for a good regression. The previous section described how the a baseline model describes aircraft performance in terms of drag, thrust, and fuel consumption. In summary, pressure ratio, temperature ratio, aircraft mass, and Mach number are the main parameters utilized by such models to perform performance calculations. Here we further investigate additional parameters from the flight data to utilize them as features. In particular, we use statistical analysis and tree-based machine learning algorithms to extract feature importance. With this approach, we aim to identify not only the essential parameters other than the baseline parameters but also how to utilize them in the machine learning models.

5.3.1 Feature extraction

5.3.1.1 Statistical analysis on fuel flow

We provide a correlation analysis to show the effect of the aircraft or environmental states on fuel flow. The process utilizes Spearman correlation as the statistical method. We select Spearman correlation because it assesses the monotonic relationship between two variables. We did not choose Pearson because it only evaluates the linear relation. Figures 11, 12, and 13 reveal the Spearman rankings of the considered parameters for fuel flow. Only the ones that have a ranking with an absolute value higher than 0.3 are selected. The rankings are listed in Tables 2, 3, and 4 for climb, cruise and descent modes respectively.

To observe the correlation of $\mathcal{B}$ related parameters on the actual fuel flow, drag, thrust, thrust coefficient C_T and throttle parameter δ_T are included. Thrust is calculated by a hybrid approach where the drag from Eq. 2.10 is combined with the true airspeed and aircraft mass from $\mathcal{Q}$ as

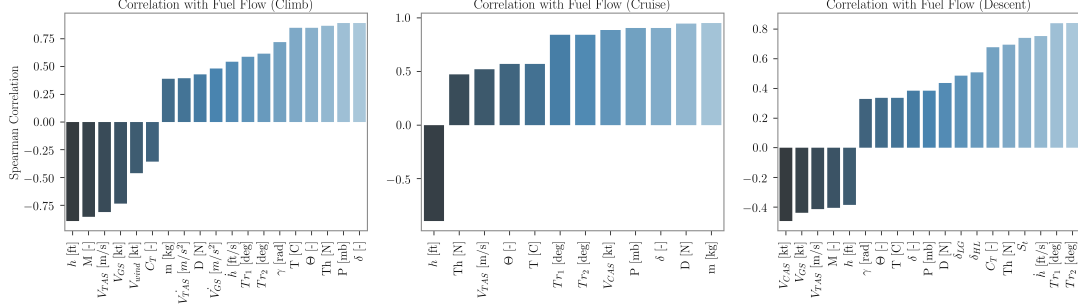


Figure 5.8 : Spearman rankings of the parameters for fuel flow correlation.

$$T = m_{\mathcal{Q}} (\dot{V}_{TAS_{\mathcal{Q}}} + g \sin \gamma_{\mathcal{Q}}) + D_{\mathcal{B}} \quad (5.18)$$

Then, the thrust coefficient C_T and the throttle parameter δ_T are calculated using (5.13) and (2.15).

Altitude or pressure-related parameters have a significant effect on fuel flow in the climb and cruise modes. Gross weight is the most influential parameter in cruise. It is expected because of two reasons: First, the data analytic conclusion is that the cruise trajectory has considerably more data, and mass have a significant variability concerning other parameters because altitude and speed-related parameters are almost constant. Second, the physical explanation is that as the aircraft cruises, it burns less fuel, and this is well reflected on the data. However, gross weight is not in the parameter list of descent. This does not imply that it is not related, but the operational evolution of the parameters such as flaps, vertical speed and flight path angle may suppress its effect. Naturally, the next essential variables for all flight modes are engine specific ones, namely throttle positions. Further, some studies reveal the effect of N1 and N2 values of the engines. However, they are not available in the data of this study. Apart from the baseline and engine specific features, we also observe some correlation of calibrated airspeed, ground speed, and wind speed with fuel flow. Even though this relation does not have any physical argument, it too is not meaningless. It is a consequence of how climb and descent trajectories evolve. An aircraft would operationally increase or decrease its ground speed as it climbs or descends to higher or lower altitudes. Moreover, the ground speed is a transformed version of the Mach number corrected with the wind. Hence, if the Mach number has any effect on the fuel flow, the ground speed is going to have a similar effect. The critical point here is we may not rely on such variables and consider them as informative features.

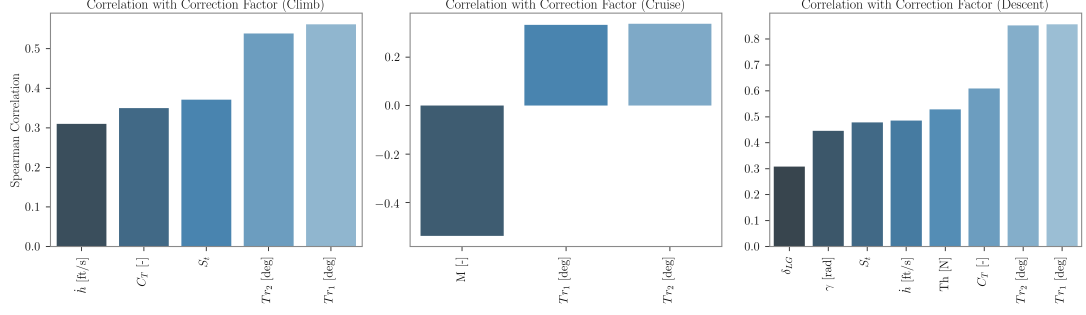


Figure 5.9 : Spearman rankings of the parameters for correction factor correlation.

5.3.1.2 Statistical analysis on the correction factor

We replicate the same statistical analysis on the correction factor. The aim is to see if the discrepancy is caused by parameters that the baseline model does not estimate correctly. Similarly, we use Spearman correlation and investigate variables that have an absolute value of ranking higher than 0.3. The most significant parameters for all modes are throttle positions. In the climb and descent modes, we also observe some importance of vertical speed and artificial thrust parameters. In the cruise mode, the only parameter other than engine variables is Mach number. Since these variables are indeed considered by the model, we can say the discrepancy is caused by parametric bias. On the other hand, the major effect of the throttle positions on the correction factor tells us about operational errors.

5.3.1.3 Machine learning based feature extraction and selection

Even though the distributions of the variables in the data and their statistical relations with the correction factor provide a good intuition on deciding which features to select, it may not be wise to use all correlated variables as inputs. This is because we are approximating a nonlinear function through tensor transformations, and if the features we select do not aggregate new information, they may act like noise on the model and cause overfitting. Hence, we applied a machine learning based feature selection process [66]. We utilized Random Forest models to extract feature importance. We train Random Forest models with 100 estimators and a max depth value of 5 and find the best features that represent the relation to the correction factor. Table 5.1 summarizes the important features extracted by the tree-based machine learning algorithm.

Table 5.1 : Features extracted by the Random Forest based methodologies.

Climb	Cruise	Descent
Latitude [deg]	Pressure ratio [-]	Longitude [deg]
Altitude [ft]	Temperature ratio [-]	Altitude [ft]
Ground speed [kt]	Ground speed [kt]	CAS [kt]
Mach [-]	Mach [-]	Mach [-]
Mass [-]	Mass [-]	Mass [-]
Throttle [deg]	Throttle [deg]	Throttle [deg]
Flap [deg]	True airspeed [m/s]	True airspeed [m/s]
Drag [N]	Drag [N]	Ground acceleration [kt/s]
Pressure ratio [-]		Pressure ratio [-]
Temperature ratio		Temperature ratio
ISA deviation [C]		Flight path angle [rad]
Vertical speed [ft/s]		Vertical speed [ft/s]

The tree-based feature extraction methodology eliminates some features and adds new ones. Most parameters are consistent with those obtained from the statistical analysis. Each mode has the baseline features and engine variables. However, the algorithm did not select any artificial-thrust related features. As mentioned before, true airspeed and ground speed are no surprises. For the climb and descent modes, latitude and longitude are also among the parameters. However, this is highly expected because the dataset has several routes. In our first experiments, we trained straightforward MLPs to estimate the correction factor using features described in Table 5.1. The models showed good regression results; however, the learning curves were not satisfactory. Adding all variables acted like noise for the models, and they all resulted in overfitting. As a result, when we analyze all feature engineering provided by statistical analysis, machine learning-based feature extraction and preliminary experiments, the common essential features are pressure ratio, temperature ratio, gross weight, Mach number and the throttle positions.

5.3.2 Proposed custom MLP architecture for estimation of the correction factor

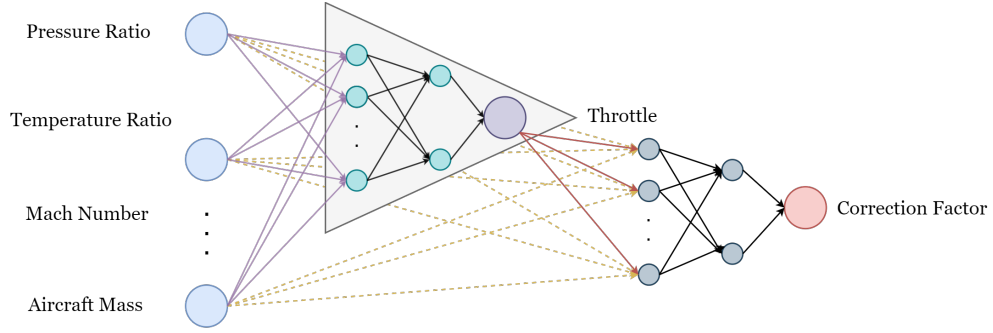


Figure 5.10 : Proposed neural network architecture.

This section describes the proposed MLP architecture for estimating the correction factor. The custom structure first approximates a mapping from the input features to the engine variable, which is the throttle position for this study. Then, this new feature is added to the main feature set to generate a function for the correction factor through the second part of the network. In other words, the model attempts to improve the correction factor by estimating proper throttle positions.

We selected standard normalization to scale the data. Let $\mathbb{X}_{(\cdot),trn}$ and $X_{(\cdot)}^{sc}$ be the training and scaled data, respectively. Let l represent the l^{th} layer of the network. The output of the layer l is

$$z^{[l](i)} = g \left(W^{[l]} x_{(\cdot),trn}^{(i),sc} + b^{[l]} \right)$$

where $x_{(\cdot),trn}^{(i),sc}$ is the i^{th} training sample of $\mathbb{X}_{(\cdot),trn}^{sc}$, the weight matrix is $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, and the bias vector is $b^{[l]} \in \mathbb{R}^{n_l}$; further, g is the activation function and n_l is the number of neurons in the l^{th} layer. Regardless of the models unique to the flight modes, the following are common. We used RELU as the activation function, except the output layer. All output layers have a linear activation function. He initialization is selected to assign the initial states of $W^{[l]}$ and $b^{[l]}$

$$W_0^{[l]} \sim \mathcal{N} \left(0, \sqrt{\frac{2}{n_l}} \right) \quad (5.19)$$

We use the mini-batch algorithm to improve the learning process. The algorithm processes $\mathbb{X}_{(\cdot),trn}^{sc}$ at each epoch by dividing it to n batches as

$$z^{[l](k)} = g \left(W^{[l]} x_{(\cdot),trn}^{(k),sc} + b^{[l]} \right) \quad (5.20)$$

where $k = \left\{ \frac{m}{n}(t-1) + 1, \frac{m}{n}(t-1) + 2, \dots, \frac{m}{n}t \right\}$ and $t = 1024$ for the cruise and $t = 256$ for the climb and descent models. To prevent overfitting, we apply dropout with value

of 0.5 and L_2 regularization, as well. As for the loss function, all models are trained on the mean absolute error. The loss function for the first part of the network is defined as

$$J_1(W, b) = \frac{1}{m} \sum_{i=1}^m |y_{Tr}^{(i)} - \hat{y}_{Tr}^{(i)}| + \frac{1}{2} \lambda W_1^2 \quad (5.21)$$

where $y_{Tr}^{(i)}$ is the actual model of the throttle as observed on the flight data, and $\hat{y}_{Tr}^{(i)}$ is the predicted throttle. The terms $J(\cdot)$ and $W(\cdot)$ denote the part of the entire network. The inputs to the first part of the network are pressure ratio, temperature ratio, gross weight and Mach number

$$u_1 = \{\delta, \theta, m, M\} \quad (5.22)$$

The second part of the network has the loss function

$$J_2(W, b) = \frac{1}{m} \sum_{i=1}^m |y_{\tau}^{(i)} - \hat{y}_{\tau}^{(i)}| + \frac{1}{2} \lambda W_2^2 \quad (5.23)$$

where y_{τ} is the correction factor. The input set of the second part of the network is

$$u_2 = u_1 \cup \hat{y}_{Tr} = \{\delta, \theta, m, M, \hat{y}_{Tr}\} \quad (5.24)$$

where u_1 is concatenated with the estimated throttle.

The learning process utilizes the Yogi optimization algorithm [67] to update weight matrices and bias vectors:

$$\zeta_{k+1} = \zeta_k + \text{Yogi} \left(\frac{\partial J_{1,2} \left(x_{(\cdot),trn}^{(k),sc} \right)}{\partial \zeta}, \alpha \right) \quad (5.25)$$

where $\zeta = \{W^{[1]}, \dots, W^{[L]}, b^{[1]}, \dots, b^{[L]}\}$. We tuned the learning rate α by checking the status of the loss function. If the loss function is not improved 10 times consecutively, the learning rate α is reduced by %90. Its initial value is chosen as 0.001. All models are trained for 2000 epochs. The size of the training, validation, and test datasets are illustrated in Table 3. All deep neural network models are implemented and trained with Tensorflow framework [68].

5.4 Experiment

The experiments in this thesis consist of training neural network models for the climb, cruise and descent modes for two datasets: The first dataset has trajectories of a specific tail-number aircraft; and the second has flights of multiple tail-numbers. As

for the benchmark, the proposed approach is compared with the baseline model, and a generic MLP model is trained with the same features plus throttle information. Such an approach is used for assessing what the gain is with the proposed approach, and how close it is to providing the throttle information directly in addition to the selected features. Table 5.2 summarizes the parameters used as inputs to the neural network models. The output is the correction factor for all of them.

Table 5.2 : Summary of the input sets of the models.

Model with engine variables	Proposed approach
Pressure ratio σ	Pressure ratio σ
Temperature ratio θ	Temperature ratio θ
Aircraft mass m	Aircraft mass m
Mach M	Mach M
Throttle positions $Tr_{1,2}$	In-estimated throttle $\hat{T}r_{1,2}$

5.4.1 Single tail number

In this experiment, we select the flight trajectories of a unique tail number aircraft. The data is composed of 98 flights. The trajectories are propagated over a four-month period, with the earliest flight on August 28, 2017 and the latest flight on December 28, 2017. In total, there are 2,865,177 points; of these, 2,571,910 are cruise, 124,329 are climb, and 168,938 are descent records. We spare %15 of the dataset as the test sets.

Table 5.3 : Size of the training and testing sets for the single tail number dataset.

	Training	Test
Climb	105,679	18,650
Cruise	2,186,123	385,787
Descent	143,597	25,341

The first model with the throttle information is composed of five hidden layers with neuron amounts of 1024, 512, 256, 256, and 128. The proposed model has two parts: the first, where the throttle is estimated has four layers with 1024, 512, 256 and 128 neurons. The second part, in which we estimate the correction factor with an input set

reinforced with the in-estimated throttle, has five layers with 1024, 512, 256, 128 and 128 neurons. All models have the same hyperparameters to demonstrate the impact of different features and architectures.

We first evaluate the models on the test points of the dataset defined in Table 5.3. Then, we evaluate on a different dataset, which is composed of various tail numbers that are not introduced to the models. This dataset has 98 flights as well, and it includes approximately the same amount of trajectory points as the datasets used for training. The idea behind such an evaluation is to determine the generalization capability of the models on different aircraft. Indeed, the estimation results would be worse than testing on the same tail number; however, we want to identify if there is any improvement with respect to the baseline model. In the evaluation process, we provide several metrics: mean absolute error on the correction factor, mean absolute error on the fuel flow, and mean absolute percentage error on the fuel flow.

Table 5.4 : Estimation results for single tail-number case.

	Model w throttle			Proposed approach			Baseline model		
	Climb	Cruise	Descent	Climb	Cruise	Descent	Climb	Cruise	Descent
MAE (Correction Factor)	0.0027	0.0043	0.0359	0.0061	0.0138	0.0732	-	-	-
MAE (Fuel flow [kg/h])	46.07	35.08	65.25	98.15	107.22	135.23	1580.47	248.85	1311.43
MSE (Fuel flow [kg/s ²])	0.0021	0.0002	0.0025	0.01396	0.0022	0.0127	0.5456	0.0091	0.3655
MAPE % (Fuel flow)	0.3412	0.443	1.741	0.9956	1.421	3.172	12.684	3.222	29.621
MAE (Fuel flow [kg/h]) on unfamiliar aircraft	344.72	119.04	233.30	1249.55	339.76	742.43	1545.68	252.28	1522.81
MAPE % (Fuel flow) on unfamiliar aircraft	2.2335	1.488	8.003	3.632	4.212	26.321	11.982	3.191	32.386

Table 5.5 : Estimation results for multiple tail-number case.

	Model w throttle			Proposed approach			Baseline model		
	Climb	Cruise	Descent	Climb	Cruise	Descent	Climb	Cruise	Descent
MAE (Correction factor)	0.0033	0.0056	0.0335	0.0065	0.0168	0.0697	-	-	-
MAE (Fuel flow [kg/h])	56.12	45.97	65.33	106.71	135.66	131.78	1547.95	252.59	1521.74
MSE (Fuel flow [kg/s ²])	0.0034	0.0013	0.0045	0.013	0.0039	0.0130	0.501	0.0038	0.475
MAPE % (Fuel flow)	0.392	0.657	1.257	0.951	1.732	2.965	12.071	3.197	32.335
MAE (Fuel flow [kg/h]) on unfamiliar aircraft	387.83	118.75	216.78	426.66	353.09	664.24	1569.57	252.28	1277.01
MAPE % (Fuel flow) on unfamiliar aircraft	2.356	1.467	8.175	3.259	4.218	27.245	12.393	3.191	27.361

Table 5.4 summarizes the results of the three models for the single tail-number case. The first model with the base features plus throttle results in the mean absolute error on the fuel flow as 46.07 [kg/h], which is 1580.47 [kg/h] in the baseline performance

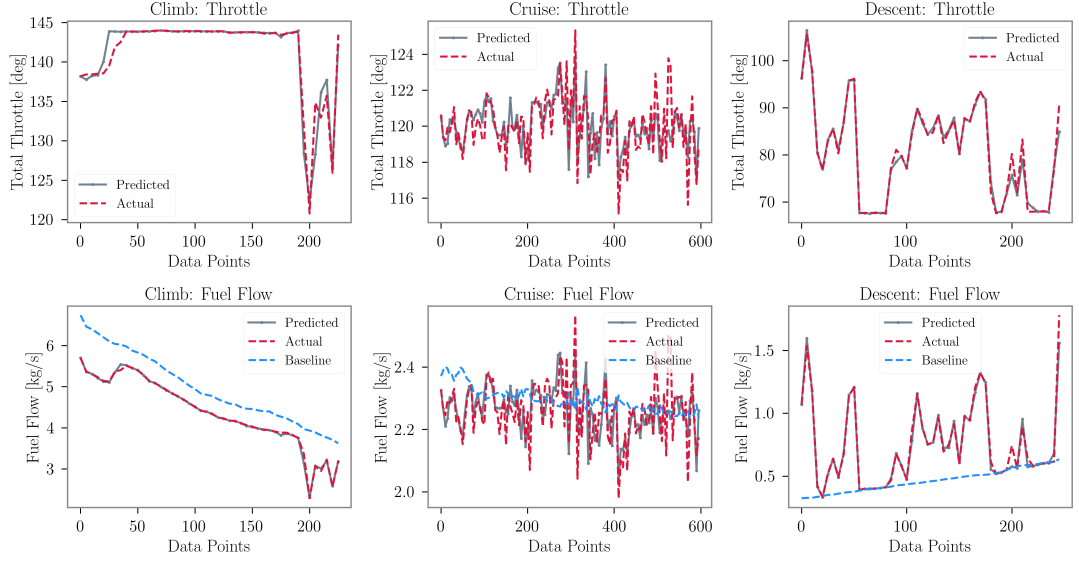


Figure 5.11 : Some snapshots from the test set illustrating the estimation results of the proposed model.

model. It is no surprise, because the issue with the climb mode was the operational error on modelling the thrust setting, whereas here, we directly gave this information to the model. In our proposed model, the estimation error is $98.15 [kg/h]$. It shows the success of the architecture by converging it to the first model. Even though the estimation errors are higher for using tail-number specific aircraft models on other aircraft, they are considerably better than the generic error of the baseline model. Thus, the proposed models not only capture systematic offsets present within the baseline models, but also the operational variations within each flight mode.

In the cruise phase, we observe a similar trend to the climb phase, where providing the throttle information has the best result. The improvement with the proposed architecture is approximately double compared to the baseline model. For an unfamiliar aircraft, the metrics are worse than both the first model and the baseline one. Nevertheless, our proposed model not only extracts the auto-throttle functionality of the given aircraft but also reduces the fuel flow error from $248.85 [kg/h]$ to $107.22 [kg/h]$.

