



HAL
open science

Analyse et détection des trajectoires d'approches atypiques des aéronefs à l'aide de l'analyse de données fonctionnelles et de l'apprentissage automatique

Gabriel Jarry

► **To cite this version:**

Gabriel Jarry. Analyse et détection des trajectoires d'approches atypiques des aéronefs à l'aide de l'analyse de données fonctionnelles et de l'apprentissage automatique. Mathématiques [math]. Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), 2020. Français. NNT: . tel-04669761

HAL Id: tel-04669761

<https://enac.hal.science/tel-04669761v1>

Submitted on 9 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *24/11/2020* par :

Gabriel JARRY

Analyse et détection des trajectoires d'approches atypiques des aéronefs à l'aide de l'analyse de données fonctionnelles et de l'apprentissage automatique

JURY

XAVIER PRATZ	Professeur Associé	Rapporteur
JACCO HOEKSTRA	Professeur d'Université	Rapporteur
AMY PRITCHETT	Professeur d'Université	Examineur
ÉRIC FÉRON	Professeur d'Université	Examineur
DANIEL DELAHAYE	Professeur d'Université	Directeur de thèse
FLORENCE NICOL	Enseignant Chercheur	Co-directeur de thèse
EMMANUEL RACHELSON	Professeur d'Université	Invité
CHRISTOPHE HURTER	Professeur d'Université	Invité
ANDRÉ VERNAY	Pilote, Expert DGAC	Invité

École doctorale et spécialité :

AA : Mathématiques Appliquées et Informatique

Unité de Recherche :

Laboratoire de Recherche ENAC

Directeur(s) de Thèse :

Daniel DELAHAYE et Florence NICOL

Rapporteurs :

Xavier PRATZ et Jacco HOEKSTRA

ABSTRACT

Improving aviation safety generally involves identifying, detecting and managing undesirable events that can lead to final events with fatalities. Previous studies conducted by the French National Supervisory Authority have led to the identification of non-compliant approaches presenting deviation from standard procedures as undesirable events.

This thesis aims to explore functional data analysis and machine learning techniques in order to provide algorithms for the detection and analysis of atypical trajectories in approach from ground side.

Four research directions are being investigated. The first axis aims to develop a post-op analysis algorithm based on functional data analysis techniques and unsupervised learning for the detection of atypical behaviours in approach. The model is confronted with the analysis of airline flight safety offices, and is applied in the particular context of the COVID-19 crisis to illustrate its potential use while the global ATM system is facing a standstill. The second axis of research addresses the generation and extraction of information from radar data using new techniques such as Machine Learning. These methodologies allow to improve the understanding and the analysis of trajectories, for example in the case of the estimation of on-board parameters from radar parameters. The third axis proposes novel data manipulation and generation techniques using the functional data analysis framework. Finally, the fourth axis focuses on extending the post-operational algorithm into real time with the use of optimal control techniques, giving directions to new situation awareness alerting systems.

Keywords: *Aviation Safety, Atypical Approaches, Machine Learning, Anomaly Detection, Functional Data Analysis, Functional Principal Component Analysis*

RÉSUMÉ

L'amélioration de la sécurité aérienne implique généralement l'identification, la détection et la gestion des événements indésirables qui peuvent conduire à des événements finaux mortels. De précédentes études menées par la DSAC, l'autorité de surveillance française, ont permis d'identifier les approches non-conformes présentant des déviations par rapport aux procédures standards comme des événements indésirables.

Cette thèse vise à explorer les techniques de l'analyse de données fonctionnelles et d'apprentissage automatique afin de fournir des algorithmes permettant la détection et l'analyse de trajectoires atypiques en approche à partir de données sol.

Quatre axes de recherche sont abordés. Le premier axe vise à développer un algorithme d'analyse post-operationnel basé sur des techniques d'analyse de données fonctionnelles et d'apprentissage non-supervisé pour la détection de comportements atypiques en approche. Le modèle sera confronté à l'analyse des bureaux de sécurité des vols des compagnies aériennes, et sera appliqué dans le contexte particulier de la période COVID-19 pour illustrer son utilisation potentielle alors que le système global ATM est confronté à une crise. Le deuxième axe de recherche s'intéresse plus particulièrement à la génération et à l'extraction d'informations à partir de données radar à l'aide de nouvelles techniques telles que l'apprentissage automatique. Ces méthodologies permettent d'améliorer la compréhension et l'analyse des trajectoires, par exemple dans le cas de l'estimation des paramètres embarqués à partir des paramètres radar. Le troisième axe, propose de nouvelles techniques de manipulation et de génération de données en utilisant le cadre de l'analyse de données fonctionnelles. Enfin, le quatrième axe se concentre sur l'extension en temps réel de l'algorithme post-operationnel grâce à l'utilisation de techniques de contrôle optimal, donnant des pistes vers de nouveaux systèmes d'alerte permettant une meilleure conscience de la situation.

Mots-clés: *Sécurité Aérienne, Approches Atypiques, Apprentissage Automatique, Détection d'Anomalies, Analyse de Données Fonctionnelles, Analyse en Composantes Principales Fonctionnelles*

ACKNOWLEDGEMENT

My thanks go first and foremost to my supervisors Daniel Delahaye and Eric Féron, who trusted me and gave me the opportunity to discover the world of research already during my master degree internships. They gave me the desire to continue doing a doctoral thesis and supported me in the development of this project.

I am grateful to Daniel for these three years, spent in a convivial, sporty atmosphere, where I have been able to grow and carry out this project. And I would like to thank Eric for his valuable advice and remarks during all this period.

I would like to warmly thank André Vernay, my DGAC operational supervisor, for his enthusiasm, his encouragement, his sense of humor, and all the operational expertise he was able to provide me with during this three years. I am also grateful to Yoni Malka, Paul-Emmanuel Thurat, Stéphane Corcos, and the entire DSAC MEAS team for their support, advice, and the pleasant atmosphere every time they welcomed me.

I would like to express my gratitude to Mathy Gonon, Rejane Lavenac, Marc Houalla, Helene Bastianelli, and Georges Welterlin, who participated in the elaboration of the thesis, and to my co-supervisor Florence Nicol who gave me helpful advice regarding the use of functional data analysis techniques.

I would like to express my thanks to Frédéric Celarié, Georges Michaud, Vincent Durel, and Pierre Jacquemin who believed in my project, and agreed to collaborate by sharing their knowledge, expertise and airline data, as well as Patrick Robbe for his always very positive feedback on my research work.

I want to thank Brice Panel, and Gaël Vincent for providing me with their expertise and knowledge of approach control at Roissy Charles de Gaulle. I wish to thank Alain Bourgin, Bertrand Georges, Didier Martin and all the DSNA environmental office team for providing expertise, radar data and for the trust they placed in me for the future. I also want to show my gratitude to Stéphane Chatty and Thibaud Figueroa for their vision of innovation and their support in promoting my work.

I would like to give a special thanks to Christophe Hurter, for the successful collaborations, the salsa parties, and for having followed me in hypnosis formation.

I am grateful to all my jogging partners Andrija Vidosavljevic, Marcel Mongeau, Philippe Monmousseau, and Daniel Delahaye, for sharing regular meetings on the canal

du midi, and to Stephane Puechmorel, and Laurent Lapasset for their always subtle humour during the gym sessions. A special thank goes to Stéphane for his incredible precise mathematical knowledge and all the advice he gave me, and to Laurent for the setting up and the management of computation clusters which allowed to strongly accelerate the learning phases and other computations of my thesis.

I would like to thank my fellow PhD students Almoctar Hassoumi, Sana Ikli and Philippe Monmousseau, with a special thoughts for Philippe, which agreed to let me give him musical blind tests. And to thank my long time friends Alex, Antoine, JB, and Nico for all the good times spent together since high school and before.

Finally, I would like to thank my family for always supporting me and providing me with a healthy, loving and secure environment, key of my success.

TABLE OF CONTENTS

ABSTRACT	i
RÉSUMÉ	ii
ACKNOWLEDGEMENT	iii
LIST OF FIGURES	x
LIST OF TABLES	xxiv
LIST OF TERMS AND ABBREVIATIONS	xxv
FOREWORD	xxvi
1 Introduction	1
2 Background	5
2.1 Approach procedures: operational and safety issues	6
2.1.1 Approach Procedures	6
2.1.2 The Role of Approach Controllers	7
2.1.3 Non-Compliant Approaches and Safety Issues	9
2.1.4 Energy Management during approach	12
2.2 Machine learning: data-based techniques	13
2.2.1 Unsupervised Learning: extracting information or structure from data	14
2.2.2 Supervised Learning: Creating Prediction Models	17
2.2.3 Reinforcement Learning	19
2.3 Functional Data Analysis: statistical tools applied to functional data	19
2.3.1 Introduction to Functional Data	20
2.3.2 The functional nature of aircraft trajectories	20
2.3.3 Functional Principal Component Analysis	22
2.3.4 Curve Registration	23
2.4 Thesis Framework	24

2.4.1	Data	24
2.4.2	Research Challenges and Contributions	25
2.5	Thesis Outline	26
3	Post-Operational Atypical Approach Detection using Unsupervised Learning and Functional Principal Component Analysis	28
3.1	Atypical Trajectory Detection Algorithm	29
3.1.1	Pre-processing and dimensionality reduction	29
3.1.2	Outlier Coefficient	32
3.1.3	Identifying Atypical Trajectories	36
3.2	Application to the aircraft atypical approaches detection and result with airline flight data monitoring and events	37
3.2.1	Energy problem and algorithm parameters	37
3.2.2	Atypical approach detection comparison with flight data monitoring and events	39
3.2.3	Discussion	47
3.3	Flight safety during Covid-19: A study of Charles de Gaulle airport atypical energy approaches	49
3.3.1	Data and Methodology	49
3.3.2	Observations	50
3.3.3	Discussion	57
3.4	Conclusions	59
4	Enhancing ground data analysis and generation with Supervised Learning	61
4.1	On the use of Generative Adversarial Networks for Aircraft Trajectory Generation and Atypical Approach Detection	62
4.1.1	Introduction and methodology	62
4.1.2	Trajectory Generation	64
4.1.3	Anomaly Detection	66
4.1.4	Latent Space Representation	71
4.1.5	Discussion	74

4.2	Approach and landing aircraft on-board parameters estimation with LSTM networks	75
4.2.1	Methodology and Data	75
4.2.2	Results	78
4.2.3	Generalization	80
4.2.4	Use cases	83
4.2.5	Discussion	86
4.3	Conclusions	87
5	Toward novel interactive manipulation and generation techniques using Functional Principal Component Analysis	88
5.1	Interactive Trajectory Modification and Generation with Functional Principal Component Analysis	89
5.1.1	Related Work	89
5.1.2	Methodology and Pipeline	91
5.1.3	Use cases	101
5.1.4	Discussion	105
5.2	Interactive Shape Based Brushing Technique for Trail Sets	107
5.2.1	Related Work	109
5.2.2	Interaction Pipeline	111
5.2.3	Design space	115
5.2.4	Use cases	118
5.2.5	Discussion	124
5.3	Conclusions	126
6	Directions to real time atypical approach detection using Dubins Path	128
6.1	Methodology and Background	129
6.1.1	Dubin curves problem for trajectory generation	129
6.1.2	Real-time atypical scoring methodology	130
6.1.3	Suggested trajectory methodology	132
6.2	Study	132
6.2.1	Asiana flight 214	133
6.2.2	Pegasus Airlines flight 2193	136

6.2.3	Hermes Airlines flight 7817	139
6.2.4	Air India Express flight 1344	141
6.3	Discussion	142
6.4	Conclusions	144
7	Thesis Conclusions and Perspectives	145
7.1	Research and Operational Contributions	145
7.2	Perspectives	146
7.2.1	Research Perspectives	146
7.2.2	Operational Perspectives	147
	LIST OF PUBLICATIONS	148
	REFERENCES	150
Appendices		
Appendix A	Operational boundaries for flight parameters	174
Appendix B	Kernel Density Estimation	177
Appendix C	Neural Networks	179
C.1	Introduction	179
C.2	Multilayer perceptron and learning	180
C.3	Deep Learning and complex architectures	181
Appendix D	Pearson’s χ^2 Independence Test	182
Appendix E	Trajectory Approach Analysis (TAPAS): A post-operational aircraft approach analysis tool	184
Appendix F	Interactive Trajectory Modification and Generation with Functional Principal Component Analysis: User Interface	189
Appendix G	Predicting Passenger Flow at Charles De Gaulle Airport Security Checkpoints	191
G.1	Introduction	191
G.1.1	Motivation	191

	G.1.2 State of the art	192
G.2	Model building	193
	G.2.1 Machine Learning and Long Short-Term Memory Neural Network	193
	G.2.2 From data to features	194
	G.2.3 Network Architecture and Learning	195
	G.2.4 Penalized Loss	196
	G.2.5 Model Summary	197
G.3	Model comparison	197
	G.3.1 First results	199
	G.3.2 Case Study	203
	G.3.3 Daily analysis	203
	G.3.4 Hourly analysis	204
G.4	Discussion and Conclusion	209

LIST OF FIGURES

1.1	Accident rates and on-board fatalities per Year [1]	1
1.2	Accident rates and on-board fatalities per flight phases [1]	2
1.3	Heinrich Pyramid and Safety Philosophy	3
2.1	Diagram illustrating the segments of a conventional approach procedures	7
2.2	ILS Approach chart on runway 08R at LFPG Airport	9
2.3	This figure describes Compliant Approach Criteria and illustrates Sta- bilized Approaches.	11
2.4	Illustration of Glide Interception Detection Tool set up at CDG airport. .	12
2.5	Simple FPCA decomposition basis illustration	23
2.6	Curve registration process and warping function illustration	24
3.1	GLOSH outlier scoring for different distribution and HDBSCAN pa- rameter values	34
3.2	α -outlier coefficient for different distribution and α values	36
3.3	Atypical approach detection methodology illustration	37
3.4	Paris Orly runway 26 atypical approaches altitude and ground speed profiles	40
3.5	The three phases on a flight landing at Paris Orly Airport.	41
3.6	B737 data set atypical phases: appearance and disappearance locations. Each atypical phase is represented by a horizontal line. The x-axis gives the beginning and the end of the atypical phase and the y-axis indicates the beginning of the atypical phase. The color corresponds to the three flight phases detailed previously (25NM-15NM in blue, 15NM-5NM in yellow, 5NM-THR in purple).	42

3.7	B737 and A320 data set atypical phases: appearance and disappearance locations summary. The number in each column represents the number of atypical flights remaining or appearing during this segment. The color corresponds to the three flight phases detailed previously (25NM-15NM in blue, 15NM-5NM in yellow, 5NM-THR in purple).	43
3.8	A320 and B737 flights landing configuration time distribution and atypical ratio. Short time configuration are associated with a higher atypical ratio	46
3.9	A320 and B737 flights landing gear action appearance distribution in the landing sequence and the atypical ratio. Gear down action being applied with less flaps extension than expected is associated with a higher atypical ratio	47
3.10	A320 and B737 flights last landing configuration element distance to the mean distribution and the atypical ratio. Last landing configuration element occurring later than the mean distance is associated with higher atypical ratio	48
3.11	QFU usage at Roissy Charles de Gaulle airport before and during COVID-19. Each runway has a different colour, and the percentage corresponds to the ratio of runway use to total flights for the period. The COVID-19 period shows a change in the use of runways, whereas before COVID-19 both pairs are used with the outer runways for landing and mostly in a west-facing configuration. During the COVID-19 period, the south pair was quickly closed and a majority configuration facing east was observed linked to weather conditions.	50
3.12	Distribution of trajectory atypical rate per runway QFU before and during COVID-19	51
3.13	Example of a bayonet procedure at Roissy Charles de Gaulle airport. The aircraft was initially lined up on final for the outer runway 09L and was cleared to change and land on the inner runway 09R. This procedure is generally used to minimize taxi time.	52
3.14	Lateral profile of atypical flight before (left) and during (right) COVID-19 period at Charles de Gaulle Airport Runway 09L and 09R	52

3.15	Distribution of localizer interception distance (NM) for typical and atypical flight under and before COVID-19 for runway QFU 09L and 09R. For the COVID-19 period, there is the emergence of glide interceptions from above that were not or only slightly present in the period before COVID-19. The color of trajectories is proportional to the local atypical energy coefficient (green for 0, orange for 0.5 and red for 1). The colored dashed lines are published operational limits and deviation from them.	53
3.16	Vertical profile of atypical flights before (left) and during (right) COVID-19 period at Charles de Gaulle Airport Runway 09L and 09R	54
3.17	Distribution of altitude areas for typical and atypical flights under or before COVID-19 for runway QFU 09L and 09R. Green category corresponds to flights without any point above or under the altitude orange dashed line limit. Orange category corresponds to flight with at least on point in the altitude orange to red dashed line area, and Red category to flights with at least on point in the altitude Red dashed line area. These operational limits are detailed in Appendix A	55
3.18	Distribution of ground speed areas for typical and atypical flights under or before COVID-19 for runway QFU 09L and 09R. Green category corresponds to flights without any point above or under the orange ground speed dashed line limit. Orange category corresponds to flight with at least on point in the orange to red ground speed dashed line area and Red category to flight with at least on point in the Red ground speed dashed line area. These operational limits are detailed in Appendix A	56
3.19	Distribution of the typical ratio of the 7 major airlines at CDG under or before COVID-19 for runway QFU 09L and 09R	56
3.20	Distribution of the atypical ratio of the 7 major aircraft type families operating at CDG under or before COVID-19 for runway QFU 09L and 09R	57

3.21	Lateral, altitude and ground speed profiles of a late speed reduction A320 landing on runway 09L. The top left graph shows the lateral trajectory, the top right graph shows the vertical profile and the bottom graph shows the ground speed profile. It can be seen that the aircraft maintains 230kts ground speed up to 5NM before sharply reducing speed. The color of the trajectory is proportional to the local atypical energy coefficient (green for 0, orange for 0.5 and red for 1). The colored dashed lines are fixed operational limits to allow a better understanding of the situation.	58
3.22	Lateral, altitude and ground speed profiles of an A320 glide interception from above on runway 09L. The top left graph shows the lateral trajectory, the top right graph shows the vertical profile and the bottom graph shows the ground speed profile. It is observed that the aircraft shortened its trajectory, which led it to a glide path interception from above, which is not recovered until 5NM from the runway threshold. The color of the trajectory is proportional to the local atypical energy coefficient (green for 0, orange for 0.5 and red for 1). The colored dashed lines are fixed operational limits to allow a better understanding of the situation.	59
4.1	Generative Adversarial Network Structure	63
4.2	Discriminator and Generator network architectures. On the left a), is illustrated the discriminator structure, and on the right b) the generator structure. Both structures use convolution or convolution transpose layers.	65
4.3	1000 trajectories at Paris Orly Airport generated with a GAN. In blue are represented the original trajectories and in green the generated trajectories. At the top, the longitudinal path is represented, in the center, the altitude profile, and at the bottom the ground speed profile.	67
4.4	Extended structure of a Generative Adversarial Network with an Encoder	68
4.5	Trajectory colored with the score given by the discriminator on the original dataset.	69
4.6	Normalized distribution of the encoder-generator reconstruction errors for the specific total energy trajectories at Paris Orly Airport	70
4.7	Specific total energy trajectories at Paris Orly Airport	71

4.8	Altitude and speed profile of the highest reconstruction error flight presenting a Glide Interception From Above. The flight presents a glide interception from above with high ground speed, and a little late power reduction. The colored dots correspond to the atypical FPCA coefficients of the total energy defined in Chapter 2. The dashed colored lines correspond to operational limits: nominal (green), warning (orange), and critical (red). For more details see Appendix A	72
4.9	Latent space representation and real trajectories altitude and lateral profiles colored with a linear interpolation for the first FPCA axis. Flights which values on the first FPCA dimension are close to 1, correspond to continuous descent profiles or those with glide interceptions from above. The values close to -1 are trajectories with large levelled-off flights	73
4.10	Simplified structure of a LSTM cell. It is composed of three gates that enable the network to learn which features of the time series is important to keep or forget	76
4.11	Neural network architectures used to predict aircraft on-board parameters. On the left, is represented the LSTM architecture, and on the right the MLP Dense architecture.	77
4.12	Fuel flow rate prediction on a flight of the test data set	80
4.13	Flap configuration prediction on a flight of the test data set	80
4.14	Landing gear configuration prediction on a flight of the test data set . . .	81
4.15	Fuel flow rate prediction on a flight of the B737 test data set	82
4.16	Fuel flow rate prediction on a flight of the A330 test data set	84
4.17	Lateral, altitude, ground speed profiles and predicted parameters of an A320 glide interception from above at Roissy Charles de Gaulle airport, right y-axis information are the following Gear (0, 1 : up, down), Speed brakes (0, 1 : not commanded, commanded), Flaps (0, 1, 2, 3, 4 : Flaps 0, 1, 2, 3, Full)	85

4.18	A320 approach replay landing at Roissy Charles de Gaulle airport with extended parameters and environmental efficiency. On top of the figure is represented the altitude profile. The color below the curve is a linear mapping of the noise efficiency score. In the middle are represented both the ground speed and the estimated flaps and gear configuration. At the bottom is represented the estimated fuel flow with the color below the curve being the linear mapping of the fuel efficiency score	86
5.1	Fisheye deformation applied to air traffic trajectories over France	91
5.2	Bring and go algorithm. Multiple links leaving a node in the same compass direction are collected into bundles by routing through a dummy junction node. The process is repeated until the number of nodes of bundles leaving all nodes or junctions in the same compass falls below a given threshold	91
5.3	Data cleaning or outlier removal process illustration	93
5.4	Good and bad usage of curve modifications for aircraft approach trajectory altitude profile. At the top, both the mean and the principal components were modified. At the bottom, only the mean was modified. The behavior observed while modifying only the mean curve presents artifacts. Level-off flight was expected, but the trajectories present descent phases. These behaviors are not nominal and underline that the process was not executed properly.	95
5.5	Interpolation operator between the curves of equation $y(t)=0$ and $y(t)=1$, for $t_1=0.25$ and $t_2=0.75$	95
5.6	Translation and Rotation operators illustration on 2D curves. The curve of coordinates $(t, y(t)=0.25)$ is first translated with vector $(0, 0.25)$, finally the rotation operator a is applied at $t=0.5$ and with angle $\theta = \frac{\pi}{4}$. . .	96
5.7	Temporal compression and cut operators illustration. The curve of equation $y=\max(0.25,\min(0.75,x))$ is compressed between $t_1=0.25$ and $t_2=0.75$ with $\alpha = 2$. Second the curve is cut at $t_1=0.4$ by $\delta t = 0.1$	97
5.8	Neighborhood generation illustration. The variance is computed along each dimension. A sample is generated by drawing a vector in a variance based ellipsoid neighborhood of an existing sample	98

5.9	Illustration of the different generation processes for the principal component score distribution. In red, a Simple Gaussian model is illustrated. In blue, the Mixture Gaussian model is represented. Finally, in green, the neighbourhood model is shown. Applying a methodology consists approximately to draw a sample in one of the corresponding circles . . .	99
5.10	Illustration of the Edge Bundling (i.e. trail visual simplification) process for landing trajectories at Charles de Gaulle Airport	100
5.11	Modification and generation pipeline. On the left, the input curve dataset passes through an initial clustering step. After this, trajectory registration based on landmarks is applied. Then, the FPCA process is computed for the first time for each cluster. This FPCA process gives two pieces/elements of information: the principal component scores (top figure) and the mean curve with the principal component functions (bottom figure). The principal component scores are used in two situations: First for clustering refinement and data cleaning, second for the generation process. Generation consists in estimating the principal component score distribution and in generating new samples following the estimated distribution. During the modification operations, the mean and the principal component functions are processed. It consists in applying modification operators (rotation, translation, dilatation) in order to obtain the desired distortion. Finally, the Inverse FPCA process enables the trajectory to be reconstructed with the new distribution (i.e. increase or decrease in trajectory number) and behavior. The user chooses if they wish to add a visual simplification, otherwise, the trajectories are fully reconstructed.	100
5.12	Clustering task on landing trajectories at Bordeaux airport. On the top image is illustrated the result of the classification of the Bordeaux airport approaches from the 2017 data records. The two bottom graphs represent the first four principal component score distributions.	102

5.13	Noise NA62 indicator map of the real aircraft traffic and simulated traffic obtained with the modification pipeline. At the bottom, the same indicator map for the simulated traffic obtained with the pipeline. The modification consists in raising by 1000ft the level-off flight of landing aircraft before landing. This modification was applied following the Grenelle de l'environnement for noise reduction purposes.	103
5.14	Departure procedure deformation at Nantes airport using Bordeaux Departure data set. On top, the original trajectories at Bordeaux Airport are represented. At the bottom, the modified trajectories at Nantes Airport. In addition, the expected procedure following different aeronautical way points is shown in red.	104
5.15	Brushing interaction on simple curves. The curves brushed by the red line are displayed in red. The remaining curves are colored in blue . . .	108
5.16	Interaction pipeline to filtered items thanks to the user brush input. The pipeline extracts the brushed items but also the shape of the brush which will be then used as a comparison with the brushed items (metrics stage). Then, brushed items are stored in bins displayed with small multiples where the user can interactivity refine the selection (i.e. binning and filtering stage). Finally, the user can adjust the selection with additional brushing interactions. Note that both the PC and FPCA can be used for the binning process, but separately. Each small multiple comes from one metric exclusively.	112
5.17	Example of the two metrics for shape comparison usage. The user brushed around curve 3 and thus also selected curves 1 and 2. Thanks to the Pearson computation, the associated small multiples show that only curve 2 is correlated to the shape of the brush. Curve 3 is anti correlated since it in an opposite direction to the shape of the brush. The FPCA computation does not take into account the direction but rather the curvature similarity. As such, only shape 3 is considered as highly similar to the brush shape input.	113
5.18	Diagram of the transformation from the trail space to the 2D points space using FPCA	114

5.19	(a) The user brushes the trajectories in order to select those from the south-east of France to Paris. (b) He selects the most correlated value to take into account the direction of the trails	116
5.20	One day's aircraft trajectories in French airspace, including taxiing, taking-off, cruise, final approach and landing phases, selecting specific trajectories using the standard brushing technique will yield inaccurate results due to the large number of trajectories, occlusions, and closeness in the spatial representation.	118
5.21	(a) The user filters the trajectories that land on the north runway in Paris by brushing following the "C" shape. This retrieves the flights that come from the east. (b) He changes the selection on the small multiples, to select all the dissimilar Shapes, resulting in the trajectories that land on the north landing strip strip but that do not come from the east.	119
5.22	(a) Selected eye movements between the PFD and ND, (b) Selected eye movements in the vicinity of the PFD , (c) Only one saccade between the ND and the FCU, (d) Eye movement from the PFD to the LAI passing by the FCU.	122
5.23	Animated transition between the X/Y gaze view to the temporal view. This helps to detected how the selected eye movement occurred over time.	123
5.24	Three different trajectories containing three different event sequences from [2]. The user is able to select all trajectories passing through particular locations	124
6.1	Control model for Dubins path	130
6.2	Dubins path between two points A and B	131
6.3	The different cases of the methodology	132
6.4	Asiana Airline Flight 214 lateral, altitude and speed profiles, when the aircraft was in downwind leg. The tool indicates that a direct to the interception chevrons is critical (in red) and would cause the plane to be in over-energy due to high potential energy figure (a). The tool suggests a trajectory extending the tailwind leg (in green)	133

6.5	Asiana Airline Flight 214 lateral, altitude and speed profiles. The plane has just intercepted the runway centerline (a) and is in good condition, it is on the glide path (b) and presents a nominal speed (c).	133
6.6	Asiana Airline Flight 214 lateral, altitude and speed profiles. The aircraft is 5NM from the runway threshold (a), above the glide path (b) with a ground speed of 180kts due to air traffic constraints (c). The plane is in over-energy due to both high potential and kinetic energy. . .	134
6.7	Asiana Airline Flight 214 lateral, altitude and speed profiles. The plane is at 500ft ground level, its speed (c) and altitude (b) have returned to nominal values but its energy variation is atypical (with a coefficient of 0.97) and directs it towards an under-energy	134
6.8	Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The plane turns in base leg (a). Its ground speed increases (c) and it passes over the glide path (b).The tool suggests to slightly lengthen the track (in green).	136
6.9	Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The plane is on the base segment (a). Its ground speed is back to normal (c) but it is high on the glide path (c). The tool still suggests extending the trajectory (in green).	136
6.10	Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The plane is on final approach (a), it is high on the glide path (b). Its ground speed is increasing (c) and its atypical score is 0.87.	137
6.11	Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The aircraft is 1NM from the runway threshold. The glide path management presents large variations, its ground speed is 190kts the atypicality did not decrease throughout the final.	137
6.12	Hermes Airline Flight 7817 lateral, altitude and speed profiles. The crew follows the localizer interception heading given by the Air Traffic Controller and requests permission to deviate its trajectory by 10° left to avoid a cloud a). The aircraft's ground speed is slightly high (c) and it is still under the glide path (b).The tool indicates that the aircraft is in a typical situation.	139