The reduction in the error over the baseline model is considerably large in the descent, where it is reduced from $1311.43 [kg/h]$ to $135.0 [kg/h]$. The proposed method again has better results than the baseline model on unfamiliar aircraft. However, the mean absolute error rises from $\%3.172$ to $\%26.321$ compared to the single aircraft case. Different from the climb and cruise cases, the baseline model underestimates the total

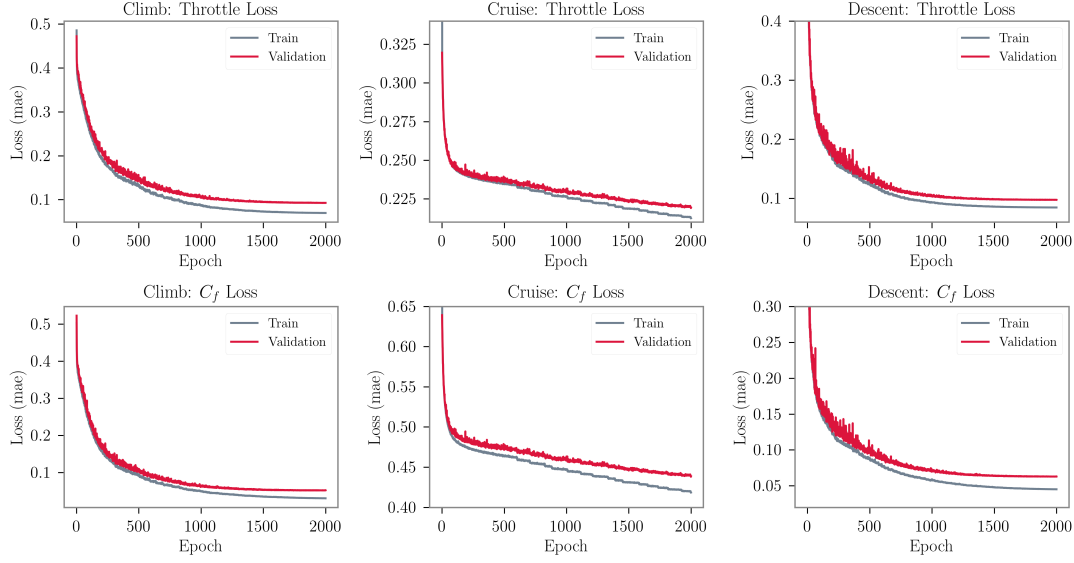


Figure 5.12 : Evolutions of the loss functions of the intermediate throttle and the correction factor.

fuel flow for the descent. It is expected because the baseline model assumes a low idle thrust setting. However, the data shows that the aircraft increase throttle especially during the approach phase.

Figure 5.12 demonstrates the evolution of the loss functions with the epoch. Climb and descent models show very good curves of learning, where the validation curve is slightly over the training one. They stand for good generalization capability and there is no overfit or underfit. The case in the cruise is a bit different. Both curves did not completely settle to their final values. To prevent overfitting, we did not proceed after 2,000 epochs. Figure 5.11 depicts some examples of the regressions calculated by the proposed model. They all show good capabilities at capturing tendencies in both throttle and fuel flow profiles. Each figure belongs to a part of a single flight. We can see that the proposed model can predict the throttle and improve fuel flow prediction.

The experiments on specific tail-numbers show the importance of tailoring performance models to the individual aircraft and the impact of dataset shifting. When models are tested on unfamiliar datasets, the performances get worse. This is highly likely to be the cumulative sum of the error in both the throttle and the correction factor. In that sense, we can say that the baseline model has better capability at generalizing aircraft performance, as it calculates similar values for both single and multiple aircraft cases. Still, the proposed method shows improvement for all cases.

5.4.2 Multiple tail numbers

This section provides the estimation results for the second dataset, including multiple tail numbers. The data size is approximately the same as the single tail number case. We aim to quantify the effect of using multiple aircraft with different performances. Further, we test the models trained with shuffled tail numbers on a dataset with a single tail number not included in the training.

The tendencies in the multiple tail number case is similar to the single tail one. The error in fuel flow is reduced from a mean absolute value of 1545.97 [kg/h] to 106.71 [kg/h]. As for the generalization capabilities, the baseline model is consistent with a similar amount of mean absolute errors in the training and testing sets of both single tail and multiple tail cases. In the test set of the unfamiliar aircraft, the mean absolute percentage error of the proposed model is %3.259, where it was %3.652 for the single tail number case. It shows us that using a dataset with shuffled tail numbers would result in an increased capability of generalization, with a penalty of a slight increase in the mean absolute error.

We obtain similar conclusions for the cruise and descent phases as well. The results show a significant amount of reduction in the error over the baseline model. For the descent model, the mean absolute error is reduced from 1521.74 kg/h to 131.78 kg/h. The cruise model shows an approximately %50 improvement. The generalization capability is similar to the single tail number case, for the cruise model. It is improved slightly in the descent model, where the mean absolute error is reduced from 742.43 kg/h to 664.24 kg/h.



6. PHYSICS-GUIDED MACHINE LEARNING FOR AIRCRAFT PERFORMANCE MODELING

Physical modelling, such as aircraft fuel consumption modeling, is used extensively to design and to predict processes in a wide range of engineering applications including flight planning [56, 59]. Although the underlying flight dynamic models are based on physical principles and physical laws, these are approximations of actual processes with added errors and biases based on a series of underlying assumptions and simplifications [9, 10, 69]. In addition, these models also contain a number of parameters, the values of which have to be calculated with scarce observed data, further decreasing their accuracy, largely due to the variability of the underlying physical-rules in both space and time [70].

Machine Learning (ML) algorithms have been shown to produce models that capture actual physical processes in a wide range of engineering disciplines including aircraft performance modeling [11, 71]. In that aspect, the most critical aspect in such flight performance modeling is to extract the actual fuel usage based on the flying conditions such as Mach number, altitude number and environmental conditions including disturbances such as wind [72]. As such, ML algorithms, are shown to be able to automatically extract complicated relationships from data [73]. A significant deduction for this conclusion is that ML models, given sufficient data, can find formation and patterns in data where the underlying complexity prevents the precise physics-based modeling of a system's actual process characteristics. However, the validity of such ML driven models across across the whole operational state-space (in this case the whole flight envelope) is complicated by a few critical factors [74]. Specifically, even though state-of-the-art ML models can seize entangled spatial-temporal correlations and relations, they require a vast amount of labeled data for training and testing, seldom available in real applications. [75] In addition to this, ML algorithms and methods often give scientifically discrepant results. They can only effectively capture relations in the possible training data and hence have a poor out-of-sample generalization capability

[76–78]. This is indeed a fact which is also valid for the fuel burn modeling of an aircraft [79].

In this chapter, we design a novel physics guided machine learning process for such data-driven aircraft fuel consumption modeling. Specifically, we guide and design the underlying neural networks with the actual physic laws that govern the fuel consumption dynamics. Even though conventional supervised learning algorithms perform well on this problem, we show that their applicability is limited by the actual flight envelope of the underlying training data. With this approach, we show that we improve consistency in unseen flight regimes, thus extending the validity of the machine learning model to the whole flight envelope. To the best of our knowledge, this is the first work that successfully designs and demonstrates a physics-guided deep learning framework in fuel consumption modeling for an aircraft.

In order to obtain more accurate results and reliable out-of-sample generalization, the vital intention is to merge physics-based models with ML algorithms to leverage their complementary capabilities. Such combined ML-physics models are expected to thoroughly seize the dynamics of scientific systems and improve the knowledge of underlying physical laws [80]. There are several ways to inject physical laws, knowledge, or information into ML models to build physics aware ML models [81]. But physical information often shows a high degree of complexity due to connections among many physical variables diversifying over space and time at various ranges [82]. Conventional ML models can fall short of directly obtaining such relations from data, primarily when given limited measurement data [83]. This scarce data problem is one cause for the failure of the generalization to the circumstances in unseen training data. As a result, several novel research has been used to include physical information into training loss functions to support ML models to seize generalizable patterns consistent with underlying physical laws and governing equations. One of the most common ways to make machine learning models consistent with physical laws is by extending the loss function of the machine learning models to include physical constraints and other physical information [84]. Although the concept of integrating scientific knowledge and machine learning models has only become a popular topic of scientific research in the last few years, there is already extensive literature on this topic [85–88]. In the last decade, there has been an increase in utilizing operational

flight data, namely Quick Access Recorder (QAR) or Flight Data Recorder (FDR) [19] for many applications such as performance monitoring, anomaly detection, or weather forecasting [20–23]. These data consist of historical logs of all parameters that can be measured or observed through on-board sensors and systems. Even though they do not have information on the thrust, drag or lift, they record critical performance indicators such as vertical speed, gross weight and fuel flow. This capability makes them highly suitable for the supervised learning of aircraft performance. Chati and Balakrishnan used FDR data in tree-based learning and Gaussian process algorithms to model fuel flow [24–26]. Baklacioglu [27] combined genetic algorithms and neural networks for the same purpose. Baumann and Klingauf [28] proposed another scheme for supervised learning of fuel consumption with FDR. The works of Huang et al. [29] and Khadilkar [30] are other examples that have used operational data. These studies proved that aircraft performance can be represented through machine learning techniques. However, they did not investigate model performances in flight regimes that are not covered by the training data. Thus, the applicability of the proposed approaches to the complete flight envelope remains an open issue.

This thesis derives from our previous study on tail-specific fuel consumption modeling [11]. Previously, we proved that using proper neural network architectures and hyper-parameters, it is possible to build satisfactory fuel consumption models capturing actual profiles. The simplicity in fuel consumption modeling is that the parameters it depends on are distinct. In a nutshell, fuel consumption varies with the amount of thrust to be produced, altitude, and Mach number. However, the inter-dependencies are non-linear and complex. As illustrated in previous chapters, deep neural networks are very efficient for this problem because neural networks can be used as universal function approximation under certain conditions [26, 89]. However, extrapolating to the flight or atmospheric conditions that are not in the data is challenging and out-of-distribution generalization capability of neural network models or even any machine learning algorithms may not be enough to approximate the latent or hidden functions related to problem. Granted that the data comprehend samples from operational limit conditions, a black-box model could be extrapolated to some degree. However, it is not the case in real operations because the regimes aircraft fly are determined by either air traffic regulations or airline preferences. Moreover,

in a level flight, aircraft speed (Mach number) is set to a fixed value, and altitude is constant. Commercial aircraft usually cruise at altitudes proportional to one thousand feet. Therefore, even though the data cover all cruise levels, there would still be a lack of altitude variance. For instance, operational data of an aircraft with a maximum operable altitude of forty-thousand feet would have forty-one distinct altitudes at best. The case with the Mach number is even more challenging, because the most efficient Mach speeds for jet aircraft usually converge to a very narrow region at high altitudes. For example, narrow-body aircraft generally cruise at 0.78 Mach, whereas wide-body aircraft fly 0.83 [90]. Additionally, airlines prefer high altitude cruise for fuel-efficiency. Therefore, the Mach number variation observed in operational data is limited.

This thesis aims to overcome this problem by introducing the cruise fuel flow dynamics into the deep neural networks. First, we analyze the model-based approaches and data to identify a physical intuition that we can embed as a criterion. Then, we generate artificial data-sets that the operational data do not cover. We train our deep neural network such that it learns a mapping for fuel consumption from the labeled data, and captures the physics-guidance we introduce through unlabeled artificial data. This thesis's contributions are: First, we select a physical-guidance function that is valid for all feasible flight conditions. Second, our methodology can be implemented for any parameter that lacks variation in data. Last, our framework produces physically-consistent outputs for unseen flight regimes.

6.1 Problem Formulation

We are interested in estimating fuel consumption from engines for a complete flight envelope. The aim is to find the fuel consumption estimator $f_{FF} : X \rightarrow Y$, where X is the set of inputs, and Y is the target variable, which is fuel consumption from engines in mass per time. We aim to find a proper function approximation such that:

$$f_{FF}(X) = \hat{Y} \quad (6.1)$$

$$f_{FF} = \arg \min_f J_E(Y, \hat{Y}) + \lambda_S J_S(f) + \lambda_{PHY} J_{PHY}(\hat{Y}) \quad (6.2)$$

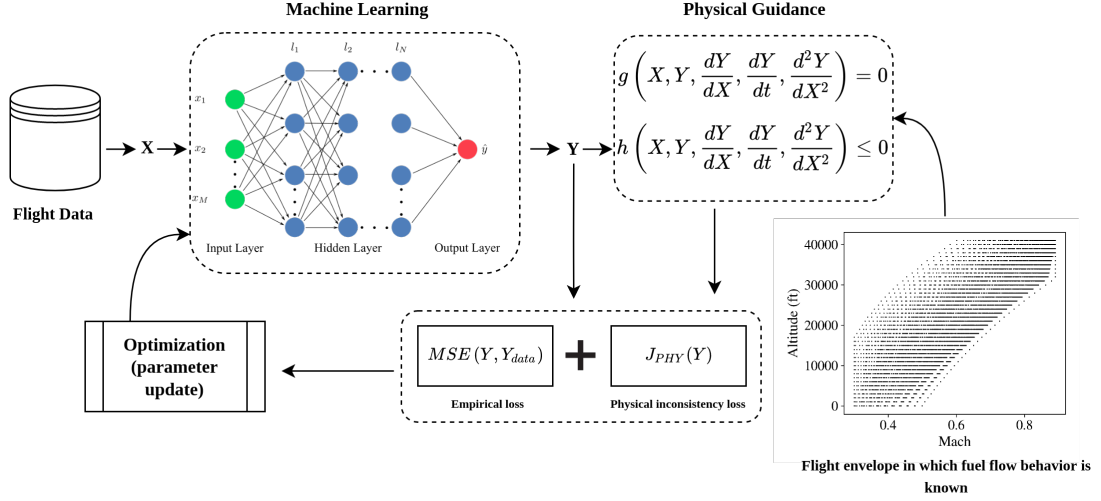


Figure 6.1 : The hybrid system consisting of neural networks, and physics-guidance. where $\hat{Y}$ is the predicted fuel consumption, J_E is the empirical error, and J_S is the model complexity (regularization term). Considering only J_E and J_S is the formulation of a generic supervised learning problem. However, a model trained in this scheme does not assure physically consistent fuel consumption predictions or known-physical laws for the complete state space. Hence, we introduce the physical inconsistency loss denoted by J_{PHY} . It quantifies how physical laws, constraints or relationships are violated, provides physical intuition and knowledge for the model to improve generalization capability. We can define these as:

$$g\left(X, Y, \frac{dY}{dX}, \frac{dY}{dt}, \frac{d^2Y}{dX^2}, \dots\right) = 0 \quad (6.3)$$

$$h\left(X, Y, \frac{dY}{dX}, \frac{dY}{dt}, \frac{d^2Y}{dX^2}, \dots\right) \leq 0 \quad (6.4)$$

In equations 6.3 and 6.4, g and h are known or derivable functions to express the physical constraints, laws or relations. By using these type of equations or inequalities, additional physics based knowledge can be injected in to ML model to learn by introducing an additional physical inconsistency loss.

The physical inconsistency loss J_{PHY} penalizes model predictions that contravene these constraints. Finally, the hyper-parameters $\lambda_{(.)}$ are the coefficients of physical loss components contributing to the penalization of the loss, related to specific physical-knowledge or law. By adjusting coefficients of these terms, level of the involvement of the physical laws or knowledge of the system can be adjusted to the machine learning model. Obtaining the values of $\lambda_{(.)}$ coefficients is generally possible with experimentally or trial-and-error methods. Adjusting these values provides the

injection degree of the physical intuition into ML model. The function approximation process is illustrated in Figure 6.1. Inputs to the neural network are features related to problem, labeled data, and unlabeled data in which we have knowledge of fuel consumption behavior. One contributor to the main loss function is the empirical error, where we measure how accurate the model is at predicting fuel consumption. As for the unlabeled data, even though we cannot measure any prediction error, we can check whether the output satisfies some physical constraints such as positiveness, monotonic increase/decrease, convexity or physical conservation laws [91]. The most crucial point is that all the terms in the loss function have to be differentiable to calculate gradients with respect to the model parameters. One way of expressing and implementing the constraints in a loss function is to use well-known activation functions such as ReLU, ELU, TanH, and Sigmoid. These activation functions' differentiability property helps to build a strong framework to express the physical constraints and information.

Combining all the required loss terms that include physical knowledge and constraints, the final loss function can be used for the optimization of the model parameters by using convex or other heuristic optimization techniques. However, one important issue must be noted that even though using this type of implementation has several advantages, the most crucial drawback would be making the loss function more complex. As a result, the optimization of the model parameters with the physics-based loss function can eventually be complicated, and thus leading to longer convergence time in the training of the model. As such, convergence to local optimal solutions during training is possible. To solve these problems, second order optimization algorithms such as Limited Memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) algorithm [92] or usage of combination of different optimization algorithms sequentially and complementary manner can be utilized [93, 94].

6.2 Fuel Flow Dynamics

In this section, we first provide neural network's Multi-Layer Perceptron (MLP) architecture and its mathematical background as a universal function approximator for regression problems, and then describe the flight physics governing fuel consumption dynamics.

6.3 Methodology

This section explains the physical guidance extracted from the fuel consumption analysis and how we implement it into the machine learning framework. First, we derive physics-based loss terms as a guide to the neural network design to capture the underlying physical trends in addition to the data-based correlations. Second, we explain how the physics-based loss terms are integrated through unlabeled artificial data.

6.3.1 Physics-based loss function design for fuel consumption

In this part of the study, we seek for a physical relationship for fuel flow that satisfies Equations 6.3, and 6.4. Equaling a physical quantity to zero is one of the possible approaches, e.g., Typical conservation of mass, conservation of momentum or energy equations with dissipating terms geared towards modeling cumulative effects such as total fuel consumption over distance time. However, we aim to model instantaneous fuel flow given flight and atmospheric conditions. Therefore, we base our approach on intuitive, instantaneous relationships, namely fuel flow's monotonic increase, and increasing power. It is important to note that the main aim here is to derive a representative equation for the power without utilizing BADA or OEM models. This enables our proposed model to be self-governing, i.e., independent from external performance models and parameters. Starting with the physical principle that

$$F \propto Th_{req}M \quad (6.5)$$

and using the Mach - true airspeed formula in Equation 2.35, Equation 6.5 can be written as,

$$F = \lambda \frac{Th_{req}\mathcal{V}}{\sqrt{\theta}} \quad (6.6)$$

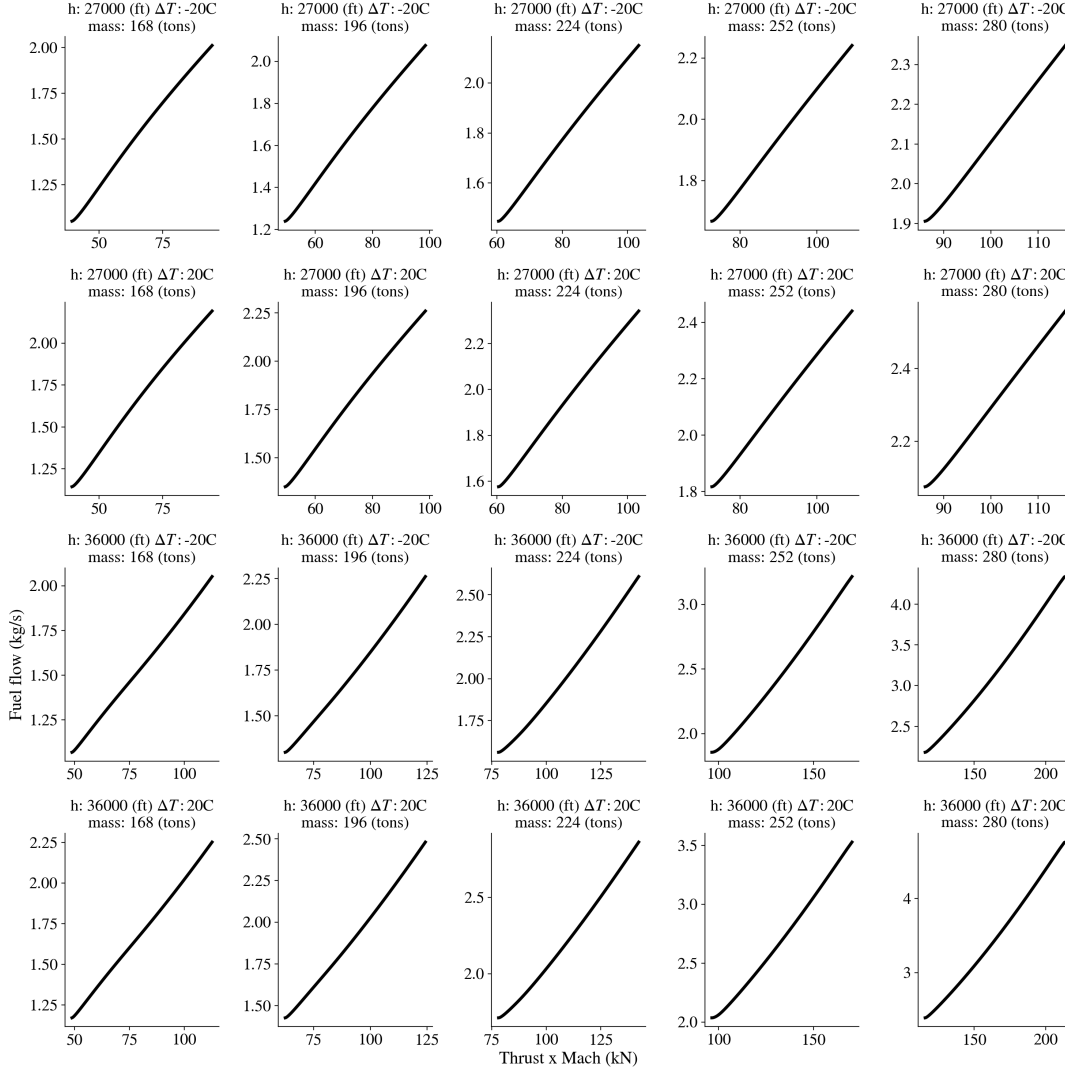


Figure 6.2 : Thrust times Mach versus fuel consumption for various flight regimes.

where the nominator part $T_{req}\mathcal{V}$ is the required power. The sole variable in the denominator is the temperature. In purely physical terms, given ambient conditions, increasing the power requires more fuel to be injected to the air flow. Hence, fuel flow also increases to maintain the new power setting.