- 6.13 Hermes Airline Fight 7817 lateral, altitude and speed profiles. The aircraft intercepts the runway extended centerline. The deviation of 10/degree right, shortened the track a), the plane is now above the glide path b). The aircraft's ground speed is around 250kts (c). The tool indicates an atypical situation due to an excess of potential energy. . 139
- 6.14 Hermes Airline Fight 7817 lateral, altitude and speed profiles. The plane has just recovered the glide path a). However, the potential energy b) has been transformed into kinetic energy and there is a slight increase of the ground speed towards 260kts c). The atypicality increases around 0.88 due to an excess of kinetic energy. 140
- 6.15 Hermes Airline Fight 7817 crash lateral, altitude and speed profiles. The aircraft is about 3NM from the runway threshold a). Its atypicality peaks at 1.0 because of a very high ground speed of 210 kts c), although decreasing. The aircraft is not stabilized with a very high speed and nevertheless continues its approach. 140
- 6.16 Air India Express Fight 1344 lateral, altitude and speed profiles. The aircraft intercepts the runway extended center line (a). The aircraft's ground speed and altitude profile are nominal (a and c). The tool indicates that the aircraft is in a typical situation. 142
- 6.17 Air India Express Fight 1344 lateral, altitude and speed profiles. The altitude profile is nominal however, its ground speed is around 200kts with no sign of reduction. The atypical coefficient has increased up to 0.97 underlying an atypical variation leading the aircraft to high energy. 142
- 6.18 Air India Express Fight 1344 lateral, altitude and speed profiles. The aircraft descends on the glide path, the ground speed has started to reduce down to 175kts. The atypical coefficient is back to green areas(0.09). The reduction of the energy returns the aircraft to a less atypical state 143

6.19	Air India Express Flight 1344 lateral, altitude and speed profiles. The ADS-B data stopped at 1.6NM to the runway threshold. The aircraft is still on the glide path. However, the ground speed is not reducing anymore and stagnates at 175kts. The atypical coefficient increased up to 0.96 again underlying a high kinetic energy	143
A.1	Longitudinal operational limits illustration. The green line corresponds to the extended runway center line. The orange and red line correspond to deviation from the runway center line and from the interception chevrons.	175
A.2	Altitude operational limits illustration. The green lines correspond to the glide path and the minimum published interception altitude. The orange and red line correspond to deviation from the glide path and the minimum published interception altitude.	175
A.3	Ground speed operational limits illustration. The green lines correspond to nominal ground speed areas. The orange and red lines correspond to deviations from the green line.	176
C.1	Formal neuron illustration. A weighted sum of the input goes through an activation function to compute the output.	180
C.2	Multilayer perceptron architecture. Each neuron uses as input the outputs of the previous layer and computes the input for the following layer.	181
D.1	P-value function of the the χ^2 cumulative test statistic for several degree of freedom k	183
E.1	Data section of TAPAS prototype software. It provides general information on the data set such as the aircraft type or the airports studied. . .	185
E.2	Statistics section of TAPAS prototype software. It provides information on operational statistics such as the compliance or the average number of monitoring events per 100 flights.	185
E.3	Event section of TAPAS prototype software. It provides further information and detail on the monitoring events occurrences.	186
E.4	Flight Study section of TAPAS prototype software. It enables the analysis of a particular trajectory.	186

E.5	First graph of TAPAS section Flight Study, displaying the longitudinal path and lateral compliance criteria.	187
E.6	Second graph of TAPAS section Flight Study, displaying the altitude profile and the atypical coefficient.	188
E.7	Third graph of TAPAS section Flight Study, displaying the ground speed profile, the computed air speed profile, and the atypical coefficient. . . .	188
E.8	Fourth graph of TAPAS section Flight Study, displaying the engine power, the flaps settings, the landing gear setting, and the airbrake use. .	188
F.1	User interface of the Trajectory Modification and Generation with Functional Principal Component Analysis prototype software.	189
G.1	Overview map of Charles De Gaulle terminals	192
G.2	Simplified illustration of the structure of a LSTM cell.	194
G.3	Description of the neural network LSTM200 architecture used in the experiments.	196
G.4	Model of the passenger flow at a security checkpoint.	199
G.5	Comparison of the models using or not the hour in the training set. Green color cells correspond to the model kept in the following study. Bold cells correspond to the best models.	200
G.6	Comparison per checkpoints of different mathematical metrics for the three considered models.	201
G.7	Comparison per checkpoints of different operational metrics for the three considered models.	202
G.8	Heatmap visualization of the performance difference between the LSTM models and the current predictive model.	203
G.9	Daily correlation distribution per day of the week for C2G-Depart and C2E-Puits2E.	204
G.10	Hourly passenger error boxplots comparison between the current model and the neural net trained with a mean squared error loss function at two different checkpoints.	205
G.11	Hourly average wait time boxplots comparison between the current model and the neural net trained with a mean squared error loss function at two different checkpoints.	206

G.12 Hourly comparison of the predicted number of passengers between the current model and the neural net at C2E-Puits2E on January 16th 2019.	207
G.13 Hourly comparison between the current model and the neural net trained with a mean squared error loss function at C2G-Depart on January 16th 2019	208

LIST OF TABLES

3.1	\hat{c}_{30} coefficient at $2\sigma_2$ and $3\sigma_2$ frontiers for different values of $\frac{\sigma_1}{\sigma_2}$ ratio . . .	35
3.2	\hat{c}_{30} coefficient for different ratio between the maximum of density over the sample set and the considered point	35
3.3	Contingency matrix between atypical categories and medium-high event flights underlying the distribution of the flights into each category over the data set	44
3.4	Association coefficients and tests between atypical and medium-high event flights. Four indicators are given, the Pearson correlation score, the Yule's Q score, the Cohen's Kappa, and the score and p-value of a χ^2 independence test.	44
3.5	Contingency matrix between atypical categories and unstable flights underlying the distribution of the flights into each category over the data set	44
3.6	Association coefficients and tests between atypical and unstable flights. Four indicators are given, the Pearson correlation score, the Yule's Q score, the Cohen's Kappa, and the score and p-value of a χ^2 independence test.	45
3.7	Contingency matrix between atypical categories and long touchdown flights underlying the distribution of the flights into each category over the data set	45
3.8	Association coefficients and tests between atypical and long touchdown flights. Four indicators are given, the Pearson correlation score, the Yule's Q score, the Cohen's Kappa, and the score and p-value of a χ^2 independence test.	45

4.1	Predictive performance of the LSTM and the Dense models on A320 test data for the fuel flow rate. Each entry shows the mean and the standard deviation (within parentheses) of the evaluation metric across all the flights in the test data set.	79
4.2	Predictive performance of the LSTM models on A320 test data for the flap and landing gear setting	80
4.3	Predictive performance of the LSTM model on B737 test data for the fuel flow rate and landing gear at LFPO and GMAD airport	82
4.4	Fuel flow BADA coefficient ratio between A320 and A330 depending on the flight level	83
4.5	Predictive performance of the scaled LSTM model on A330 test data for the fuel flow rate at LFPO airport	84
G.1	Summary of the three models used in the appendix	197

FOREWORD

This thesis is funded by the Direction Générale de l'Aviation Civile (DGAC), the french civil aviation administration. The thesis is operationally supervised by the Safety Management coordination Office of the Direction de la Sécurité de l'Aviation Civile (DSAC), the French Safety Authority. Scientific supervision is provided by the École Nationale l'Aviation Civile (ENAC) Operation Research and Machine Learning laboratory.

The aims of this thesis are to develop post-operational algorithms for the detection and analysis of safety events and precursors during the approach and landing phases, and to investigate potential extension towards real time detection tools or situation awareness alerting systems.

The thesis adopts the point of view of an Air Navigation Service Provider (ANSP) or National Safety Agency (NSA), which implies working with ground available data such as Radar or ADS-B data. In this thesis, the data can be open sources such as Automatic Dependent Surveillance-Broadcast (ADS-B) data or obtained through non-disclosure agreements. In particular, the radar data obtained from the Direction des Services de la Navigation Aérienne (DSNA), the french ANSP, have been subject to non-disclosure agreements. Similarly, data from Flight Data Recorder (FDR) and safety events from airlines are subject to confidentiality agreements.



CHAPTER 1

Introduction

Since the 1950s, there have been major advances in aviation safety. These improvements have led to a drastic reduction in the annual accident and fatality rate, while at the same time the number of flights and passengers has risen sharply. Figure 1.1 illustrates the accident rates for one million departures and the total number fatalities per year from a Boeing Statistical Summary [1].

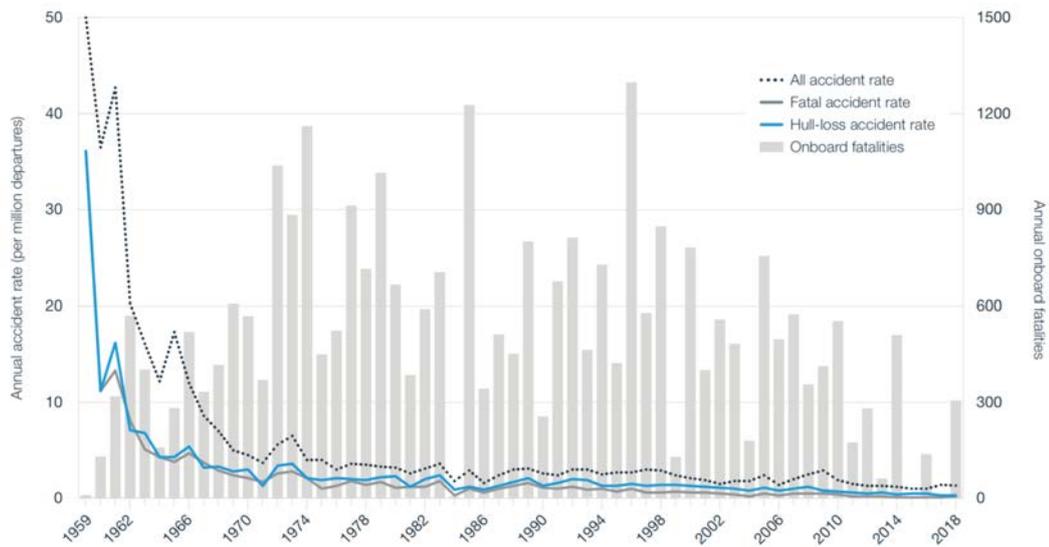


Fig. 1.1 Accident rates and on-board fatalities per Year [1]

The ethical, political and human issues surrounding air safety are substantial and involve numerous standards with appropriate monitoring and surveillance of the various air transportation system stakeholders. The National Safety Agency (NSA) ensures at a national level that safety requirements and recommendations are met. The French Safety Authority called Direction de la Sécurité de l'Aviation Civile (DSAC), is the competent authority for surveillance and certification in the field of civil aviation. In order to meet the requirements of the International Civil Aviation Organization (ICAO), the DSAC launched a safety state programme in 2006, which includes a strategic plan for the improvement of aviation safety, the current version of which is entitled "Horizon

2023” [3].

Improving safety has long been based on an effective incident and accident analysis system combined with a just culture that encourages stakeholders to report occurrences so that they can be analyzed and safety recommendations made if necessary. The just culture is part of larger safety management systems. Specific entities such as flight safety offices for airlines, or service quality offices for Air Navigation Service Provider (ANSP), have the role of analysing the flights that have been operated in order to ensure compliance with procedures and operational manuals.

Flight operations are usually divided into several phases: taxi on the ground, take-off, climb, cruise, descent, approach, landing and then a final taxi phase. However, the historical accidents are not evenly distributed in these different phases. Indeed, accident that occur during the approach and landing phases account for 47% of the total number of accidents and 40% of fatalities each year [4]. Similar results are given by Boeing Statistical Summary [1], and illustrated in Figure 1.2. Consequently, these critical phases are the subject of further analysis and recommendations.



Fig. 1.2 Accident rates and on-board fatalities per flight phases [1]

Some events in these critical phases such as unstablized approach have been identified within the context of the French safety state programme as undesirable events that might lead to final events such as runway excursion, control flight into terrain or loss of control in-flight. The stabilization criteria aim at deciding whether to continue an approach and land or abort and go-around. Unstabilized approaches are also the subject of studies within the scope of the Flight Safety Foundation (FSF), the Data4Safety (D4S) programme of the European Union Aviation Safety Agency (EASA), or Eurocontrol working groups such as the European Action Plan for the Prevention of Runway Excursions (EAPPRE). Unstabilized approaches are subject to vigilance because they have been detected in different accidents such as the Air Nostrum flight 8313 on July 30, 2011 [5], or the Asiana Airline Flight 214 at San Francisco

Airport on July 6, 2013 [6]. In addition, it also has been shown that of the 4% of unstabilized flights, only 3% go-around [7].

Identifying, detecting and managing undesirable events, potential precursors of final events, to prevent accidents or incidents is at the very heart of aviation safety. Sometimes represented as a pyramid called the Heinrich pyramid as shown in Figure 1.3, it illustrates the fact that the severity of events is generally inversely proportional to their frequency. The philosophy of safety is to reduce the base of the pyramid by avoiding precursor events, to induce a decrease in more serious events.

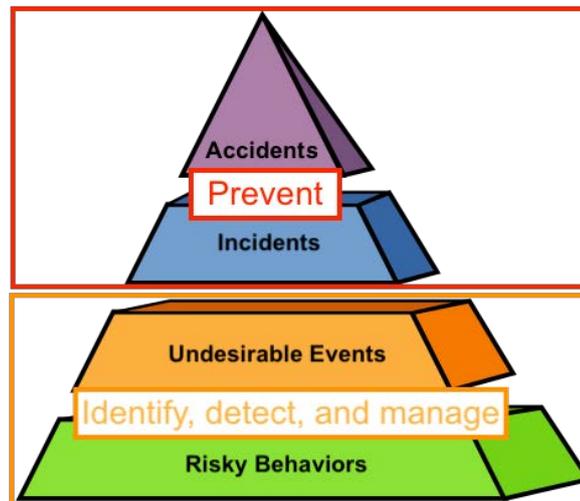


Fig. 1.3 Heinrich Pyramid and Safety Philosophy

The years 2010 were marked by the rise of new technologies such as Big Data or Machine Learning. The boom in storage and computing capacity has led to the rebirth of old models like neural networks that are now showing their full extent and capacity. The world of science and artificial intelligence is facing a paradigm shift. From an era where mathematical models based on equations and demonstration were built to explain the world and its interactions, the machine learning and its statistical approach promises to reconstruct models from the data itself if enough samples are provided.

While the latest models of flight recorders on aircraft such as the A350 record several thousand parameters at frequencies of several points per second, big data and machine learning methods show a real interest in their ability to abstract and explore high dimensional objects. It is therefore natural to ask whether these new technologies can contribute to the improvement of aviation safety in a world where certification is a prerequisite for any operational use of critical applications.

Today, there is no certification process for applications using artificial intelligence such as Machine Learning. However, the recent publication of a road map towards the certification of such systems [8] by EASA, or the creation of the European Aviation Artificial Intelligence High Level Group at Eurocontrol [9], show a trend and a willingness

of aviation institutions to promote the use of artificial intelligence in order to improve the performance and safety of the Air Traffic Management (ATM) system.

CHAPTER 2

Background

”Faut arrêter ces conneries de nord et de sud ! Une fois pour toutes, le nord, suivant comment on est tourné, ça change tout !”

Perceval - Kaamelott, Livre I, Ambidextrie

Contents

2.1	Approach procedures: operational and safety issues	6
2.1.1	Approach Procedures	6
2.1.2	The Role of Approach Controllers	7
2.1.3	Non-Compliant Approaches and Safety Issues	9
2.1.4	Energy Management during approach	12
2.2	Machine learning: data-based techniques	13
2.2.1	Unsupervised Learning: extracting information or structure from data	14
2.2.2	Supervised Learning: Creating Prediction Models	17
2.2.3	Reinforcement Learning	19
2.3	Functional Data Analysis: statistical tools applied to functional data .	19
2.3.1	Introduction to Functional Data	20
2.3.2	The functional nature of aircraft trajectories	20
2.3.3	Functional Principal Component Analysis	22
2.3.4	Curve Registration	23
2.4	Thesis Framework	24
2.4.1	Data	24
2.4.2	Research Challenges and Contributions	25
2.5	Thesis Outline	26

This chapter provides background on aircraft approach procedures, machine learning, and functional data analysis. It also presents the contributions and outlines of the thesis

2.1 Approach procedures: operational and safety issues

This section describes aeronautical approaches with their operational and safety issues. A part of the information and definitions explained in this subsection is mainly extracted from Volume 2 of the ICAO Aircraft Operations Manual. [10], dedicated to the construction of visual and instrument flight procedures.

2.1.1 Approach Procedures

An approach procedure defines the set of trajectories and altitudes to be respected in order to ensure the landing of an aircraft. It can be carried out visually or by instruments. During this phase, the aircraft reduces its speed and takes configuration to prepare for landing. For a visual approach, the pilot must be able to maintain visual reference to the runway throughout the landing phase. The pilot himself ensures the proper management of the approach by maintaining the runway centreline and the glide path. Sometimes, a Precision Approach Path Indicator (PAPI) gives pilots information on the proper management of the glide path. The instrument approach is based on standard approach procedures and is guided by the flight instruments. It follows the Standard Terminal Arrival Route (STAR) and can be said to be conventional if it uses ground stations such as ILS, VOR, or NDB systems. It is called RNAV in the context of surface navigation¹. Moreover, it can be said of precision if there is a positioning device in the vertical plane, or non-precision approach if not. The structure of approach procedures is generally divided into four segments:

1. The initial approach
2. The intermediate approach
3. The final approach
4. The missed approach

The initial approach segment is the portion of an instrument approach procedure between two points or beacons called Initial Approach Fix (IAF) and Intermediate Fix (IF). In this segment, the aircraft moves from en route to approach. The intermediate approach segment is the portion of an instrument approach procedure between the IF and another point called the Final Approach Fix (FAF). It is intended to allow the aircraft to adopt its landing configuration and adjust its airspeed to begin the final approach. The final approach segment is that part of an instrument approach procedure between the FAF and the Minimum Decision Altitude / Height (MDA/H) or Decision Altitude

¹Surface navigation is an instrument flight method that allows an aircraft to use any trajectory within a network of waypoints rather than navigating directly between ground stations.

/ Height (DA/H) depending on the type of approach. During a precision approach, if visual cues are not acquired at DA/H, the pilot will abort the approach and follow the published missed approach procedure or continue. On a conventional approach, if visual cues are not acquired at MDA/H, the pilot maintains altitude to the Missed Approach Point (MAPt) and then follows the published missed approach procedure. The missed approach segment is the portion of an instrument approach procedure during which an aircraft executes a go-around. It starts either at the DA/H for a precision approach or at the MAPt for a conventional approach and directs the aircraft either to the start of a holding procedure, to a new approach procedure or to a diversion point to another airport. Figure 2.1 summarizes the different segments of a conventional approach procedure.

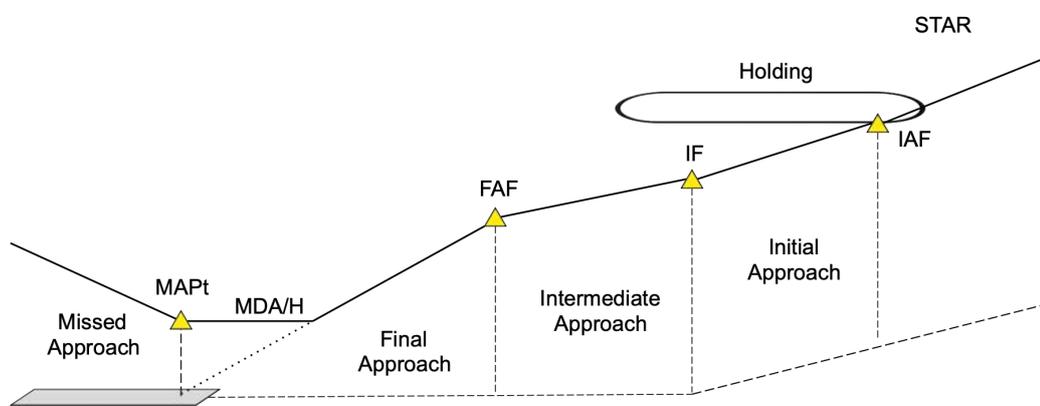


Fig. 2.1 Diagram illustrating the segments of a conventional approach procedures

2.1.2 The Role of Approach Controllers

The purpose of approach control is to provide air navigation services to aircraft operating in controlled airspace surrounding airports. Approach controllers manage both departing aircraft that wish to climb into higher airspace, transiting aircraft, and arriving aircraft that wish to land. Their objective is to ensure the smooth flow and organisation of air traffic in the corresponding airspace. One of the problems that controllers face is the scheduling and timing of landings. It is a consequence of the convergence towards the airport of aircraft arriving from various directions. As previously discussed, arriving aircraft follow STARs and then approach procedures that guide them to the runways. The controllers manage the merging of the different streams using radar guidance. It generally prevents aircraft from having to wait in the air. The aircraft are thus aligned one after the other on the runway centreline in order to land.

Air navigation services are required to ensure separation between aircraft. This separation may be horizontal or vertical. A horizontal separation of 5NM is applied between two aircraft flying at the same level for en route airspace. It may be longitudinal

if the aircraft follow each other or lateral if they cross each other. This distance is generally reduced to 3NM in Terminal Manoeuvring Areas (TMA). The vertical separation in France is 1000ft below flight level² 410. Thus, if horizontal separation cannot be ensured, the controller will perform a vertical separation. Conversely, if vertical separation is no longer assured, he will perform a horizontal separation. In radar control, the controller observes all traffic and must ensure that aircraft maintain radar separation. He must anticipate and prevent potential conflicts. He may interact with aircraft and ask them to reduce speed, change altitude or direction.

The approach phase is a transition zone. Aircraft arriving from all directions are separated geographically but must be sequenced when landing. There is a shift from a geographical separation to a temporal separation. The temporal separation is all the more important as the performance of an aerodrome is measured at its runway cadence. In other words, the number of aircraft that can land on its runways in one hour. Sequencing is done in such a way as to respect separation constraints but also to avoid wake turbulence from one aircraft to another. There is therefore a compromise and a balance to be found between separation and runway cadence. In practice, controllers try to place aircraft at separation minima in order to optimise the runway cadence, but always keep in mind the safety aspects essential to the smooth running of the approach phase. The points of convergence are sensitive points. In fact, incorrect guidance or incorrect decision making can lead to the loss of a runway slot and the subsequent reintroduction of the aircraft into the landing sequence. It can also lead to operational safety problems if the aircraft is unable to complete the approach in the proposed conditions.

Charles de Gaulles Airport situation As an example, approach control at Roissy Charles de Gaulle airport (CDG) is a sensitive task since the airport handles more than 100,000 passengers a year, i.e. about 700 to 800 landings a day. It also takes place in a complex airspace with the presence of Le Bourget airport to the south-west and Orly airport to the south. The approaches to Le Bourget and CDG are carried out by the same control centre and the same controllers due to their proximity. Le Bourget receives mainly business flights.

The airport consists of two parallel north-south runway pairs which are shown on the map in Figure 2.2. In nominal use, the outer runways of each dual runways (09L-27R and 08R-26L) are used for landings and the inner runways (09R-27L and 08L-26R) are used for take-offs. This choice allows independence of the runways for landing because they are sufficiently spaced. There are therefore no constraints and two aircraft can land simultaneously on the north and south doublets. However, the horizontal or vertical separation is still valid beyond 4NM upstream of the runway threshold. For this reason, current operations recommend different interception altitudes for ILS approaches at

²flight level (FL) is an aircraft's altitude at standard air pressure, expressed in hundreds of feet.

FAF for the north and south runways (5000ft and 4000ft respectively). In this context, controllers are particularly interested in configurations where so-called "twin aircraft" are observed. That is to say, symmetrical aircraft, one to the north, the other to the south, for which the respective interceptions of the runway axis occur almost simultaneously. This causes separation problems if they are flying at the same altitude, and implies increased attention for the controllers.

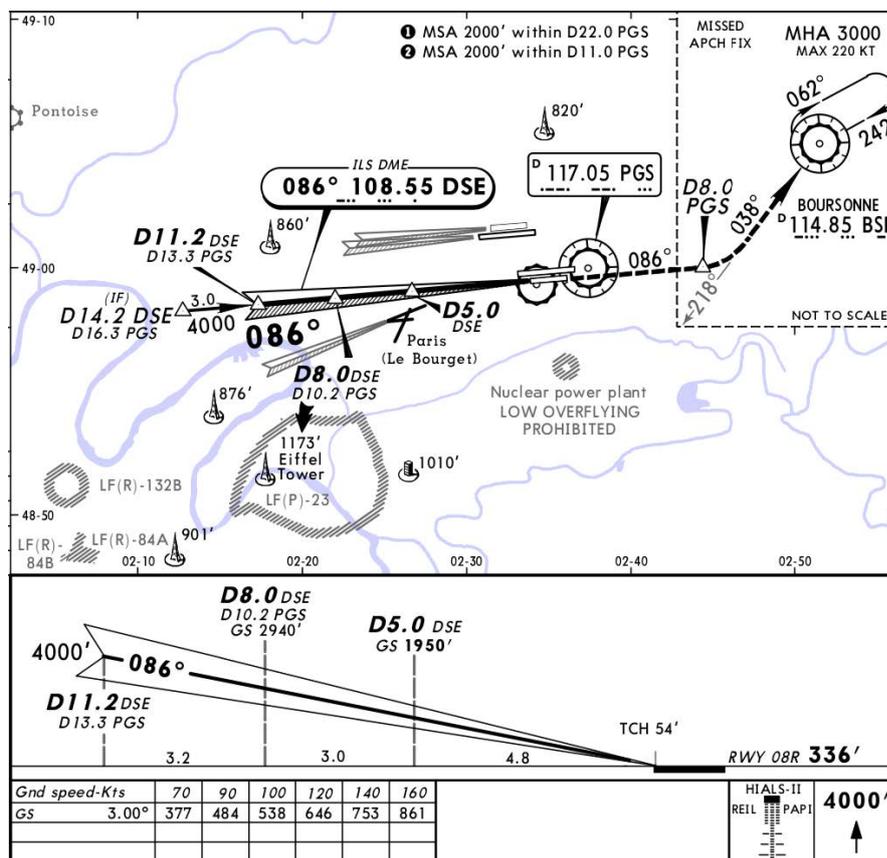


Fig. 2.2 ILS Approach chart on runway 08R at LFPG Airport

2.1.3 Non-Compliant Approaches and Safety Issues

In order to meet the requirements of the International Civil Aviation Organization (ICAO), the French Safety Authority launched a safety state programme in 2006, which includes a strategic plan for the improvement of aviation safety, the current version of which is entitled "Horizon 2023" [3]. In the 2009 action plan, a risk map was developed. In this framework, two types of events are described. Firstly, ultimate events, which are accidents within the meaning of Annex 13 of the ICAO [11], such as Control Flight Into Terrain (CFIT), runway excursions and damage or injury in flight or on the ground. Secondly, undesirable events that are undesirable with respect to the expected services. An undesirable event can be of a technical, procedural or human nature such as runway in-

cursions, unstabilized approaches, work related events, depressurization etc. Risk management in the context of the Safety State Programme aims at reducing the occurrence of undesirable events and prevent them from leading to ultimate events. Undesirable events are related to ultimate events by the probability that they will induce the event. In addition, they are characterized by their criticality and frequency of occurrence. The overall risk of an undesirable event is determined by the product of its frequency and its defined criticality.

The objective of the risk map is to establish the links between ultimate events and adverse events. Actions can then be put in place to reduce the frequency of occurrence of undesirable events and therefore the potential risk of inducing an ultimate event. The second version or action plan 2009-2013 introduces the notion of non-compliant approaches (NCAs) describe as the following: a situation in which an aircraft in IFR flight is conducting an approach for which the final approach joining conditions do not comply with those prescribed in the operational documentation (regulations, Aeronautical Information Publications, airlines operations manual, air traffic control operations manual). NCAs can take place when the aircraft is under radar vectoring, conducting an instrument approach or a visual approach. It may have on-board and/or ATC origins. It can be detected by the crew or by ATCs, in particular on the basis of radar information. NCAs may be the precursor of an unstabilised approach [12].

Practical criteria have also been introduced, notably through the positioning of chevrons upstream of the FAF. The interception of the final approach track must be carried out upstream of the chevrons with a maximum interception angle of 45 degrees or 30 degrees for certain procedures. The chevrons are positioned to allow a minimum 30-second levelled-off flight that should allow the aircraft to configure itself correctly and perform a glide slope interception from below. The ultimate goal is for the aircraft to be stabilized and in the correct configuration at the stabilized approach decision point (usually 1000ft or 500ft depending on airline policy). These criteria are illustrated in figure 2.3.

The introduction of NCAs has led to various studies and NCAs have also been detected during accidents or incidents. For example, Air Nostrum flight 8313 on July 30, 2011 to Barcelona suffered structural damage following a hard landing. Stormy weather conditions and the presence of clouds on final impeded a visual landing. Very high descent rates (up to about 3000ft/min) on final were detected and the aircraft crossed the runway threshold at a height of 315ft [5]. Another example is the Asiana Airlines Flight 214 crash on July 6, 2013 in San Francisco, which resulted in 3 deaths and 181 injuries, including 49 serious ones. The plane initially had a high altitude and high speed, then the energy state shifted to a speed and altitude too low compared to usual approach. The late go-around following the stall warning did not prevent the crash [6].

In order to be able to study the proportion of non-compliant approaches in all flights,

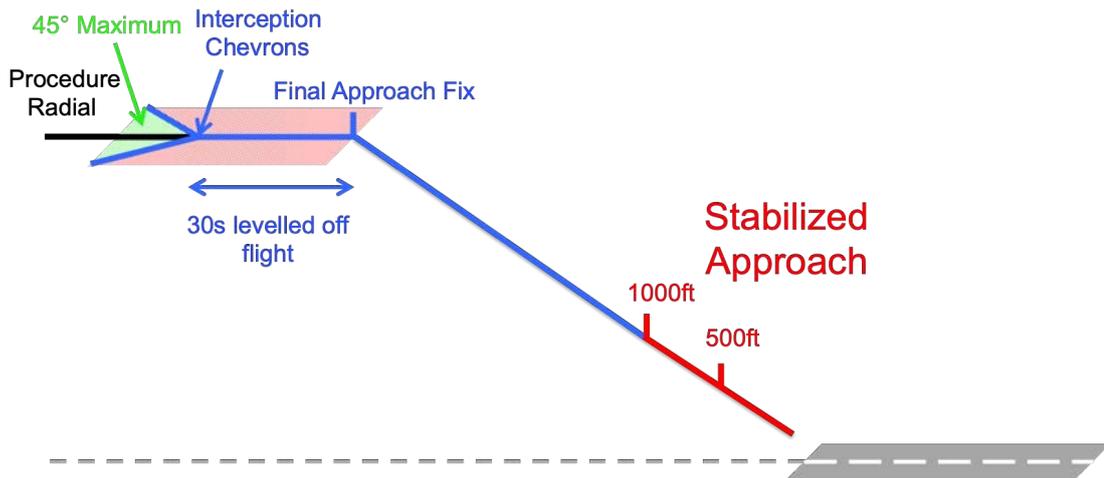


Fig. 2.3 This figure describes Compliant Approach Criteria and illustrates Stabilized Approaches.

a detection function using radar recordings has been developed and added to an analysis tool of the French ANSP (DSNA). This function allows the detection of NCAs according to pre-established criteria and an internal study was set up on flights operated at CDG airport. Of all the flights in the period, 22% were detected as non-compliant. In addition, 8 to 10% of the non-compliant flights showed significant non-compliance.

Glide Interceptions From Above (GIFAs) were identified as critical for compliance and approach stabilization [13]. A GIFA detection tool was therefore implemented at CDG airport. The tool alerts controllers when aircraft on final approach enter areas with dimensions defined in Figure 2.4 and located above the glide slope. The zones were defined to detect the three types of GIFAs highlighted by the analysis of non-compliant approaches. The first type concerns aircraft above the glide slope already before the published interception altitude. This corresponds, for example, to a control constraint that only allows the aircraft to descend late for separation reasons. For the second type, the aircraft has properly followed the levelled-off flight at the published interception altitude but it does not descend directly after intercepting the glide slope. This may be due to a misunderstanding of the phraseology. In the United States for example, controllers clear aircraft to intercept the final approach track and clear them again to descend on the glide path. In France, both clearances are given simultaneously. Therefore, it may happen that pilots wait for the controller's confirmation before descending and then passes over the glide slope. The third type corresponds to approaches where the aircraft has flown the levelled-off flight and then initiated its descent correctly on the glide slope. However, for reasons such as over speed, or reduction due to separation with the preceding aircraft, it performs a deceleration levelled-off flight and passes above the glide slope. In summary the tool is composed of four 3D-volumes, the first three volumes are warning volumes for which ATCOs advise pilots that they are too high on glide and

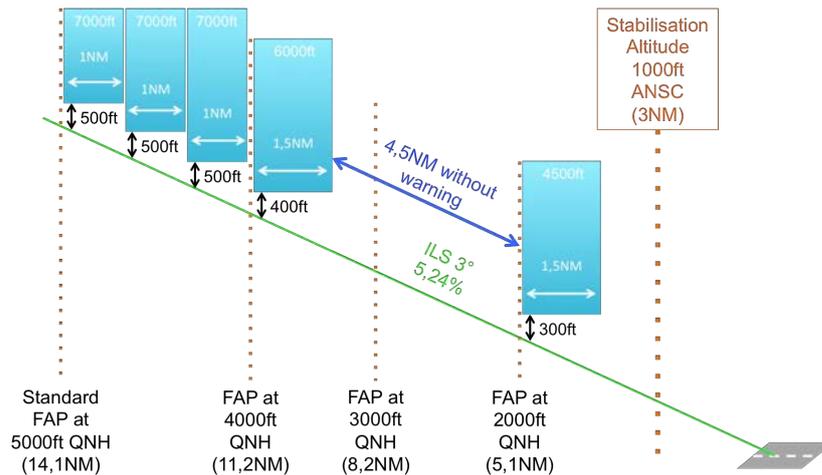


Fig. 2.4 Illustration of Glide Interception Detection Tool set up at CDG airport.

might suggest a recovery slope. The final volume is a decision volume, where ATCOs might suggest the pilot to interrupt the approach if they are still too high. The results of the experiments is positive, GIFAs are detected and an appropriate measure are taken. Approximately five flights of the 700 per day raise an alarm and in about half of the cases, ATCOs suggest a recovery slope.

2.1.4 Energy Management during approach

Naturally, it appears that approach and landing safety is intrinsically linked to the aircraft energy management in the physical sense. The objective is that an aircraft with a given altitude and speed (FL250 and 340kts for example) should exit the runway with a controlled speed and that the whole process should be as safe as possible. The total energy model of an aircraft is decomposed into kinetic energy and potential energy. From a force balance perspective, energy is added to or removed from the system through thrust and drag. In addition, when the lift induced by the aircraft speed compensates for the weight, it allows the aircraft to lift up.

The approach phase is an energy reduction phase. The plane descends and reduces its speed. During this phase, the thrust is generally reduced to the maximum in idle mode and the aircraft uses aerodynamic drag to reduce the total energy. The ability to lose energy depends on the aerodynamic configuration of the airplane such as the extension or not of the flaps and slats, the deployment of landing gear and the use of speed brakes, but also on the aerodynamics of the aircraft itself expressed through the glide ratio, which can vary greatly from one type of aircraft to another. The decrease in energy can result in a decrease in kinetic energy (i.e. speed) and/or potential energy (i.e. altitude). However, during descent phase, if the decrease in energy induced by the drag forces does not compensate for the variation in potential energy, this may imply that the

aircraft will have difficulties reducing its speed or even will have its speed increased. This has historically led to the presence of deceleration levelled-off flight in approach procedures to allow aircraft to reduce airspeed before final approach.

Typical speed reductions with different configuration has been given [14]. Nominal deceleration in level flight, with approach flaps extended is 10 knots (kts) to 15 kts per Nautical Miles (NM), and during extension of the landing gear and landing flaps is 20 kts to 30 kts per NM. Nominal deceleration on a three-degree glide path, with approach flaps and landing gear down, during extension of landing flaps is around 10 kts to 20 kts per NM; Decelerating on a three-degree glide path in a clean configuration is not possible usually. The speed reductions correspond to standard conditions, they can be deteriorated by bad weather conditions such as tailwind on final approach.

It should also be added that aircraft do not operate alone but in a pilot/controller system with strong constraints such as separation standards. This can be reflected in control constraints requiring the aircraft to maintain speeds up to a certain distance from the runway threshold. Speed constraint recommendations were put forward to allow all aircraft to reach their approach speed at the stabilization altitude. The speed constraints should not imply a speed of 180kts beyond 8NM or a speed of 150kts beyond 4NM [14].

With regard to this summary of operational and safety issues during approach and landing, directions for the thesis are to investigate the development of tools capable of analyzing energy management and detecting abnormal or atypical management that might lead to safety issues.

2.2 Machine learning: data-based techniques

The objective of this section is to describe the aims of Machine Learning and to present the potential underlying application in the aviation safety framework. A major part of the theoretical information presented in this section comes from the books of Hastie et al. [15], Vapnik et al. [16, 17], and Silverman et al. [18].

Machine learning involves teaching algorithms to complete tasks that a human could perform, using a set of data. Machine Learning is generally divided into three major task categories, they are presented in the following ordered by the increasing amount of information required for their implementation. The first category, called unsupervised learning, involves extracting information or structure from a set of input data in order to analyze or represent it. Tasks resulting from unsupervised learning include data partitioning, also known as clustering, density estimation and dimensionality reduction. The second category, called supervised learning, consists of building a model to esti-

mate one or more output variables from one or more input variables. The construction of this model goes through a learning phase on a set of known pairs (input, output). Once the learning phase is over, the model is frozen and can be used to predict the outputs on a new input sample whose output is unknown. The last category called reinforcement learning aims at building a model estimating the best transition strategy in a given environment to achieve a goal. The use of such models makes it possible to choose at any given moment which action is the best to take to achieve the final objective.

2.2.1 Unsupervised Learning: extracting information or structure from data

Unsupervised learning involves extracting information or structure from a data set in order to analyze or manipulate it. The development of such models only requires access to the input data. It is from the very structure of this data that the algorithm will extract information. Unsupervised learning allows to answer to clustering, density estimation or dimensionality reduction problems. It is generally difficult to evaluate a score on the results since the desired result is not known a priori.

Density Estimation Density estimation consists in modelling the statistical distribution of the data in its representation space. It is thus possible to know if an element is in a dense area with many neighbouring elements, or if on the contrary, it is in a low-density area and therefore isolated from the other elements. Density estimation methods include Gaussian mixtures from the Expectation Minimization (EM) algorithm [19], or neighborhood based methods such as Kernel Density Estimation [18]. For more detail with the Kernel Density Estimation see appendix B.

Density estimation was, for example, used to develop bundling techniques for visualisation [20]. It presents major links with anomaly detection and safety since it enables characterizing low density areas that can easily be assimilated to unusual events.

Data Partitioning or Data Clustering A widely used task consists in grouping together data that resemble each other to form groups or clusters. It is then possible to infer properties for each group or to analyze them independently. The clustering task can be performed using different methods such as centroid-based methods (k-means [21], k-medoid), hierarchical clustering methods, expectation maximization (EM) algorithms [19], proximity-based algorithms (DBSCAN [22] or its extended version HDBSCAN [23], OPTICS [24]), or even neural network methods (self-adaptive maps [25]).

In the following paragraphs, the DBSCAN algorithm [22] and its extension to hierarchical clustering are detailed. DBSCAN uses two parameters: a distance, which defines the maximum distance between two points in a cluster, and a number n , which corresponds to the minimum number of points in a partition to be considered as a cluster. In practice, the algorithm takes a point and examines its neighborhood within the

minimum distance to build a first partition. Then, the process is applied iteratively to all points in the partition until no more points can be added. At the end, if there are more than n points in the final partition, it is considered as a cluster. Unlike k-mean, the number of clusters is not defined a priori but depends on the parameters chosen.

DBSCAN was extended into a hierarchical clustering algorithm called HDBSCAN by Cambello et al. [23], using a technique to extract a flat clustering based on the stability of clusters. The algorithm is divided into five steps. It first transforms the space according to density. Secondly, it builds the minimum spanning tree of the distance weighted graph. Thirdly, a cluster hierarchy of the connected components is constructed. Then, the cluster hierarchy based on the minimum cluster size is condensed. Finally, it extracts the stable clusters from the condensed tree. Additionally, an outlier scoring is also integrated to HDBSCAN. After the clustering process, it is possible to consider as outlier the elements that fall outside the clusters, i.e. the elements that are far from any cluster. An algorithm called Global-Local Outlier Score from Hierarchies (GLOSH) is used and gives a score between 0 for nominal and 1 for outliers [23]. It compares the density of a point to the density of any points in the associated current and child cluster. Samples with substantially lower density than the cluster density are likely to be considered as outliers.

In the aeronautical field, clustering techniques have been used for example to analyse air traffic flows [26, 27], or to monitor air-spaces [28].

Dimensionality Reduction Dimensionality reduction consists of taking data from a large dimensional space and replacing it with data from a smaller dimensional space. The reason why such an operation is useful is that smaller-dimensional data can be processed more quickly and because this operation is crucial to combat the scourge of dimensionality. The scourge of dimensionality is a phenomenon appearing when the number of dimensions increases. It implies that the volume of space grows rapidly and the data becomes isolated and scattered. This is problematic for methods, such as supervised learning, that require a significant amount of data to be valid, making them inefficient or even ineffective.

It is usually divided into two groups of techniques called feature selection and feature extraction. Feature selection consists in selecting a subset of relevant features generally through a searching phase and a scoring. Feature extraction consists in creating new and more relevant variables based on the initial variables. It exists different algorithms among which, the Principal Component Analysis (PCA [29, 30], the Singular Value Decomposition (SVD) [31], the Independent Component Analysis (ICA) [32], the Non-negative Matrix Factorization (NMF) [33], or the Latent Dirichlet Allocation (LDA) [34]. These algorithms have also been extended to the study of functional data with the so called Functional Principal Component Analysis. More details will be given

in the Functional Data Analysis section.

In the following paragraph, the PCA methodology is explained. PCA is a powerful statistic method that summarizes a significant amount of data information by creating new variables as the linear combination of existent variables. It is an orthogonal projection that concentrates the majority of the data variance in the first components of the redescription space. It enables simpler representation and analysis of complex or even large dimension variables.

Let $x_i, i = 1, \dots, p$ be a sampling of p statistic variables observed over n samples. A weighting vector ξ such like the redescription of x_i is f_i is defined as:

$$f_i = \sum_j \xi_j x_{ij} = \xi^T x_i \quad (2.1)$$

In practice, the redescription space is defined by using a projection on the eigenbasis space of the observations covariance matrix ordered by decreasing eigenvalues.

Application to anomaly detection In the context of this thesis, the theme of anomaly detection is preponderant. Anomaly detection, sometimes referred to as outlier detection, involves finding samples in a data-set that do not appears to be similar to the majority of the others. Various survey have been published in the context of anomaly detection, giving a complete overview of possible techniques [35, 36, 37]. Anomaly Detection can be performed by unsupervised learning using the tasks or combination of tasks detailed above.

A first approach consists in using probability density estimation to locate very low density areas assimilated to anomalies. The simplest models are Gaussian models, the belonging to the model is tested through statistical tests such as box-plot [38] or Grubb tests [39, 40, 41]. More complex models were also used such as Gaussian mixtures [42, 43], or non-parametric models, which do not assume prior knowledge on the data distribution with histogram analysis [44, 45] or kernel density estimation [46, 47, 48].

Clustering based Anomaly detection is also widespread. An anomaly detection algorithm called Orca [49] was developed using k-nearest neighbour approach with a modified pruning technique. Anomaly detection is also found through centroid-based algorithms such as k-means [50, 51], which however implies to provide the number of clusters in advance, or k-medoid in the fraud detection [52]. Other clustering algorithms such as hierarchical clustering have been used for anomaly detection [53] or in a phase preceding anomaly detection [54, 55]. Hierarchical clustering aims at building a cluster hierarchy in order to give a finer representation of the data structure. This structure is often represented in the form of a dendogram. In addition, density based algorithm such as DBSCAN were used to perform anomaly detection.

Finally, dimensionality reduction techniques are generally used in the upstream

phase in order to extract features, on which it is easier to apply clustering, density estimation or similarity measures since it is usually low-dimensional space. In particular, PCA was used for the traffic anomaly detection [56, 57] or network anomaly detection [58, 59]

In the field of aviation safety, different research has already been conducted on the detection of anomalies. In particular, Li et al. have developed an anomaly detection algorithms based on dimensional reduction (PCA) and cluster outlier detection [60, 61, 62]. This approach has shown very good results in post-opt analysis framework and the methodology can be applied to large data sets. However, it requires a complete trajectory and cannot be applied directly for real-time analysis. In addition, it is possible to extended the philosophy by integrating functional properties of the trajectory data within the Functional Data Analysis (FDA) framework explained further in the section 2.3.

2.2.2 Supervised Learning: Creating Prediction Models

Supervised learning consists in building a model that estimates output variables from input variables. It is called regression when the output space is continuous and classification if it is discrete. It can be seen as an approximator h of an unknown function f of which only samples are observed :

$$f : \mathcal{X} \longrightarrow \mathcal{Y} \quad (2.2)$$

This function f goes from input space \mathcal{X} to output space \mathcal{Y} . The approximator h usually depends on adaptive parameters θ . It is then noted h_θ . The learning step is performed on a learning set \mathcal{D} which consists of sample pairs $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. This phase is a statistical optimization of the algorithm parameters θ in order to minimize the estimation error called true risk :

$$R(h) = \int_{\mathcal{X} \times \mathcal{Y}} l(h(X), Y) dP(X, Y) \quad (2.3)$$

where P is the data distribution and l is a loss function, such as the mean square error or the cross-entropy error. In practice, the distribution of the data is unknown, so the algorithm seeks to minimize the empirical risk over the training set :

$$R_{emp}(h) = \sum_{(X, Y) \in \mathcal{D}} l(h(X), Y) \quad (2.4)$$

Supervised learning models are multiple. There are kernel based algorithm such as Support Vector Machine (SVM) [16], decision tree based algorithm such as Classification And Regression Trees (CART) [63] or Random Forest, stochastic models like the

Gaussian Process Regressors [64], or even bio-inspired models like Neural Networks (NN) [25], each of them having specificities. The SVMs have shown good performance on small learning sets [16]. Techniques based on decision trees bring some explicability to the results [63]. Neural networks and their dedicated architectures offering high performance have revolutionized some fields such as image processing thanks to Convolutional Neural Network (CNN) [65], or temporal object processing such as speech recognition thanks to Recurrent Neural Networks (RNN) [66].

Anomaly detection with Supervised Learning Supervised learning methods can also be used to perform anomaly detection. This generally follows a semi-supervised philosophy [67, 68]. A model is built on the nominal elements, if the data does not follow the model then it is detected as an anomaly. Among the algorithms, there are the one class svm [69] that builds a boundary of normal elements, an element beyond the boundary is considered outlier, auto-encoders whose reconstruction error gives a metric on the similarity with the learned model [70, 71], or Generative Adversarial Networks (GAN) whose discriminant and generator can be used to perform anomaly detection [72, 73, 74]. This methodology will be further detailed in Chapter 4.1.

In particular, one class SVM have been apply to aircraft on-board anomaly detection with the Multiple Kernel Anomaly Detection algorithm (MKAD) [75]. This is a post-operational analysis algorithm that presented a breakthrough in the analysis of heterogeneous data and has been used to detect anomalies in aircraft approach on-board parameters.

Aircraft parameter estimation with Supervised Learning Nowadays, air traffic controllers and air traffic managers still rely on ground accessible data to understand and analyze aircraft behaviors and to evaluate air traffic management system performance. Nevertheless, aircraft flight dynamic is defined with various on-board parameters. Therefore, the estimation of these parameters could be beneficial to enhance the air traffic system performance and evaluation. Estimation, sometimes called regression, is a well known mathematical problem. It is divided into two main approaches: the physical model-driven estimation, and the data-driven regression within the supervised learning framework.

Physical model-driven parameter estimation consists in using physical equations and models to estimate target features. Usually, differential equation problems are solved or assumptions are taken to give simplified estimations. In aviation, these models have been studied for a long time in different areas such as aerodynamic [76], or fuel-efficiency [77]. These models may need significant computational resource.

On the other hand, data-driven models or supervised learning models, try to estimate the underlying physical models using available data. The learning phase can be

costly in computational resource, however, the prediction phase is then instantaneous, which presents interesting properties in real-time operations. Many studies have been conducted to predict on-board parameters using ground-based surveillance systems. Estimation of the True Air Speed and the wind was performed using radar data [78, 79]. Estimation of the aircraft mass was lead using Bayesian inference methods [80], and other models such as Gaussian Process regression [81] and tree-based classification [82] was studied to predict the aircraft fuel flow rate. Finally, other algorithms such as neural networks have also been investigated to estimate the fuel flow rate with access to many on-board parameters [83].

2.2.3 Reinforcement Learning

Reinforcement learning differs from the previous two approaches in that it aims to determine an optimal transition policy in an environment [84, 85]. The user transits between different states of the environment and is given a reward at each stage. The goal is to find the policy that maximizes the rewards over the entire episode. Reinforcement learning comes from optimal control but uses data and simulations to build the models.

It has been applied in various fields [86, 87] in particular in robotics. More recently, a programm called AlphaGo was developed and has proven to beats human beings at the go board-game [88]. Another program called AlphaStar, beats human beings at the real time strategy video game Star Craft II [89].

The use of reinforcement learning algorithms opens up a number of possibilities for ATM, in particular it can help developing decision support tool for ATCos [90, 91, 92] and could give directions towards the automation of certain tasks [93].

Machine Learning presents many opportunities to detect atypical approaches. However, the techniques developed have very rarely integrated functional data analysis methods and do not directly integrate a continuous localization of anomalies leading to real-time extensions. In addition, parameter prediction via supervised learning models provide opportunities to analyse, understand and leverage trajectories with the aim of improving the ATM system.

2.3 Functional Data Analysis: statistical tools applied to functional data

The objective of this section is to describe how the tools developed in the field of Functional Data Analysis (FDA) can be applied to radar data. The underlying nature of radar tracks is functional. It connects an interval, most often the time to fix ideas, to a state space (position, speed,...). In practice, radar trajectories are observed in a discrete way. It is therefore necessary to have adapted tools to reconstruct functional data from discrete observations. It may also be necessary to smooth the data if they are noisy.

This section presents some methods from the literature allowing the representation of trajectories as functional objects as well as a statistical analysis tool adapted to these data: the functional principal components analysis.

2.3.1 Introduction to Functional Data

The analysis of functional data is one of the fields of statistics which aims to study data from random variables with values in a functional space [94]. These methods began to develop in the 1980s thanks to the work of Deville [95] and Dauxois [96, 97]. They have evolved a lot in recent years and especially in the 2000s thanks to the seminal works of Ramsay and Silverman [94, 98, 99]. Theoretical aspects and applications have also been published by Ferraty [100, 101].

There are many applications for functional data. Ullah et al. [102] survey many application domains for functional data analysis and highlight the multidisciplinary challenge of these methods. In particular, they identify applications in biomedical, medical, biological, financial, aeronautical and demographic fields. In aeronautics, functional data analysis is widespread. In his thesis [103], Gregorutti uses data from flight recorders and proposes a tool for analyzing safety and predicting the risk of long or hard landing type events. It exploits the data by a functional analysis and a decomposition on wavelet bases. It also presents methods for the selection of variables from supervised learning as well as random forest algorithms for prediction. Suyundikov [104] proposes a partitioning of multivariate functional data from approach trajectories using functional principal component analysis in Sobolev spaces. It uses the properties of wavelet databases to simply represent its trajectories in a Sobolev space and then performs the principal component analysis and data partitioning. Tastambekov proposes in his thesis [105], methods for aircraft trajectory prediction based on local functional regression. He uses wavelet based decomposition and classification algorithms such as the k-nearest neighbor algorithm. In all this work, Functional Data Analysis made it possible to extract a structure and an adequate representation of aircraft trajectories.

2.3.2 The functional nature of aircraft trajectories

Functional Data Analysis considers data as functions. In aeronautics, trajectories are naturally smooth, hence the existence of the derivatives of the curves must be assumed to ensure trajectory continuity and smoothness. Trajectories are therefore usually modeled in a Hilbert space \mathcal{H}^m of square integrable functions where all the derivatives until the order m are square integrable. Let $f, g \in L^2(J)$ be two functions, the inner product associated with the Hilbert space is:

$$\langle f, g \rangle = \int_J f(t) g(t) dt$$

The various FDA methods focus on the statistical analysis of a set of curves. In practice, discretizations of these functions at time t_j , $j = 1, \dots, M$ are observed. In aeronautics, approach trajectories map a time interval to a state space \mathbb{R}^d . Curve radar data are discretely recorded every 4 seconds. Trajectories are observed on a time interval $[0, T_i]$, which can be different for each trajectory. For this reason, the first step of Functional Data Analysis consists in recovering the functional nature of curve data from discretized observations by using a decomposition on a functional basis. A system of basis functions is a set of known independent functions ϕ_k . A particular property of basis functions is that every other function can be approximated with a linear combination of a sufficient number K of these basis functions. There are different basis function systems in the literature such as polynomial basis, Fourier basis, wavelet basis or smoothing spline basis [99]. Let $\mathcal{B} = \{\phi_1, \phi_2, \dots\}$ be a basis function system on a functional infinite dimension space, and X a functional variable in this space. The functional representation of X in the basis is:

$$\hat{X}(t) = \sum_{k=1}^{+\infty} c_k \phi_k(t) = \mathbf{c}^T \boldsymbol{\phi} \quad (2.5)$$

Then, it is assumed that when k is greater than of a sufficiently large integer N , residuals are considered as noise. An object of infinite dimension Hilbert space can then be manipulated as a finite dimension vector: the vector c_k of the decomposition coefficients. The order N approximation \hat{X} of X is:

$$\hat{X}(t) = \sum_{k=1}^N c_k \phi_k(t) = \mathbf{c}^T \boldsymbol{\phi} \quad (2.6)$$

where \mathbf{c} is the vector of length K with the c_k coefficients, and $\boldsymbol{\phi}$ the vector of ϕ_k basis functions. To estimate the c_k coefficients in ϕ_k basis, the ordinary mean square method is recommended. It consists in solving the following minimization problem:

$$\text{Min}_{c_k} \sum_{i=1}^M \left\| y_i - \sum_{k=1}^N c_k \phi_k(t_i) \right\|^2 \quad (2.7)$$

where $y_i = x(t_i)$ is the observation of a curve x at time t_i . Usually, a smoothness regularization penalty is added to ensure the smoothness of the functions. In general, the smoothness is ensured until the order m . The penalty is therefore:

$$\int_J \|D^m \mathbf{c}^T \boldsymbol{\phi}(s)\|^2 ds \quad (2.8)$$

where D^m is the order m derivative operator. With this model, the minimization problem becomes:

$$\text{Min}_{c_k} \left(\sum_{i=1}^M \|y_i - \sum_{k=1}^N c_k \phi_k(t_i)\|^2 + \lambda \int_J \|D^m \mathbf{c}^T \boldsymbol{\phi}(s)\|^2 ds \right) \quad (2.9)$$

2.3.3 Functional Principal Component Analysis

Functional Data Analysis proposes extension of multivariate statistical methods to the functional setting. In particular, Multivariate Principal Component Analysis (PCA) [30] was extended to functional data.

PCA was extended to the functional setting, called Functional Principal Component Analysis (FPCA) by Deville [95] and Dauxois [96, 97]. When data are functions sampled from an underlying stochastic process, FPCA enables dimensionality reduction by estimating the Karhunen-Loève decomposition. Let's consider a sample of trajectory $X_{i=1\dots L}$. With this decomposition, each trajectory can be represented by its decomposition coefficients on the principal component basis and considered as a small dimension vector. This process is described in Equation 2.10 known as the Karhunen-Loève expansion:

$$X(t) = \bar{\gamma}(t) + \sum_{j=1}^{+\infty} b_j \phi_j(t) \quad (2.10)$$

It consists in considering each curve X as the weighted sum, where b_j are real-valued random variable, $\bar{\gamma}$ the empirical mean curve estimation and ϕ_j the principal components determined using the covariance operator of \mathbf{X} :

$$\hat{H}(t, s) = \frac{1}{L} \sum_{i=1}^L X_i(t) X_i(s)^T \quad (2.11)$$

By solving the Fredholm's functional equations:

$$\int_J \hat{H}(t, s) \phi_i(s) ds = \lambda_i \phi_i(t) \quad (2.12)$$

In other words, ϕ are the (vector-valued) eigenfunctions of the covariance operator with eigenvalues λ_i . The random variables b_i are called the scores of the decomposition. The difference with simple functional basis decomposition is that the functional principal components decomposition maximizes the variance on the first components. It thus provides the representation with the N minimum for a given noise level.