We validate Equation 6.5 through BADA4. Figure 6.2 depicts fuel flow with growing $Th_{req}M$ term in cruise, given several flight conditions. Note that these are BADA4 model outputs, as well as Boeing's performance model's. Each subplot comprises Mach sensitivity similar to thrust and fuel consumption analyses in Section 6.2. All flight conditions yield increasing fuel flow profiles with increasing $Th_{req}M$. For the rest of the section, we approximate $Th_{req}M$ through empirical equations. First, we use the generic drag polar in Equation 2.12 to approximate the drag coefficient. The thrust required becomes;

$$Th_{req} = c_0 \delta M^2 (\mathcal{C}_0 + \mathcal{C}_2 C_L^2) \quad (6.7)$$

where $c_0 = 0.5 \kappa p_0 S$. Because all regimes show linear tendencies, we can consider fuel consumption as a multiplication of thrust times Mach with a scalar λ :

$$F = \lambda Th_{req} M \quad (6.8)$$

$$F = \lambda c_0 \delta M^3 (\mathcal{C}_0 + \mathcal{C}_2 C_L^2) \quad (6.9)$$

As seen in Figure 6.2, the scalar λ is not always constant. The slope is different at each atmospheric condition - mass combination:

$$\lambda = f(\delta, \theta, m) \quad (6.10)$$

Because the dependency on the mass is only due to drag polar, we can assume λ as a function of pressure and temperature ratios:

$$\lambda = f(\delta, \theta) \quad (6.11)$$

We can consider the scalar λ as a ratio between a constant scalar $\bar{\lambda}$, and pressure ratio multiplied by square root of temperature ratio. This assumption is based on Equation 2.32.

$$\lambda = \frac{\bar{\lambda}}{\delta \sqrt{\theta}} \quad (6.12)$$

Similar to the drag equation, the lift equation is:

$$L = c_0 \delta M^2 C_L \quad (6.13)$$

where lift equals weight in cruise. Hence, the lift coefficient can be calculated as:

$$C_L = \frac{W}{c_0 \delta M^2} \quad (6.14)$$

Finally, combining all equations above, the generalized fuel consumption equation is found as:

$$\mathcal{T} = \frac{M}{\sqrt{\theta}} \left(c_1 M^2 + c_2 \frac{m^2}{\delta^2 M^2} \right) \quad (6.15)$$

where $c_1 = \mathcal{C}_0 c_0$ and $c_2 = \mathcal{C}_2 / c_0$ are constants, and customized to the aircraft type. We utilize this generalized fuel consumption equation as the primary physical guide to fuel usage for various cruise flight and environmental conditions.

6.3.2 Implementation of the physics guided loss function

To generate physical guidance through loss functions, we first generate N_e sets consisting of several flight and atmospheric conditions as illustrated in Figure 6.2. The set is denoted as $\mathcal{R} = \{r_n \in \mathbb{R}^{N_r \times M_r}\}_{n=1}^{N_e}$, where M_r is the number of features, and N_r is the number columns in r_n . Columns of each set r_n are the input features; pressure ratio δ , temperature ratio θ , Mach number M , and aircraft mass m_{ac} . The formulation to generate these sets is to keep parameters constant, except the Mach number M :

$$r_n = \begin{bmatrix} \delta_n & \theta_n & M_n^{min} & m_{ac,n} \\ \delta_n & \theta_n & M_n^{min} + \Delta M & m_{ac,n} \\ \delta_n & \theta_n & M_n^{min} + 2\Delta M & m_{ac,n} \\ \vdots & \vdots & \vdots & \vdots \\ \delta_n & \theta_n & M_n^{max} & m_{ac,n} \end{bmatrix} \quad (6.16)$$

where M_n^{min} and M_n^{max} are the minimum and maximum operable Mach speeds at the n^{th} set given as $\{\delta_n, \theta_n, m_{ac,n}\}$. The thrust times Mach set associated to this set r_n is denoted by $\mathcal{T}_n = \{tm_n^{[i]}\}_{i=1}^{N_r}$. It is calculated through Equation 6.15, given i^{th} row of r_n :

$$tm_n^{[i]} = \frac{M_n^{[i]}}{\sqrt{\theta_n^{[i]}}} \left(c_1 (M_n^{[i]})^2 + c_2 \frac{(m_{ac,n}^{[i]})^2}{(\delta_n^{[i]})^2 (M_n^{[i]})^2} \right) \quad (6.17)$$

Then, we re-order the associated thrust times Mach such that it monotonically increases:

$$\mathcal{T}_n' = \{tm_n^{[i]} : tm_n^{[i+1]} - tm_n^{[i]} \geq 0\} \quad (6.18)$$

Using the index order in $\mathcal{T}_n'$, we organize the matrix r_n in the same way. The new version is r_n' :

$$r_n' = \{r_n^{[i]} : \mathcal{T}(r_n^{[i+1]}) > \mathcal{T}(r_n^{[i]})\} \quad (6.19)$$

Fuel consumption predicted by the model for given r_n' is $\hat{Y}_n$, where $\hat{Y}_n = f_{FF}(r_n', \xi) = \{\hat{y}_n^{[i]}\}_{i=1}^{N_b}$. Considering the monotonic increase relationship illustrated in Figure 6.2, we expect $\hat{Y}_n$ to be in an increasing order. To model this, we divide the predicted fuel consumption $\hat{Y}$ into two parts as below:

$$\hat{Y}_n^+ = \{\hat{y}_n^{[2]}, \hat{y}_n^{[3]}, \dots, \hat{y}_n^{[N_b]}\} \quad (6.20)$$

$$\hat{Y}_n^- = \{\hat{y}_n^{[1]}, \hat{y}_n^{[2]}, \dots, \hat{y}_n^{[N_b-1]}\} \quad (6.21)$$

If the model satisfies the physical laws, element-wise subtraction of $\hat{Y}_n^-$ from $\hat{Y}_n^+$ should always be positive. Hence, we penalize model predictions that violate it. The

most practical way to do it is applying *ReLU* activation function, which is defined as:

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (6.22)$$

Hence, the first physical loss function related to the monotonicity can be written as:

$$J_{PHY,1} = \frac{1}{N_e} \sum_{n=1}^{N_e} \sum_{i=1}^{N_r-1} \frac{1}{N_r^+} ReLU \left(\hat{y}_n^{[i]} - \hat{y}_n^{[i+1]} \right) \quad (6.23)$$

where N_r^+ stands for the number of positive outputs of $ReLU(x)$, and $N_b^+ \leq N_r$. Previous studies usually divide the sum by N_b . However, in our case using such approach causes the loss function to be so small. Therefore it becomes negligible compared to the empirical error. The second physics-guided loss prevents model to predict negative fuel flows, and is defined as:

$$J_{PHY,2} = \frac{1}{N_e} \frac{1}{N_r} \sum_{n=1}^{N_e} \sum_{i=1}^{N_r} ReLU \left(-\hat{y}_n^{[i]} \right) \quad (6.24)$$

The last physics-based loss function is a heuristic limitation to the fuel consumption the model predicts. We limit the difference between the maximum and minimum fuel consumption for a given set of r_n with a pre-defined value, denoted by $\mathcal{F}_{ref}$. Its magnitude hinges on domain expertise from pilots and aircraft performance engineers.

$$J_{PHY,3} = \frac{1}{N_e} \sum_{n=1}^{N_e} ReLU \left(\max \hat{Y}_n - \min \hat{Y}_n - \mathcal{F}_{ref} \right) \quad (6.25)$$

The last two physics-guided losses are more like modifications to the first one. The first intuition enables the model to capture a linear relationship for fuel flow, but it does not include lower and upper limits. That is where the remaining loss terms appear. Finally, the combined physical loss function is the sum of Equations 6.23, 6.24, and 6.25:

$$J_{PHY} = \sum_i \lambda_{PHY,i} J_{PHY,i} \quad (6.26)$$

Combined with the empirical error, which is the mean squared error in this study, the final loss function is given in Equation 6.27 :

$$\begin{aligned} J = & \lambda_e \frac{1}{N_d} \sum_{i=1}^{N_d} \left(\hat{y}_d^{[i]} - y_d^{[i]} \right)^2 + \lambda_s ||W||_2^2 + \frac{1}{N_e} \sum_{n=1}^{N_e} \sum_{i=1}^{N_r-1} \frac{1}{N_r^+} ReLU \left(\hat{y}_n^{[i]} - \hat{y}_n^{[i+1]} \right) \\ & + \frac{1}{N_e} \frac{1}{N_r} \sum_{n=1}^{N_e} \sum_{i=1}^{N_r} ReLU \left(-\hat{y}_n^{[i]} \right) + \frac{1}{N_e} \sum_{n=1}^{N_e} ReLU \left(\max \hat{Y}_n - \min \hat{Y}_n - \mathcal{F}_{ref} \right) \end{aligned} \quad (6.27)$$

This loss function becomes the main learning function in data-driven neural network design process. In particular, this specific design allows us to capture data correlations and nonlinear relationships inline with the general physical principles that extend beyond the flight envelope as captured by data. In the next section, we demonstrate this methodology by using actual Quick Access Recorder (QAR) data set from a major European flag carrier airline. The results show that the proposed approach allows us to develop more precise fuel consumption (and thus flight performance models) over a wide range of the flight envelope in comparison to standard supervised learning approaches and existing BADA4 model.

6.4 Experiment

The dataset consists of myriads of features sampled at 1 Hz, which is the maximum recording frequency in flight data recording systems [95]. The dataset of interest for this study includes information about dynamic states of an aircraft (airspeed, altitude, position, vertical speed, and heading), body axis states (angle of attack, and longitudinal and lateral acceleration), performance states (throttle settings, fuel flow from engines, and mass), configuration states (high lift devices, landing gear, and speed brake) and environmental states (wind speed/direction, static and total temperature of air, and air pressure). Even though these states are the most critical variables used in this kind of research, a QAR device usually records more than 1000 features. Table 6.1 summarizes the variables selected from the QAR dataset used in this thesis. Because each individual aircraft differs at performance, we have used QAR dataset of a unique tail-number. Figure 6.3 depicts geospatial distribution of the flight routes in the training data.

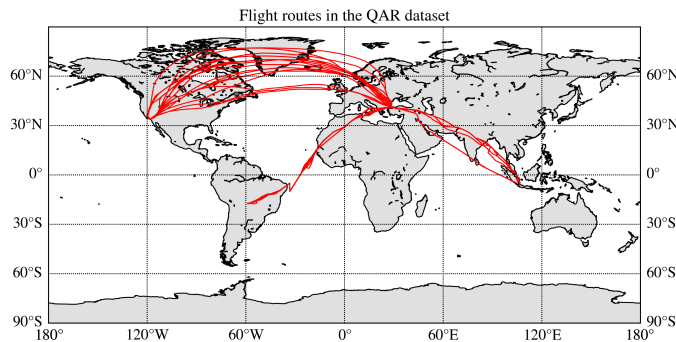


Figure 6.3 : Lateral profiles of the flights in the dataset.

Table 6.1 : Summary of parameters directly used from QAR dataset.

h	Barometric altitude [ft]
T_c	Static air temperature [C]
V_{CAS}	Calibrated airspeed [kt]
M	Mach speed
m_{ac}	Aircraft mass [kg]
F_{eng}	Fuel flow from engines [lb/hr]
$Tr_{1,2}$	Throttle positions of engines 1 and 2
V_{wind}	Wind speed [kt]
χ_{Wind}	Wind direction [deg]
χ_{ac}	True heading [deg]
δ_{flap}	Flap deflection [deg]
δ_{lg}	Landing gear status $\{0, 1\}$
δ_{sb}	Speed break deflection [deg]
F_{apu}	APU fuel consumption [lb/hr]

During training, dataset of a particular tail-number aircraft is utilized. The data has 98 flights, 81 long haul and 17 short haul, with an amount of $2.6M$ cruise points. Because the original QAR files are flight-based, they are comprised of the whole flight from take-off to landing, including taxi phases. Therefore, ground, climb and descent points should be separated to have solely cruise trajectories. We use the altitude gradient to check whether a point belongs to a level flight. Let $\mathbb{X}_{crz}$ be the cruise data:

$$\mathbb{X}_{crz} = \left\{ x^{[i]} : |\dot{h}^{[i]}| \leq 1 ft/s \right\}_{i=1}^{N_d} \quad (6.28)$$

where $x^{[i]} = \{\delta^{[i]}, \theta^{[i]}, M^{[i]}, m_{ac}^{[i]}\}$, and $h^{[i]}$ is the corresponding altitude at i^{th} element. As for the train-test split, we spared the first 90 flights for training, and the remaining 8 flights for testing. We selected standard scaling to scale the data. Let $\mathbb{X}_{trn}$, $\mathbb{X}_{tst}$ and $X_{(.)}^{sc}$ be the training, test, and scaled data, respectively.

The complementary flight regime set $\mathcal{R}$ is independent from the data. Its elements are separate from the cruise data $\mathbb{X}_{crz}$. There is no experiment in this thesis to find the optimum number of elements of $\mathcal{R}$. We selected two hundred distinct flight regimes consisting different values of altitude h , ISA temperature deviation ΔT , and

Table 6.2 : Operational altitude, ISA deviation, and aircraft mass limits.

Parameter	Min	Max
Altitude $h(ft)$	0	41000
ISA deviation $\Delta T(C)$	-77	50
Mass $m_{ac}(kg)$	167,000	353,000

aircraft mass m_{ac} . Altitude and ISA deviation are sufficient to calculate pressure and temperature ratio through the following equations:

$$\theta = \frac{T_{amb}}{288.15} = \frac{288.15 + \Delta T_{ISA} - 0.0019812h}{288.15} \quad (6.29)$$

$$\delta = \left(\frac{T_{amb} - \Delta T_{ISA}}{288.15} \right)^{5.25588} \quad (6.30)$$

For each combination of δ , θ , and m_{ac} , we calculate the corresponding Mach number limits. The minimum Mach number is in which the aircraft is cruising at the maximum lift coefficient available, $C_{L,max}$. This study utilizes BADA4 buffet limit model to calculate $C_{L,max}$, and for now, it is the only point wherein this methodology requires an external performance model. We calculate the maximum Mach number given the maximum operable calibrated airspeed $V_{CAS,max}$:

$$M_{max} = \sqrt{5 \left[\left(\frac{1}{\delta} \left\{ \left[1 + 0.2 \left(\frac{V_{CAS,max}}{661.4786} \right)^2 \right]^{3.5} - 1 \right\} + 1 \right)^{\frac{1}{3.5}} - 1 \right]} \quad (6.31)$$

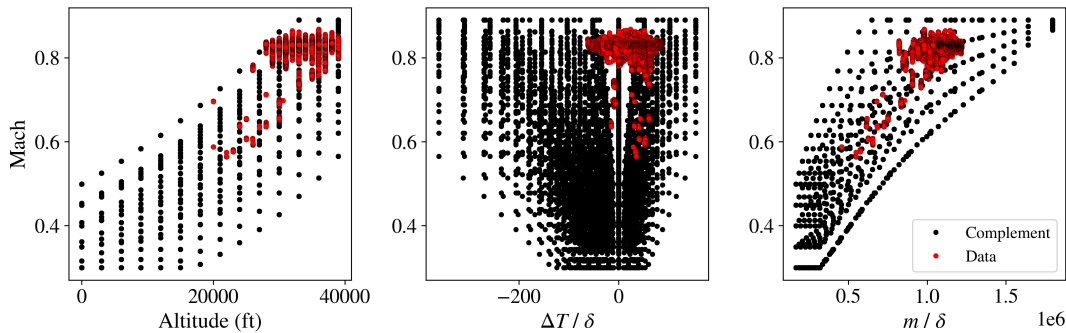


Figure 6.4 : Full flight envelope compared to the regimes represented in the data.

For the aircraft in this study, the maximum operable calibrated airspeed is $V_{CAS,max} = 330[kt]$. The minimum Mach speed is connected with altitude and aircraft mass.

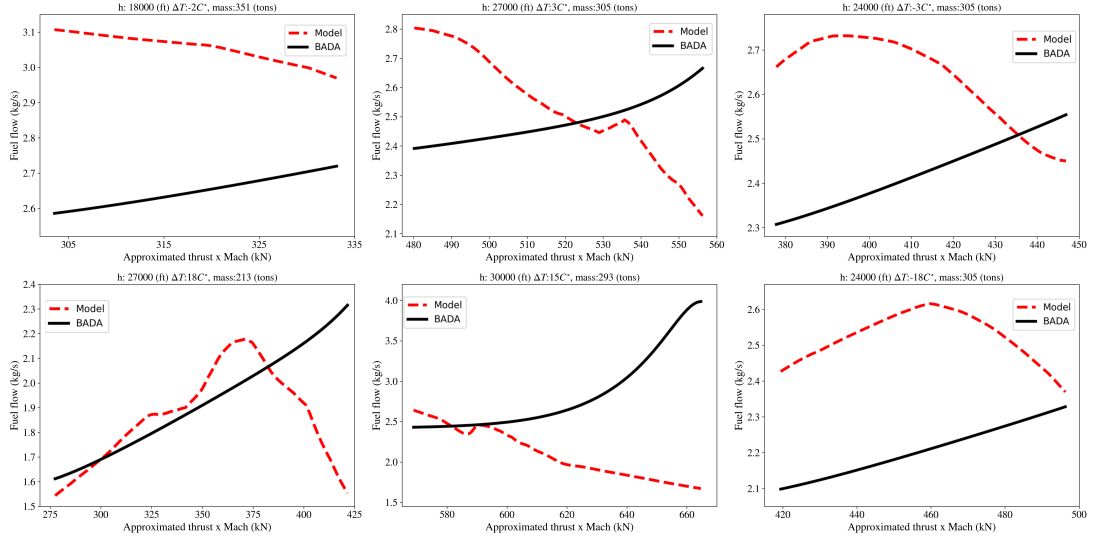


Figure 6.5 : Mach sensitivity of the model without including physics.

The flight regime sets in $\mathcal{R}$ are selected considering the aircraft limits, which are open-access information. The lower and upper limits of altitude, ISA deviation and aircraft mass are given in Table 6.4. The elements of $\mathcal{R}$ are triple combinations of altitude, ISA deviation, and aircraft mass within these boundaries. Then, BADA4 and Equation 6.31 calculate the Mach speed limits for each r_n . Two examples are given below:

$$r_1 = \begin{bmatrix} 0.35 & 0.84 & 0.65 & 280,000 \\ 0.35 & 0.84 & 0.653 & 280,000 \\ 0.35 & 0.84 & 0.657 & 280,000 \\ \vdots & \vdots & \vdots & \vdots \\ 0.35 & 0.84 & 0.837 & 280,000 \\ 0.35 & 0.84 & 0.84 & 280,000 \end{bmatrix} \quad r_2 = \begin{bmatrix} 0.22 & 0.76 & 0.76 & 245,000 \\ 0.22 & 0.76 & 0.764 & 245,000 \\ 0.22 & 0.76 & 0.769 & 245,000 \\ \vdots & \vdots & \vdots & \vdots \\ 0.22 & 0.76 & 0.883 & 245,000 \\ 0.22 & 0.76 & 0.89 & 245,000 \end{bmatrix}$$

Figure 6.4 illustrates the envelope covered in the data, and the flight regimes included. On the left, there is altitude versus Mach, and it shows that most of the flights are above 30,000 feet. The rest is highly sparse and there is almost none at altitudes below 20,000 feet. The middle sub-figure represents Mach distribution over ISA deviation. To generalize it, we divided ISA deviation by pressure ratio. Otherwise, there is no inter-dependency between ISA deviation and Mach. Under the limits, an aircraft can fly at all Mach speeds with all ISA deviation values. In the data, observed ISA deviations are between -25°C and 20°C . We expanded this regime to have more temperature ratio values to provide to the algorithm. Lastly, the plot on the right depicts Mach distribution over aircraft mass. Likewise, aircraft mass is divided by pressure ratio for generalization. As an aircraft gets lighter after burning fuel, it either increase

its speed or altitude for flight efficiency. In real-world operations, this appears as step climbs after cruising some distances. Therefore, the Mach variance in the data is low, because the aircraft usually maintains the speed, but increase the altitude. This has another effect as well, which is the rarity of observing high aircraft masses at higher altitudes. Whenever an aircraft of this study does its third or fourth step climb, it is ordinarily lighter than 260 tons. Therefore, many of the feasible altitude - mass pairs are not available in data.