Figure 2.5 illustrates a very simple example. Let's consider a set of curves with a constant mean $\bar{\mu}$. These curves present only two possible variations at time t_a and t_b . The FPCA decomposition gives two principal components functions ϕ_1, ϕ_2 . Each original curve can be written as a linear combination : $\bar{\mu} + b_1 \cdot \phi_1 + b_2 \cdot \phi_2$, and represented by their principal score vector (b_1, b_2) .

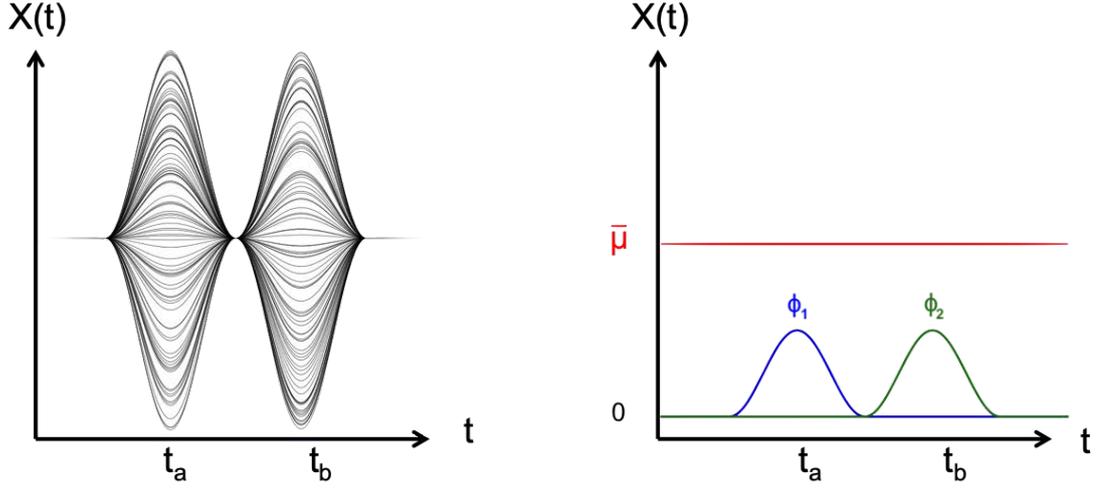


Fig. 2.5 Simple FPCA decomposition basis illustration

2.3.4 Curve Registration

In the functional data model, curves present two types of variability: phase variations and amplitude variations. This well-known literature model is referred to as the curve registration. The objective of functional principal component analysis is to capture the amplitude variation. When data are not registered, the principal component functions also capture the phase variability. To this end, it is recommended to use registration [106]. Among the different possible registration algorithms, there is the landmark registration. It uses landmarks that are characteristic points that are present in every curve. For example, these could be points where the derivatives are equal to zero. For aircraft trajectories, it is usual to use turning points. The registration process will then align together the corresponding landmarks of each curve using warping functions. Warping functions can be defined by a monotonic interpolation of a landmark's desired time in function of its original time. For each trajectory γ_i the associated warping function h_i from $[0, 1]$ in $[0, 1]$ is computed such that each landmark is aligned for all the registered curves. The general formula for registration is:

$$\gamma_i^*(t) = \gamma_i(h_i(t)) \quad (2.13)$$

Figure 2.6 illustrates the registration of one curve parameter. The registration uses landmarks based on the curve turning points, which aligns curves together on a reference time t_{ref} . The impact on explained variance could be very large. It usually increases the variance explained by the first components and gives a more consistent representation. It is important to underline that registration does not change the 2D or 3D curve shape but changes the speed at which we run through the curve. In addition, one must apply the same registration to all curve parameter dimensions. In the

context of trajectory data, and in particular for the energy management analysis during approach, the registration will not be applied, because on the speed changes induced by such a process, but a starting and a final point consistent for all the trajectories will be considered. For example, the last 25NM before the runway threshold, of each trajectory.

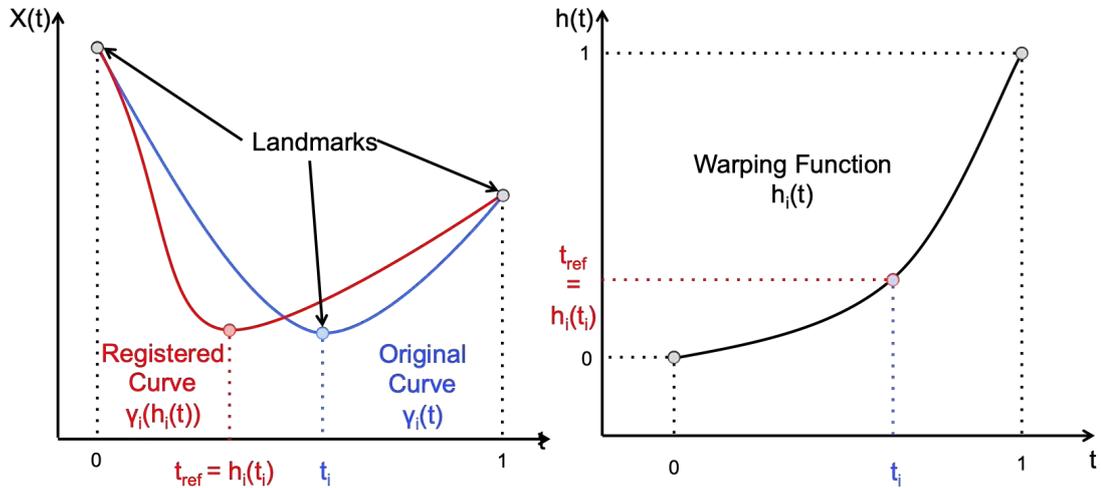


Fig. 2.6 Curve registration process and warping function illustration

The integration of functional data models through decomposition on functional bases or the use of statistical tools such as functional principal component analysis seems relevant in the context of radar trajectory analysis. Indeed, it provides a robust framework of representation, and dedicated tools for the dimension reduction of functional data. In addition, these methods have had very little use in interactive manipulation, modification or generation of data and thus provide relevant complementary research directions.

2.4 Thesis Framework

2.4.1 Data

This thesis is positioned in a data-oriented point of view. The objective is to build models from data. These models should allow to analyze and detect samples with atypical behaviors. The data must be available from the ground because the thesis is positioned from an ANSP point of view. Since the beginning of air traffic control, different methods have been developed to allow downlink transmission of aircraft parameter data. In addition to the primary radars that give the position of the aircraft, there are two main categories of technologies that enables downlink transmissions. First, the interrogation systems, such as secondary radars, interrogate aircraft transponders, which returns information such as the position and altitude or the aircraft identification. Second, the broadcast systems such as the Automatic Dependent Surveillance-Broadcast (ADS-B),

where the aircraft computes its position itself using Global Navigation Satellite System (GNSS) and broadcast it periodically.

The available or directly computed parameters are the aircraft 3D position, its ground and vertical speed, and its track. Other parameters are available with an extension of secondary radars called radar mode S and with ADS-B receivers. However, all aircraft are not yet equipped and therefore these parameters are not fully available today. Therefore, a constraint of this thesis is to use as input parameters those always available by an ANSP using secondary radar (3D Position, track, ground and vertical speed). In France, the secondary radars have a refresh rate of 4 seconds. Consequently, all the data in this thesis are re-sampled to fit with this refresh rate.

2.4.2 Research Challenges and Contributions

This thesis takes up different challenges and aims to respond to different research problems.

The review of operational and scientific techniques for the detection of anomalies has shown different perspectives of research. In the operational field, detection techniques are generally based on exceeding operational thresholds that do not take into account the variation of the studied parameters. The scientific literature presents many solutions for FDR data using both kernel-based or cluster-based techniques. These techniques generally do not present a direct localization of atypical events and require the whole trajectory to be applied. This does not allow a real time extension. Moreover, anomaly detection does not directly involve any link with safety. It is therefore important to study the potential links between an anomaly detection algorithm and known safety events, which leads to the first two objectives of the thesis.

- Develop a post-operational anomaly detection algorithm that locates atypicalities and should allow a real time extension.
- Analyze the links between the detection anomalies and the known security events.

In addition, studies could be carried out to show how these technologies enhanced the monitoring of large data sets.

The recent literature of machine learning shows numerous research opportunities. In particular the Generative Adversarial Network (GAN), offers a new solution for anomaly detection and data generation and has not yet been investigated in the context of aircraft trajectories. In addition, supervised prediction models such as Long Short Term Memory (LSTM) networks offer relevant alternatives to the estimation of aircraft parameters. This requires access to FDR data, but the flexibility of such models opens up interesting perspectives for improving the analysis of radar trajectories and the

evaluation of ATM system performance. The second objective of this thesis is to evaluate the use of methods such as GAN or LSTM networks in order to propose alternative methods for the detection of anomalies or the estimation of aircraft parameters.

Literature around FDA also shows opportunities as a result of the work carried out by ENAC colleagues working on the Information Visualisation [107]. Indeed the functional principal component analysis provides a representation space allowing manipulation, deformation and regeneration of functional data sets. In the context of machine learning, alternatives for the generation and data sets augmentation are relevant. Similarly, data pre-processing is an important phase, interactions such as brushing can facilitate data analysis or selection. Thus, a third objective consists in defining pipelines dedicated to the interactive manipulation, generation and deformation of functional data sets.

Finally, real-time detection is a more complex task, as it is often subject to safety implementation constraints. However, it has been shown that glide inception from above detection systems exist at Roissy Charles de Gaulle. Therefore, this thesis aims to investigate potential detection tools for atypical approaches not only constrained to the altitude parameter.

The objectives of this thesis are summarized below :

- Develop a post-operational atypical trajectory detection tool that is able to localize atypical segments. The algorithm should allow a real time extension.
- Study the link between atypical flights and safety related events, and highlight how this methodology enhanced the monitoring of atypical events over large data sets.
- Investigate the use of machine learning techniques such as GAN or LSTM networks to provide alternatives anomaly detection and to enhance the analysis of radar trajectory.
- Propose dedicated frameworks using Principal Component Functional Analysis to support interactive manipulation, modification and generation of functional trajectories.
- Investigate real-time atypical trajectory detection in order to propose directions toward novel situation awareness systems.

2.5 Thesis Outline

The thesis is divided into four chapters, which are summarised as follows. It should be noted that a broad overview of the main topics covered in this PhD thesis has already

been presented in the previous sections. A more in-depth and specific overview for each individual topic is included at the beginning of the chapter dealing with it if needed.

- Chapter 3 presents and describes a developed post-operational atypical trajectory detection methodology. The methodology is confronted with the flight data analysis performed by airlines. Finally the methodology is applied to monitor energy management in the context of the COVID-19 crisis .
- Chapter 4 focuses on the use of machine learning and Functional Data Analysis techniques in order to enhance the analysis, the manipulation and the generation of radar data. First, the use of Generative Adversarial Network for generation and anomaly detection in aircraft approach trajectories is presented. Second, the use of supervised learning techniques to predict aircraft flight deck parameters using radar data is explored.
- Chapter 5 explores novel interactive manipulation and generation techniques using Functional Principal Component Analysis. This research was lead in collaboration with the ENAC colleagues working on Information Visualization (InfoVis). First, a pipeline to support data manipulation and generation using FPCA is proposed. Second a novel interactive brushing techniques to manipulate trajectory based data sets is presented.
- Chapter 6 extends to real time the post-operational methodology presented in Chapter 3. Additionally, a suggested trajectory generation process is proposed and the methodology is illustrated over several air crashes.

CHAPTER 3

Post-Operational Atypical Approach Detection using Unsupervised Learning and Functional Principal Component Analysis

”Lorsqu’on le tient par la partie sporadique, ou boulière, le fenouil est un objet redondant.”

Karadoc - Kaamelott, Livre IV, Unagi IV

Contents

3.1	Atypical Trajectory Detection Algorithm	29
3.1.1	Pre-processing and dimensionality reduction	29
3.1.2	Outlier Coefficient	32
3.1.3	Identifying Atypical Trajectories	36
3.2	Application to the aircraft atypical approaches detection and result with airline flight data monitoring and events	37
3.2.1	Energy problem and algorithm parameters	37
3.2.2	Atypical approach detection comparison with flight data monitoring and events	39
3.2.3	Discussion	47
3.3	Flight safety during Covid-19: A study of Charles de Gaulle airport atypical energy approaches	49
3.3.1	Data and Methodology	49
3.3.2	Observations	50
3.3.3	Discussion	57
3.4	Conclusions	59

This chapter presents and describes the developed post-operational atypicality detection methodology. It is divided into three parts: first the algorithm is presented and illustrated, then the methodology is confronted with the flight data analysis performed by airlines, finally an example of application in the context of the coronavirus crisis is detailed.

In the framework of this thesis, the research focuses on the analysis and detection of aircraft atypical approaches from ground side. The objective is not only to detect atypical flights but also to locate areas of atypicality. These atypicalities must include atypicalities due to extreme values and also those resulting from atypical variations. In addition, the initial hypothesis is that it is desired to detect flights that do not behave like others.

The functional data analysis framework presented in the previous chapter provides a robust framework for the representation of functional data such as radar data which includes both values and variations. It will be applied as a pre-processing of the data before the actual anomaly detection phase. As a contribution, the anomaly detection phase is applied not to a flight or a complete flight phase as usually in the literature but on a sliding window allowing to obtain a fine localization of the atypical areas. In addition, the reduction of dimensionality is achieved through the Functional Principal Component Analysis process. Finally, a dedicated outlier scoring is proposed. The methodology is presented in a generic way, but will subsequently be applied to the detection of atypical aircraft energy management during approach in the context of the covid crisis, and confronted with airlines flight data monitoring.

This chapter is in part the result of different publications: a first conference publication presenting the methodology [108], which was selected for an extended version into a special issue journal paper [109]. A second conference publication was made, presenting the tool developed with the methodology and the results of the study with the airlines flight data monitoring [110]. And finally, a journal paper was published and presents a study of atypical approaches at Charles De Gaulle Airport during the COVID-19 crisis [111].

3.1 Atypical Trajectory Detection Algorithm

3.1.1 Pre-processing and dimensionality reduction

Suppose that a sample $X_{i=1\dots L}$ of trajectories $X(t)$ is observed. Each trajectory is an observation of L variables at timestamps $t_{i,j,j=1\dots n}$, the timestamps may be different from one trajectory to another. It is assumed that the timestamps t_{i0} and t_{in} are identical for all trajectory i , the assumption is justified since it is usually desired to observe the first/last minutes or nautical miles of a trajectory. In the case where this assumption is not realized it is possible to apply the so-called registration process [94].

First of all, a pre-processing of the data is performed. A Z-normalization of each variable is applied to prevent one variable from capturing the majority of the variance in the FPCA process [94]. Z-normalization consists of subtracting the mean and dividing the result by the standard deviation to return to a centered normal distribution. Then, considering a window width ν and a window shift δ , the data is divided into several

windows on each of which the functional principal component analysis and anomaly detection will be applied.

This sliding window should allow to answer several problems. First of all, it should allow to detect both local events and global event. Indeed, the models are built on restricted segments, and the combination of all shifts would give a global view of the atypicality and should allow the detection of atypicalities that are more global or that are maintained over time. The latter are the ones that may pose safety problems if they persist when the aircraft approaches the runway. Secondly, it opens the way to real-time detection, which will be discussed in Chapter 6. Indeed, the models thus constructed can obviously be applied in post-ops but also can be applied in real time in a dedicated framework.

The next step consists in applying a FPCA decomposition. It implies first to decompose the trajectory sub-window into a functional spline basis. This allows in particular to remove the problems of different timestamps from one trajectory to another and also allows to modulate a smoothing of the trajectories if necessary. Following the notation presented in the Chapter 2.3, a function X can be modelled using reproducing kernel theory [112], and expressed as a linear combination of kernels K :

$$X(t) = \alpha_0 + \alpha_1 t + \sum_{j=0}^{j=n} \alpha_{j+2} K(t, t_j) \quad (3.1)$$

Where α_j are real-value coefficient chosen to minimize equation 2.9, and K a cubic spline kernel defined as :

$$K(t, s) = \frac{\max((t - s)^3, 0)}{6} - \frac{t^2 s}{2} - \frac{t^3}{6} \quad (3.2)$$

X are square integrable since they are in the Hilbert space \mathcal{H}^m , with $f(0) = f'(0)$. Consequently they obey to the so-called reproduction property:

$$X(t) = \int_J \frac{\partial^2 K}{\partial s^2}(t, s) X''(s) ds \quad (3.3)$$

Let $a = (\alpha_0, \dots, \alpha_{n+2})$ be the vector of unknown coefficients to be minimized to solve the problem in Eqn. 2.9, with $\mathbf{c} = a$. Using the expression of X as a sum of the kernel functions in Eqn. 3.2 and the property in Eqn. 3.3, the minimization error in Eqn. 2.9 can be written as a matrix-vector:

$$e = \|X - Ga\| + \lambda a^T \tilde{G} a \quad (3.4)$$

With the matrix G and \tilde{G} being:

$$G = \begin{pmatrix} 1 & t_1 & K(t_1, t_1) & \dots & K(t_1, t_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & K(t_n, t_1) & \dots & K(t_n, t_n) \end{pmatrix} \quad (3.5)$$

$$\tilde{G} = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & K(t_n, t_1) & \dots & K(t_n, t_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & K(t_n, t_1) & \dots & K(t_n, t_n) \end{pmatrix} \quad (3.6)$$

To minimize Eqn. 2.9, the gradient from Eqn. 3.4 is equated to zero, which leads to:

$$(G^T G + \lambda \tilde{G})a = G^T X \quad (3.7)$$

This corresponds to a linear system with positive-definite matrix of unknown vector a . Besides, X curves often have a different number of points and are not observed at the same timestamps t_j . This implies a different expression for each curve. To solve the problem, the spline basis used in the calculation is fixed. Then, the best approximation of the curves is found by assuming that the spline kernels $K(t, \gamma_i)$ are located at p nodes $(\gamma_1, \dots, \gamma_p)$ in the study interval J . The problem is thus solved with the same linear system with the modified matrices:

$$G = \begin{pmatrix} 1 & t_1 & K(t_1, \gamma_1) & \dots & K(t_1, \gamma_p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & K(t_n, \gamma_1) & \dots & K(t_n, \gamma_p) \end{pmatrix} \quad (3.8)$$

$$\tilde{G} = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & K(\gamma_n, \gamma_1) & \dots & K(\gamma_n, \gamma_p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & K(\gamma_n, \gamma_1) & \dots & K(\gamma_n, \gamma_p) \end{pmatrix} \quad (3.9)$$

The \tilde{G} matrix is calculated only once since it depends only on the fixed locations of the nodes γ_i .

Then, it implies to apply the FPCA decomposition to the spline coefficients. Let $\Phi_K(t)$ be the vector of the p spline kernels introduced previously:

$$\Phi_K(t) = \begin{pmatrix} 1 \\ t \\ K(t, \gamma_1) \\ \vdots \\ K(t, \gamma_p) \end{pmatrix} \quad (3.10)$$

Each function X_i can be written with its a_i coefficient as:

$$X_i = a_i^T \Phi_K \quad (3.11)$$

The co-variance operator 2.11 becomes:

$$\hat{H}(t, s) = \frac{1}{L} \sum_{i=1}^L X_i(t) X_i(s)^T = \Phi_K(t)^T H_a \Phi_K(s) \quad (3.12)$$

with $H_a = \frac{1}{L} \sum_{i=1}^L a_i a_i^T$. Then the Fredholm's equation 2.12 can be written in the matrix form as:

$$H_\phi H_a H_\phi b_i = \lambda_i H_\phi b_i. \quad (3.13)$$

where $H_\phi = \int_J \Phi_K(s) \Phi_K(s)^T ds$

Finally, considering the Cholesky decomposition of $H_\phi = L^T L$, with L being a lower-triangular and invertible matrix, it leads to the following eigen problem:

$$(L H_a L^T) L b_i = \lambda_i L b_i. \quad (3.14)$$

which is solved for $L b_i$. The b_i coefficients of the principal curve ϕ_i are finally obtained by simple back-substitution (since L is lower-triangular).

At this stage the trajectories have first been normalized. Then they were cut into segments of the same window width. Then, the trajectory portions were decomposed on a spline basis and the resulting coefficients were fed into the functional principal component analysis algorithm giving the final coefficients.

3.1.2 Outlier Coefficient

The objective is now to give a coefficient that indicates whether or not the portion of the trajectory behaves like the other samples. In order to obtain a continuity and not to state a binary atypicality at this stage, the coefficient is supposed to be continuous between 0 and 1. The closer the score is to 1 the more the sample is atypical.

The coefficient was initially computed using a hierarchical clustering process: HDBSCAN (which does not assume the number of clusters a priori and is based on the distribution density) and the GLOSH algorithm as presented in the literature [23].

However, this algorithm has shown several limitations. First, it balances the different clusters by considering the maximum density of each cluster individually. This does not correspond to the desired detection of atypicality since if there is a cluster of lower density implying less widespread use, this must be taken into account in the calculation of the coefficient since the objective is to detect flights that don't behave like the others. Finally, in a practical way the use of HDBSCAN and GLOSH algorithms via the dedicated Python library showed potential non robust behaviors as illustrated in Figure 3.1. Different distributions were represented, in the first row a 2D Gaussian, in the second row two intermingled 2D Gaussians with the same number of samples, and in the third row two separated 2D Gaussians with an unbalanced proportion of samples (7 times less element for the extended Gaussian). In the columns different values of the minimum number of elements per cluster (the meta parameter requested by the HDBSCAN algorithm) have been tested. The results of the second column are the best and the most consistent with an atypical coefficient that would inform whether a point is in a distribution and would be given a score of 0 (green), or at the frontier of a distribution (score of 0.5 orange), or statistically outside the distribution (score of 1 red). However, it is observed, for the third row, non consistent behaviours and outliers within the distribution. In addition, the models created are not designed to be saved and reused later on other samples, which is necessary in particular for a real time extension.

Therefore, it is proposed to give an outlier scoring directly link to the probability density distribution and the maximum of density. Suppose that d is the probability density function, d_{max} the maximum density of the distribution, α an exponent coefficient, and x the current sample. The α -outlier coefficient c_α of x is define as follows:

$$c_\alpha(x) = \left(1 - \frac{d(x)}{d_{max}}\right)^\alpha \quad (3.15)$$

Let's now consider a normal 1D distribution $\mathcal{N}(\mu, \sigma^2)$ to fix the idea. The probability density is given by:

$$d_{\mathcal{N}(\mu, \sigma^2)}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}, \quad x \in \mathbb{R} \quad (3.16)$$

And therefore, since the maximum density corresponds to $x = \mu$, the c_α outlier coefficient is:

$$c_{\alpha, \mathcal{N}(\mu, \sigma^2)}(x) = \left(1 - e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right)^\alpha \quad (3.17)$$

Statistically, 95% of the data is within 2 standard deviations $[\mu - 2\sigma; \mu + 2\sigma]$ and 99.7% of the data is within 3 standard deviations $[\mu - 3\sigma; \mu + 3\sigma]$. The coefficients associated with the 2σ and 3σ frontiers and more generally $k\sigma$ frontiers have interesting

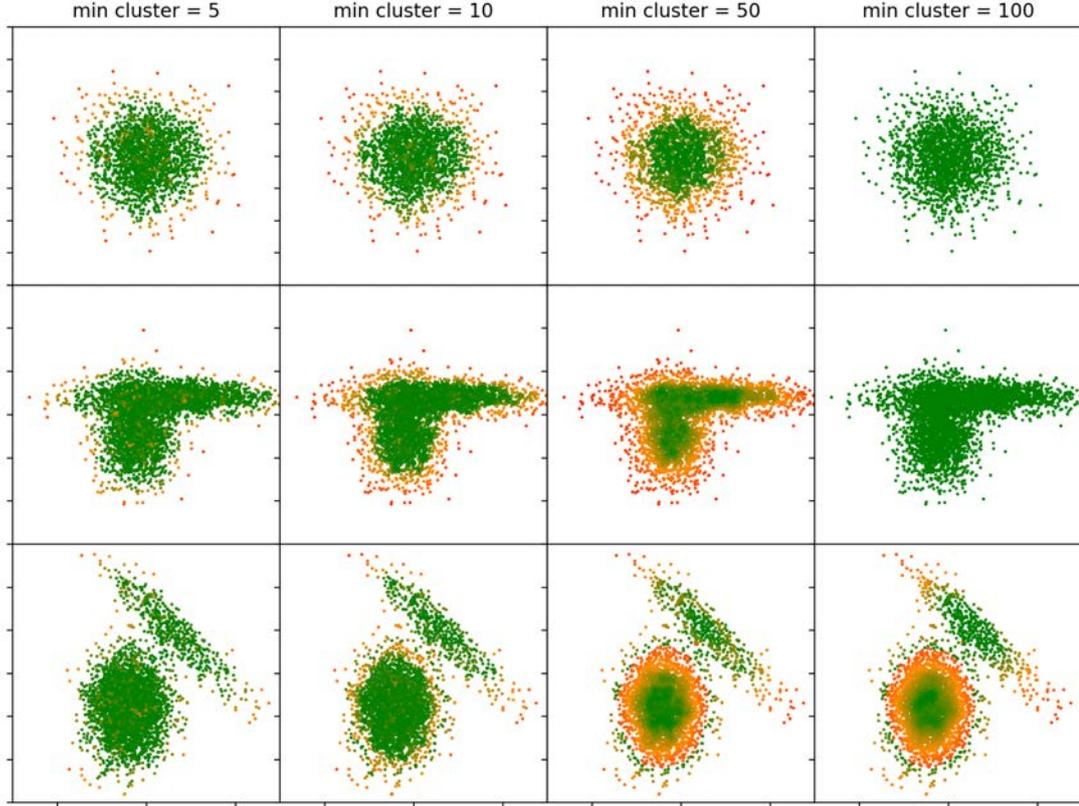


Fig. 3.1 GLOSH outlier scoring for different distribution and HDBSCAN parameter values

properties because they are independent of the parameters of the distribution and there values are :

$$c_{\alpha, \mathcal{N}(\mu, \sigma^2)}(\mu \pm 2\sigma) = (1 - e^{-2})^\alpha \quad (3.18)$$

$$c_{\alpha, \mathcal{N}(\mu, \sigma^2)}(\mu \pm 3\sigma) = (1 - e^{-\frac{9}{2}})^\alpha \quad (3.19)$$

Then, α can be chosen in order to position the two values in the $[0,1]$ interval. In this thesis, the extreme elements are targeted since it is assumed that they may induce safety issues. The 2σ frontier is a good candidate to underline the beginning of atypical behaviours, it should be close to 0. Besides, the 3σ should already be atypical with a high coefficient between 0.5 and 1. Empirically, $\alpha = 30$ is chosen because it gives a 2σ frontier coefficient of 0.013 and a 3σ frontier coefficient of 0.715. Other choices could be made depending on the desired behaviour.

Empirically, the FPCA coefficients appears to follow Gaussian mixture distributions. Therefore, let's now consider a more realistic hypothesis: two non-overlapping Gaussian distributions $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$. Let also suppose that $\sigma_1 < \sigma_2$, which trivially implies that the maximum of density corresponds to μ_1 . The α -outlier coeffi-

cient is the same for $\mathcal{N}(\mu_1, \sigma_1^2)$ but changes for $\mathcal{N}(\mu_2, \sigma_2^2)$ as follow:

$$c_{\alpha, \mathcal{N}(\mu_2, \sigma_2^2)}(x) = \left(1 - \frac{\sigma_1}{\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}\right)^\alpha \quad (3.20)$$

In Table 3.1 are illustrated different values of the coefficient at $2\sigma_2$ and $3\sigma_2$ frontier of $\mathcal{N}(\mu_2, \sigma_2^2)$ for different values of $\frac{\sigma_1}{\sigma_2}$ ratio, and $\alpha = 30$. The result are still consistent with the expected behavior. Distributions that are more spread out and therefore less dense compared to the main distribution will have a slightly higher coefficient because they are observed globally.