The tolerance value $\mathcal{F}_{tol}$ is selected to be $4,000[kg/h]$. This magnitude is heuristic, and based on simulation results using BADA4. Hyper-parameters are $\lambda_e = 1.0$, $\lambda_s = 10^{-6}$, $\lambda_{PHY,1} = 0.8$, $\lambda_{PHY,2} = 0.2$, and $\lambda_{PHY,3} = 0.8$. Note that the elements of $\mathcal{R}$ are also scaled through Equation 3.32. For more efficient stochastic gradient descent, the training data $\mathbb{X}_{trn}^{sc}$ is divided into mini-batches with a size of 1024. The initialization [61] is selected to assign the initial states of $W^{[l]}$ and $b^{[l]}$. The learning process utilizes the *AdaBound* optimization algorithm [96] to update weight matrices W and bias vectors b at each step k :

$$\xi_{k+1} = \xi_k + AdaBound \left(\frac{\partial J(x_{trn}^{[k],sc}, \mathcal{R})}{\partial \xi}, \alpha \right) \quad (6.32)$$

We tuned the learning rate α by checking the status of the loss function. If the loss function is not improved 10 times consecutively, the learning rate α is reduced by %90. Its initial value is chosen as 0.001. The models are trained for 1000 epochs. Five hidden layers are used and the neuron numbers are $\{1024, 512, 256, 128, 32\}$. The deep neural network models are implemented and trained with PyTorch framework [97].

We compare the model with other linear and non-linear function approximation algorithms. These are linear regression *LR*, support vector regression *SVR*, neural network with one hidden layer *NN*, and deep neural network *DNN* without including physics loss. Additionally, fuel consumption calculations from BADA4 is provided. The model denoted by *BADA4** is the baseline BADA4 fuel consumption calibrated through linear regression. Two different error metrics, namely the mean absolute error and the mean absolute percentage error are presented in Table 6.3. The results indicate that deep neural networks yield the lowest prediction errors. This implies the necessity of universal function approximators for capturing fuel consumption dynamics better.

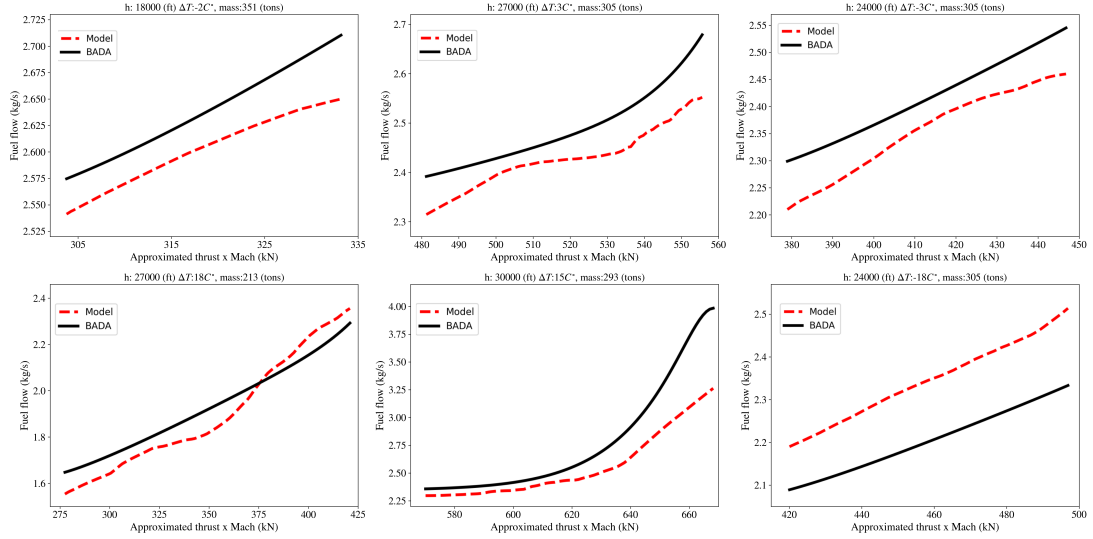


Figure 6.6 : Mach sensitivity of the model with including physics.

Table 6.3 : Comparison of prediction results on the test set.

Criteria	BADA4	BADA4*	SVR	LR	NN	DNN	Proposed
MAE [kg/h]	291	215	195	201	172	127	133
MAPE %	3.712	2.632	2.677	2.695	1.897	1.521	1.568

$$MAE = \frac{1}{N_d} \sum_i^{N_d} \left| \hat{y}_d^{[i]} - y_d^{[i]} \right| \quad (6.33)$$

$$MAPE = 100 \times \frac{1}{N_d} \sum_i^{N_d} \left| \frac{\hat{y}_d^{[i]} - y_d^{[i]}}{y_d^{[i]}} \right| \quad (6.34)$$

Fuel consumption prediction errors of the generic deep neural network and our physics informed network are shown to be very close on the test set. The main discrepancy between these designs lies in their respective physical consistencies. As illustrated in Figure 6.2, we expect any physically consistent model to produce an increasing fuel consumption profile with an increasing thrust times Mach input, given an altitude, a ISA deviation, and a reference mass. Figure 6.5 demonstrates fuel consumption predicted by the generic deep neural network model for six different cases of flight conditions. Even though this model accurately captures the fuel consumption trajectories on the test set, it performs poorly in terms of physical consistency. All fuel consumption profiles in these six regimes have decaying parts, with growing thrust times Mach. Moreover, the behavior is not homogeneous, i.e. the model does not always yield monotonic decreasing curves.

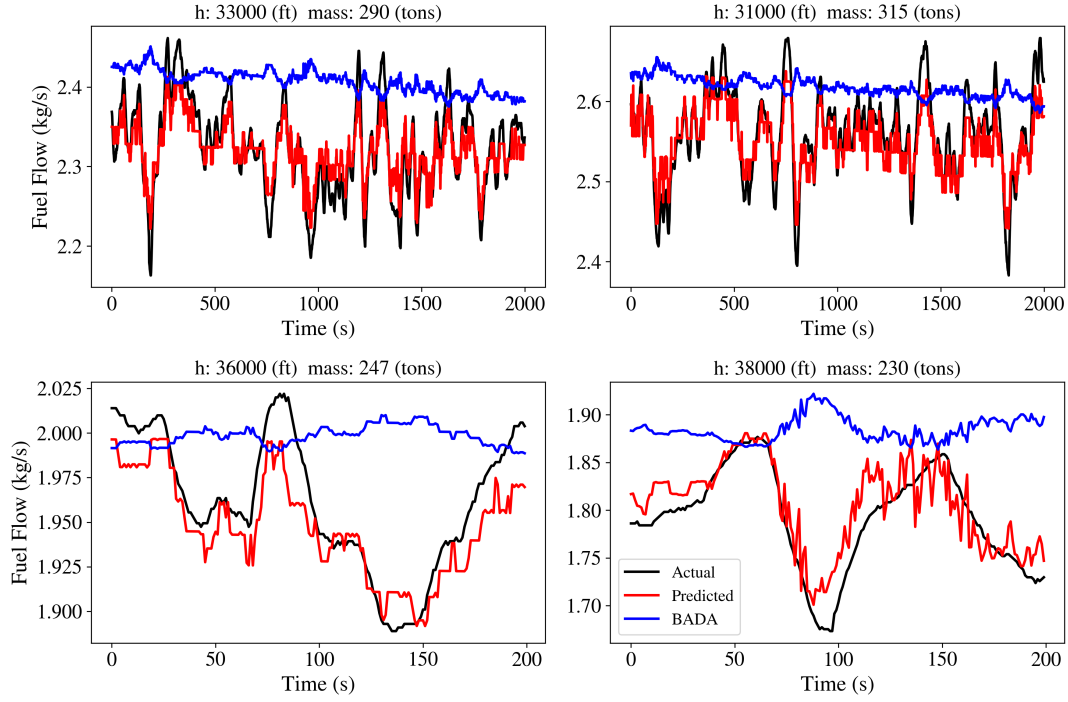


Figure 6.7 : Fuel consumption prediction with the trained physics-guided NN model.

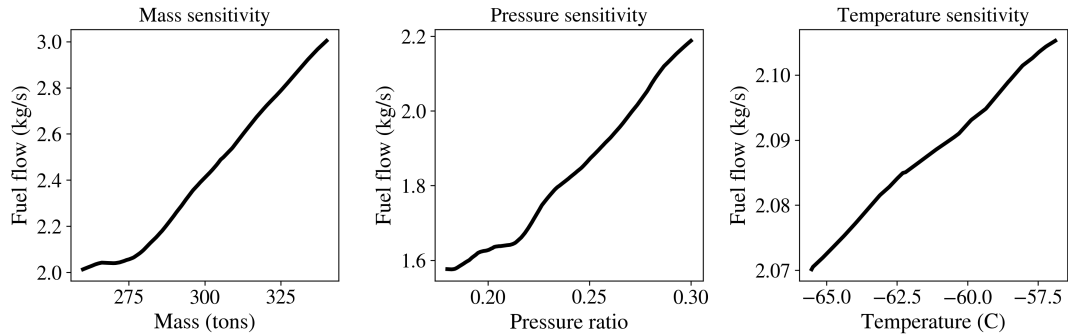


Figure 6.8 : Model sensitivity to pressure ratio, temperature ratio, and mass.

On the other hand, our proposed model, which includes physical loss functions, complies with the physical intuition. Figure 6.6 reveals that fuel consumption trajectories predicted with the proposed neural network model grow as thrust times Mach values increase. They are not always entirely linear, but do not violate the physics. Additionally, in many cases fuel consumption curves predicted with the trained model is under the ones computed by BADA4. This shows that model-based approaches tend to overestimate fuel consumption for this tail-number aircraft. There are some exceptions, like the example in right-bottom plot in Figure 6.6. However, we do not have labeled data for this flight regime to justify whether our prediction is very close to the actual. Still, as can be seen in Figure 6.7, the proposed model adequately

predicts the fuel consumption profiles of the test set. Compared to our model, BADA4 overestimates fuel consumption in the test set. Furthermore, our model is able to capture the unknown patterns that BADA4 does not cover. Zoomed regression plots show that the model is able to predict short-period fluctuations. These are due to autopilot modules that control the angle of attack and throttle to maintain level flight. Because NNs are universal function approximators, they can establish such mappings that curve fitting algorithms fail.

Additionally, we show the model's sensitivity to the other parameters; namely the pressure ratio, temperature ratio, and mass in Figure 6.8. Because these variables have enough variance in the data, the sensitivity lines are meaningful without including additional loss terms. Among these parameters, temperature is output of a forecast in flight planning. It usually appears at high accuracy, but it is still possible to correct weather uncertainty using data-driven techniques, as well [71]. In conclusion, appending additional loss terms to the empirical error does not diminish the supervised learning performance. Properly selected loss functions even enable the deep neural network to capture further dynamics and flight regimes that data sets do not actually cover.



7. WIND MODEL CALIBRATION

An airline's operational deficiencies can be categorized by their sources. Deficiencies sourced by other Air Traffic Management (ATM) stakeholders are basically considered to be Air Traffic Control (ATC) decisions and restrictions, airport and ground service deficiencies. As for an airline's internal sources of deficiencies, we could count deficiencies in fleet selection, flight scheduling and maintenance planning on macroscopic level, which involve fleet-based operations. On microscopic level, which we define as the issues on an individual aircraft, we could point deficiencies due to uncertainties in the flight planning process. Based on these ATM rules and uncertainties, airlines have to generate the most efficient trajectories. However, they are not able to control or intervene all these deficiency factors. As for the correctable ones, we focus on two distinct problems: Uncertainty of environmental states, which refers to wind uncertainty, is a critical issue because wind is one of the fundamental inputs to the trajectory generation process. In addition to proper estimation of wind, characteristics of the aircraft must be known. This refers to high-fidelity digital-twin aircraft performance models tailored to individual aircraft.

Aircraft performance models and estimates of key environmental factors such as wind play a crucial role in designing efficient flight trajectories for key segments of long-haul flights [38, 98]. Wind has a high impact on ground speed, which determines the trip time and the total amount of fuel consumed. Aircraft performance models compute fuel consumption rates for given flight conditions. Moreover, ground-based flight planning tools utilize these two key factors to generate optimum speed and altitude schedules. The process of generating and optimizing 4D trajectories is summarized in Figure 7.1. Even though the baseline performance models and wind forecast systems yield estimations reasonably close to the actual values, they still result in errors due to some uncertainties and assumptions. This study aims to compensate for errors in wind speed estimations through machine learning methodologies.

During flight planning, airlines utilize the wind forecasts provided by meteorological services. One of the fundamental component for providing this information is Global Forecast System [99], [100]. A GFS process is comprised of two stages: First, a data assimilation is performed from many measurement sources such as aircraft, radiosonde [101]. Then the climate models utilize this data as an input to Numerical Weather Prediction algorithms [102]. Outputs of these forecasts are predictions of a myriad of meteorological parameters along earth for several isobaric surfaces. Even though these models are quite successful at capturing the meteorological trends, they still come with errors. The studies in [103, 104] considered this problem by investigating discrepancies between GFS forecasts and actual wind acquired through ADS-C and Mode-S messages from aircraft. These kinds of studies steered the problem to the improvement of wind forecasts or generating them through aircraft trajectory data, which have been widely studied. Improving the baseline forecasts with aircraft measurements first implemented in Rapid Cycle Update system in [105]. Radar track data, ADS-B and ADS-C messages are utilized to generate local wind maps in [106–108]. The authors of [109] increased the accuracy of predicted trajectories by estimating wind speed through sampling-based filtering systems. A recent work done in [110] showed potential improvements in forecasts by modeling their uncertainty through linear regression methods. Some studies revealed the importance of wind information on actual flight operations. Flight planning and optimization based on wind information are studied in [111] and [112]. Finally, the effect of wind uncertainty on flight planning is analyzed in [113, 114].

The aim of this study is not to build models doing the job that GFS already does. Instead, we focus on improving the baseline wind estimations of GFS by reducing their errors through machine learning techniques. Nonlinear function approximations between a proper set of inputs including GFS forecasts of u and v components of wind and actual wind measurements are achieved via deep neural networks.

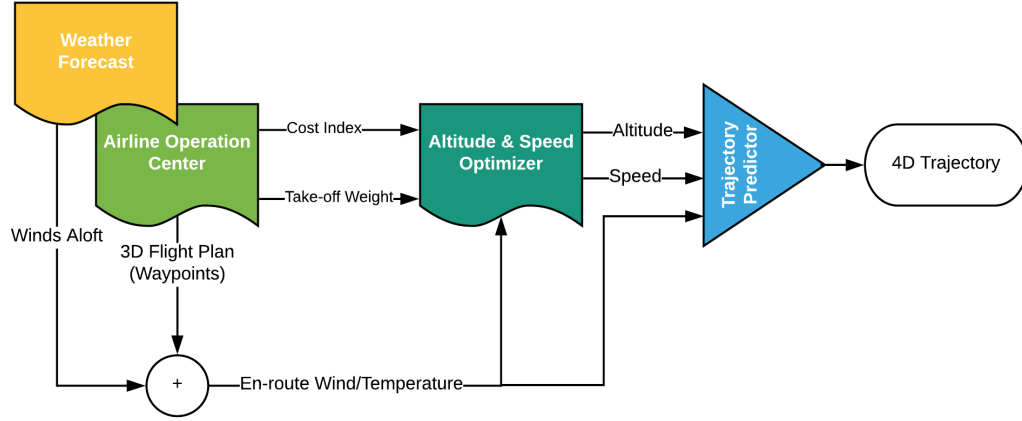


Figure 7.1 : The summary of 4D trajectory generation.

7.1 Problem Definition

This thesis considers the problem of proper estimation of trip fuel and trip time. In that sense, we are investigating how much is the gain when aircraft performance models and wind estimations are improved. General form of the time-invariant dynamics of the aircraft is considered as $\dot{x}(t) = f(x(t), u(t))$ and $x(0) = x_0$ where the $x(t) \in X \subseteq \mathbb{R}^n$, $u(t) \in U \subseteq \mathbb{R}^m$ such that $n, m \in \mathbb{N}$ and the state $x_0 \in X$ is called the initial state of the ownship. Aircraft states are defined as $x = \{p_x, p_y, h, V_{TAS}, \chi, m\}$ and input space is $u = \{\chi, \gamma, Th\}$. The equations of aircraft motion are:

$$\dot{p}_x = V_{TAS} \cos \chi \cos \gamma_{Q_j} + w_x \quad (7.1)$$

$$\dot{p}_y = V_{TAS} \sin \chi \cos \gamma_{Q_j} + w_y \quad (7.2)$$

$$\dot{h} = V_{TAS} \sin \gamma + w_z \quad (7.3)$$

$$\dot{V}_{TAS} = -\frac{C_D S \rho V_{TAS}^2}{2m} - g \sin \gamma_{Q_j} + \frac{Th_{Q_j}}{m} \quad (7.4)$$

$$\dot{\chi} = \frac{C_L S \rho V_{TAS}}{2m} \frac{\sin \mu}{\cos \gamma_{Q_j}} \quad (7.5)$$

$$\dot{m} = -F_{Q_j} \quad (7.6)$$

where p_x and p_y are the aircraft coordinates, h is the geopotential pressure altitude, V_{TAS} is the true airspeed, χ is the heading and m is the aircraft mass. C_L and C_D are the coefficients of lift and drag, respectively. The air density is denoted by ρ and the wing area is represented by S . As for the control inputs, Th is the thrust force and γ is the flight path angle. $\{w_x, w_y, w_z\} \in V_{wind}$ are x , y and z components of wind speed

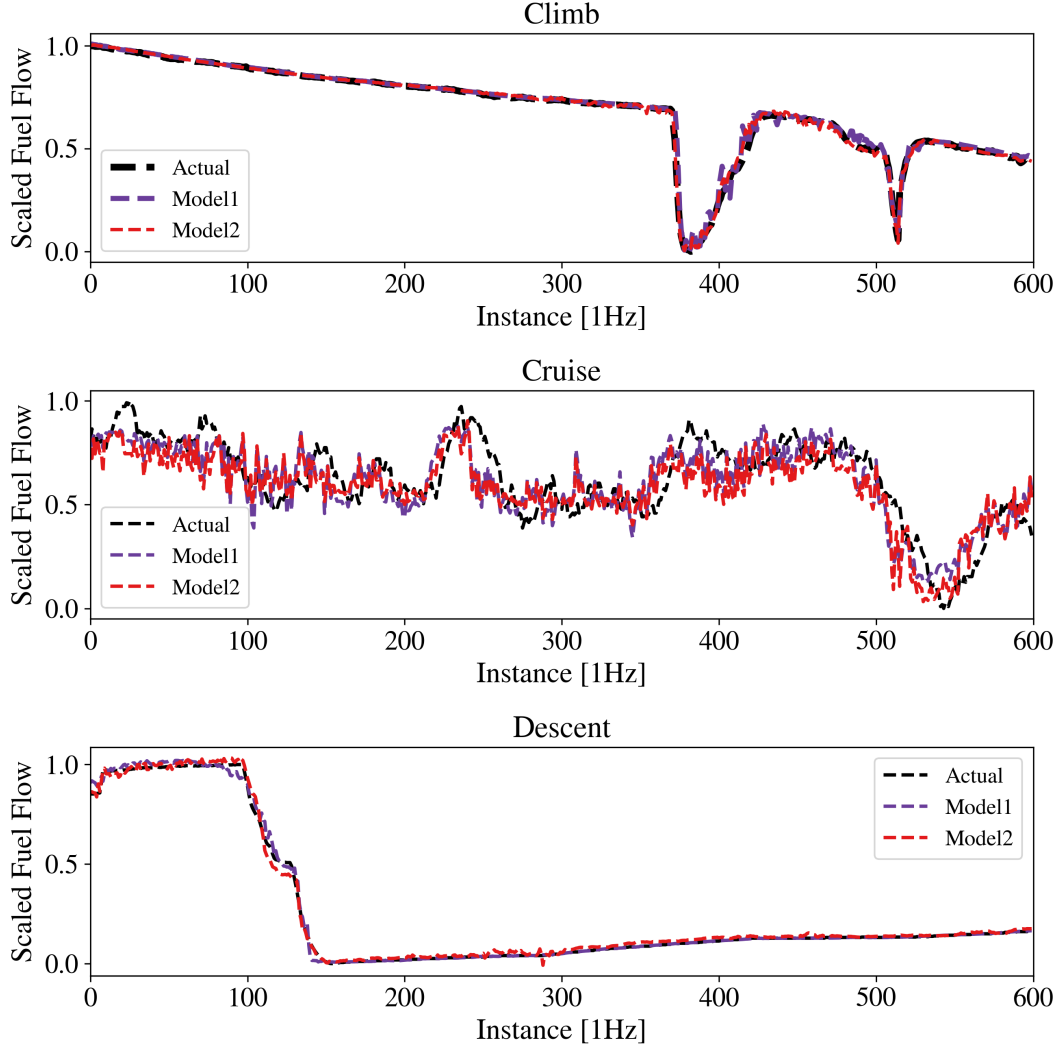


Figure 7.2 : Digital twin performance model through machine learning based estimations.