$\frac{\sigma_1}{\sigma_2}$	$c_{30, \mathcal{N}(\mu_2, \sigma_2^2)}(\mu_2 \pm 2\sigma_2)$	$c_{30, \mathcal{N}(\mu_2, \sigma_2^2)}(\mu_2 \pm 3\sigma_2)$
2	0.12	0.846
3	0.25	0.895
4	0.356	0.920
5	0.439	0.935

Table 3.1 \hat{c}_{30} coefficient at $2\sigma_2$ and $3\sigma_2$ frontiers for different values of $\frac{\sigma_1}{\sigma_2}$ ratio

Over the different experiments carried out with aircraft trajectories, the probability distributions resulting from the FPCA decomposition scores appear to be 2D or 3D Gaussian mixtures. This hypothesis is then assumed for the probability density estimation.

In practice, the probability density can only be estimated empirically. In this thesis, the Kernel Density Estimation algorithm [18] is proposed to be used with Gaussian kernels. The maximum of density, is then determined empirically within the given FPCA scores data set. Let's call the data set \mathcal{S} . Let \hat{d} be the estimated probability density through the kernel density estimation, \hat{c}_α is the estimated α -outlier coefficient with the following equation :

$$\hat{c}_\alpha(x) = \left(1 - \frac{\hat{d}(x)}{\max_{y \in \mathcal{S}} \hat{d}(y)}\right)^\alpha \quad (3.21)$$

In table 3.2 are illustrated the given values of \hat{c}_{30} considering different ratio of density compared to the estimated maximum of density. It is observed that a point with a density hundred times lower than the maximum of density has a coefficient similar to the 3σ frontier coefficient in the 1D-normal distribution hypothesis.

$\frac{\max_{y \in \mathcal{S}} \hat{d}(y)}{\hat{d}(x)}$	5	10	20	50	100
\hat{c}_{30}	0.001	0.042	0.215	0.545	0.74

Table 3.2 \hat{c}_{30} coefficient for different ratio between the maximum of density over the sample set and the considered point

In the following of the thesis, \hat{c}_α and c_α will be confounded for a simpler understanding. In Figure 3.2 the c_α coefficient is illustrated for different values of α over the same distribution than in Figure 3.1. The color is still a linear interpolation between green for a coefficient of 0, orange for a coefficient of 0.5, and red for a coefficient of 1. As expected, point at the frontier or outside of the distribution are given a non-zero coefficient in particular for $\alpha = 30$.

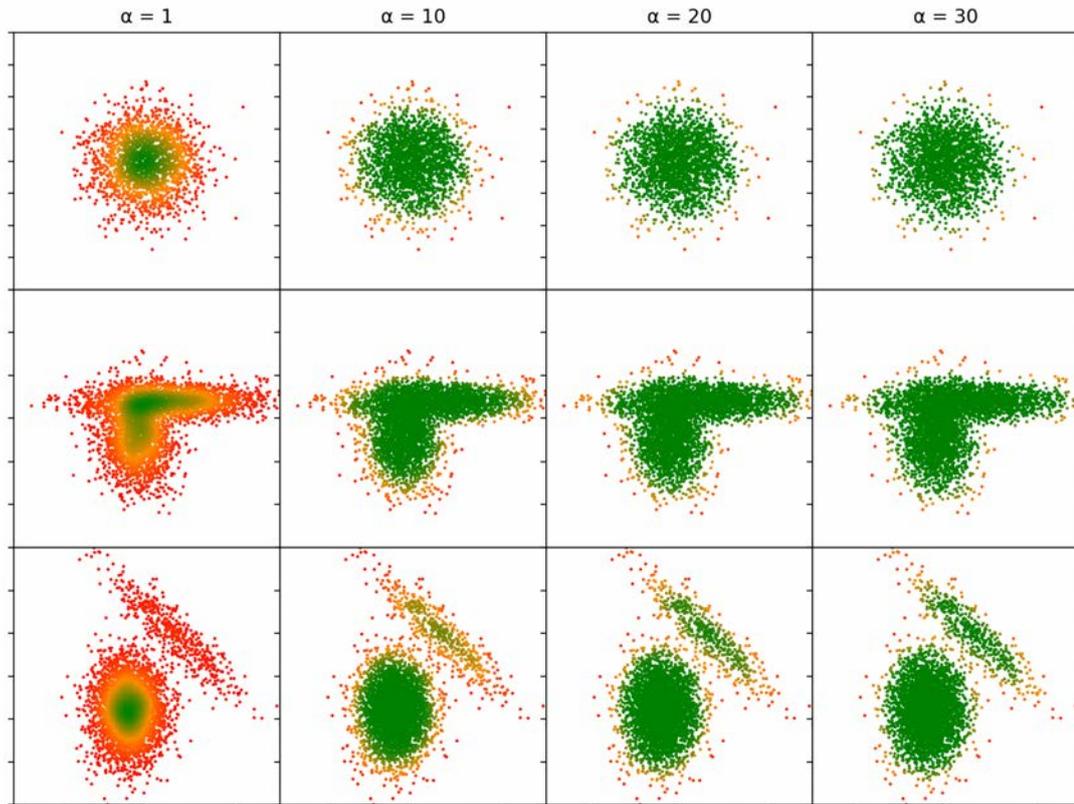


Fig. 3.2 α -outlier coefficient for different distribution and α values

Finally, the coefficient associated to a particular window slide is attributed to the timestamp corresponding to the middle of the slide. To enhance smoothness of the coefficient, a moving average over the hole sequence could be applied optionally. As a summary, the overall process on a window slide is illustrated in Figure 3.3.

3.1.3 Identifying Atypical Trajectories

The trajectory is now presenting a continuous coefficient determining local atypicality. In order to detect global atypicality, it is proposed to consider as atypical, trajectories whose coefficient c_α is higher than a threshold τ during more than a reference duration λ . A good candidate value for c_{30} , for example, is $\tau = 0.7$ equivalent to the 3σ frontier in the 1D Gaussian distribution hypothesis. In case this hypothesis is false, a score of 0.7 approximately corresponds to an area with a probability density 100 times lower

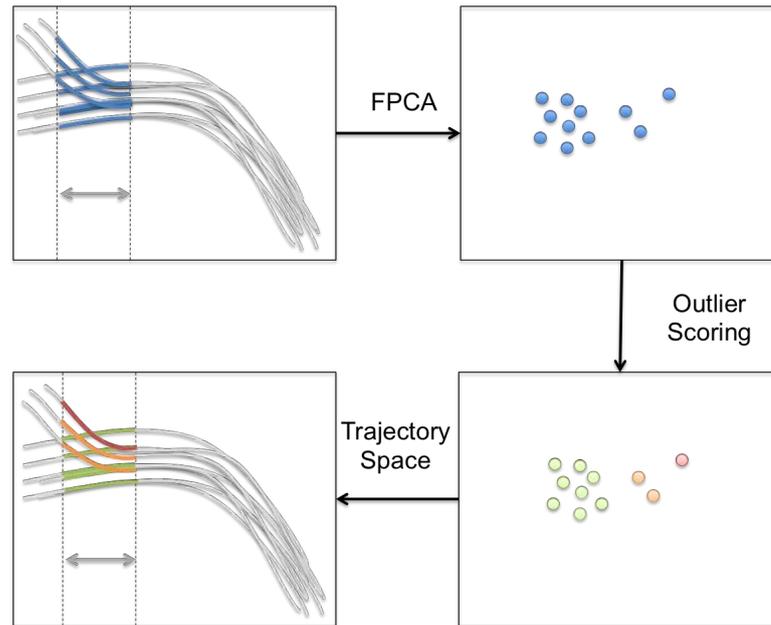


Fig. 3.3 Atypical approach detection methodology illustration

than the maximum data set probability density, which is still consistent. The value of λ depends on the type of behaviours to be detected and must be chosen to make operational sense. Potential choices in the case of aircraft approaches will be detailed in the following sections. As a summary, a trajectory, whose coefficient c_α is higher than τ for a duration higher than λ , will be refer to as (α, τ, λ) -atypical trajectories and more generally thereafter atypical when there is no ambiguity on the parameters.

3.2 Application to the aircraft atypical approaches detection and result with airline flight data monitoring and events

3.2.1 Energy problem and algorithm parameters

Focusing on the operational problem of detecting atypical approach trajectories, it has been underlined in chapter 2, the importance of energy management during approach and landing. For an aircraft with a given altitude and speed, the objective is that it should end up at the end of the runway with a controlled speed and that the energy management should have been done in the safest possible way. The principle and philosophy of the algorithm presented in the previous section, takes up the following relatively factual hypothesis: overall the pilot-controller system is safe. It is therefore quite intuitive to build a model on a historical set in order to detect the elements at the margin to study them and thus observe if they present safety issues.

One of the major issues for aircraft while landing is excess of energy. This may be found, for example, in situations where an aircraft is too high on glide slope owing to Glide Interception From Above, resulting in high potential energy, or in high speed

owing to tail wind in final approach, or late power reduction, both resulting in high kinetic energy. Low energy is also a problem as the aircraft is in near stall situations. The functional properties of the algorithm should make it possible to statistically detect both high and low energy, but also atypical energy variations and thus probably to anticipate cases of high and low energy.

It would have been possible to provide the different radar parameters of the trajectories to the algorithm, but the idea is to focus on the energy management of the aircraft. Therefore, a feature engineering is performed in order to give an estimation of the total energy of the aircraft in the runway coordinate system to the algorithm. Radar records do not contain aircraft mass, nevertheless since the study only concerns the last phases of flight before landing, the mass could be assumed to be constant during this phase. In addition, the distribution of weights for a type of aircraft during landing is generally a Gaussian not very dispersed, it can be assumed that for a majority of flights, the weight is relatively similar. There is also a substantial literature on the estimation of aircraft weight, this types of model could be incorporated in order to enhance the detection.

To estimate the Total Specific Energy E_T (per unit of mass), The height must be computed by correcting the altitude with the local QNH¹. The Total Specific Energy is defined as follow:

$$E_T = E_c + E_p; E_c = \frac{1}{2} \cdot (G_s^2 + V_z^2); E_p = g \cdot h \quad (3.22)$$

Where E_p is the specific potential energy, E_c the specific kinetic energy, G_s is the ground speed, V_z the vertical speed, h the height and g the gravity constant.

In general, literature methodologies index trajectories by time, however in the context of energy management it is relevant to index trajectories by the distance to the runway threshold. Indeed, the aircraft has this distance to dissipate energy and energy levels should be compared at a constant distance. Moreover, considering distance allows to take into account the problems of tailwind on final. Suppose that for two aircraft all parameters are constant except the tailwind component, an aircraft with a large tail wind component will be in a higher energy level and consequently can be detected. Finally, in the context where one objective is to extend the tool for real time detection, the computation of the remaining distance is exact when the aircraft is aligned with the extended runway center line, and could easily be estimated for the rest of the approach. This will be further detailed in Chapter 6. This methodology has the advantage that it can be applied regardless of the data source (radar, ADS-B or Flight Deck data) since the required parameters (aircraft position, altitude, ground speed and vertical speed) are

¹In aeronautics, the QNH is the "corrected barometric pressure for instrumental, temperature and gravity errors and adjusted to Mean Sea Level (MSL) according to the characteristics of the standard atmosphere". In practice, since QNH is given in reference to an airport runway threshold, when an altimeter is set to QNH, it displays the geographical altitude when the aircraft is on the runway threshold

available in each of the data sources.

Taking into account the parameters of the sliding window, what values should be assigned ? Concerning the window shift, the radar data generally have a refresh rate of 4s, for an aircraft at 180kts, it corresponds to 0.2NM. This is a good candidate since it almost corresponds to the refresh of the radar. Regarding the size of the window, it should be large enough to incorporate variation trends but not too large otherwise the local detection effect is lost. A value of 2NM corresponding to 10 radar points and around 40s of flight assuming an aircraft flying at 180kts is initially chosen.

3.2.2 Atypical approach detection comparison with flight data monitoring and events

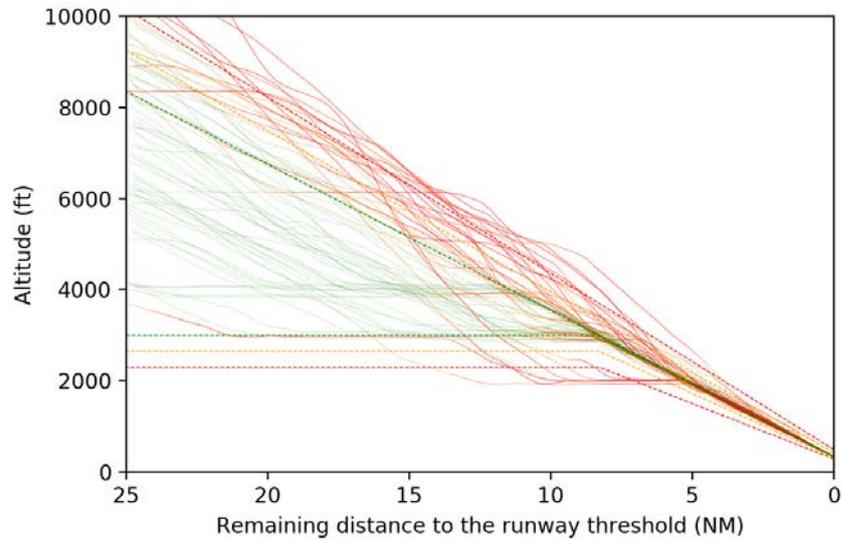
In order to confront the results of the algorithm with safety aspects, studies have been conducted with several flight safety offices of French airlines. They provided the on-board data from the flight data recorders (from which it is possible to extract the radar equivalent trajectory) as well as the analysis and the safety events from the flight data monitoring.

The first data set is composed of 16,471 A320 flights, the second data set is composed of 73,896 B737 flights. The data sets are composed of various airport destinations, but both airlines are based at Paris Orly Airport (LFPO), which implies that a significant number of landings are made at this airport.

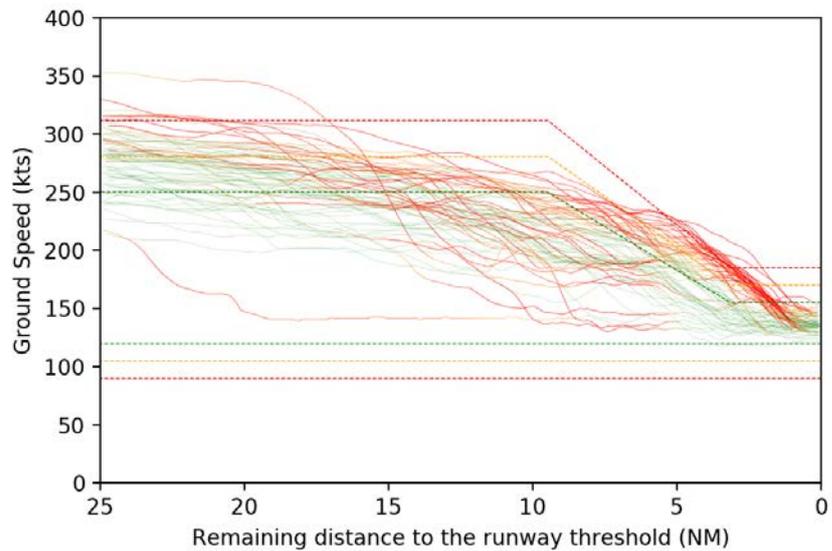
For each data set, an atypical model of the data is built with the parameters detailed above, and applied to the trajectories. It results in 827 atypical flights (1.1%) for the B737 data set, and 348 atypical flights (2.1%) for the A320 data set.

Observations In order to illustrate the typology of atypical flights detected, the results for A320 landings at Paris-Orly runway 26 is studied. Over 4401 flights the algorithm detects 71 atypical flights. Their altitude profiles and ground speed profile are illustrated in Figure 3.4. The color of the trajectories is linearly proportional to the atypical coefficient (green for a coefficient of 0, orange for a coefficient of 0.5 and red for a coefficient of 1). In addition, operational boundaries are represented by colored dashed line (green for nominal, orange for warning, and red for critical). These boundaries enable analysing the causes of energy atypicalities. For more detail on their definition see Appendix A.

The atypical flights present two main behaviours. Atypicalities localized before interception of the axis and glide path (left side of the altitude profiles in Figure 3.4a) are generally due to excessive potential energy and glide interceptions from above. Those more downstream located on the final approach (right side of the ground speed profiles in Figure 3.4b) are due to a late reduction in ground speed, some flights have ground speeds of the order of 200 kts a few nautical miles from the runway threshold.



(a) Altitude profiles



(b) Ground Speed profiles

Fig. 3.4 Paris Orly runway 26 atypical approaches altitude and ground speed profiles

Appearance and disappearance of Atypical Phases The first study lead aims to determine how the phases of atypicality in approach are distributed. As the coefficient is continuous, it is possible to determine the beginning (when the atypical coefficient exceeds the τ threshold) and the end (when the atypical coefficient falls below the τ threshold) of the atypical phases.

In order to analyze the approach, three main phases were created according to the distance remaining to the runway threshold. The 25NM to 15NM phase, which generally corresponds to the end of the downwind leg and the beginning of the base leg where the aircraft has not yet intercepted the runway centreline. The 15NM to 5NM phase, which generally corresponds to the phase of compliance with the Localizer Inception and glide path monitoring. The phase from 5NM to the thresholds corresponding to

the stabilization and final approach, this phase is particularly monitored by the airlines because it is the critical phase. As an example, the three phases were illustrated on a flight landing at Paris Orly Airport in Figure 3.5.



Fig. 3.5 The three phases on a flight landing at Paris Orly Airport.

Figure 3.6 shows the different phases of atypicality detected by the algorithm on the B737 data set. The graph represents each atypical phase by a horizontal line. The x-axis gives the beginning and the end of the atypical phase and the y-axis indicates the beginning of the atypical phase. The colors used corresponds to the three flight phases detailed above. The results are also summarised in Figure 3.7 for both data sets. It can be seen that, in the vast majority of cases, the atypicalities that begin in the 25NM to 15NM phase disappear before the 5NM to the runway threshold phase. However, the atypicalities appearing in the 15NM to 5NM phase have not disappeared at 5NM in around half of the cases, generally disappearing at a distance, which corresponds to the stabilisation height operated by the airline. In addition, there is a non-negligible number of energy atypicality appearing in the last 5NM nautical miles. These cases will be illustrated later on and generally correspond to high speed approaches with a late aerodynamic configuration involving a late speed reduction as well.

These results are very relevant from a safety point of view. Flights whose atypicality begins in the 15NM to 5NM phase implies that it would be possible to detect and take them into account as soon as they appear. Consequently, it would potentially prevent an aircraft from ending up downstream of the approach in more critical areas, as they are close to the runway, with atypical energy situations such as high energies. These situation may involve safety issues such as aircraft unstable approaches or runway excursions.

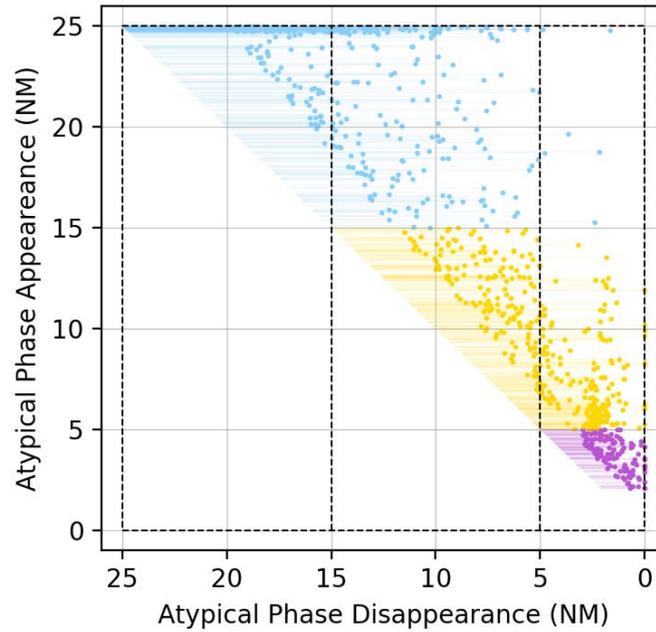


Fig. 3.6 B737 data set atypical phases: appearance and disappearance locations. Each atypical phase is represented by a horizontal line. The x-axis gives the beginning and the end of the atypical phase and the y-axis indicates the beginning of the atypical phase. The color corresponds to the three flight phases detailed previously (25NM-15NM in blue, 15NM-5NM in yellow, 5NM-THR in purple).

Atypical Flight and safety Events In this paragraph, the atypical results are compared with the flight safety events monitored by the airlines. The events are specific to each airline and generally correspond to the exceeding of a threshold on a given parameter, which was set by the flight safety officer. Usually, different thresholds are set to quantify the criticality level of the event. The two airlines from which this analysis is derived have three levels of criticality: low, medium and high.

In order to establish potential links between atypicalities and safety events, the contingency matrix of the events and different indicators will be observed. The indicators are the Pearson correlation coefficient, the Yule's Q coefficient which is a special case of Goodman and Kruskal's gamma [113], the Cohen's κ coefficient [114], and the score and the p-value from a χ^2 independence test. For more detail on the χ^2 independence test see Appendix D

The data set of B737 flights is considered because it includes the largest number of flights, and therefore presents more statistically representative results. As the atypicality is studied on the last 25NM and safety events are generally observed on the last nautical miles, the results will be given for only three atypical flight groups, the flights presenting an atypicality between between 15NM and 5NM, those between 5NM and the runway threshold, and those atypical whatever the phase.

The first link studied is the link between the atypicality and the flights presenting at least one safety event of medium or high criticality, the results are summarised in Table

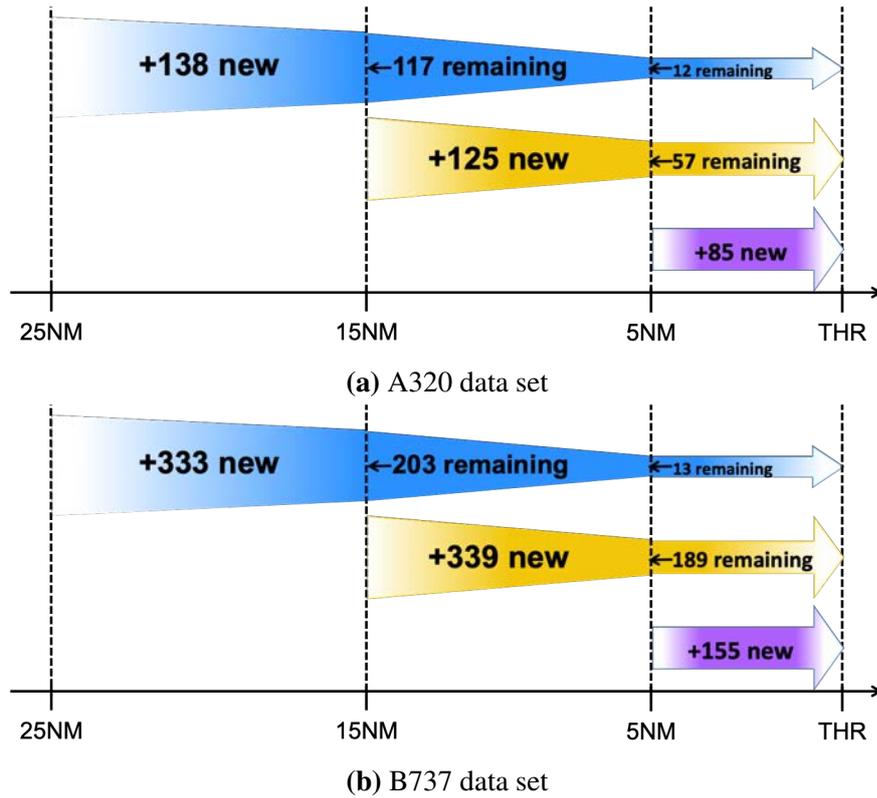


Fig. 3.7 B737 and A320 data set atypical phases: appearance and disappearance locations summary. The number in each column represents the number of atypical flights remaining or appearing during this segment. The color corresponds to the three flight phases detailed previously (25NM-15NM in blue, 15NM-5NM in yellow, 5NM-THR in purple).

3.4. For all the atypical phases studied, a correlation is observed with a maximum of 0.0967 for the atypical flight between 5NM and the runway Threshold. The correlation does not imply causality, it implies that statistically when one variable show an increase the other also has a tendency to increase over the observed data set. To give an example, the ratio of flights with at least one medium or high criticality event is 3.13% for typical flights from 5NM to the runway threshold, and the same ratio is 28.01% for atypical flights. It can be hypothesized that energy atypicalities resulting from kinetic over-energy generally imply an excessive speed, which is monitored by the airlines. The χ^2 test reject the H_0 hypothesis of independence. Non-independence means that the realization of one variable affects the probability distribution of the other.

Second, the link between atypical flights and flights presenting unstabilized approaches is studied and summarized in Table 3.6. Similar results are observed. The correlation coefficient is even greater and is 0.17 for atypical flights between 5NM and the runway threshold. In particular, the unstabilized approach rate is 3.58% for typical flights, and 50.98% for atypical flights.

Finally, the link with long touchdowns is summarized in Table 3.8. Here the link

Atypical 15NM to 5NM vs Medium-High	Category	No M-H	M-H
	Typical	71 867	2 320
	Atypical	429	107
Atypical 5NM to THR vs Medium-High	Category	No M-H	M-H
	Typical	72 039	2 327
	Atypical	257	100
Atypical vs Medium-High	Category	No M-H	M-H
	Typical	71 627	2 269
	Atypical	669	158

Table 3.3 Contingency matrix between atypical categories and medium-high event flights underlying the distribution of the flights into each category over the data set

Indicator	Atypical 15NM to 5NM vs Medium-High	Atypical 5NM to THR vs Medium-High	Atypical vs Medium-High
Pearson Correlation	0.0801	0.0967	0.0946
Yule's Q	0.7708	0.8466	0.7634
Cohen's Kappa	0.0611	0.0640	0.0819
χ^2 Score	474.62	692.08	664.04
χ^2 p-value	$\approx 0 (\leq 10^{-50})$	$\approx 0 (\leq 10^{-50})$	$\approx 0 (\leq 10^{-50})$

Table 3.4 Association coefficients and tests between atypical and medium-high event flights. Four indicators are given, the Pearson correlation score, the Yule's Q score, the Cohen's Kappa, and the score and p-value of a χ^2 independence test.

Atypical 15NM to 5NM vs Unstable	Category	Stable	Unstable
	Typical	71 493	2 694
	Atypical	379	157
Atypical 5NM to THR vs Unstable	Category	Stable	Unstable
	Typical	71 704	2 662
	Atypical	175	182
Atypical vs Unstable	Category	Stable	Unstable
	Typical	71 269	2 627
	Atypical	603	224

Table 3.5 Contingency matrix between atypical categories and unstable flights underlying the distribution of the flights into each category over the data set

is also present but in a weaker way with a Pearson's coefficient of 0.0174 for atypical flights between 5NM and the runway threshold. In particular, the long touchdown rate is 4.36% for typical flights between 5NM and the runway threshold, and 9.52% for atypical flights. The χ^2 Independence test for atypical flights between 15NM and 5NM gives a p-value allowing to reject (H_0) only with a risk of error of the order of 2%.

Thus, the results underline the existence of links between the atypicality of the tra-

Indicator	Atypical 15NM to 5NM vs Unstable	Atypical 5NM to THR vs Unstable	Atypical vs Unstable
Pearson Correlation	0.1130	0.1708	0.1285
Yule's Q	0.8332	0.9308	0.8194
Cohen's Kappa	0.0816	0.1058	0.1064
χ^2 Score	947.77	2161.50	1227.55
χ^2 p-value	$\approx 0 (\leq 10^{-50})$	$\approx 0 (\leq 10^{-50})$	$\approx 0 (\leq 10^{-50})$

Table 3.6 Association coefficients and tests between atypical and unstable flights. Four indicators are given, the Pearson correlation score, the Yule's Q score, the Cohen's Kappa, and the score and p-value of a χ^2 independence test.