$\vec{V}_{wind}$. Q_j stands for the flight mode aircraft is operating and $Q = \{Q_j | j \in \mathbb{N}\}$ is the countable set consists of these modes. The parameters with index Q_j indicates that their formulations differ with flight mode, which are climb, cruise and descent.

Let $\mathbb{X} \in \mathbb{R}^N$ be the set of inputs to the 4D trajectory prediction system defined in Figure 7.1. Let Γ and Ω represent the aircraft performance model and wind information. Note that $\Gamma, \Omega \in X$. The 4D trajectory predicted with the input set $\mathbb{X}$ is denoted as $x_p(t)$. The desired condition is to obtain $x_p(t)$ close to the actual trajectory $x_a(t)$, such that $\|x_p(t) - x_a(t)\|_2$ is minimized. This study focuses on improving the estimations of Γ and Ω . Aircraft performance model Γ consists of drag, thrust and fuel flow, $\Gamma : x \rightarrow \{C_L, C_D, Th_{Q_j}, F_{Q_j}\}$. Wind information Ω is the same as V_{wind} . In this thesis, we define Ω in terms of u and v components of wind, where u is the component of

the horizontal wind towards east and v is towards north. These are the same with w_x and w_y . We use this notation to be compatible with meteorological data. The vertical component of the wind is omitted.

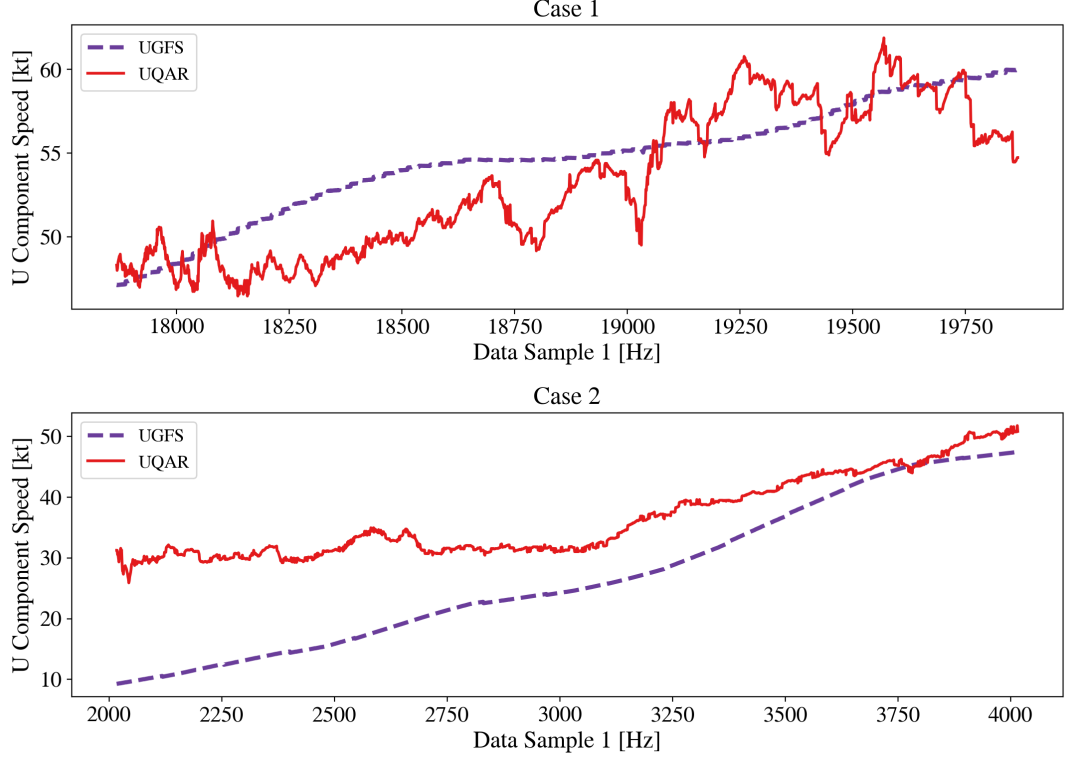


Figure 7.3 : Example of a comparison between GFS and QAR wind speed components for long haul case.

The objective is to find proper functions that map baseline aircraft performance models and wind estimations to the actual ones, or somewhat update them. For the aircraft performance model, the update function is defined as:

$$f_{Q_j} : \mathbb{R}^{\rho(\cdot)} \rightarrow \mathbb{R} \quad (7.7)$$

where $Q_j \in Q$ and $\rho(\cdot)$ assigns the dimension of the input set for a given flight mode. The functions f_{Q_j} are represented by multilayer perceptrons and composed of several functions:

$$f_{Q_j} = f_{Q_j}^{(1)} \circ f_{Q_j}^{(2)} \dots \circ f_{Q_j}^{(L)} \quad (7.8)$$

where L is the number of layers in the neural network. A multilayer perceptron described in Eq. 7.8 defines the following mapping:

$$\Gamma_{Q_j}^* = f_{Q_j}(\Gamma_{Q_j}; \xi) \quad (7.9)$$

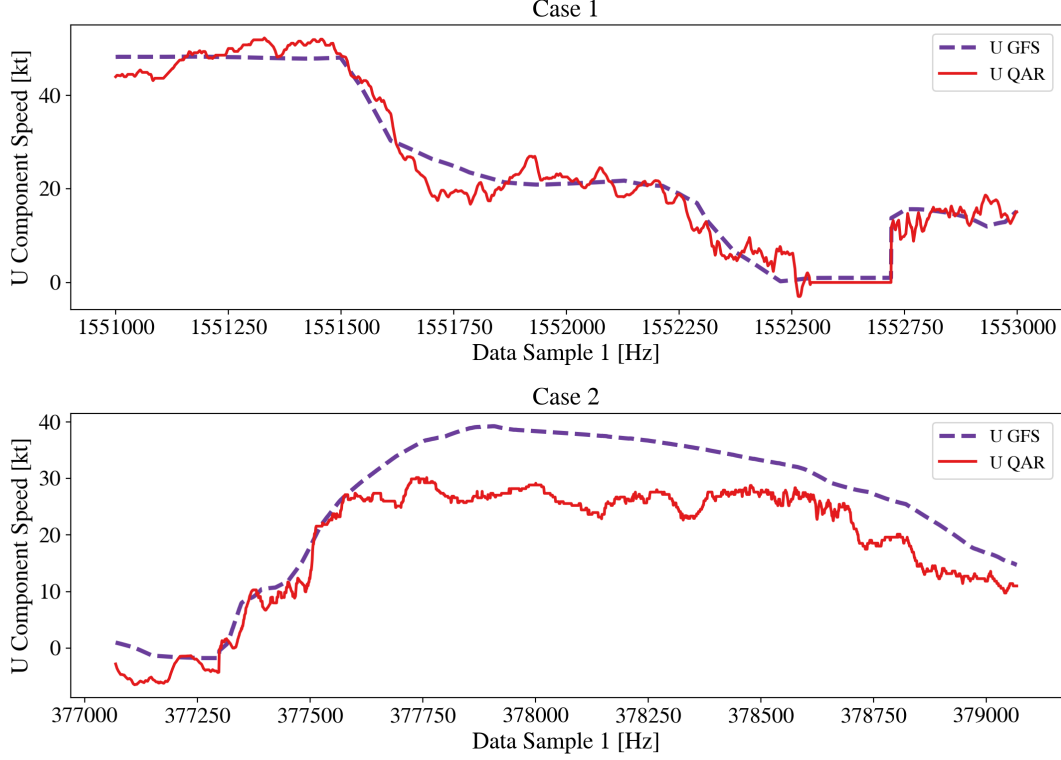


Figure 7.4 : Example of a comparison between GFS and QAR wind speed components for short haul case.

where $\Gamma_{Q_j}^*$ is the updated model for the flight mode Q_j . The flight mode set is $Q = \{climb, cruise, descent\}$. During training process, the model learns the value of ξ such that it results in the best function approximation. This supervised learning utilizes maximum likelihood estimation and optimizes the following cost function:

$$J_{Q_j}(\xi) = -\mathbb{E}_{x_{Q_j}, y_{Q_j} \sim \hat{p}_{data}} \log p_{model}(y_{Q_j} | x_{Q_j}) \quad (7.10)$$

where $p_{model}(x_{Q_j}; \xi)$ maps any configuration x to a real number estimating the true probability $p_{data}(x)$, which is obtained through actual flown trajectories. As for the wind information Ω , the following function is aimed to be obtained:

$$g : \mathbb{R}^N \rightarrow \mathbb{R}^2 \quad (7.11)$$

For the wind case, the input set is fixed and its dimension is N . It is composed of 3D location, u and v components of the wind for the last available forecast and the time difference from aircraft measurement at this location. In other words, the function g aims to capture meteorological trends and update the latest GFS forecast for a given time horizon.

$$\Omega^* = g(\Omega; \xi) \quad (7.12)$$

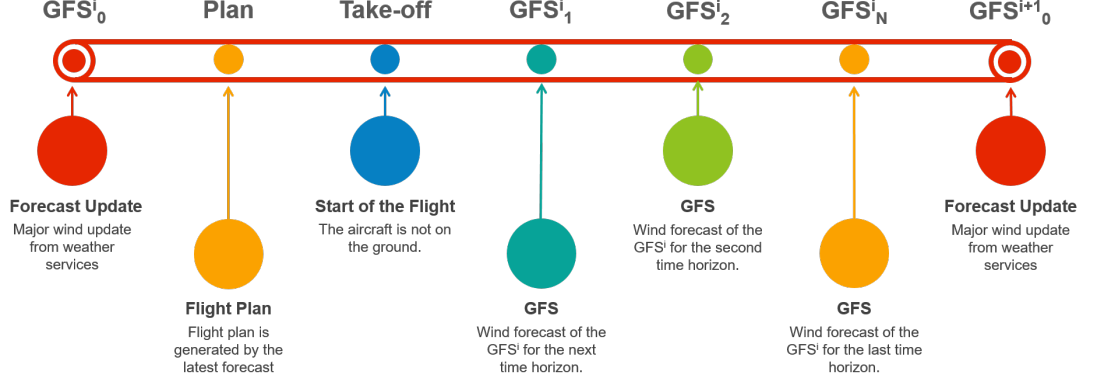


Figure 7.5 : Timeline chart for wind information during flight planning phase.

The new 4D trajectory is then predicted by the updated input set $\mathbb{V}^* = \{\mathbb{V} \setminus \{\Gamma, \Omega\}\} \cup \{\Gamma^*, \Omega^*\}$. As indicated in Figures 7.4 and 7.3, wind forecasts already capture the trend of the ground truth. The objective here is to reduce the prediction error.

7.2 Meteorological Data Used in the Study

The objective is to map the latest available wind forecast during flight planning to the ground truth for a given time horizon. In this aspect we use two datasets: GFS forecasts of National Centers for Environmental Prediction (NCEP) for the baseline wind information, wind speed and direction measurements in QAR trajectories as the actual ones or labels. We utilize historical GFS data archived in Research Data Archive of the National Center for Atmospheric Research (NCAR) in Boulder, Colorado. [115]. This forecast is updated at every 6 hours and comprised of forecasts for the next +6, +12, ..., +384 hours. We only considered the forecast for the next +6 hours of the latest GFS. As indicated in Figure 7.5, the latest forecast stands for the one just prior to the dispatching or planning time. The meteorological data utilized in this study is in *GRIB2* format, which is a specific compressing type for meteorological operations. The selected parameters from this archive are u and v components of wind and static air temperature T . For the rest of the thesis, values of these parameters are denoted by $(\cdot)_{GFS}$ for GFS data and $(\cdot)_{QAR}$ for QAR data.

Each variable is represented by a 3D matrix where u_{GFS}, v_{GFS} and $T_{GFS} \in \mathbb{R}^{M \times N \times P}$. M is the number of latitude points λ_i . Number of longitude points ϕ_i is denoted by N . In vertical space, P represents the number of pressure altitude levels. Since we consider both regional and trans-atlantic flights, the GFS data used in this

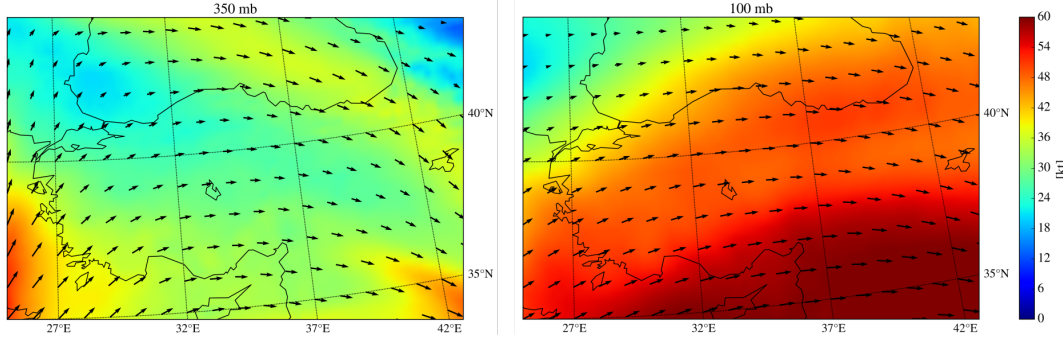


Figure 7.6 : An example of a wind map depicting wind speed and direction for different pressure levels.

study covers all regions where $\lambda = \{-90^\circ, -89.75^\circ, \dots, 0^\circ, 89.75^\circ, 90^\circ\}$ and $\phi = \{-180^\circ, -179.75^\circ, \dots, 0^\circ, 179.75^\circ, 180^\circ\}$. Vertical levels starts from isobaric surface $1000[mbar]$ and ends at $100[mbar]$. These bounds approximately correspond to $350[ft]$ and $47,000[ft]$. The resolution is $25[mbar]$ until $900[mbar]$ and $50[mbar]$ for the rest of the isobaric surface levels. For any pressure level i , u component of wind speed $u_{GFS}^{(i)}$, v component of wind speed $v_{GFS}^{(i)}$ and temperature $T_{GFS}^{(i)}$ in a GFS file are described as:

$$u_{GFS}^{(i)} = \begin{bmatrix} u_{\lambda_1, \psi_1}^{(i)} & u_{\lambda_2, \psi_1}^{(i)} & \cdots & u_{\lambda_M, \psi_1}^{(i)} \\ u_{\lambda_1, \psi_2}^{(i)} & u_{\lambda_2, \psi_2}^{(i)} & \cdots & u_{\lambda_M, \psi_2}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{\lambda_1, \psi_M}^{(i)} & u_{\lambda_2, \psi_M}^{(i)} & \cdots & u_{\lambda_M, \psi_N}^{(i)} \end{bmatrix} \quad (7.13)$$

$$v_{GFS}^{(i)} = \begin{bmatrix} v_{\lambda_1, \psi_1}^{(i)} & v_{\lambda_2, \psi_1}^{(i)} & \cdots & v_{\lambda_M, \psi_1}^{(i)} \\ v_{\lambda_1, \psi_2}^{(i)} & v_{\lambda_2, \psi_2}^{(i)} & \cdots & v_{\lambda_M, \psi_2}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\lambda_1, \psi_M}^{(i)} & v_{\lambda_2, \psi_M}^{(i)} & \cdots & v_{\lambda_M, \psi_N}^{(i)} \end{bmatrix} \quad (7.14)$$

$$T_{GFS}^{(i)} = \begin{bmatrix} T_{\lambda_1, \psi_1}^{(i)} & T_{\lambda_2, \psi_1}^{(i)} & \cdots & T_{\lambda_M, \psi_1}^{(i)} \\ T_{\lambda_1, \psi_2}^{(i)} & T_{\lambda_2, \psi_2}^{(i)} & \cdots & T_{\lambda_M, \psi_2}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{\lambda_1, \psi_M}^{(i)} & T_{\lambda_2, \psi_M}^{(i)} & \cdots & T_{\lambda_M, \psi_N}^{(i)} \end{bmatrix} \quad (7.15)$$

where u and v are in terms of $[m/s]$ and T is in $[K]$. Figure 7.6 illustrates an example of wind map for pressure levels $350[mbar]$ and $100[mbar]$. The forecast for the investigated flight is denoted as Ω_{GFS} and computed by interpolating u_{GFS} , v_{GFS} and T_{GFS} through the flown trajectory from QAR using a linear interpolation function A

such that:

$$\Omega GFS = A(\lambda_{QAR}, \psi_{QAR}, h_{QAR}, u_{GFS}, v_{GFS}, T_{GFS}) \quad (7.16)$$

The wind is recorded in a QAR trajectory in terms of speed and direction. In order to be compatible with the meteorological data, wind speed and direction are converted to u and v components via following equations:

$$u_{QAR} = -w_{s,QAR} \sin(w_{d,QAR}) \quad (7.17)$$

$$v_{QAR} = -w_{s,QAR} \cos(w_{d,QAR}) \quad (7.18)$$

where $w_{s,QAR}$ is the wind speed and $w_{d,QAR}$ is the wind direction. *Mod* function is applied because wind direction is recorded between the interval $[-180^\circ, 180^\circ]$ for some QAR flights. Additionally, we derived the forecast age, which is defined as the difference between the timestamp in which the forecast is generated and the timestamp that the wind is measured. It is denoted by η . In summary, data prepared for the machine learning process is composed of the parameters given in Table 1.

Table 7.1 : Summary of parameters to be utilized for machine learning process.

λ_{QAR}	Latitude position in QAR [<i>deg</i>]
ψ_{QAR}	Longitude position in QAR [<i>deg</i>]
h_{QAR}	Altitude in QAR [<i>ft</i>]
u_{QAR}	U component of the wind in QAR [<i>kt</i>]
v_{QAR}	V component of the wind in QAR [<i>kt</i>]
u_{GFS}	U component of the wind in GFS [<i>kt</i>]
v_{GFS}	V component of the wind in GFS [<i>kt</i>]
T_{GFS}	Static air temperature in GFS [<i>K</i>]
η	Forecast age for a given location [<i>s</i>]

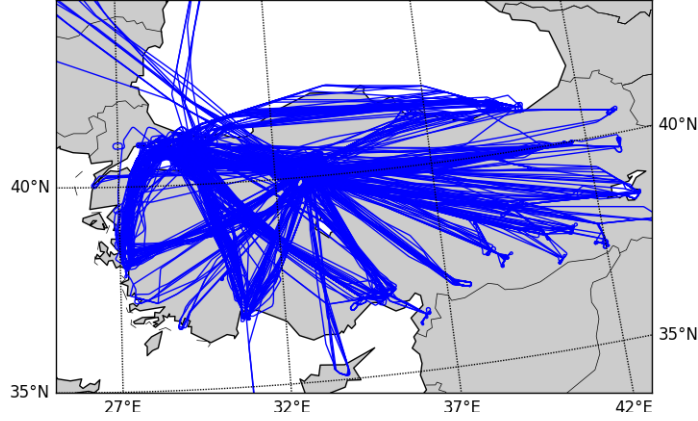


Figure 7.7 : Trajectories of short haul flights used in the study.