Atypical 15NM to 5NM vs Long touchdown	Category	No long touchdown	Long touchdown
	Typical	70 949	3 238
	Atypical	501	35
Atypical 5NM to THR vs Long touchdown	Category	No long touchdown	Long touchdown
	Typical	71 127	3 239
	Atypical	323	34
Atypical vs Long touchdown	Category	No long touchdown	Long touchdown
	Typical	70 685	3 211
	Atypical	765	62

Table 3.7 Contingency matrix between atypical categories and long touchdown flights underlying the distribution of the flights into each category over the data set

Indicator	Atypical 15NM to 5NM vs Long Touchdown	Atypical 5NM to THR vs Long Touchdown	Atypical vs Long Touchdown
Pearson Correlation	0.0089	0.0174	0.0161
Yule's Q	0.2097	0.3960	0.2816
Cohen's Kappa	0.0061	0.0102	0.01279
χ^2 Score	5.45	21.44	18.65
χ^2 p-value	0.01956	3.65×10^{-5}	1.57×10^{-5}

Table 3.8 Association coefficients and tests between atypical and long touchdown flights. Four indicators are given, the Pearson correlation score, the Yule's Q score, the Cohen's Kappa, and the score and p-value of a χ^2 independence test.

jectories and the events monitored by the airlines' flight safety offices. The nature of the safety events is usually different from the atypicality since the events are generally observed on a very short window in the last 5NM of the trajectory, whereas the atypicality

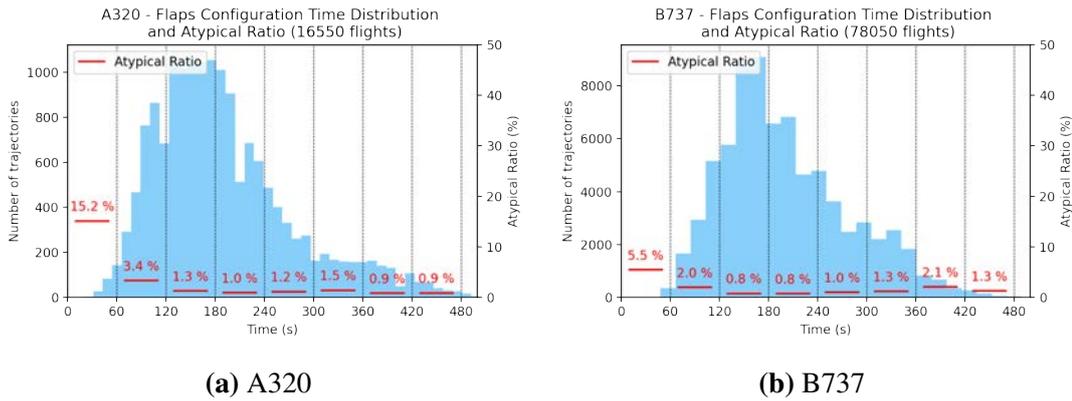


Fig. 3.8 A320 and B737 flights landing configuration time distribution and atypical ratio. Short time configuration are associated with a higher atypical ratio

is studied continuously on the last 25NM. The hypothesis that certain atypicalities are taken into account and managed could explain the strength of the link, which increases as the aircraft approaches the threshold of the runway. Finally, the objective of atypicality is not to detect already existing events but to study potential precursors of those events. It appears from this study that energy atypicalities in the last 15NM up to the runway threshold can lead to an increased ratio of safety events such as non-stabilization on final approach.

3.2.2.1 Inappropriate Control Inputs

In this section inappropriate control inputs are analysed with the atypical flight ratio. Inappropriate control inputs are actions that do not correspond to those recommended by the standard operating procedure (SOP) documentation

First, the distribution of landing configuration times is analyzed to switch from flaps and gear up to flaps full and gear down. It is illustrated in Figure 3.8 for the two aircraft types A320 and B737. The distribution is shown in blue (left axis) and the ratio of atypical flights is shown in red by grouping flights per minute (right axis). The average landing configuration time is between two and three minutes, while the fastest time is 24 seconds. Short landing configuration times can result in a high workload for pilots. The atypical ratio has been calculated by groups of minutes (less than one minute, between one and two minutes, etc.). It is observed that the atypical ratio increases to 15.1% (A320 data set), and 5.5% (B737 data set) for the group of flight that configure within less than one minute. This group is mainly composed of fast approaches or late reduction flights.

Second, the landing gear down setting appearance distribution in the landing configuration is illustrated in Figure 3.9. For the A320, it is recommended to apply gear down while flaps 2 configuration is out. This is well described with the distribution in Figure 3.9 since this is what a great majority of flight did. However, a non-negligible

number of flights applied gear down while being flaps 1 or flaps 0. The correlation with the atypical ratio is meaningful with 9.8% for the flaps 1 group and 29.6% for the flaps 0 group. It describes underlying energy managements. Indeed it mainly corresponds to atypicality in the 25NM to 15NM phase, and pilots anticipated landing gear configuration in order to dissipate an excess of energy. Similar behaviours are observed for the B737 distribution.

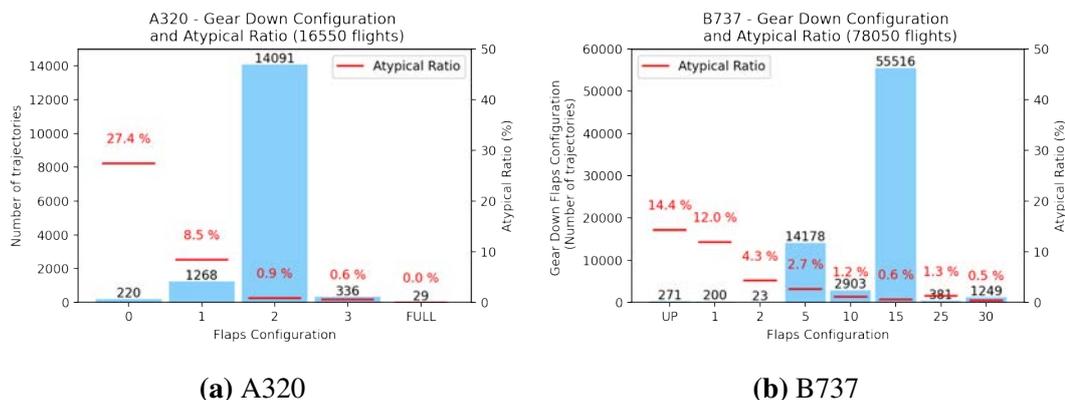


Fig. 3.9 A320 and B737 flights landing gear action appearance distribution in the landing sequence and the atypical ratio. Gear down action being applied with less flaps extension than expected is associated with a higher atypical ratio

Finally, the distribution of distance in nautical miles to the mean last landing configuration element distance is studied in Figure 3.10. Atypical ratios were computed every nautical mile. When the last landing configuration element appears between 1NM and 2NM after the mean distance, the atypical ration increases up to 6.2% (respectively 4.4%), and up to 27.7% (respectively 28.2%) between 2NM and 3NM for the A320 (respectively B737) data set. It usually corresponds to late configurations and late power reductions.

3.2.3 Discussion

As previously introduced in chapter 2, national and international safety authorities would like to detect and propose appropriate corrections to undesirable events as soon as possible when they occur. The FPCA decomposition provides a possible solution since it not only takes into consideration the amplitude but also the variation of the curves. Using simple operational criteria would lead, for example, to the detection of an over potential energy only when the parameter exceeds a defined limit. However, the proposed atypical scoring can anticipate non-nominal values considering the curve variation. Since the model is built on several flights, it will consider a global nominal behaviour, which consists in a reduction of the total energy. If this particular aircraft keep a constant potential energy, its energy variation will not follow the group behav-

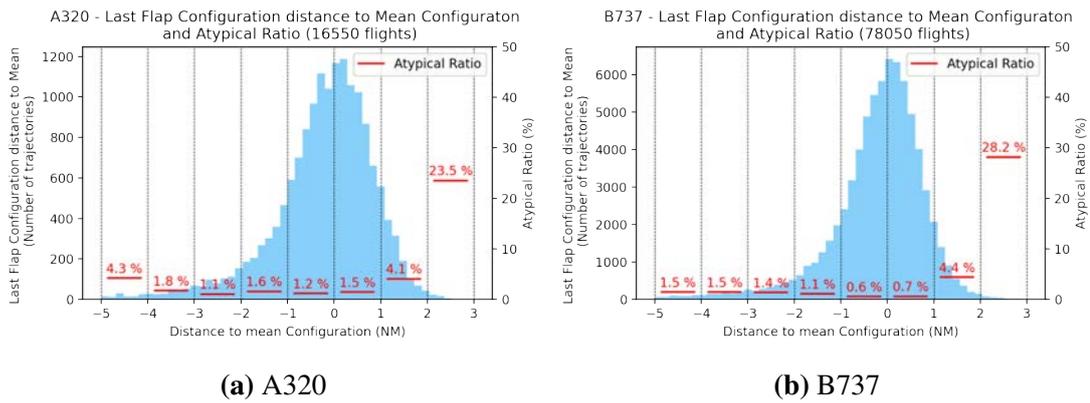


Fig. 3.10 A320 and B737 flights last landing configuration element distance to the mean distribution and the atypical ratio. Last landing configuration element occurring later than the mean distance is associated with higher atypical ratio

ior and will be detected as an atypical flight. Consequently, the first advantage of this method is that it anticipates non-nominal values considering the curve variation.

Secondly, determining safety event precursors with ground available data is a complex problem. This methodology contribution is to provide potential precursors for flight safety events, considering only one assumption: atypical flights do not behave like the others. This assumption implies a dedicated analysis of the extracted flights to decide if they contain safety issues or not. Nevertheless, it has the advantage of providing flights that might not have been monitored with the current flight data analysis methods. Indeed, current safety analyses mainly focus on the stabilization phase, i.e. the last five nautical miles of the approach. In addition, only few studies have been conducted in an approach path management perspective. Underlying non-monitored behaviors enables better situation awareness. Airline safety offices or air navigation service providers could, for example, create safety advises when they consider that the behaviors present a potential threat.

Finally, the study with airline company data shows that there are links between atypical energy management and airlines safety events. Many of the atypical areas usually disappear before the aircraft stabilizes, so it can be assumed that they have been addressed and properly managed. Nevertheless, there are energy atypicalities, in particular those beginning in the 15NM to 5NM phase, that persist and are generally associated with non-stabilization. This tool therefore provides opportunities to detect atypicalities in energy management from ground side. These atypicalities present real safety issues. In addition, directions towards a real-time detection tool will be developed in chapter 6.

3.3 Flight safety during Covid-19: A study of Charles de Gaulle airport atypical energy approaches

During the COVID-19 period and particularly during lockdown, deviations from nominal operations have shown to become more frequent. To confirm this observation this paper proposes to evaluate the impact of COVID-19, and more generally of crises that lead to a sharp drop in traffic, on the pilot/controller system, especially during the critical approach and landing phases. To study the influence of this type of crisis on flight operations at Charles De Gaulle airport, an existing energy atypicality metric is applied on a reference period before COVID-19 and compared to the COVID-19 period. Whereas the traffic at Charles De Gaulle airport has decreased by around 90% on April 2020, the obtained statistics underlined an increase in the atypical flight ratio of around 50%. This trend can be explained in part by the appearance of glide interceptions from above as a result of trajectory shortenings, and an increase in the proportion of high speed approaches

3.3.1 Data and Methodology

The data studied in this section are radar data at Roissy-Charles-de-Gaulle Airport (CDG) from the French Air Navigation Service Provider (DSNA). Among all airports whose operations are assigned to DSNA, CDG was chosen because it was the only airport that maintained sufficient traffic during the COVID-19 outbreak to obtain statistically representative results. The data was collected during two time periods: a reference period before COVID-19 from 01 to 31 May 2019 (accounting for 21 895 landings), and a study period under COVID-19 from 16 March to 20 April 2020 (accounting for 4 583 landings). The reference period covering May 1st to May 31st was the available period that best respected the seasonality studied during the COVID-19 period. The data contains landing trajectories (longitude, latitude, altitude, ground speed, vertical speed) as well as meta-information such as aircraft type, airline, and the runway used.

The methodology consists of constructing an energy atypicality model from the available data covering the reference period before COVID-19 and applying it to the two periods respectively. As a reminder, the model continuously gives an atypicality score between 0 and 1 along the trajectory. The closer the score is to 1, the more the energy management at this point does not behave like the majority of the flights in the learning set. A threshold is then set so that statistical studies can be carried out. Then, a flight will be considered atypical if it scores above 0.7 during more than 2NM. A score greater than 0.7 is equivalent to a deviation from the mean functional principal components analysis coefficient greater than 3σ under Gaussian distribution hypothesis.

In addition, other classical operational metrics are computed, such as the interception distance on the runway centreline, or the exceeding of altitude limits (based on the

interception altitude and deviation from 3° glide path) and ground speed limits (based on approach speed nominal interval and on 3° glide path deceleration [14]). Three categories, green, orange, and red, are defined using altitude and speed limits as explained in Appendix A. These categories enable analysing the causes of energy atypicalities.

3.3.2 Observations

The first observation that can be made about the COVID-19 period is that the distribution of the Magnetic bearing of the runway in use (QFU) usage is changing as illustrated in Figure 3.11. For the period before COVID-19, the two runway pairs are used with the outer runways for landing and a majority of landings facing west (runway 26L and 27R). During the COVID-19 period, the South runway pair was quickly closed and the North runway pair was mainly used, particularly in the eastward facing configuration (runway 09L and 09R accounting for 70% of the total operations), probably due to weather conditions.

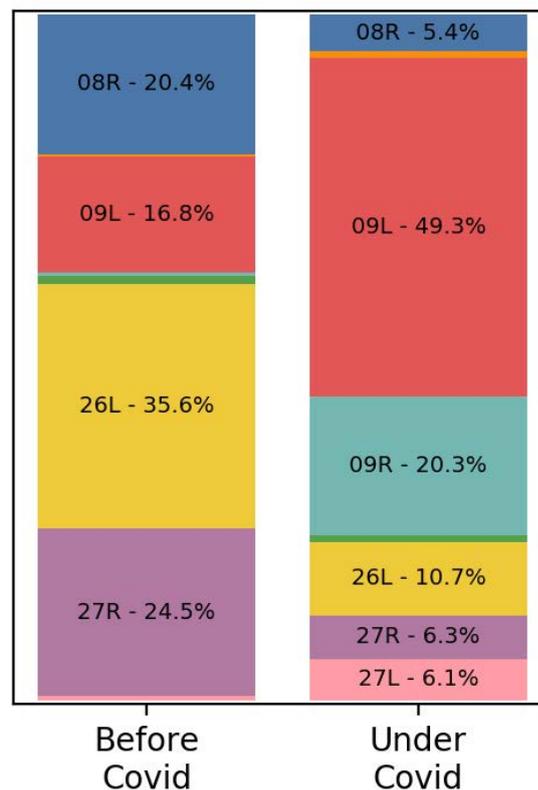


Fig. 3.11 QFU usage at Roissy Charles de Gaulle airport before and during COVID-19. Each runway has a different colour, and the percentage corresponds to the ratio of runway use to total flights for the period. The COVID-19 period shows a change in the use of runways, whereas before COVID-19 both pairs are used with the outer runways for landing and mostly in a west-facing configuration. During the COVID-19 period, the south pair was quickly closed and a majority configuration facing east was observed linked to weather conditions.

The reference period shows a rate of atypical flights at 6%. During the COVID-19 period, the rate goes up to 9.4%, representing a 56% increase. This increase can be explained in part by the change in configuration and the increased use of runway 09R, which already had a high rate of atypical flight before COVID-19 as shown in Figure 3.12. In addition, the high ratio on inner runways (26R, 27L, 08L, 09R) is generally explained by the use of a procedure known as the "bayonet" procedure, which is illustrated in Figure 3.13. The aircraft is initially lined up on final for the outer runway and may be cleared to change and land on the inner runway generally to minimize taxi time. This procedure generally induces a phase of energy atypicality because the glide paths are not aligned.

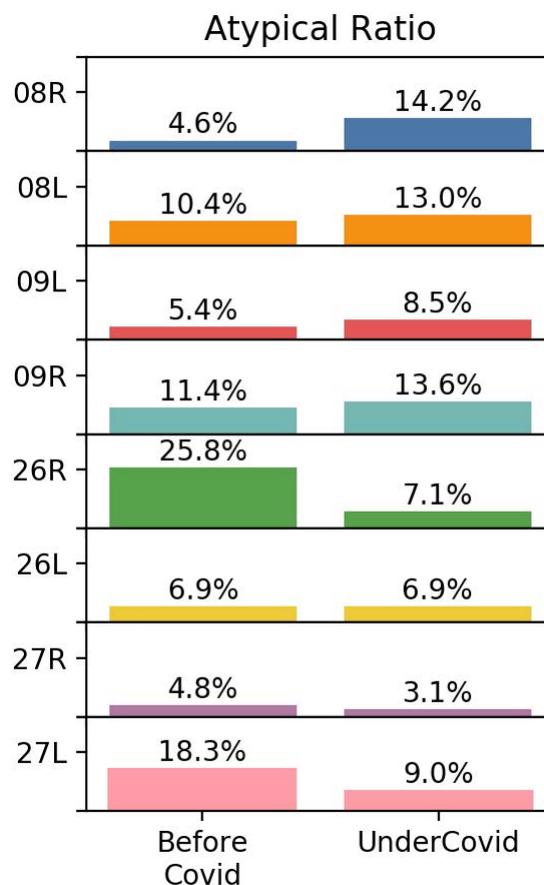


Fig. 3.12 Distribution of trajectory atypical rate per runway QFU before and during COVID-19

In view of these initial results, which show a majority use of the north runway doublet facing east, the following paragraphs focuses on a detailed study of the approach and landings on runways 09L and 09R before and during COVID-19. The total flight volume is relatively similar: there are 3 809 flights before COVID-19 and 3 189 flights during COVID-19, which can be compared;

First, the lateral management of trajectories will be studied. Figure 3.14 shows the lateral profiles of all atypical flights before (top figure) and during (bottom figure)

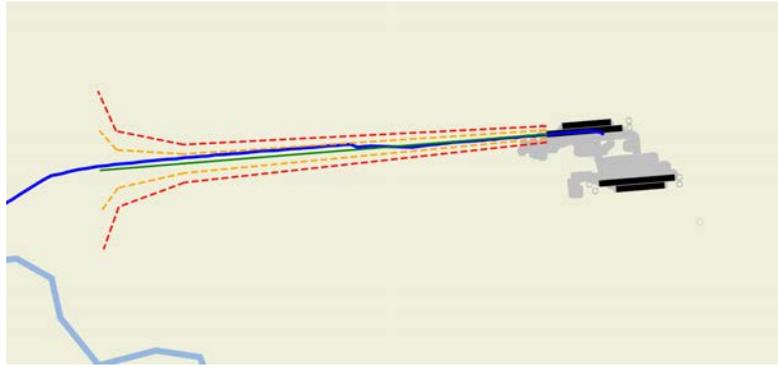


Fig. 3.13 Example of a bayonet procedure at Roissy Charles de Gaulle airport. The aircraft was initially lined up on final for the outer runway 09L and was cleared to change and land on the inner runway 09R. This procedure is generally used to minimize taxi time.



Fig. 3.14 Lateral profile of atypical flight before (left) and during (right) COVID-19 period at Charles de Gaulle Airport Runway 09L and 09R

COVID-19 period at Charles de Gaulle Airport landing on runway 09L and 09R. It can be seen that over the period before COVID-19 the energy atypicality is localized on the final phase when the aircraft is aligned with the runway and contrary to the period during the COVID-19, there is no shortening of the trajectory.

During the COVID-19 period, the phases of atypicality start earlier during downwind leg or base leg. In addition, the appearance of shortening of the trajectory is observed. The shortening of trajectories with a southerly approach is not at all present in normal situations since the approach to Le Bourget Airport is located in this area. However, during this period of COVID-19, Le Bourget airport was closed, the constraint was no longer present, and this type of approach was possible. This trend is confirmed by the distribution of interception distance from runway threshold illustrated in Figure 3.15, where the interception distance distribution shifts from the 15NM to 20NM area to the 10NM to 15NM area and even less for atypical flights. Furthermore, it is illustrated that the phases of atypicality during the COVID-19 start upstream while the aircraft are still on base leg or downwind.

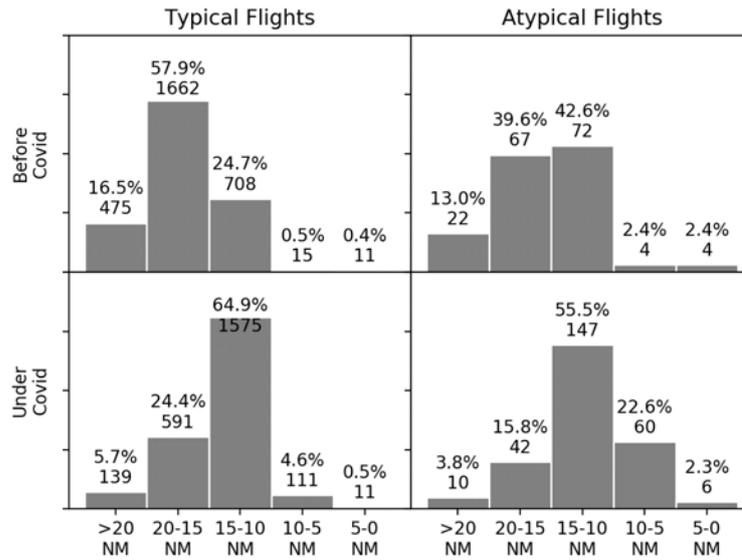


Fig. 3.15 Distribution of localizer interception distance (NM) for typical and atypical flight under and before COVID-19 for runway QFU 09L and 09R. For the COVID-19 period, there is the emergence of glide interceptions from above that were not or only slightly present in the period before COVID-19. The color of trajectories is proportional to the local atypical energy coefficient (green for 0, orange for 0.5 and red for 1). The colored dashed lines are published operational limits and deviation from them.

The shortening of trajectories may appear to be consistent with the drastic reduction in traffic, which facilitates direct trajectories. In addition, it implies an anticipation on energy management since shortening the trajectory also reduces the distance remaining to the threshold. The large phases of energy atypicality imply that this induced over-energy has not always been taken into account. This is confirmed by the altitude profiles illustrated in Figure 3.16. On top of the Figure are illustrated vertical profiles on runway 09L and 09R of atypical flights before COVID-19 and at the bottom vertical profiles under COVID-19. There is an appearance of altitude profiles with glide interception from above that was not or only slightly present in the period before COVID-19. The distribution of altitude warning areas illustrated in Figure 3.17, confirms this trend. The ratio of atypical flights in the orange and red altitude areas increased for the COVID-19 period from 40% to 53%. In addition, there are two clusters for atypical flights. Those in the green area, for which the atypicality is not due to the potential energy, the factor comes from the kinetic energy, and those in the red area with high potential energy and possibly also a high kinetic energy.

This is reflected in the distribution of flights by ground speed limit areas illustrated in Figure 3.18. Already before the COVID-19, the atypical flights have a high ratio of flights with ground speeds in the orange and red areas and this ratio increases during the COVID-19 period. Overall, the arrivals in 09L and 09R are quite fast already before the COVID-19 and the shortening of trajectories and glide interceptions from above

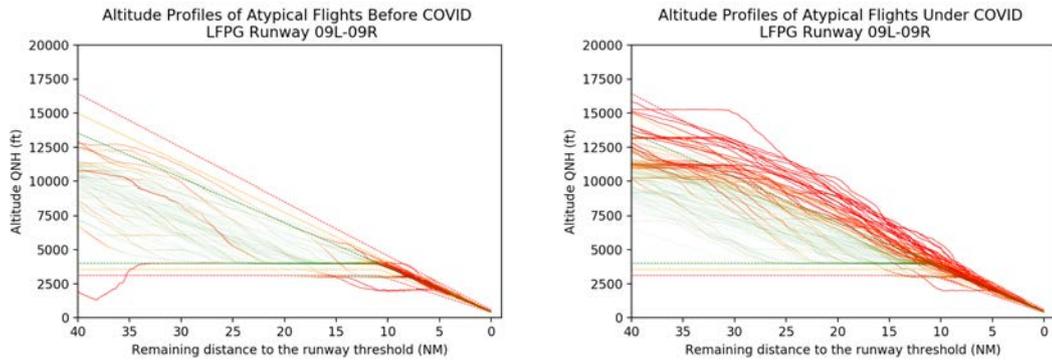


Fig. 3.16 Vertical profile of atypical flights before (left) and during (right) COVID-19 period at Charles de Gaulle Airport Runway 09L and 09R

probably reinforced this trend. In addition, the number of flights in the orange and red zones during COVID-19 has increased, despite the fact that there are slightly fewer flights operating on runways 09L and 09R during this period.

Energy management seems to be different depending on the airline. Figure 3.19 illustrates the atypicality ratio before and during COVID-19 for the seven major airlines operating at CDG. The airlines are anonymized and ordered by atypical ratio. Some airlines have a low ratio during COVID-19 period, the low ratio is maintained or slightly increased for the COVID-19 period (airlines A3 and A7). Other airlines like A8 have a high ratio initially that increases during the COVID-19 period, and some airlines have a high ratio that decreases (A1) or increases (A2) during the COVID-19 period. The level of implementation of energy monitoring and training might be different from one airline to another, which may affect the observed atypical ratio.

An analysis by aircraft type family can also be carried out. The atypical ratio of the seven major aircraft type families operating at CDG is illustrated in Figure 3.20. The result indicates that all aircraft type families have an atypicality ratio that increases during the COVID-19 period except for the B777 family. This result is linked to the airline results presented above. B777 is operated by different airlines and in particular A1, whose average atypical ratio decreased during this period.

Two specific flights are now analysed to explore the behaviours observed in the previous study.

High Speed Approach The first flight is an approach to runway 09L of an A320 aircraft. A rapid arrival behaviour is observed. Figure 3.21 shows on top the lateral trajectory and the altitude profile, and the speed profile at the bottom. The lateral and vertical profiles of the trajectory are compliant with the interception altitude and chevrons and follow properly the 3° glide path. The ground speed profile shows that the ground speed was maintained at 230kts up to 5NM and then was sharply reduced just before stabi-

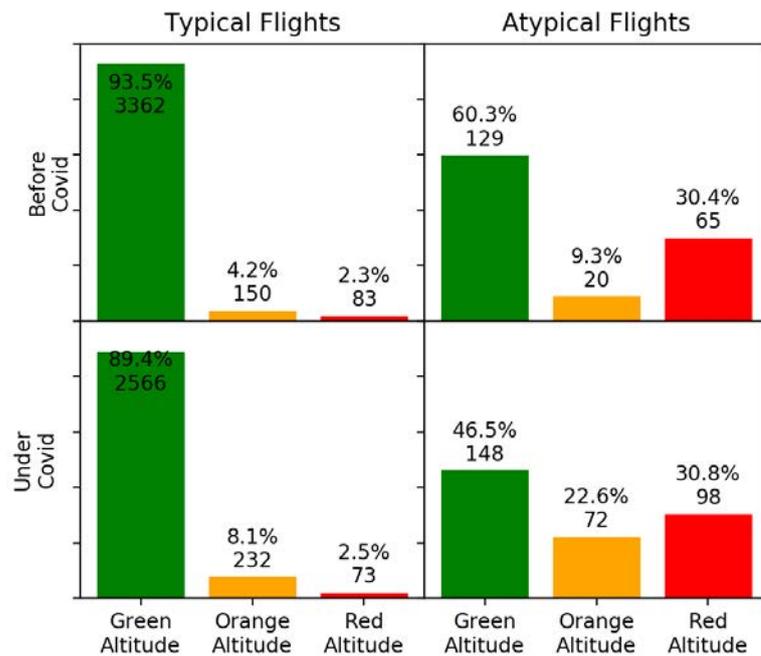


Fig. 3.17 Distribution of altitude areas for typical and atypical flights under or before COVID-19 for runway QFU 09L and 09R. Green category corresponds to flights without any point above or under the altitude orange dashed line limit. Orange category corresponds to flight with at least on point in the altitude orange to red dashed line area, and Red category to flights with at least on point in the altitude Red dashed line area. These operational limits are detailed in Appendix A

lization. It is observed that the atypical areas starts at 10NM since the aircraft keeps its kinetic energy whereas the usual behaviour at 10NM is a reduction in kinetic energy.

This type of approach is atypical in the sense that the speed reduction appeared much later than normal, which also raises questions about safety. The aircraft was probably stabilized, nevertheless the interest of the tool is to highlight events that are not always monitored so that safety aspects can be checked. The ground speed in final is an important criteria, because high-speed combined with bad weather conditions has been observed in crashes such as the runway overrun of the Hermes Airline Flight 7817 at Lyon Saint-Exupéry on March 29, 2013 [115], or the Pegasus Airlines Flight 2193, which overran Istanbul runway on February 5, 2020 [116, 117].

Approach with Glide Interception From Above The second case, illustrated in Figure 3.22, is an A320 approach on runway 09R with a trajectory shortening associated with a glide slope interception from above. The aircraft was initially on a downwind leg, the base leg turn is anticipated, and the aircraft takes an interception heading that brings it well beyond the interceptions chevrons. The ground speed remains in the nominal, but the shortening of trajectory results in a late glide interception from above recovered at only 5NM to the runway threshold.