7.3 Machine Learning Based Wind Estimation

The total dataset is comprised of two sets including short-haul flights of a narrow-body aircraft and long-haul flights of a wide-body aircraft. The wind information in QAR trajectories of these flights are considered to be ground truth. It is assumed that the flight plans are generated 3-hours prior take-off times. Hence, the last available forecast for these timestamps are treated as the baseline wind estimations. In total, we have 1200 short-haul flights operated between 16-23 November 2017 with 4.5M data points of trajectory. QAR trajectories of long-haul flights are from 27 August 2017 to 29 December 2017 and have 33M data points. Both trajectory sets are depicted in Figures 7.7 and 7.8. As for each dataset, we spared %10 of the data as the test sets. %5 of the remaining is utilized as the validation set during training. The parameters listed in Table 1 are used as features and labels. It is aimed to find the proper function f that gives the following mapping:

$$f : \{P_{QAR}, u_{GFS}, v_{GFS}, T_{GFS}, \eta(P_{QAR})\} \rightarrow \{u_{QAR}, v_{QAR}\} \quad (7.19)$$

where $P_{QAR} = \{\lambda_{QAR}, \psi_{QAR}, h_{QAR}\}$. After training sets and test sets are generated, standard scaling operation is applied. Let $\mathbb{X}_{(\cdot),trn}$ be the training data and $\mathbb{X}_{(\cdot)}^{sc}$ denote the scaled data:

$$\mathbb{X}_{(\cdot),trn}^{sc} = \frac{\mathbb{X}_{(\cdot),trn} - \mathbb{E}[\mathbb{X}_{(\cdot),trn}]}{\sqrt{\mathbb{E}[(\mathbb{X}_{(\cdot),trn} - \mathbb{E}[\mathbb{X}_{(\cdot),trn}])^2]}} \quad (7.20)$$

Note that the scaling is done through column by column. The same scaling with the same parameters is then applied to the test set:

$$\mathbb{X}_{(\cdot),tst}^{sc} = \frac{\mathbb{X}_{(\cdot),tst} - \mathbb{E}[\mathbb{X}_{(\cdot),trn}]}{\sqrt{\mathbb{E}[(\mathbb{X}_{(\cdot),trn} - \mathbb{E}[\mathbb{X}_{(\cdot),trn}])^2]}} \quad (7.21)$$

Models are designed based on Deep Neural Network architecture. An example architecture of a Deep Neural Network with M features and N layers is illustrated in. Let l represent l^{th} layer in the network. The output of the layer is calculated as:

$$z^{[l](i)} = g(W^{[l]}x_{(\cdot),trn}^{(i),sc} + b^{[l]}) \quad (7.22)$$

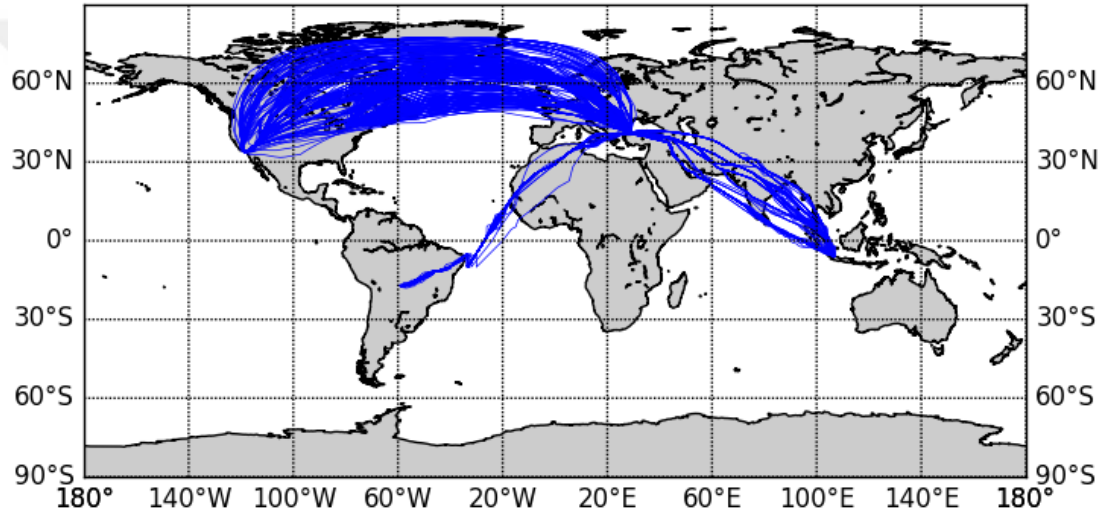


Figure 7.8 : Trajectories of long haul flights used in the study.

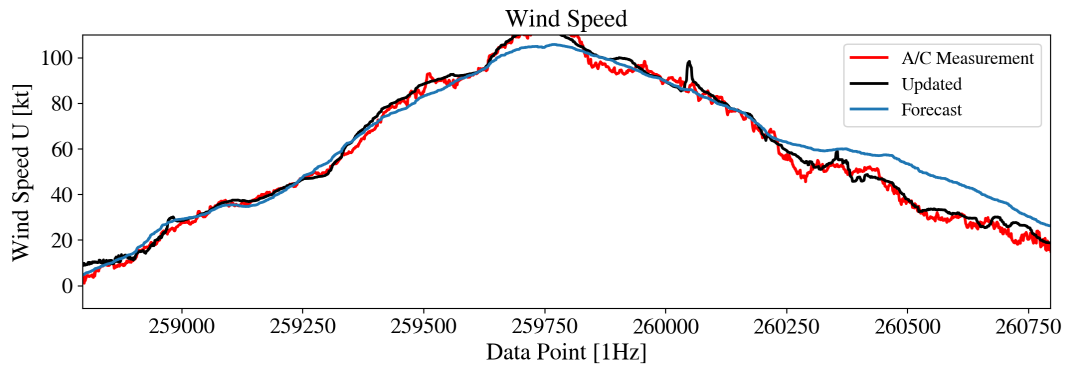


Figure 7.9 : u component of wind speed after update.

where $x_{(\cdot),trn}^{(i),sc}$ is the i^{th} training sample from $\mathbb{X}_{(\cdot),trn}^{sc}$, the weight matrix is denoted by $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, $b^{[l]} \in \mathbb{R}^{n_l}$ is the bias vector, g is the activation function and n_l is the number of neurons in the l^{th} layer. In the last layer, which is the output layer, final value

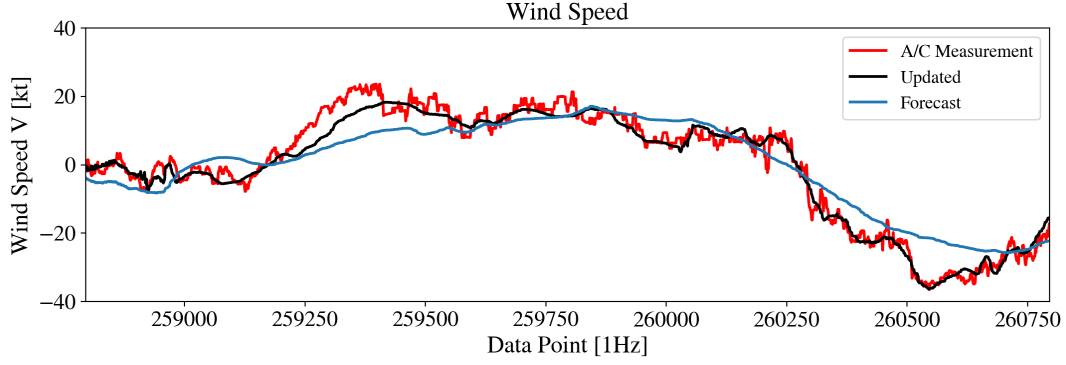


Figure 7.10 : v component of wind speed after update.

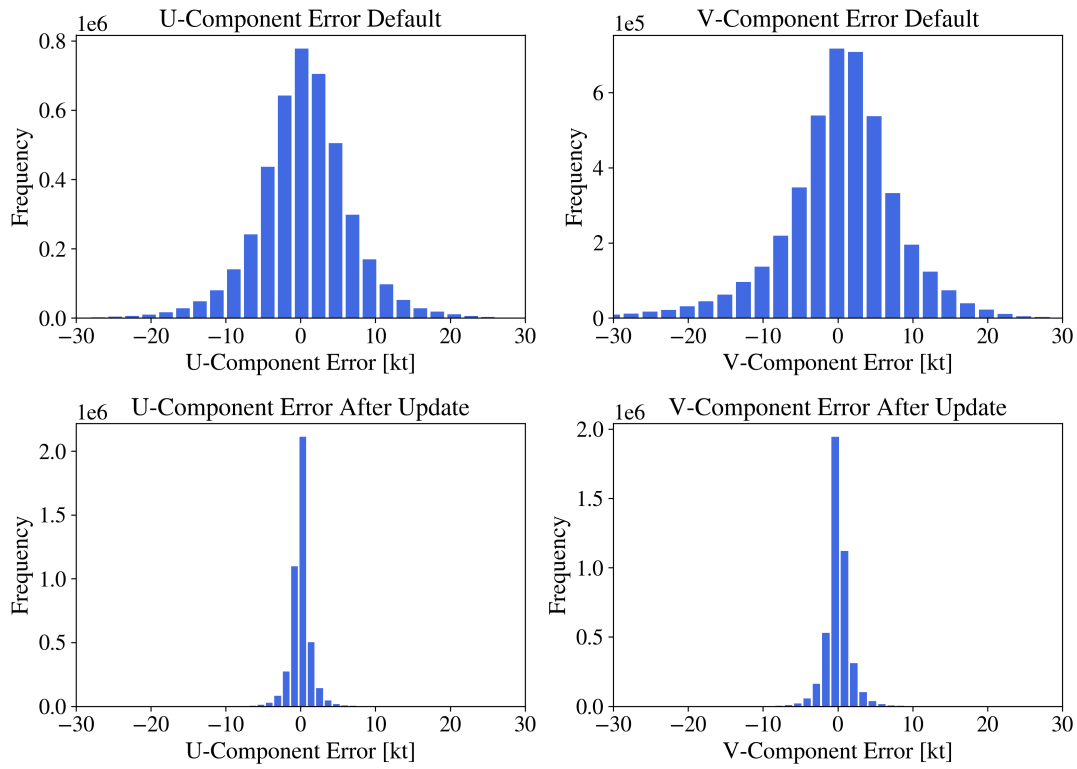


Figure 7.11 : Reduction of wind uncertainty for short haul flights.

$\hat{y}^{(i)}$ is calculated. The other common equations employed in all models and training related hyper parameters are summarized below.

RELU is selected as the activation function for all layer except the output layer:

$$g(z) = \max(0, z) \quad (7.23)$$

In order to assign initial values to weights $W^{[l]}$ at each layer l , He initialization [61] is selected.

$$W_0^{[l]} \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_l}}\right) \quad (7.24)$$

Mini-batch training [62] is used to train the deep neural networks. $\mathbb{X}_{(\cdot),trn}^{sc}$ is processed by being divided into n subsets with equal length.

$$z^{[l](k)} = g(W^{[l]}x_{(\cdot),trn}^{(k),sc} + b^{[l]}) \quad (7.25)$$

where $k = \{\frac{m}{n}(t-1) + 1, \frac{m}{n}(t-1) + 2, \dots, \frac{m}{n}t\}$ and t is the iteration number. Batch size is selected as $\frac{m}{n} = 1024$. The loss function is selected as the mean absolute error:

$$J(W, b) = \frac{1}{m} \sum_{i=1}^m |y^{(i)} - \hat{y}^{(i)}| \quad (7.26)$$

where $y^{(i)}$ is the actual output value in the data for the given input $x^{(i)}$.

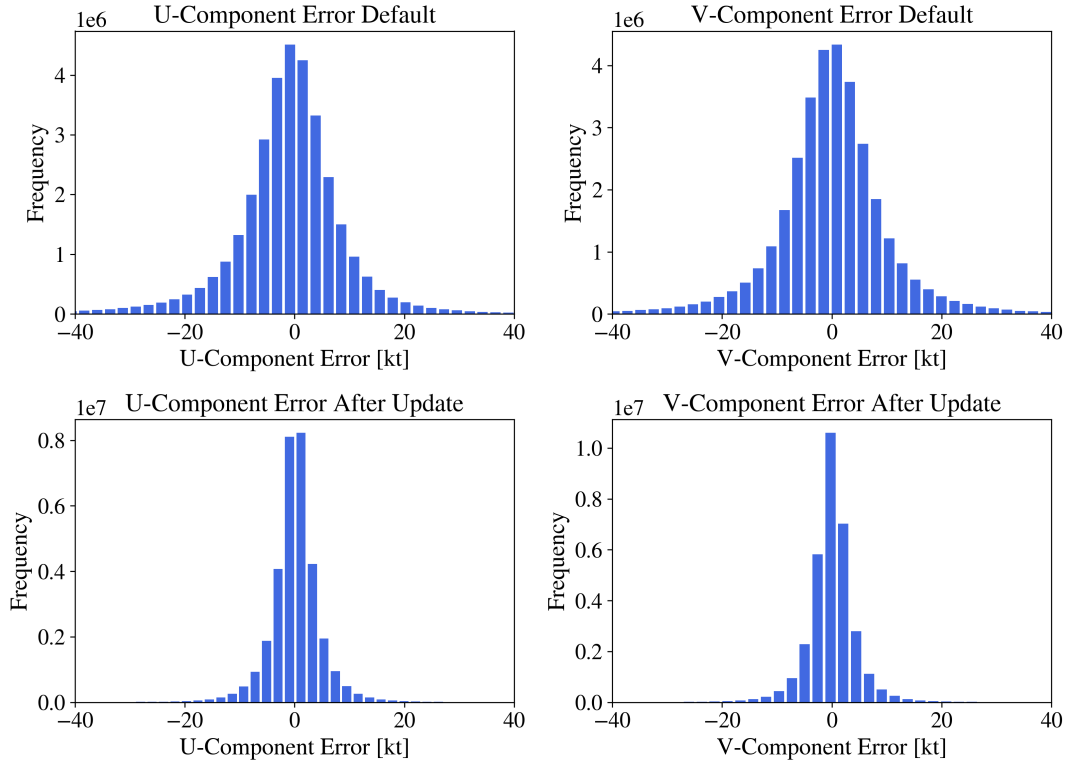


Figure 7.12 : Reduction of wind uncertainty for long haul flights.

In this process, α is the learning rate, β_1 , β_2 and $\varepsilon = 10^{-3}$ are the hyper-parameters to adjust the decay rates of $\frac{dJ}{dW}$ and $(\frac{dJ}{dW})^2$ (also for b). We tuned the learning rate α by checking the status of loss function defined in 7.26. If the loss function is not improved 10 times consecutively, α is reduced by %90. Its initial value is chosen as 0.001. All models are trained for 2000 epochs. Size of train, validation and test datasets are illustrated in Table 3. All Deep Neural Network models are implemented and trained with Tensorflow framework [68]. Hidden layer size hyper-parameter is selected to L2 ($\lambda = 0.001$) regularization methods [64] are also implemented and used separately.

Figures 7.11 and 7.12 show that we can achieve considerable reduction in wind uncertainty both from a mean error and variance sense. Specifically, for short-haul flights, the standard deviations of forecast errors in u and v components are reduced from 6.25 and 8.38 knots to 1.37 and 1.81 knots, respectively. The same reduction is from 11.02 and 10.89 knots to 4.88 and 4.76 knots in the long haul flights. In addition, the expected values of errors in u and v components are reduced an order of magnitude for both long-haul and short-haul flights. Besides, Figures 7.9 and 7.10 compare wind speeds before and after update for a randomly selected long haul trajectory.

7.4 Conclusion

In this chapter, we have applied deep neural networks for proper estimation of u and v component of wind speed, for given latitude, longitude, altitude using wind information from the latest available GFS prediction including the age of this forecast. Wind speed and direction recorded in QAR trajectories are used as the truth data for training deep neural networks for wind estimation. Towards this goal, we have utilized two data sets from a major European Flag carrier. First set is 1 week of domestic flights in Eurasia region with 4M wind measurements. The second set is 4 months of transatlantic flights with 33M wind measurements. The rationale behind such division was to investigate whether it is possible to capture the meteorological trends for both small region (i.e. short time horizon (dense data)) and large region (i.e. long time horizon (sparse data)). Results show significant amount of reduction in wind uncertainty.

Updating baseline estimations of key factors such as aircraft performance and wind models be enabler for more efficient flights. In this sense, trip time and trip fuel of an aircraft could be more accurately predicted. Such achievement not only provides proper estimation of take-off weight but also eliminates the redundant fuel load, which results in less fuel burn by not causing aircraft to carry it.

8. FLIGHT PLANNING APPLICATIONS

To understand both fuel usage and the associated economic effect of updating the baseline performance model, we consider actual filed flight plans. In this context, we emulate a flight planning process with two different aircraft performance settings: First, we calculate the trip fuel using the baseline model for a given reference trajectory. Second, we replicate the process with the updated model. As for the flight plans, we use dispatched filed plans by a major European Airline Integrated Operation Center. An example snapshot from a dispatched flight plan is shown in Fig. 8.1. An operational flight plan consists of a set of waypoints describing the flight route. This trajectory is an abstraction of the trajectory to be flown. The waypoints listed are top of climb/descend locations, entries/exits of airspaces, specific geographic locations, or part of sequences of departure/arrival procedures. These data are loaded into the Flight Management Computer (FMC) of the aircraft to be executed. Figure 8.2 shows an image of an FMC listing these planned waypoints, as obtained from an operating Boeing 777-300ER.

OPERATIONAL FLIGHT PLAN PAGE 4/11 RLSD 12APR18 0609.48Z										
AWY MOCA	WPT NAME/FIR LAT/LONG	FRQ FL TRO SHR	MT TT VAR	WIND SAT TDV	TAS MN G/S	DIST REMD ACCD	TIME ACCT REMT	ETA ATA REV	RQRD ACCF FOB	
UL602 040	FIR EDVV HANNOVER UIR FIR N51200E009076	320 347 02	299 302 03E	172/058 M53 M5	478 .827 511	24 3453 1076	3 0218 0719		62877 23919	
UL602 038	FIR EDGG LANGEN FIR N51265E008505	320 329 05	299 301 02E	168/076 M54 M6	478 .829 514	13 3440 1089	2 0220 0717		62674 24122	
UL602 040	HMM D115.65 HAMM N51514E007425	320 329 05	299 301 02E	168/075 M54 M6	478 .829 525	49 3391 1138	5 0225 0712		61910 24886	
UL602 021	REBGU N52044E006582	320 328 01	295 296 01E	136/041 M54 M6	478 .829 516	30 3361 1168	4 0229 0708		61436 25360	
UL602 020	RELBI EHAA AMSTERDAM FIR N52071E006488	320 328 01	293 295 02E	136/041 M54 M6	478 .829 515	6 3355 1174	0 0229 0708		61341 25455	

Figure 8.1 : An example of trajectory in a dispatched flight plan.

At each waypoint, the ground-based model calculates the optimal speed for the given altitude and mass, usually utilizing the performance models provided by the aircraft manufacturers. A performance update is achieved using the performance factor, which is also noted in a flight plan and an example is illustrated in Fig. 8.2. In Fig. 8.1,

DISPATCH LOAD:			
		PAYLOAD:	50446
EZFW:	224015	MZFW: (S)	237682
ETOW:	310010	MTOW: (S)	351534
ELDW:	233386	MLDW: (S)	251290
REMF:	9371	MIN DIV:	6752
FMS INIT LOAD:			
LTBA/KJFK			
LDG ELEV:0013FT		PRF FACTOR%:2.9	
CI: 46		TROPQ:40775	



Figure 8.2 : Performance factor (left) and planned waypoints in a Flight Management Computer (right).

the flight altitudes, the speeds, and remaining fuel with consumed fuel are denoted with blue, green, and red boxes, respectively. Based on the fuel flow model, fuel consumption is calculated using these values, and then, the model integrates it with the next waypoint. We follow the same process but with the fuel flow model developed in this study.

Result: Trajectory of aircraft mass through the proposed algorithm

$$x_p = \Gamma(x'_p)$$

$$\tau_{q_j} \leftarrow \text{train}(\mathbf{X}^{q_j}, \mathcal{B})$$

$$\tau_{q_j} \leftarrow \{\tau_{q_j}^{cmb}, \tau_{q_j}^{crz}, \tau_{q_j}^{des}\}$$

$$m^{[0]} \leftarrow m_p^{[0]}, m_p \in x_p$$

$$\text{Index } k \leftarrow 0, \text{ Traveled distance } d \leftarrow 0, \text{ Unit distance of } 1 \text{ [nm]} \delta x \leftarrow 1$$

while $d < d_f$ **do**

$$\delta t \leftarrow \delta x / V_{gs}^{[k]}, V_{gs} \in x_p$$

$$f_m \leftarrow \text{FlightMode}()$$

$$F_{\mathcal{B}} \leftarrow \mathcal{B}^{f_m}(x_p^{[k]})$$

$$F_{\mathcal{B}}^* \leftarrow \tau_{q_j}^{f_m}(x_p^{[k]}) F_{\mathcal{B}}$$

$$m^{[k+1]} \leftarrow m^{[k]} - F_{\mathcal{B}}^* \delta t$$

$$d \leftarrow d + \delta x, k \leftarrow k + 1$$

end

Algorithm 1: Re-calculating fuel consumption for a given flight plan.

Algorithm 1 presents the process of computing aircraft mass using the proposed fuel flow estimation. Here, a flight plan is denoted by x_p . To convert flight plan data to QAR-like form, we first create an array of traveled distance with 1 nm intervals, $d = \{d^{[0]}, d^{[0]} + \delta x, d^{[1]} + \delta x, \dots, d_f\}$, where $d^{[0]} = 0$ and d_f is the distance planned to be traveled. Then, we generate a linear interpolation function Γ that computes altitude, true airspeed, ground speed, temperature, and Mach for each $d^{[k]}$. The algorithm starts with the same initial mass m_0 as the flight plan does. Then, at each step, the mass is reduced by the fuel flow $F_{\mathcal{B}}^*$, which is the corrected one. The $\text{FlightMode}()$ function

Table 8.1 : Flight plan analysis showing the mean and max fuel loading differences for two cases of under usage of fuel and extra fuel usage.

	Count	Max difference [kg]	Mean difference [kg]
Under burn	163	3738	1039
Extra burn	75	3085	854

returns the flight mode f_m that the aircraft is in, by checking the gradient of altitude over the traveled distance. Based on that, the corresponding model $\tau_{qj}^{f_m}$ for climb, cruise, or descent is selected to correct fuel flow.

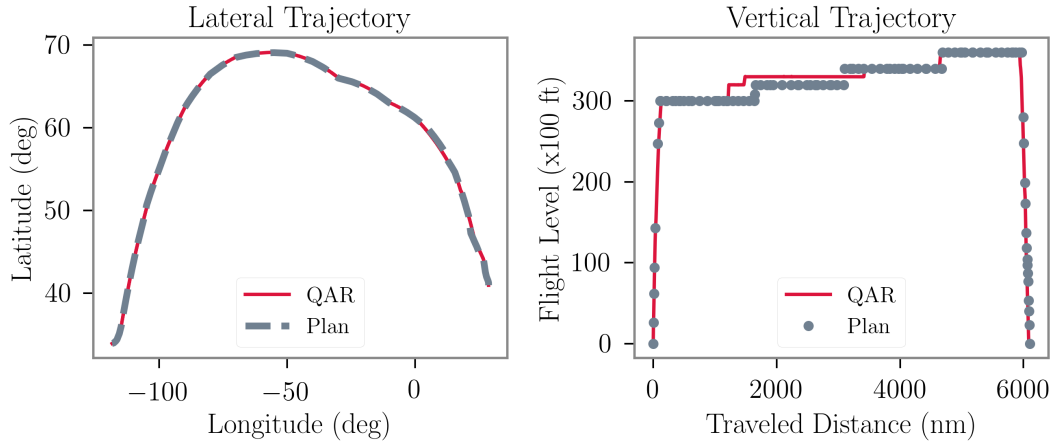


Figure 8.3 : Comparison between the planned and the actual trajectories. Existing ground based flight planning tools estimate that the fuel burn is 104,782 kg, where the actual flight reveals a fuel burn of 105,872 kg.

The selected unique-tail aircraft has 88 long-haul flights in the QAR dataset and 238 plans of both short and long-distance flights. We built a digital-twin performance update model through the QAR flights and use this tail-number specific model to re-calculate the total fuel burn for the flight plans. The fuel price is selected as on January 2020 from the IATA fuel price monitor [3] as 82.32 [\$/bbl], which equals 0.65 [\$/kg].

Figure 8.4 illustrates the level of fuel loading difference versus the number of flights, and Table 8.1 summarizes two distinct classes of fuel usage difference. In Table 8.1, the first class corresponds to the flights for which the proposed algorithm calculated less fuel burn than the planing software suggested. The second class corresponds to the cases in which the proposed model calculated a higher amount of fuel to execute the filed flight plans in comparison to the ground planning tool. These two classes are treated separately. The first class indicates potential savings, where a properly

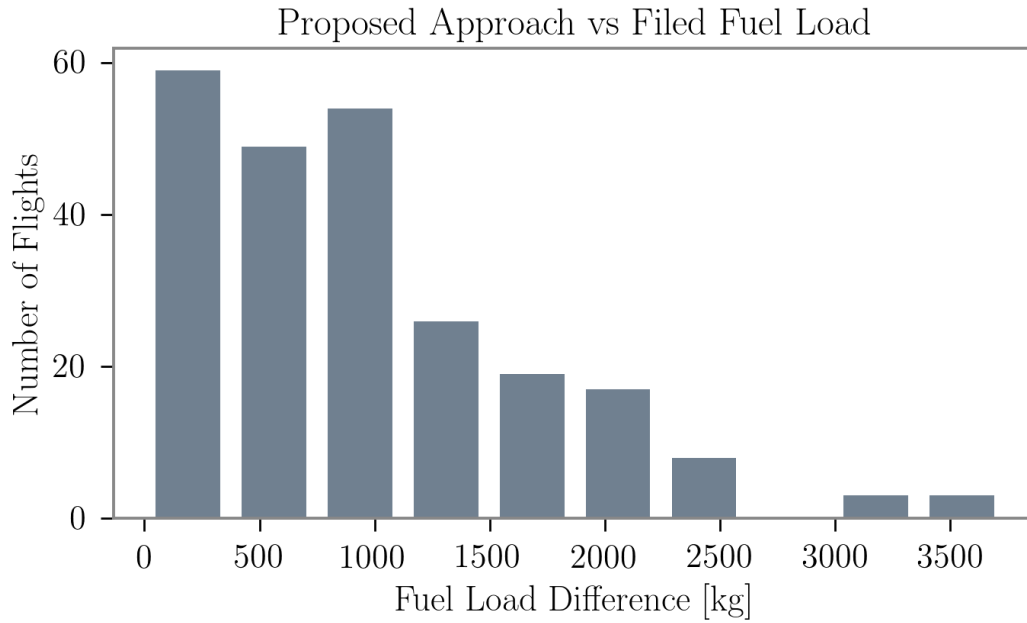


Figure 8.4 : Distribution of fuel load differences.

updated model prevents unnecessary excess fuel loading. This case is observed in 163 cases. Calculated fuel loading differences as much as 3738 kg (i.e., 3.5% of total fuel loading) were observed. Comparison of the planned and estimated fuel boarding weights (following the actual filed flight plans) for these cases show that the mean difference is 1039 [kg]. Following the industry practice of 1 : 3 ratio (i.e., burning one unit of fuel for every three units of excess fuel carried on long-haul flights), this corresponds to 235 \$ per flight) that can be saved by the proposed methodology at the ground planning phase. For a typical operation of 200 long-haul flights per day, this represents yearly savings on the order of 17 million USD at current jet fuel prices.

Our analysis also shows that in 75 flights, the flight plan tool that the airline uses actually underestimated total fuel consumption. In these flights, our approach calculated that on average an additional 854 kg of fuel per flight is required for the actual filed flight plan. This case is directly observed when we compare the planned fuel usage vs. the fuel usage of the actual flown trajectories. Figure 8.3 illustrates such an example in which we have an almost one-to-one horizontal and vertical trajectory match. In this particular flight, there is a portion of 1000 nm distance in which the aircraft had a level flight at 33,000 ft, where the planned altitude was 32,000 ft. Even though the aircraft was expected to burn less fuel (as it had flown at a higher altitude),

the actual fuel burn and our estimate is around 105,872 kg, where as the planned fuel consumption is 104,872 kg. The underestimation of fuel flow naturally results in consuming fuel from the safety margins and the fuel reserves.

This “tail-number specific” performance modeling and digital twin approach is projected to open considerable frontiers, including further reduction of fuel load safety margins and in-flight update of performance surrogate models through machine learning methods.





9. CONCLUSION

This thesis have applied state-of-the-art deep learning algorithms to digital-twin aircraft performance modeling, which is structured around neural network-based correction factors to the hybrid system formalism that follows the model description that aircraft manufacturers have generated.

In the first part, different feature and architecture combinations are considered for proper fuel flow estimation. First, the developed black-box neural network performance model with all features from the QAR dataset is observed to provide a good approximation of the actual fuel flow as the QAR data set includes throttle positions of all of the aircraft engines. A comparison of this model with real flight data shows that precise estimation of fuel flow with mean errors lower than %1 can be achieved in in-flight use cases. However, the throttle positions are not available on generic ground-based flight planning tools which rely on basic kinematic formulations and performance look-up tables. In addition, the conventional assumption of using specific throttle settings or nominal values at each of the flight modes have been shown to provide considerably different fuel flow characteristics with respect to the actual flown operation. To solve this problem, we have developed two additional different neural network models for the ground planning use case.

The second designed architecture builds a δ_T parameter estimation neural network architecture using physically consistent data regeneration using BADA model definition. This model has provided precise fuel flow estimation for climb, cruise and descent phases with mean error lower than %0.7 of the actual fuel flow. In the third designed architecture, we have designed a cascade system where throttle positions are estimated using a neural network architecture, which again uses a physically consistent data regeneration. This throttle estimation is then cascaded into the original black-box fuel flow model. The analysis has shown that this specific model provides a precise estimation of cruise phase fuel flow with mean error lower than %0.3 of the actual

fuel flow. Initial tests reflect the fact that even better accuracy can be achieved for all models as the data set size increases.

In the second part of the thesis, a data analytic digital twin is proposed to update baseline aircraft performance models with operational flight data. The methodology relies on deep learning algorithms to train a correction factor to a BADA4 based flight performance fuel consumption model. Extensive feature engineering is provided to show parameters that have a significant effect on the discrepancy between the baseline model and the data. We presented both the statistical and machine learning-based feature importance analysis to determine the correlation of variables with fuel flow and correction factor. This analysis illustrated that errors are caused by using biased polynomial coefficients and not fully covering the operational envelope of the aircraft, and thus, the engine variables. One of key aspects of our neural network design is the fact that we considered features compatible with flight planning operations and the data available. In this sense, we proposed a neural network architecture that improves the accuracy of the correction factor by providing an estimation of unavailable parameters such as throttle; further. it does this in a single neural network model. The results show a significant amount of reduction in the prediction error of fuel flow concerning the baseline model.

In that respect, tail-number specific surrogate models provides a key enabling technology towards efficient flight planning across the entire flight envelope. However, the main challenge of using machine learning methods to develop aircraft performance is that they result in poor estimations outside the flight envelope for which they are trained in. In that sense, extensive flight data is needed for the models to predict performance over the entire flight envelope. This is not something available through daily operational data, which are restricted to limited flight regimes. As such, extensive data from maintenance logs (such as engine overhauls) would be needed to further quantify performance deterioration, and tie it to through-life factors other than age of the aircraft. In addition to the above critical points and as part of on-going work, the authors are extending the methodology to other aircraft types. Further optimization of the digital twin architectures with different neural-network configurations and hyper-parameters is an ongoing effort.

The third part of this thesis aimed to improve model's physical consistency for the flight envelope that is not covered by data. In particular, we have focused on the cruise segment of flight, which is the critical segment for fuel efficient flight planning in long-haul flights. The results show that the standard machine learning algorithms as developed within the literature, although providing high precision input-output relationships within the given flight envelope of the data, fail to capture the fundamental physical principles of fuel consumption once utilized over the whole flight envelope. Thus the applicability of such neural network models for flight planning over the whole flight envelope is questionable. As to solve this critical issue, we have designed a novel physics guided deep learning method to capture not only the nonlinear relationships between the key variables within the flight data, but also the physics of flight and fuel consumption as denoted by model-based approaches. Our proposed method relies on the introduction of learning loss functions which embed the underlying physical principles and aircraft constraints. The resulting neural network structures are shown to produce high precision fuel consumption models within the flown flight regimes and physically consistent solutions across the whole flight envelope. It is important to note that without the availability of data in unforeseen flight regimes, it is impossible to fully quantify the precision of the model over the whole flight envelope. However, our results show that our deep learning model produces fuel consumption predictions which are inline with the BADA4 calculations in unseen flight regimes. Thus, including key physical principles in the training/learning phase of purely data-driven models increases accuracy, explainability and generalization capability of the developed models. To the best of our knowledge, this is the first work that successfully designs and demonstrates a physics-guided deep learning framework in fuel consumption modeling for an aircraft.

The most challenging part of this physics-guided modelling is selecting the loss terms' weights. We envision seeking methods such as Bayesian optimization to standardize this procedure. Additionally, even though our model yields physically coherent outputs, accuracy on unlabeled flight regimes is still an open issue. One possible approach could be providing initial guesses using existing performance models, but it would diminish the independence. In conclusion, current work focuses on the effects of using particular deep learning architectures on the fuel flow estimation accuracy.

Specifically, we investigate the neural architecture search algorithms' performance for this problem. As such, novel deep learning models considering the input feature interactions more effectively or further considering autoencoder-based deep embedding models are envisioned to improve fuel consumption estimation success.

Apart from aircraft performance, we have applied deep neural networks for proper estimation of u and v component of wind speed, for given latitude, longitude, altitude using wind information from the latest available GFS prediction including the age of this forecast. Wind speed and direction recorded in QAR trajectories are used as the truth data for training deep neural networks for wind estimation. Towards this goal, we have utilized two data sets from a major European Flag carrier. First set is 1 week of domestic flights in Eurasia region with 4M wind measurements. The second set is 4 months of transatlantic flights with 33M wind measurements. The rationale behind such division was to investigate whether it is possible to capture the meteorological trends for both small region (i.e. short time horizon (dense data)) and large region (i.e. long time horizon (sparse data)). Results show significant amount of reduction in wind uncertainty. Updating baseline estimations of key factors such as aircraft performance and wind models be enabler for more efficient flights. In this sense, trip time and trip fuel of an aircraft could be more accurately predicted. Such achievement not only provides proper estimation of take-off weight but also eliminates the redundant fuel load, which results in less fuel burn by not causing aircraft to carry it.

Finally, the developed digital-twin models are also applied to actual filed flight plans to illustrate potential fuel savings. Analysis for 91 actual long-haul flights illustrates that fuel savings of around 800[kg] per flight could have been achieved by the proposed methodology at the ground planning phase. For a typical operation of 100 long-haul flights per day, this represents yearly savings on the order of around 17 million USD at current jet fuel prices. This "tail-number specific" performance modeling approach is projected to open considerable frontiers, including the in-flight update of performance models through machine learning methods.

REFERENCES

- [1] **Baspinar, B., Koyuncu, E. and Inalhan, G.** (2017). Large scale data-driven delay distribution models of european air traffic flow network, *Transportation Research Procedia*, 22, 499–508.
- [2] **Sgouridis, S., Bonnefoy, P.A. and Hansman, R.J.** (2011). Air transportation in a carbon constrained world: Long-term dynamics of policies and strategies for mitigating the carbon footprint of commercial aviation, *Transportation Research Part A: Policy and Practice*, 45(10), 1077–1091.
- [3] **IATA** Jet Fuel Price Monitor, <https://www.iata.org/en/publications/economics/fuel-monitor/>, accessed: 2020-05-10.
- [4] **ICAO** Aircraft Engine Emissions Databank, <https://www.easa.europa.eu/easa-and-you/environment/icao-aircraft-engine-emissions-databank>, accessed: 2020-05-10.
- [5] **Nuic, A., Poinso, C., Iagaru, M.G., Gallo, E., Navarro, F.A. and Querejeta, C.** (2005). Advanced aircraft performance modeling for ATM: Enhancements to the BADA model, *24th Digital Avionics System Conference*, AIAA/IEEE Washington, DC, pp.1–14.
- [6] **Mouillet, V., Nuić, A., Casado, E. and Leonés, J.L.** (2018). Evaluation of the Applicability of a Modern Aircraft Performance Model to Trajectory Optimization, *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, pp.1–9.
- [7] **Gardi, A., Sabatini, R. and Kistan, T.** (2018). Multiobjective 4D trajectory optimization for integrated avionics and air traffic management systems, *IEEE Transactions on Aerospace and Electronic Systems*, 55(1), 170–181.
- [8] **Gallo, E., Navarro, F.A., Nuic, A. and Iagaru, M.** (2006). Advanced aircraft performance modeling for ATM: BADA 4.0 results, *2006 IEEE/AIAA 25TH Digital Avionics Systems Conference*, IEEE, pp.1–12.
- [9] **Rodrigues, L.** (2017). A unified optimal control approach for maximum endurance and maximum range, *IEEE Transactions on Aerospace and Electronic Systems*, 54(1), 385–391.
- [10] **Brown, A. and Anderson, D.** (2019). Trajectory optimization for high-altitude long endurance UAV maritime radar surveillance, *IEEE Transactions on Aerospace and Electronic Systems*.

- [11] **Uzun, M., Demirezen, M.U., Koyuncu, E. and Inalhan, G.** (2019). Design of a Hybrid Digital-Twin Flight Performance Model Through Machine Learning, *2019 IEEE Aerospace Conference*, IEEE, pp.1–14.
- [12] **Ghazi, G. and Botez, R.M.** (2019). Identification and validation of an engine performance database model for the flight management system, *Journal of Aerospace Information Systems*, 16(8), 307–326.
- [13] **Ghazi, G., Botez, R.M. and Domanti, S.** (2020). New Methodology for Aircraft Performance Model Identification for Flight Management System Applications, *Journal of Aerospace Information Systems*, 1–17.
- [14] **Sun, J., Ellerbroek, J. and Hoekstra, J.M.** (2019). WRAP: An open-source kinematic aircraft performance model, *Transportation Research Part C: Emerging Technologies*, 98, 118–138.
- [15] **Sun, J., Ellerbroek, J. and Hoekstra, J.** (2017). Modeling aircraft performance parameters with open ads-b data, *12th USA/Europe Air Traffic Management Research and Development Seminar*.
- [16] **Metz, I., Hoekstra, J., Ellerbroek, J. and Kügler, D.** (2016). Aircraft performance for open air traffic simulations, *AIAA Modeling and Simulation Technologies Conference*, p.3522.
- [17] **Hrastovec, M. and Solina, F.** (2016). Prediction of aircraft performances based on data collected by air traffic control centers, *Transportation Research Part C: Emerging Technologies*, 73, 167–182.
- [18] **Nuic, A.** (2010). User manual for the Base of Aircraft Data (BADA) revision 3.10, *Atmosphere*, 2010, 001.
- [19] **Todd, J.C.**, (2002), Flight data recorder system, uS Patent 6,397,128.
- [20] **Wang, L., Wu, C. and Sun, R.** (2014). An analysis of flight Quick Access Recorder (QAR) data and its applications in preventing landing incidents, *Reliability Engineering & System Safety*, 127, 86–96.
- [21] **Sembiring, J., Drees, L. and Holzapfel, F.** (2013). Extracting unmeasured parameters based on quick access recorder data using parameter-estimation method, *AIAA Atmospheric Flight Mechanics (AFM) Conference*, p.4848.
- [22] **Luo, H. and Zhong, S.** (2017). Gas turbine engine gas path anomaly detection using deep learning with Gaussian distribution, *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, IEEE, pp.1–6.
- [23] **Chati, Y.S. and Balakrishnan, H.** (2013). Aircraft engine performance study using flight data recorder archives, *2013 aviation technology, integration, and operations conference*, p.4414.
- [24] **Chati, Y.S. and Balakrishnan, H.** (2016). Statistical modeling of aircraft engine fuel flow rate, *30th Congress of the International Council of the Aeronautical Science*.

- [25] **Chati, Y.S. and Balakrishnan, H.** (2017). A Gaussian process regression approach to model aircraft engine fuel flow rate, *2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, pp.131–140.
- [26] **Chati, Y.S. and Balakrishnan, H.** (2018). Data-driven modeling of aircraft engine fuel burn in climb out and approach, *Transportation Research Record*, 2672(29), 1–11.
- [27] **Baklacioglu, T.** (2016). Modeling the fuel flow-rate of transport aircraft during flight phases using genetic algorithm-optimized neural networks, *Aerospace Science and Technology*, 49, 52–62.
- [28] **Baumann, S. and Klingauf, U.** (2020). Modeling of aircraft fuel consumption using machine learning algorithms, *CEAS Aeronautical Journal*, 11(1), 277–287.
- [29] **Huang, C., Xu, Y. and Johnson, M.E.** (2017). Statistical modeling of the fuel flow rate of GA piston engine aircraft using flight operational data, *Transportation Research Part D: Transport and Environment*, 53, 50–62.
- [30] **Khadilkar, H. and Balakrishnan, H.** (2012). Estimation of aircraft taxi fuel burn using flight data recorder archives, *Transportation Research Part D: Transport and Environment*, 17(7), 532–537.
- [31] **Directorate, A.C.S.** (2002). Getting to Grips with Aircraft Performance Monitoring.
- [32] **Maggiore, J.B. and Kinney, D.S.** (2009). Monitoring Real-Time Environmental Performance, *Boeing Aero Magazine*, (Q309).
- [33] **Anderson, D.** (2008). Cruise Performance Monitoring, *Boeing Aero Magazine*, (Q406).
- [34] **Ghazi, G., Gerardin, B., Gelhaye, M. and Botez, R.M.** (2019). New Adaptive Algorithm Development for Monitoring Aircraft Performance and Improving Flight Management System Predictions, *Journal of Aerospace Information Systems*, 1–16.
- [35] **Puranik, T.G., Harrison, E., Min, S., Chakraborty, I. and Mavris, D.N.** (2018). A framework for general aviation aircraft performance model calibration and validation, *2018 Aviation Technology, Integration, and Operations Conference*, p.3191.
- [36] **Roof, C., Hansen, A., Fleming, G., Thrasher, T., Nguyen, A., Hall, C., Dinges, E., Grandi, F., Kim, B., Usdrowski, S. et al.** (2007). Aviation environmental design tool (AEDT) system architecture, **Technical Report**, United States. Federal Aviation Administration. Office of Environment and Energy.
- [37] **Clemons, E., Reynolds, T., Badrinath, S., Chati, Y. and Balakrishnan, H.** (2018). Enhancing Aircraft Fuel Burn Modeling on the Airport Surface, *2018 Aviation Technology, Integration, and Operations Conference*, p.3991.