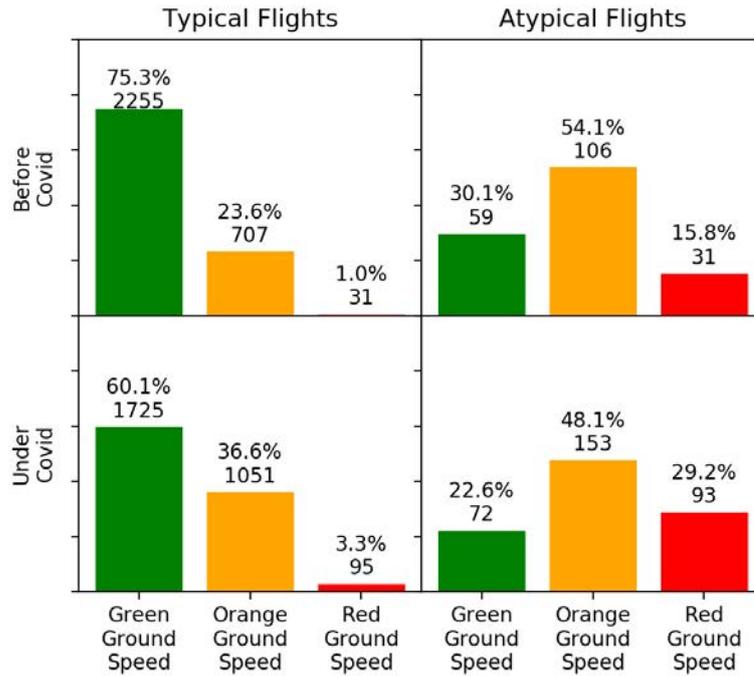


Fig. 3.18 Distribution of ground speed areas for typical and atypical flights under or before COVID-19 for runway QFU 09L and 09R. Green category corresponds to flights without any point above or under the orange ground speed dashed line limit. Orange category corresponds to flight with at least on point in the orange to red ground speed dashed line area and Red category to flight with at least on point in the Red ground speed dashed line area. These operational limits are detailed in Appendix A

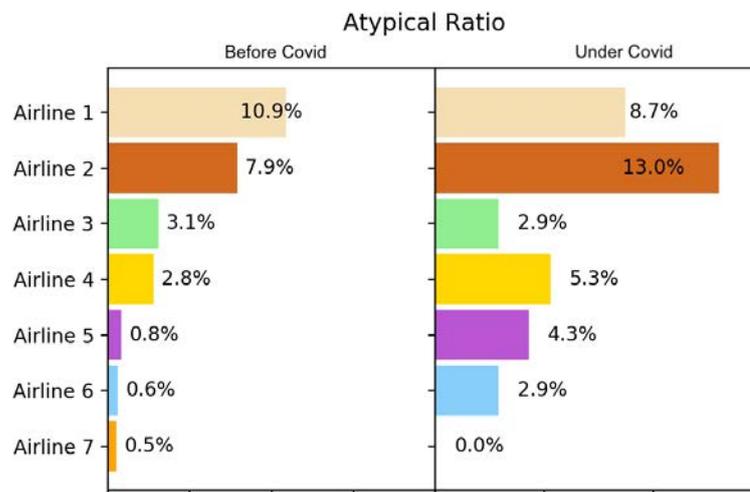


Fig. 3.19 Distribution of the typical ratio of the 7 major airlines at CDG under or before COVID-19 for runway QFU 09L and 09R

Glide Interceptions From Above and shortening of the trajectory have been observed in various air crashes. In February 2009, the Turkish airline flight 1951 intercepted the glide from above following a trajectory shortening. An altimeter defect caused it to stall on final and crash before the runway [118]. Furthermore, on July 6, 2013, a Boe-

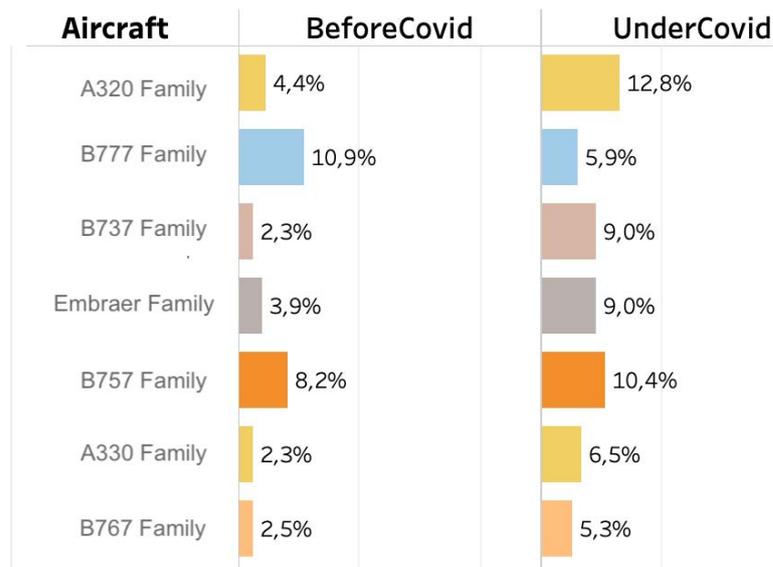


Fig. 3.20 Distribution of the atypical ratio of the 7 major aircraft type families operating at CDG under or before COVID-19 for runway QFU 09L and 09R

ing 777-200ER operating Asiana Airlines Flight 214 struck a seawall at San Francisco International Airport (SFO) in San Francisco, California. The National Transportation Safety Board in charge of the investigation, concluded that the visual approach was flown in poor condition, including a glide intercept from above, an air traffic control requirement to maintain 180kts to 5NM, and an airspeed management mode activated by the pilot leading the aircraft into a stall on short final [6].

3.3.3 Discussion

In this section, possible uses of the technology are discussed. It has been presented the use of a post-ops methodology for the analysis of energy atypicality in the context of a study of the COVID-19 period.

The construction of the model is an important criterion because it obviously influences the final atypical flight rate and whether an atypicality is detected. Nevertheless, another model has also been built using flight data on Paris Orly airport platform and the results with this other model are similarly highlighting an overall increase in atypicality during the COVID-19 period compared to the baseline period before COVID-19.

This type of analysis can be used within the framework of safety management systems and in particular within the framework of measures taken in the event of a crisis. For example, it is possible to imagine precautionary measures taken into account in the context of a low volume of air traffic, the closure of the Le Bourget aerodrome, or weather conditions involving a majority use of the doublets facing east. The various ANSP operators and airlines have already shown their interest in this type of analysis, particularly in this post-COVID-19 recovery period.

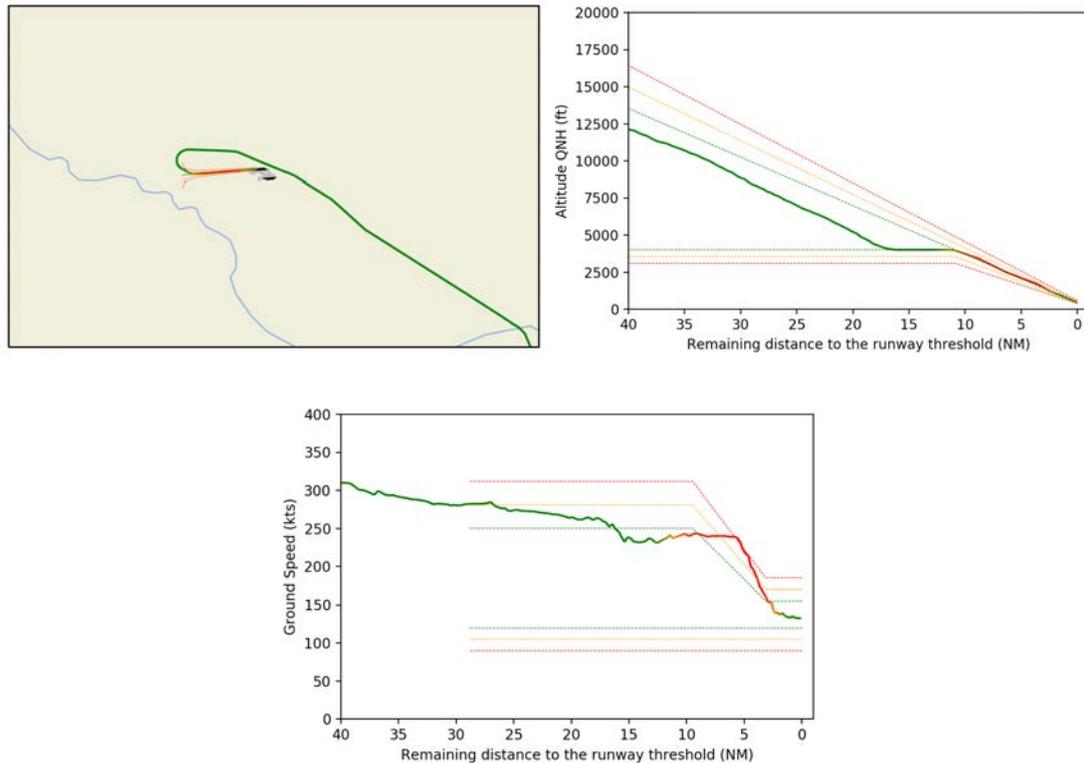


Fig. 3.21 Lateral, altitude and ground speed profiles of a late speed reduction A320 landing on runway 09L. The top left graph shows the lateral trajectory, the top right graph shows the vertical profile and the bottom graph shows the ground speed profile. It can be seen that the aircraft maintains 230kts ground speed up to 5NM before sharply reducing speed. The color of the trajectory is proportional to the local atypical energy coefficient (green for 0, orange for 0.5 and red for 1). The colored dashed lines are fixed operational limits to allow a better understanding of the situation.

It is important to specify that this methodology brings a different and new perspective for the analysis of flights but in no way replaces the existing investigation methods. Flight atypicality does not necessarily imply consequences on flight safety. It is a complementary aspect to be taken into account in a safety analysis. The results from this study show that the atypicality metric seems relatively consistent for the monitoring of an airport platform or events related to energy management.

The study shows an overall increase in approaches with energy atypicality, generally associated with shortening of trajectories, glide interceptions from above and approaches with late speed reductions. The metric seems appropriate for monitoring energy management on approach and landing. The complementary vision to the classical flight analysis techniques allows to bring an additional dimension on the situation awareness. Moreover, the results obtained seem to be consistent with the latest figures published by IATA on the sharp increase in the rate of unstabilized approaches for the period of COVID-19 [119].

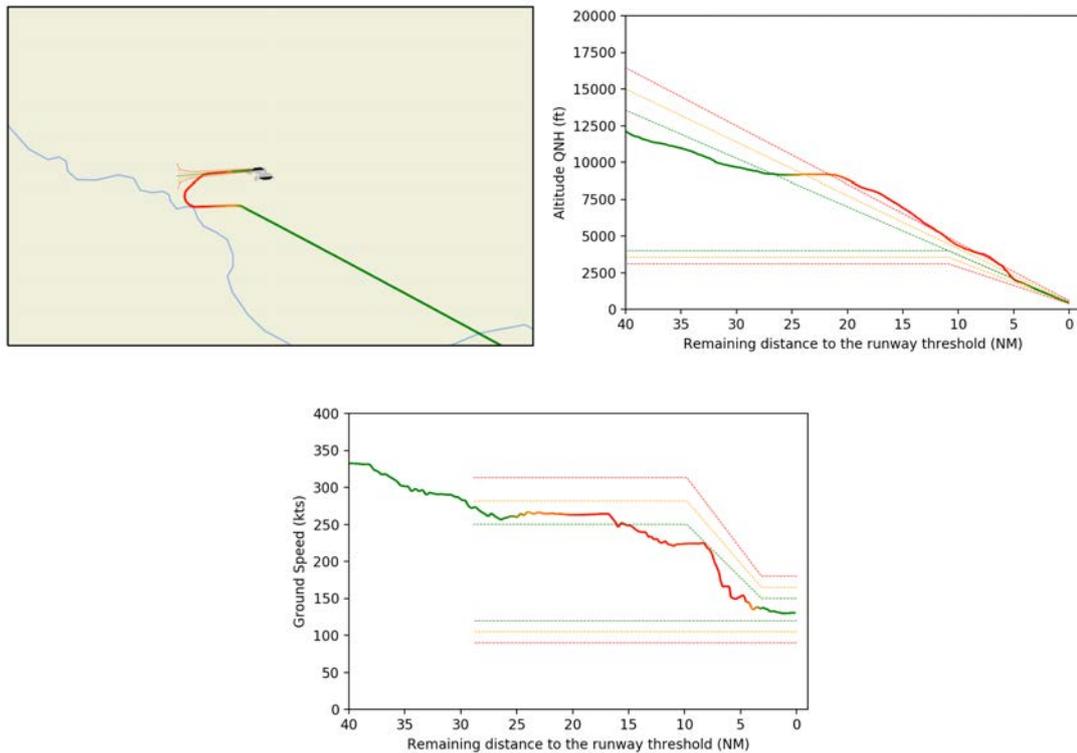


Fig. 3.22 Lateral, altitude and ground speed profiles of an A320 glide interception from above on runway 09L. The top left graph shows the lateral trajectory, the top right graph shows the vertical profile and the bottom graph shows the ground speed profile. It is observed that the aircraft shortened its trajectory, which led it to a glide path interception from above, which is not recovered until 5NM from the runway threshold. The color of the trajectory is proportional to the local atypical energy coefficient (green for 0, orange for 0.5 and red for 1). The colored dashed lines are fixed operational limits to allow a better understanding of the situation.

3.4 Conclusions

In this chapter a post-operational methodology for the detection of atypical trajectories has been presented. It is based on a functional principal component decomposition and an atypicality scoring related to the density distribution. The process is applied on a sliding window in order to be able to detect local and global events. The algorithm has been applied to the detection of atypical energetic aircraft approaches during approach from the ground. The algorithm is able to detect over-energy such as glide interceptions from above or fast approaches. It also detects atypical energy variations that lead to over or under-energy situations.

The algorithm was confronted with the analysis and events monitored in the framework of the airline flight safety offices. The study has shown statistical links between energy atypicality and safety events such as non-stabilized approaches. In particular, on the data set studied, more than one out of two atypical flights presented a non-stabilized

approach. The study also highlighted the phases of appearance and disappearance of atypicalities. The 15NM to 5NM phase could be the subject of particular attention: half of the flights presenting an atypicality keep it in the 5NM phase up to the runway threshold. Energy atypicalities are also generally corroborated with inappropriate control inputs. For example, significant atypicality rates are observed when the gear is extended in the flap-up configuration to dissipate energy, or in cases of late and rapid configuration take-up present during high speed approaches.

Finally, the algorithm was applied in the monitoring of the Roissy Charles De Gaulle airport platform during the COVID-19 period. The metric seems relevant to study the evolution of atypical behaviours during a crisis period for example. An increase in atypical energetic behaviour was observed, with a strong increase in top glide interception and fast arrivals. The methodology and the algorithm motivated the development of a prototype analysis software for airlines, air navigation service providers and supervisory authorities. It allows the efficient analysis of a set of trajectories with respect to an energy atypicality model. The software called Trajectory Approach Analysis (TAPAS) is presented in Appendix D.

The extensions of these research are multiple. In chapter 4, another anomaly detection methodology based on Generative Adversarial Networks will be presented and compared to the FPCA algorithm. In the same chapter, an estimation of the flight-deck parameters from supervised learning models will be proposed to improve trajectory analysis. Finally, chapter 6 proposes a real-time extension of the FPCA algorithm using Dubins curves.

This chapter was dedicated to the development and the application of a post-operational trajectory anomaly detection algorithm in the context of aircraft approaches and landings. The following chapter will focus on the use of machine learning techniques in order to enhance the analysis, the manipulation and the generation of radar data. The results obtained may corroborate the results of atypicality or be used in other applications.

CHAPTER 4

Enhancing ground data analysis and generation with Supervised Learning

”Quand on remporte le tour à Sloubi, on a quatorze solutions possibles : soit on annule le tour ; soit on passe ; soit on change de sens ; soit on recalcule les points ; soit on compte ; soit on divise par six ; soit on jette les bouts de bois de quinze pouces, ça c’est quand on joue avec les bouts de bois ; soit on se couche ; soit on joue sans atouts. Et après y’a les appels : plus un ; plus deux ; attrape oiseaux ; régoudon ; ou chante Sloubi.”

Perceval - Kaamelott, Livre III, Chante Sloubi

Contents

4.1	On the use of Generative Adversarial Networks for Aircraft Trajectory Generation and Atypical Approach Detection	62
4.1.1	Introduction and methodology	62
4.1.2	Trajectory Generation	64
4.1.3	Anomaly Detection	66
4.1.4	Latent Space Representation	71
4.1.5	Discussion	74
4.2	Approach and landing aircraft on-board parameters estimation with LSTM networks	75
4.2.1	Methodology and Data	75
4.2.2	Results	78
4.2.3	Generalization	80
4.2.4	Use cases	83
4.2.5	Discussion	86
4.3	Conclusions	87

This chapter proposes to explore different use of machine learning techniques in order to enhance the analysis, and the generation of radar data. It is divided in two parts resulting from different publications. The first section presents the use of Generative

Adversarial Network for generation and anomaly detection in aircraft approach trajectories. A comparison with the methodology presented in Chapter 2 will be illustrated. This research was published in a conference [120], and was selected for a publication into a book of selected papers (in going). Finally, the second section presents the use of supervised learning techniques to predict aircraft flight deck parameters using radar data and was published in a conference [121].

4.1 On the use of Generative Adversarial Networks for Aircraft Trajectory Generation and Atypical Approach Detection

4.1.1 Introduction and methodology

This section details the work conducted around the generation of trajectory and the detection of atypical trajectories using a novel machine learning technique called Generative Adversarial Network (GAN). GAN are recent neural network techniques that have already provided successful results in various fields such as image or video generation [122, 123], image resolution enhancement [124], drug discovery [125], text-to-image synthesis [126] and many others. They enable learning the data distribution by solving a min-max optimization problem between a data generator and a data classifier. The data generator tries to generate realistic data while fooling the data classifier. The classifier tries to distinguish real data from generated data. Recently, GAN have also been applied to detect anomalies in imaging data [127].

In the specific field of trajectory generation with GAN, some work has already been conducted on learning and reproducing human motion behavior [128], on robot navigation [129], or on vehicle-to-vehicle-encounters [130]. GAN do not require prior knowledge on data to learn their distribution and are therefore well suited to applications where only physical model representation is available and little is known about responses to uncertainties or external factors such as aircraft approach trajectories analysis. Furthermore, no or few investigation has been carried out on applying these techniques to aircraft trajectory generation or anomaly detection, which motivates the use of GAN. This section aims at conducting experiments with GAN to generate realistic aircraft trajectories based on airport approach and landing data. Classical trajectory generation is based on the physical aircraft model whereas here, the generation is data driven and does not use aircraft and flights physics. It can, therefore, account for external factors that impact real trajectories such as Air Traffic Control (ATC) orders, pilot behavior or meteorological phenomena. Further investigations on the use of the GAN to detect abnormal trajectory patterns are carried out and compared with the results of chapter 3.

After introducing the principles of GAN, the application of GAN to generate ap-

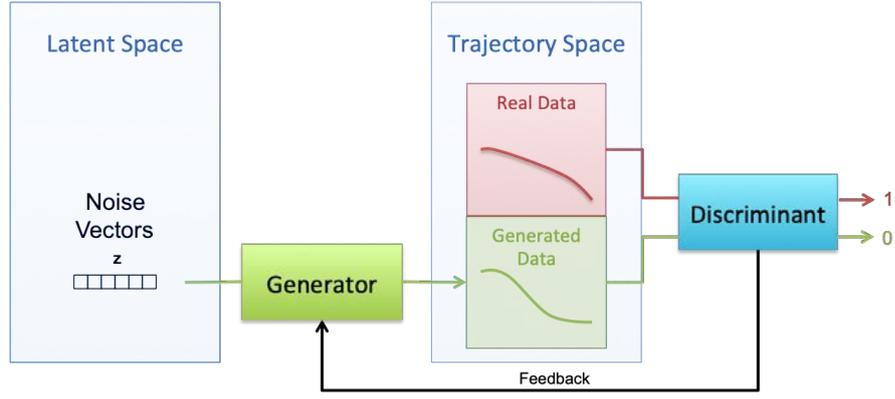


Fig. 4.1 Generative Adversarial Network Structure

proach trajectories is shown. Then, the detection of abnormal or atypical trajectories using the distribution of data learned by the network is illustrated. The relevance of the results is discussed with operational criteria and the performance of the algorithm in real operations is detailed.

GAN have recently attracted much interest in the machine learning community [122] [131]. These models have the ability to learn the distribution p_d of input data and generate new data according to the learned distribution. This is achieved through the use of a network that combines a generator G (usually a type of neural network) and a discriminator D (a classifier function). The generator G takes input noise vectors z from a low dimensional space so-called latent space, and generates new sample vectors in the data representation space (here the trajectory space). The discriminator D is trained on a given input data x to compute the probability of a sample being an input data rather than being generated by G . The process, that can be seen as a two player game is simultaneously repeated so that G minimizes $\log(1 - D(G(z)))$ (generated data that could not fool the discriminator), and D maximizes $\log(D(x))$ (real data correctly classified). The process is initiated by drawing random noise vectors z . The two player game can be summarized in the following optimization problem:

$$\min_G \max_D \mathbb{E}_{x \sim p_d} [\log(D(x))] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (4.1)$$

Figure 4.1 illustrates this principle in the specific case where x belongs to a space of trajectories. Initial vectors z are randomly generated in the latent space and mapped into the trajectory space via the Generator G . The discriminant function D returns a score value close to 1 if the generated trajectory belongs to the real trajectory data distribution or close to 0 otherwise. Next, the training phase of G receives the score feedback in order to generate a more realistic trajectory if the score is low. The process is repeated several times until an equilibrium of the min max game is found. The next section gives more details on the architecture of the generator and discriminator neural network maps used.

4.1.2 Trajectory Generation

The problem of trajectory generation is usually divided into two paradigms, model-driven generation, and data-driven generation. The approach with GAN is a data-driven generation. Since in model-driven generation, trajectories are generated with physical and dynamical models, they cannot take into consideration real-time constraints such as Air Traffic Control or even pilot behavior. Data-driven generation is supposed to provide more realistic generation considering all the parameters from real data. Model-driven generation can use real aircraft models directly, or the BADA (Base of Aircraft Data) model [132, 133] developed by Eurocontrol.

In the following, is illustrated how GAN can be used to generate aircraft landing trajectories at Paris Orly (LFPO) airport. The dataset used is composed of 4401 A320 landing trajectories on runway 26 from Flight Data Monitoring Records. The parameters selected are the longitude, the latitude, the altitude and the ground speed for the last 25NM. The initial trajectory sampling rate is one point every 4 seconds, but each trajectory is re-sampled to obtain 256 uniformly distributed points which fit a neural network structure. These parameters have been selected because they correspond to the basic parameters available in ADS-B or radar data. Thus, the results obtained can be reproduced with other data sources.

Neural Network Structures and Learning Process In order to generate aircraft trajectories, specific neural network structures were built using 1D convolutional and transpose convolutional neural networks. The neural network of the discriminant consists of four convolution layers and one fully connected layer. The neural network of the generator is built by symmetry: one fully connected layer and four convolution transpose layers with upsampling. Additional details about the dimension of each layer are given on Fig. 4.2

A uniform distribution of the noise z was arbitrarily chosen in a 4-dimensional space since the output space considers 4 dimensions (longitude, latitude, altitude, ground speed). In addition to the convolutional structure, each layer is followed by a batch normalization, max pooling, and dropout layers in order to regularize the network.

The learning task was made using Adam optimizer [134] with a decay. The learning rate starts from 10^{-3} and decreases to 10^{-7} . Networks were trained during 30 000 steps on a multi-GPU cluster. The cluster is composed of a dual ship Intel Xeon E5-2640 v4 - Deca-core (10 Core) 2,40GHz - Socket LGA 2011-v3 with 8 GPU GF GTX 1080 Ti 11 Go GDDR5X PCIe 3.0.

Generated Trajectories After the learning phase, the generator is able to compute new trajectories from sampled noise distribution. However, the obtained trajectories were noisy with mainly high-frequency noise. Therefore, a smoothing filter is applied.

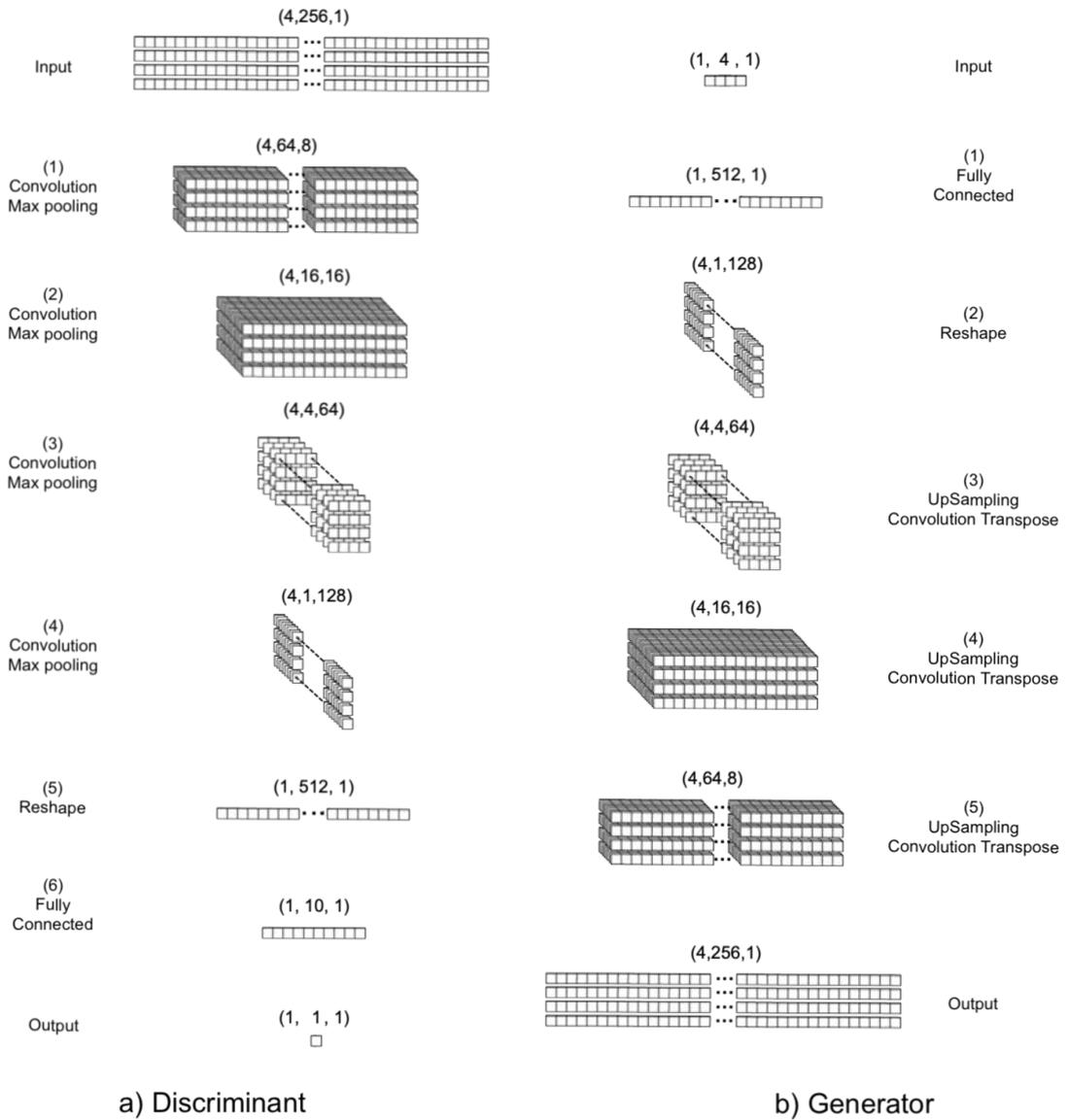


Fig. 4.2 Discriminator and Generator network architectures. On the left a), is illustrated the discriminator structure, and on the right b) the generator structure. Both structures use convolution or convolution transpose layers.

In particular, a cubic smoothing spline interpolation was computed to remove the noise from the generated trajectories. Figure 4.3 illustrates filtered generated trajectories for all the parameters.

The overall shape and distribution of the generated trajectories is satisfying since they followed the original distribution. Nevertheless, one can see that the generator is not able to capture some types of patterns. For the altitude profile, it is known that aircraft follow levelled-off path before descending on the glide path, but this is not captured by the generator. The same behavior is observed for the extended runway centre line which should be followed from 10NM to the threshold, but the generated trajectories barely followed the localizer path for the last nautical miles. This may be linked to the difficulty of convergence in GAN models. As a reminder, GAN models solve a min-max problem, which implies a very unstable optimal saddle point. The optimal solution in Equation (4.1) may not be achieved, meaning that some information such as the levelled-off pattern might not be learned during the training phase.