- [38] **Nuic, A., Poles, D. and Mouillet, V.** (2010). BADA: An advanced aircraft performance model for present and future ATM systems, *International journal of adaptive control and signal processing*, 24(10), 850–866.
- [39] **Koyuncu, E., Uzun, M. and Inalhan, G.** (2016). Cross-entropy-based cost-efficient 4D trajectory generation for airborne conflict resolution, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(9), 1605–1631.
- [40] (2002). Getting to grips with aircraft performance, **Technical Report**, Airbus customer services.
- [41] **Yechout, T.R.,** (2003). Introduction to aircraft flight mechanics, Aiaa, pp.68–75.
- [42] **Center, E.E.** (2013). User manual for the base of aircraft data (BADA) revision 3.11, *Brussels: Eurocontrol*.
- [43] **Blake, W.** (2009). *Jet Transport Performance Methods*, Boeing Commercial Airplanes.
- [44] **Winkler, D.A. and Le, T.C.** (2017). Performance of deep and shallow neural networks, the universal approximation theorem, activity cliffs, and QSAR, *Molecular informatics*, 36(1-2), 1600118.
- [45] **Chen, T. and Chen, H.** (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Transactions on Neural Networks*, 6(4), 911–917.
- [46] **Rumelhart, D.E., Hinton, G.E. and Williams, R.J.** (1986). Learning representations by back-propagating errors, *Nature*, 323(6088), 533–536, <https://doi.org/10.1038/323533a0>.
- [47] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.** (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, 15(56), 1929–1958, <http://jmlr.org/papers/v15/srivastava14a.html>.
- [48] **Kingma, D.P. and Ba, J.** (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.
- [49] **Zaheer, M., Reddi, S., Sachan, D., Kale, S. and Kumar, S.,** (2018). Adaptive Methods for Nonconvex Optimization, *Advances in Neural Information Processing Systems 31*, Curran Associates, Inc., pp.9793–9803.
- [50] **Ryerson, M.S. and Hansen, M.** (2013). Capturing the impact of fuel price on jet aircraft operating costs with Leontief technology and econometric models, *Transportation Research Part C: Emerging Technologies*, 33, 282–296.
- [51] **McConnachie, D., Wollersheim, C. and Hansman, R.J.** (2013). The impact of fuel price on airline fuel efficiency and operations, *2013 Aviation Technology, Integration, and Operations Conference*, p.4291.

- [52] **Morrison, J., Bonnefoy, P., Hansman, R.J. and Sgouridis, S.** (2010). Investigation of the impacts of effective fuel cost increase on the us air transportation network and fleet, *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, p.9202.
- [53] **Wasiuk, D., Lowenberg, M. and Shallcross, D.** (2015). An aircraft performance model implementation for the estimation of global and regional commercial aviation fuel burn and emissions, *Transportation Research Part D: Transport and Environment*, 35, 142–159.
- [54] **Chati, Y.S. and Balakrishnan, H.** (2014). Analysis of aircraft fuel burn and emissions in the landing and take off cycle using operational data, *International Conference on Research in Air Transportation*.
- [55] **Lovegren, J. and Hansman, R.** (2011). Estimation of potential aircraft fuel burn reduction in cruise via speed and altitude optimization strategies, **Technical Report**.
- [56] **Jensen, L., Hansman, R.J., Venuti, J.C. and Reynolds, T.** (2013). Commercial airline speed optimization strategies for reduced cruise fuel consumption, *2013 Aviation Technology, Integration, and Operations Conference*, p.4289.
- [57] **ICAO, I.** (2006). Aircraft Engine Emissions Databank, *International Civil Aviation Organization*.
- [58] **Nikoleris, T., Gupta, G. and Kistler, M.** (2011). Detailed estimation of fuel consumption and emissions during aircraft taxi operations at Dallas/Fort Worth International Airport, *Transportation Research Part D: Transport and Environment*, 16(4), 302–308.
- [59] **Gallo, E., Lopez-Leones, J., Vilaplana, M.A., Navarro, F.A. and Nuic, A.** (2007). Trajectory computation infrastructure based on BADA aircraft performance model, *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, IEEE, pp.1–C.
- [60] **Trani, A., Wing-Ho, F., Schilling, G., Baik, H. and Seshadri, A.** (2004). A neural network model to estimate aircraft fuel consumption, *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, p.6401.
- [61] **He, K., Zhang, X., Ren, S. and Sun, J.** (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *Proceedings of the IEEE international conference on computer vision*, pp.1026–1034.
- [62] **Ioffe, S. and Szegedy, C.** (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167*.
- [63] **Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C. and Zhang, Z.** (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, *arXiv preprint arXiv:1512.01274*.

- [64] **Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.** (2014). Dropout: a simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, 15(1), 1929–1958.
- [65] **Leonard, J. and Kramer, M.** (1990). Improvement of the backpropagation algorithm for training neural networks, *Computers & Chemical Engineering*, 14(3), 337–341.
- [66] **Kursa, M.B., Rudnicki, W.R. et al.** (2010). Feature selection with the Boruta package, *J Stat Softw*, 36(11), 1–13.
- [67] **Zaheer, M., Reddi, S., Sachan, D., Kale, S. and Kumar, S.** (2018). Adaptive methods for nonconvex optimization, *Advances in neural information processing systems*, pp.9793–9803.
- [68] **Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. et al.** (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems, *arXiv preprint arXiv:1603.04467*.
- [69] **Malaek, S., Alaeddini, A. and Gerren, D.** (2011). Optimal maneuvers for aircraft conflict resolution based on efficient genetic webs, *IEEE Transactions on Aerospace and Electronic Systems*, 47(4), 2457–2472.
- [70] **Wasiuk, D., Lowenberg, M. and Shallcross, D.** (2015). An aircraft performance model implementation for the estimation of global and regional commercial aviation fuel burn and emissions, *Transportation Research Part D: Transport and Environment*, 35, 142 – 159, <http://www.sciencedirect.com/science/article/pii/S1361920914001850>.
- [71] **Uzun, M., Demirezen, M.U., Koyuncu, E., Inalhan, G., Lopez, J. and Vilaplana, M.** (2019). Deep Learning Techniques for Improving Estimations of Key Parameters for Efficient Flight Planning, *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pp.1–8.
- [72] **Sun, J., Hoekstra, J.M. and Ellerbroek, J.** (2020). OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations, *Aerospace*, 7(8), 104, <http://dx.doi.org/10.3390/aerospace7080104>.
- [73] **Baumann, S. and Klingauf, U.** (2020). Modeling of aircraft fuel consumption using machine learning algorithms, *CEAS Aeronautical Journal*, 11(1), 277–287, <https://doi.org/10.1007/s13272-019-00422-0>.
- [74] **Swischuk, R.C. and Allaire, D.L.** A Machine Learning Approach to Aircraft Sensor Error Detection and Correction, <https://arc.aiaa.org/doi/abs/10.2514/6.2018-1164>, <https://arc.aiaa.org/doi/pdf/10.2514/6.2018-1164>.

- [75] **Behjat, A., Zeng, C., Rai, R., Matei, I., Doermann, D. and Chowdhury, S.** (2020). A physics-aware learning architecture with input transfer networks for predictive modeling, *Applied Soft Computing*, 96, 106665.
- [76] **Vabalas, A., Gowen, E., Poliakoff, E. and Casson, A.J.** (2019). Machine learning algorithm validation with a limited sample size, *PLOS ONE*, 14(11), 1–20, <https://doi.org/10.1371/journal.pone.0224365>.
- [77] **Rad, K.R., Zhou, W. and Maleki, A.** (2020). Error bounds in estimating the out-of-sample prediction error using leave-one-out cross validation in high-dimensions, *S. Chiappa and R. Calandra, editors, Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, PMLR, Online, pp.4067–4077, <http://proceedings.mlr.press/v108/rad20a.html>.
- [78] **Wu, Z., Li, X. and Du, J.** (2019). Fuel Consumption Model of Aircraft in Descent Stage Based on DBN, *IOP Conference Series: Materials Science and Engineering*, 569, 032005, <https://doi.org/10.1088/1757-899x/569/3/032005>.
- [79] **Chati, Y.S. and Balakrishnan, H.** (2018). Data-Driven Modeling of Aircraft Engine Fuel Burn in Climb Out and Approach, *Transportation Research Record*, 2672(29), 1–11, <https://doi.org/10.1177/0361198118780876>, <https://doi.org/10.1177/0361198118780876>.
- [80] **Muralidhar, N., Bu, J., Cao, Z., He, L., Ramakrishnan, N., Tafti, D. and Karpatne, A.** (2020). Physics-Guided Deep Learning for Drag Force Prediction in Dense Fluid-Particulate Systems, *Big Data*, 8(5), 431–449.
- [81] **Muralidhar, N., Islam, M.R., Marwah, M., Karpatne, A. and Ramakrishnan, N.** (2018). Incorporating prior domain knowledge into deep neural networks, *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, pp.36–45.
- [82] **Misyris, G.S., Venzke, A. and Chatzivasileiadis, S.** (2020). Physics-informed neural networks for power systems, *2020 IEEE Power & Energy Society General Meeting (PESGM)*, IEEE, pp.1–5.
- [83] **Yucesan, Y.A. and Viana, F.A.** (2020). A physics-informed neural network for wind turbine main bearing fatigue, *International Journal of Prognostics and Health Management*, 11(1), 17.
- [84] **Read, J.S., Jia, X., Willard, J., Appling, A.P., Zwart, J.A., Oliver, S.K., Karpatne, A., Hansen, G.J., Hanson, P.C., Watkins, W. et al.** (2019). Process-guided deep learning predictions of lake water temperature, *Water Resources Research*, 55(11), 9173–9190.
- [85] **Rai, R. and Sahu, C.K.** (2020). Driven by Data or Derived Through Physics? A Review of Hybrid Physics Guided Machine Learning Techniques With Cyber-Physical System (CPS) Focus, *IEEE Access*, 8, 71050–71073.

- [86] **Shukla, K., Di Leoni, P.C., Blackshire, J., Sparkman, D. and Karniadakis, G.E.** (2020). Physics-Informed Neural Network for Ultrasound Nondestructive Quantification of Surface Breaking Cracks, *Journal of Nondestructive Evaluation*, 39(3), 61.
- [87] **Lu, H. and Wang, Z.** Physics-aware Neural Network Approach to Inverse Problems in Groundwater Contamination.
- [88] **Zamzam, A.S. and Sidiropoulos, N.D.** (2020). Physics-aware neural networks for distribution system state estimation, *IEEE Transactions on Power Systems*.
- [89] **Hornik, K., Stinchcombe, M. and White, H.** (1989). Multilayer feed-forward networks are universal approximators, *Neural Networks*, 2(5), 359 – 366, <http://www.sciencedirect.com/science/article/pii/0893608089900208>.
- [90] **Roberson, B.** (2007). Fuel Conservation Strategies: cost index explained, *Boeing Aero Quarterly*, 2(2007), 26–28.
- [91] **Abu-Mostafa, Y.S.** (1990). Learning from hints in neural networks, *Journal of Complexity*, 6(2), 192 – 198, <http://www.sciencedirect.com/science/article/pii/0885064X9090006Y>.
- [92] **Liu, D.C. and Nocedal, J.** (1989). On the limited memory BFGS method for large scale optimization, *Mathematical Programming*, 45(1), 503–528, <https://doi.org/10.1007/BF01589116>.
- [93] **Rahmani, F., Lawson, K., Ouyang, W., Appling, A., Oliver, S. and Shen, C.** (2020). Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data, *Environmental Research Letters*, <http://iopscience.iop.org/article/10.1088/1748-9326/abd501>.
- [94] **Ramadhan, A., Marshall, J., Souza, A., Wagner, G.L., Ponnappati, M. and Rackauckas, C.,** (2020), Capturing missing physics in climate model parameterizations using neural differential equations, 2010.12559.
- [95] **Sudolsky, M.** (2009). ARINC 573/717, 767 and 647A: The Logical Choice for Maintenance Recording And IVHM Interface Control or Frame Updates, *Proc. of Prognostics and Health Management Society*.
- [96] **Luo, L., Xiong, Y. and Liu, Y.** (2019). Adaptive Gradient Methods with Dynamic Bound of Learning Rate, *International Conference on Learning Representations*.
- [97] **Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al.** (2019). Pytorch: An imperative style, high-performance deep learning library, *Advances in neural information processing systems*, pp.8026–8037.
- [98] **Falise, F.** (2016). AMDAR Benefits to the Air Transportation Industry, **Technical Report**, WMO Integrated Global Observing System.

- [99] **Whitaker, J.S., Hamill, T.M., Wei, X., Song, Y. and Toth, Z.** (2008). Ensemble data assimilation with the NCEP global forecast system, *Monthly Weather Review*, 136(2), 463–482.
- [100] **Benjamin, S.G., Weygandt, S.S., Brown, J.M., Hu, M., Alexander, C.R., Smirnova, T.G., Olson, J.B., James, E.P., Dowell, D.C., Grell, G.A. et al.** (2016). A North American hourly assimilation and model forecast cycle: The Rapid Refresh, *Monthly Weather Review*, 144(4), 1669–1694.
- [101] **Derber, J. and Bouttier, F.** (1999). A reformulation of the background error covariance in the ECMWF global data assimilation system, *Tellus A: Dynamic Meteorology and Oceanography*, 51(2), 195–221.
- [102] **Richardson, L.F.** (2007). *Weather prediction by numerical process*, Cambridge University Press.
- [103] **Haan, S.d., Bailey, L. and Können, J.** (2013). Quality assessment of Automatic Dependent Surveillance Contract (ADS-C) wind and temperature observation from commercial aircraft, *Atmospheric Measurement Techniques*, 6(2), 199–206.
- [104] **de Haan, S.** (2011). High-resolution wind and temperature observations from aircraft tracked by Mode-S air traffic control radar, *Journal of Geophysical Research: Atmospheres*, 116(D10).
- [105] **Cole, R.E., Green, S.M. and Jardin, M.R.** (2000). Improving RUC-1 wind estimates by incorporating near-real-time aircraft reports, *Weather and Forecasting*, 15(4), 447–460.
- [106] **De Jong, P., Laan, J.v.d., Veld, A.i., Van Paassen, M. and Mulder, M.** (2014). Wind-profile estimation using airborne sensors, *Journal of Aircraft*, 51(6), 1852–1863.
- [107] **Sun, J., Vû, H., Ellerbroek, J. and Hoekstra, J.** (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, 30th.
- [108] **Delahaye, D. and Puechmorel, S.** (2008). Aircraft local wind estimation from radar tracker data, *2008 10th International Conference on Control, Automation, Robotics and Vision*, IEEE, pp.1033–1038.
- [109] **Blom, H. and Lygeros, J.** (2006). *Stochastic Hybrid Systems: Theory and Safety Critical Applications*, Springer.
- [110] **Cabos, R., Hecker, P., Kneuper, N. and Schiefele, J.** (2017). Wind forecast uncertainty prediction using Machine Learning techniques on Big Weather Data, *17th AIAA Aviation Technology, Integration, and Operations Conference*, p.3077.
- [111] **Jensen, L., Tran, H. and Hansman, J.** (2015). Cruise Fuel Reduction Potential from Altitude and Speed Optimization in Global Airline Operations, *Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015)*, Lisbon, Portugal.

- [112] **Villarroel, J. and Rodrigues, L.** (2016). An optimal control framework for the climb and descent economy modes of flight management systems, *IEEE Transactions on Aerospace and Electronic Systems*, 52(3), 1227–1240.
- [113] **Rivas, D., Vazquez, R. and Franco, A.** (2016). Probabilistic Analysis of Aircraft Fuel Consumption Using Ensemble Weather Forecasts, *7th International Conference on Research in Air Transportation (ICRAT)*, Philadelphia, PA, pp.1–8.
- [114] **González Arribas, D., Soler Arnedo, M.F. and Sanjurjo Rivo, M.** (2016). Wind-based robust trajectory optimization using meteorological ensemble probabilistic forecasts.
- [115] (2015), NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive, <https://doi.org/10.5065/D65D8PWK>.



CURRICULUM VITAE

Name Surname: Mevlüt UZUN

EDUCATION:

- **B.Sc.:** 2013, Istanbul Technical University, Faculty of Aeronautics and Astronautics, Aeronautical Engineering
- **B.Sc.:** 2015, Istanbul Technical University, Faculty of Electrical and Electronics Engineering, Control Engineering
- **M.Sc.:** 2016, Istanbul Technical University, Graduate School of Science Engineering and Technology, Aerospace Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2019-2020 Visiting Researcher, Department of Operational Flight Efficiency, Boeing Research and Technology.
- 2015-2021 Teaching Assistant, Department of Aeronautical Engineering, Istanbul Technical University.
- 2013-2020 Research Assistant, Controls and Avionics Research Group, ITU Aerospace Research Center.
- 2019-2020 TUBITAK International Doctoral Research Fellowship.
- 2017-2018 Outstanding Researcher Award, Istanbul Technical University
- 2014 The Boeing Company International Scholarship

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Uzun, M.**, Demirezen, U., & Inalhan, G. (2020). Physics Guided Neural Network for Data-Driven Aircraft Performance Modeling, Aerospace 2021, 8, 44..
- **Uzun, M.**, Demirezen, U., & Inalhan, G. (2020). Deep Learning Based Digital Twin for Updating Baseline Aircraft Performance Models, IEEE Transactions on Aerospace and Electronic Systems (Under Review).
- **Uzun, M.**, Demirezen, M. U., Koyuncu, E., Inalhan, G., Lopez, J., & Vilaplana, M. (2019, September). Deep Learning Techniques for Improving Estimations of Key Parameters for Efficient Flight Planning. In 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC) (pp. 1-8). IEEE, San Diego, CA, US.
- **Uzun, M.**, Demirezen, M. U., Koyuncu, E., & Inalhan, G. (2019, March). Design of a hybrid digital-twin flight performance model through machine learning. In 2019 IEEE Aerospace Conference (pp. 1-14). IEEE, Big Sky, MT, US.
- Alizadeh, A., **Uzun, M.**, Koyuncu, E., Inalhan, G., “Optimal En-Route Trajectory Planning Based on Wind Information”, 15th IFAC Symposium on Control in Transportation Systems, CTS 2018, Savona, Italy, June 6-8, 2018

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Uzun, M., & Koyuncu, E.** (2017). Data-driven Trajectory Uncertainty Quantification for Climbing Aircraft to Improve Ground-Based Trajectory Prediction. *Anadolu University of Sciences & Technology-A: Applied Sciences & Engineering*, 18(2).
- Koyuncu, E., **Uzun, M.**, & Inalhan, G. (2016). Cross-entropy-based cost-efficient 4D trajectory generation for airborne conflict resolution. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(9), 1605-1631.
- Roghani, S. E. S., Koyuncu, E., & **Uzun, M.** (2019, March). Trajectory Generation and Regeneration for Constrained Differentially Flat Control Systems. In *2019 IEEE Aerospace Conference*, IEEE, Big Sky, MT, US.
- Altun, A., Başpınar, B., **Uzun, M.**, Koyuncu, E. & Inalhan, G. (2019, September). The Propagation of Trajectory Uncertainties on Air Traffic Flow Network (In Turkish). 8th. National Aerospace Conference, Samsun, Turkey.
- Alizadeh, A., **Uzun, M.**, & Koyuncu, E. (2018). Optimal Trajectory Planning based on Wind-Optimal Cost Index. In *2018 AIAA Aviation Technology, Integration, and Operations Conference* (p. 2880), Atlanta, GA, US.
- Casado, E., La Civita, M., **Uzun, M.**, Koyuncu, E., & Inalhan, G. (2018). Estimated Time of Arrival Sensitivity to Aircraft Intent Uncertainty. *IFAC-PapersOnLine*, 51(9), 162-167.
- **Uzun, M.**, Başpınar, B., Koyuncu, E., & Inalhan, G. (2017, September). Takeoff weight error recovery for tactical trajectory prediction automaton of air traffic control operator. In *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th* (pp. 1-7). IEEE.
- Başpınar, B., **Uzun, M.**, Guven, A. F., Basturk, T., Tasdelen, I., Koyuncu, E., & Inalhan, G. (2017, September). A 4D trajectory generation infrastructure tool for controller working position. In *Digital Avionics Systems Conference (DASC), 2017 IEEE/AIAA 36th* (pp. 1-10). IEEE.
- **Uzun, M.**, Koyuncu, E. Inalhan, G. (2016, September). Flight Deck Centered Trajectory Generation (In Turkish). 6th National Aerospace Conference, Kocaeli, Turkey.
- **Uzun, M.**, Guner, G., Koyuncu, E., & Inalhan, G. (2015, September). Flight Deck Centered Tactical 4D Trajectory Planning and Collision Avoidance with Flight Envelope Sampling. In *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems* (pp. 53-64). ACM.
- **Uzun, M.**, Güner, G., Koyuncu, E., & Inalhan, G. (2014, May). Integrated Flight Deck Testbed with Next Generation Visual Decision Support Tools. In *Proceedings of the International Conference for Research in Air Transportation*, 2014.

- **Uzun, M.** (2015, September), Flight Deck Centered Efficient Trajectory Planning, Base of Aircraft Data User Group Meeting, Boeing Premises, Washington D.C, VA, US.
- **Uzun, M.** (2017, May), Use of BADA 4 in Quantifying Uncertainties in Trajectory Prediction, Base of Aircraft Data User Group Meeting, NATS Premises, Prestwick, Scotland, UK.
- **Uzun, M.** (2018, September), Digital Twin Flight Modeling Through Machine Learning, Base of Aircraft Data User Group Meeting, NAV Centre, Cornwall, Canada.
- **Uzun, M.,** Demirezen, U., Koyuncu, E., & İnalhan, G. Lopez, J., & Vilaplana, M. (2019, August). Digital-Twin Aircraft Performance and Wind Modeling Using Machine Learning. In Boeing Technical Excellence Conference, Anaheim, CA, US.
- **Uzun, M.,** Demirezen, U., & İnalhan, G., (September 2018) Digital-twin Flight Modeling Through Machine Learning For Trajectory Error Estimation And Recovery, In FAA-EUROCONTROL Technical Interchange Meeting / ART Workshop ‘Machine Learning and Artificial Intelligence’, Brétigny, France.