This effect is not problematic for anomaly detection. Even if the generator cannot reconstruct data with high levels of accuracy, it has still learned the general distribution of the data. Thus data that does not have characteristics similar to the learned data will have a large reconstruction error and will be detected as anomaly. Furthermore, anomaly detection using the discriminator is not affected at all. The generation with GAN will be compared in the following section 5.1 with generation using Functional Principal Component Analysis framework.

4.1.3 Anomaly Detection

In the following paragraphs, is illustrated how GAN provide solutions to the anomaly detection problem. The two ways of applying anomaly detection with GAN are first explained. Then, each way is applied and illustrated on operational data.

As explained in Subsection 4.1.1, GAN combine a generator G , and a discriminator D . After the learning task, the discriminator has been trained to recognize real data from generated data. Consequently, the first approach to anomaly detection consists in using the score of D . Indeed, the closer the score is to 0, the less realistic the data is supposed to be, or in other words, the less likely it is to belong to the original distribution.

Another way to perform anomaly detection with GAN is to build an encoder E (usually another neural network). The encoder embeds samples from the trajectory space to the latent space. The encoder is illustrated in Figure 4.4. It can be automatically tuned during the GAN training (this setup is known as BIGAN), or after the training (Encoder). The anomaly detection can be applied to a dataset with the following process : first, encode each trajectory in the latent space with the encoder E , next, rebuild the trajectories through the generator, and finally compute a distance between the original and

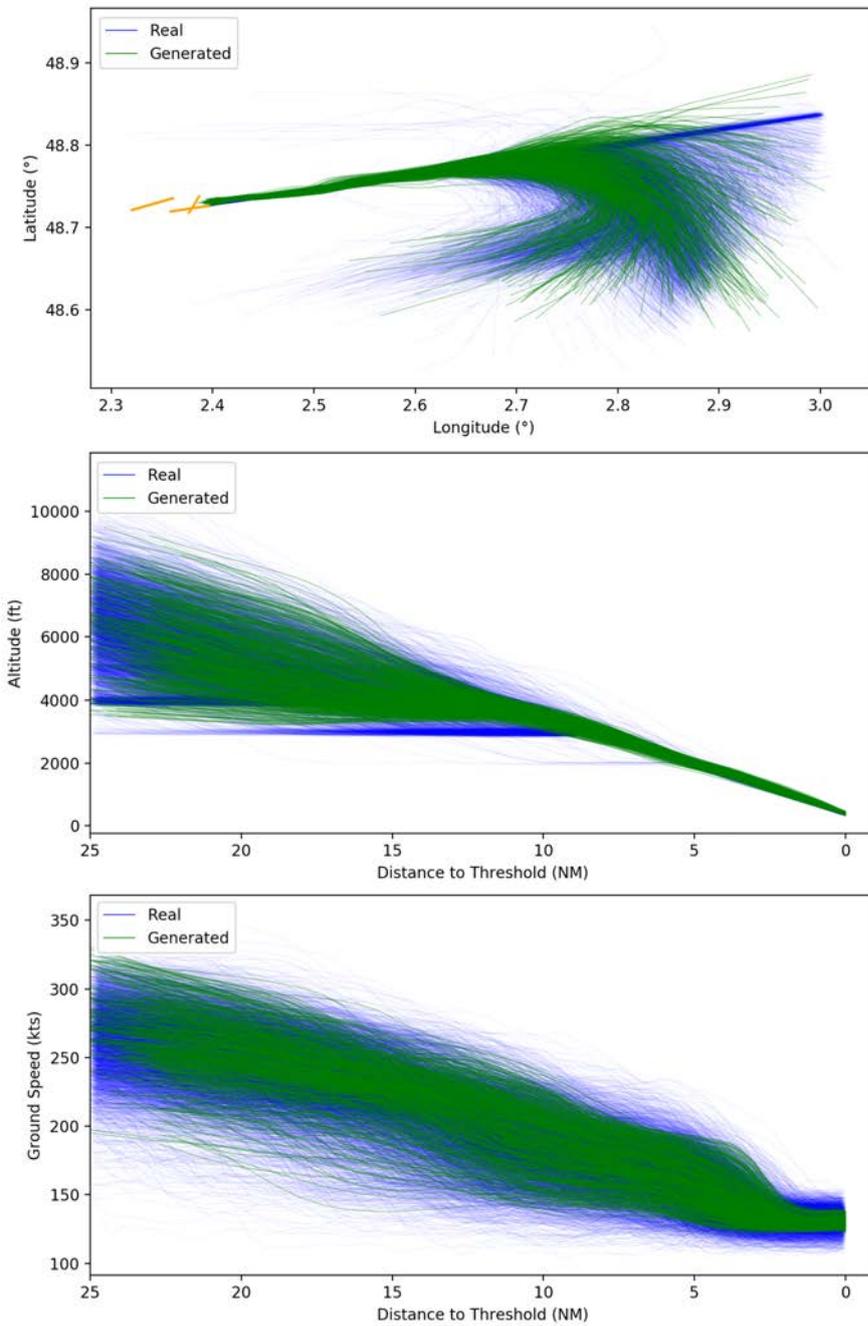


Fig. 4.3 1000 trajectories at Paris Orly Airport generated with a GAN. In blue are represented the original trajectories and in green the generated trajectories. At the top, the longitudinal path is represented, in the center, the altitude profile, and at the bottom the ground speed profile.

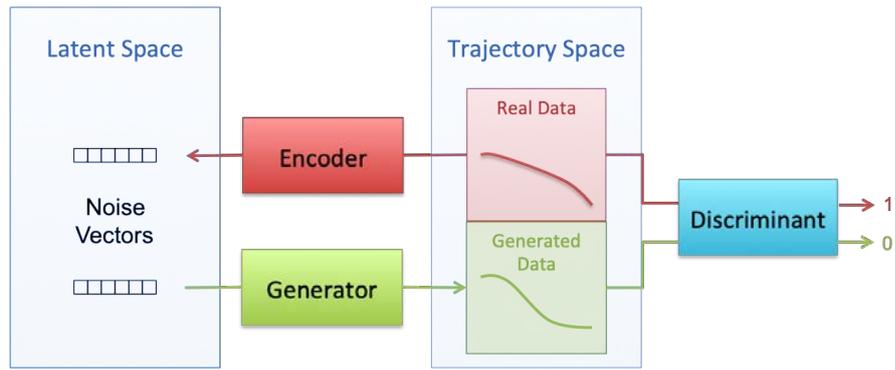


Fig. 4.4 Extended structure of a Generative Adversarial Network with an Encoder

reconstructed trajectories. The most distant trajectories can be considered as anomalies since the generator was not able to rebuild the trajectory properly. Indeed, if a trajectory does not belong to the trajectory distribution learned by the generator, the reconstruction error will be high. This approach is very similar to auto-encoder anomaly detection [70, 71]. Nevertheless, GAN are richer since they also provide trajectory generation. The encoder network structure is similar to the discriminant network detailed in Figure 4.2. However, the last layer is sized to correspond to the latent space dimension.

Anomaly Detection using the Discriminator The first approach to performing anomaly detection is to use the discriminator. It is trained to distinguish real samples from the original data set and generates samples from the generator. Therefore, its natural behavior tends to give a score next to 1 for trajectories that are similar to the original data set and a score close to 0 for atypical trajectories.

This method of anomaly detection is applied to the original dataset of the Paris Orly Airport trajectories and the results are shown on Figure 4.5. Red lines correspond to trajectories with the minimum discriminator score for the dataset, green lines to the maximum discriminant score, and orange lines to intermediate values.

The anomaly detection with the discriminator shows interesting results for the longitudinal trajectories and for the altitude profiles. The typical altitude profile (in green) follows a levelled-off path before descending on the glide path, which corresponds to the published procedure. On the other hand, the atypical profiles present high altitude or even Glide Interception From Above. 2D trajectories are illustrated at the top of Figure 4.5. It seems that typical 2D paths are approaches from the south and east, while less typical come from the west and very atypical from the north with holding patterns. The ground speed profiles, at the bottom of Figure 4.5, do not show any specific results. This could be explained by the fact that the main distribution (in green) shows a sparse distribution of ground speed.

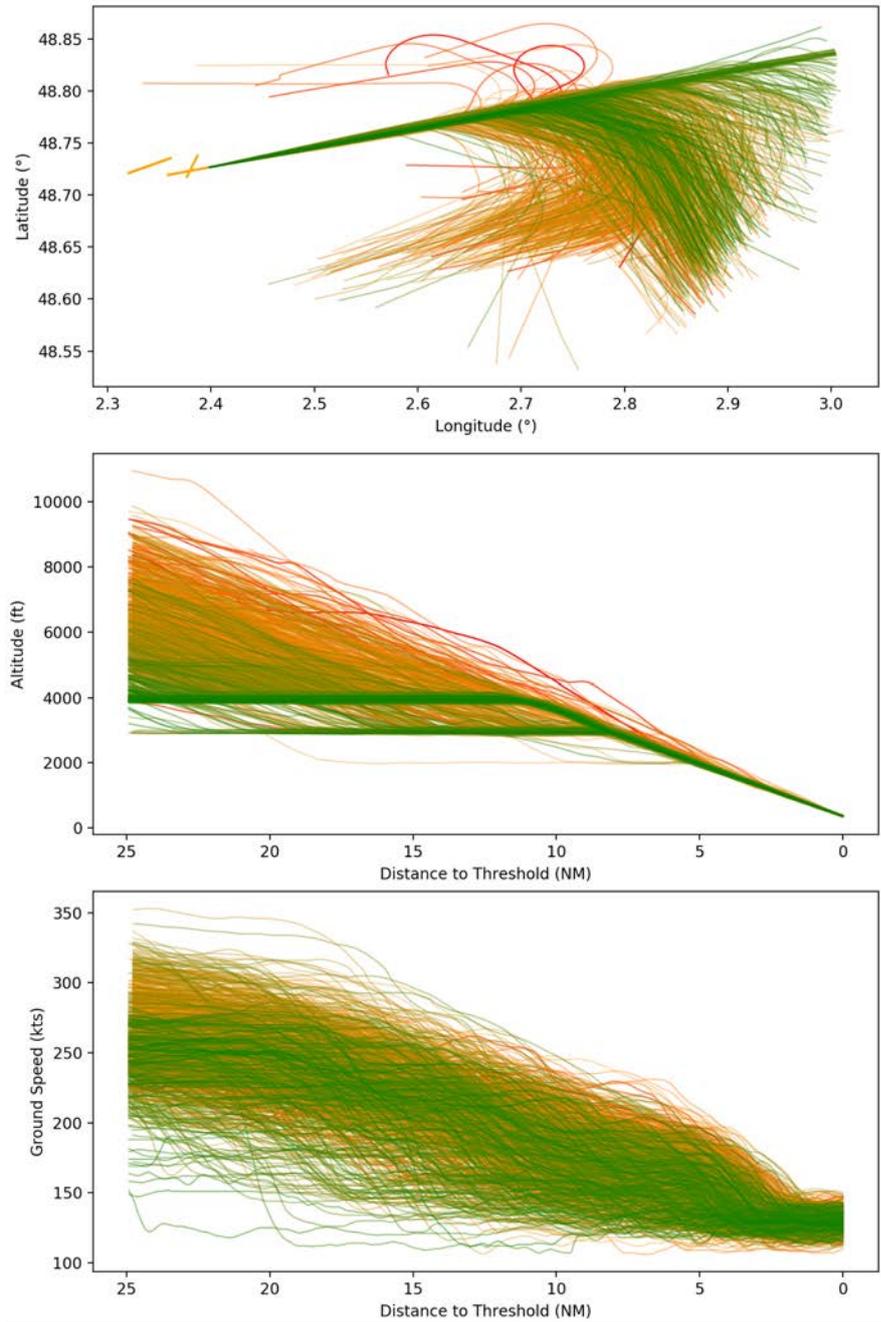


Fig. 4.5 Trajectory colored with the score given by the discriminator on the original dataset.

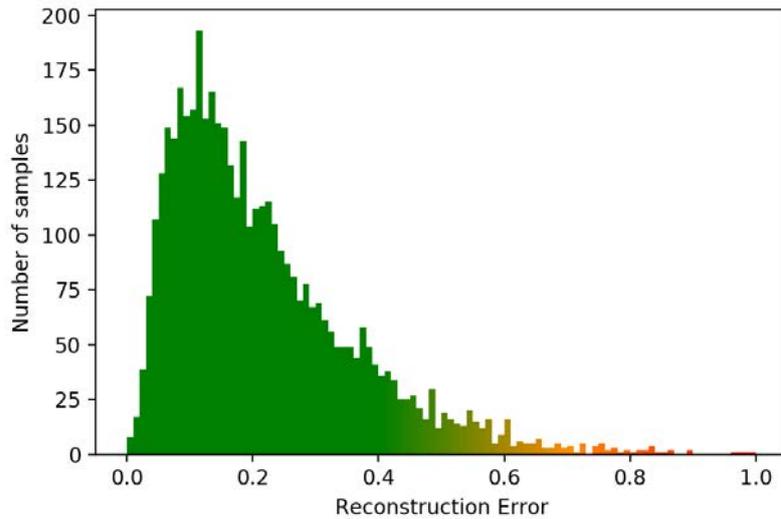


Fig. 4.6 Normalized distribution of the encoder-generator reconstruction errors for the specific total energy trajectories at Paris Orly Airport

Anomaly Detection using the Encoder This paragraph illustrates the use of the encoder to detect anomalies. The encoder is tuned automatically after the GAN training phase. The encoder embedded the trajectory samples to the latent space. The anomaly detection is performed in three steps. First, each trajectory is embedded to the latent space with the encoder. Second, all trajectories are rebuilt through the generator. Third, the reconstruction error (L2 Norm) is computed between the original trajectories and the rebuilt trajectories. Finally, trajectories with high reconstruction errors are considered as atypical.

In order to be able to compare results with the functional principal component analysis method explained in chapter 2, the anomaly detection is applied to specific total energy trajectories. In this purpose, another network is trained to generate and encode specific total energy trajectories extracted from Paris Orly landing trajectories. Figure 4.6 illustrates the normalized distribution of reconstruction errors. The color variation is from green for small errors to red for large reconstruction errors. This corresponding color (and reconstruction error) is also used to represent the specific total energy trajectories in Figure 4.7.

The anomalies are mainly composed of high energy profile trajectories due to glide interceptions from above (in red at the top of the figure). The flight with the highest reconstruction error is selected and discussed in the following. The comparison with the atypical coefficient algorithm using FPCA from chapter 3 is detailed below. Figure 4.8 illustrates its altitude speed profiles. The colored dots correspond to atypical coefficients between 0 for typical and 1 for atypical. Between, 25NM and 15NM, one can observe a large atypical area due to high energy (high altitude and high speed).

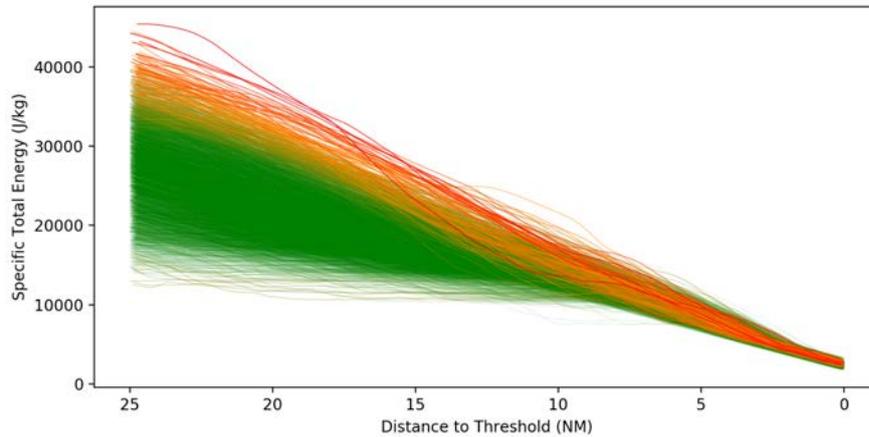


Fig. 4.7 Specific total energy trajectories at Paris Orly Airport

FPCA algorithm and GAN anomaly detection presents similar results. The 10 highest reconstruction error flights from the GAN anomaly detection experiments, were also considered as atypical by the FPCA algorithm

4.1.4 Latent Space Representation

The encoder enables the trajectories to be embedded in latent space. Each trajectory is then represented as a single point in a low dimensional space. Therefore, this enables a simpler representation of a group of samples with a dimensionality reduction. The embedding of the original trajectories in two first dimensions in the latent space is represented at the top of Figure 4.9. The corresponding trajectories are illustrated in the center and the altitude profile at the bottom. In order to analyse the variability captured by the first dimension, a color map (blue, orange, green) is applied on the re-description space. The same color map is also displayed for the real trajectory.

It can be seen that this dimension mainly encodes the variability of altitude profiles. Flights which values on the first FPCA dimension are close to 1, correspond to continuous descent profiles or those with glide interceptions from above. The values close to -1 are trajectories with large levelled-off flights.

This representation enables different applications such as clustering, data analysis, or linear interpolation. For example, one may use this representation for approach procedure detection. Suppose we focus on a QFU approach with separate published procedures, a GNSS RNAV, a visual approach and a VOR/DME approach. These three types of procedures have a characteristic footprint. Approaches of the same type will be grouped together and could be easily discriminated in the latent space.

The latent space also enables a linear interpolation. Linear interpolation is a continuous deformation between two Objects. It is a complicated mathematical problem for trajectories or infinite dimension objects. The use of GAN makes it possible as the latent space is a low-dimensional normalized space. It would therefore consists in inter-

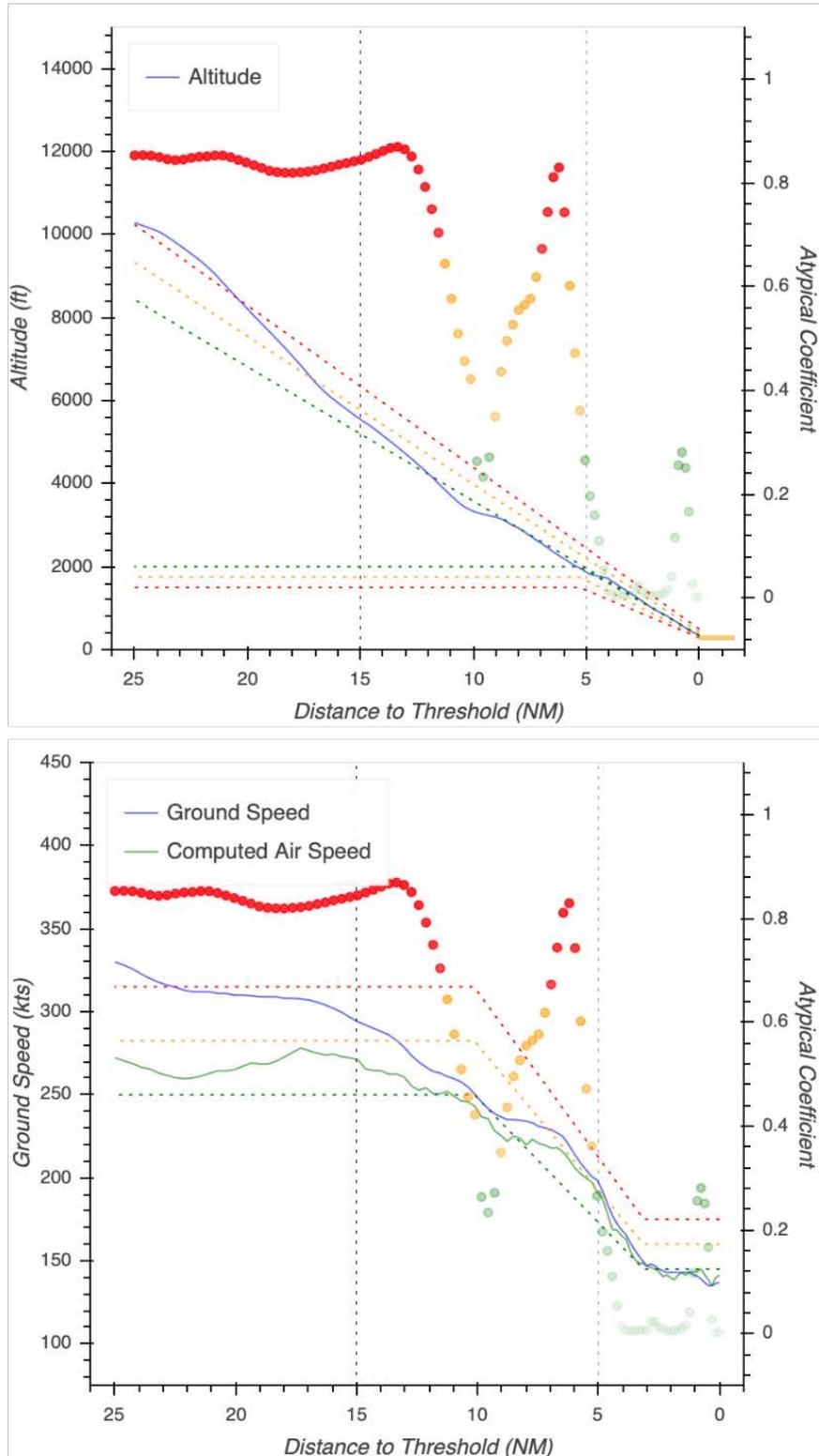


Fig. 4.8 Altitude and speed profile of the highest reconstruction error flight presenting a Glide Interception From Above. The flight presents a glide interception from above with high ground speed, and a little late power reduction. The colored dots correspond to the atypical FPCA coefficients of the total energy defined in Chapter 2. The dashed colored lines correspond to operational limits: nominal (green), warning (orange), and critical (red). For more details see Appendix A

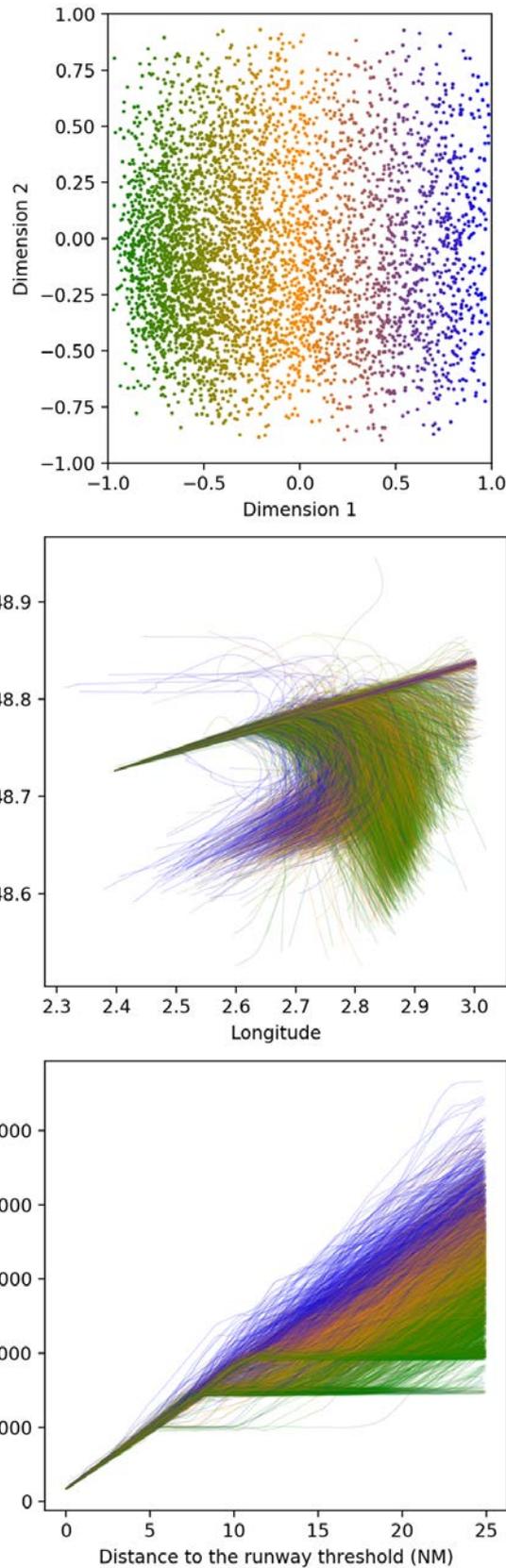


Fig. 4.9 Latent space representation and real trajectories altitude and lateral profiles colored with a linear interpolation for the first FPCA axis. Flights which values on the first FPCA dimension are close to 1, correspond to continuous descent profiles or those with glide interceptions from above. The values close to -1 are trajectories with large levelled-off flights

polating by a straight line the two points representing the trajectories in the latent space. Then, a discretization of this line could be used to rebuild the corresponding trajectories with the generator. It will then provide a smooth interpolation between the two initial curves.

4.1.5 Discussion

In the following, operational and scientific insights are provided for use of GAN. There are several possible methodologies to detect anomalies on large dimension spaces. The GAN had not been used for the moment in this aeronautical safety framework. The operational objective is the analysis and identification of precursors to potential safety incidents or events. Atypical energy managements have shown a strong correlation with flight safety events as illustrated in chapter 2, therefore GAN could be used post-operationally to analyse trajectories.

In addition, it is easy to understand the use of the total energy. Flying an aircraft consists in managing altitude and speed, and therefore its total energy. The final approach phase is a phase of decreasing energy. The plane goes from high speed at high altitude to landing where its potential energy and speed must cancel each other out. Energy transfers generally imply that a decrease in potential energy results in an increase in kinetic energy. On a standard 3° glide path, aircraft are facing the constraint of simultaneously reducing kinetic energy and potential energy. This constraint implies good energy management upstream, generally by configuring the flaps and landing gear to increase the drag and decrease the total energy, or by using a level-off deceleration flight to reduce the speed before descending on the glide path. Some aircraft have aerodynamic characteristics such that deceleration on final approach can be very difficult, especially in poor weather conditions such as tailwind. Therefore, there is a real operational interest in detecting atypical energy management that could lead to incidents or accidents.

There are many similarities between GAN, and the FPCA methodology presented in chapter 3. The ultimate goal is to estimate a large dimension data distribution. GAN bring the generative aspect and the possibility to use both the discriminator and the generator for anomaly detection. However, like auto-encoders, they are subject to the convergence of the learning phase (even stronger for GAN). The FPCA methodology has the advantage of being deterministic. The use of GAN model on a sliding window as proposed in chapter 3, seems much more complex and involves ensuring the convergence of each model resulting from a sliding window.

Further work should include the analysis of tailored network architectures and learning, or extensions to WassersteinGAN [135] that can learn data from multimodal distributions.

These experiments show that the generated trajectories follow realistic patterns. This confirms that GAN are promising alternatives to model-based trajectory simulators. The resulting generated trajectories are based on past historical data and therefore account for external factors that are often difficult to embed in physical models. Further experiments were also provided with GAN to detect atypical trajectories. The comparison with the algorithm presented in chapter 3 also confirms that reported anomalies are relevant. This research was one of the first attempts to generate aircraft trajectories with such generative machine learning tools. The next section will investigate the use of other supervised learning models such as LSTM in order to predict flight deck parameters with radar data.

4.2 Approach and landing aircraft on-board parameters estimation with LSTM networks

This section investigates the use of supervised learning model in order to estimate flight deck parameters using radar data. As presented in Chapter 2.2, different machine learning models were used to predict aircraft flight deck parameters. Besides, a recent type of neural network called Long Short Term Memory has proven to be very efficient on time series data to predict trajectories [136] or hard landings [137]. To the best of the authors' knowledge, LSTM neural networks have not already been used to estimate aircraft on-board parameters such as fuel flow rate or flap and landing gear settings. Therefore, this section investigates the use of such kind of neural network structure on its potential use to enhance the analysis of the ATM system.

4.2.1 Methodology and Data

This subsection presents the machine learning models chosen for the following experiments as well as the data considered.

Machine Learning and Multilayer Perceptron Neural Network A learning process consists in using data analysis methods and artificial intelligence to predict the behavior of a system. The aim is to define a model that will fit as best as possible the system. Machine learning algorithms define learning models h_θ , with parameters θ , that approximate the system function. The learning process is done upon a finite training set \mathcal{D} , and aims at minimizing the empirical risk over the training set by tuning the parameters θ of the learning model. The empirical risk is evaluated with an error function, the least square error being the usual choice for regression [16, 138, 139].

There are various learning models, and in this section the choice of a particular neural network named LSTM is made and compared to a classical Multi-Layer Perceptron

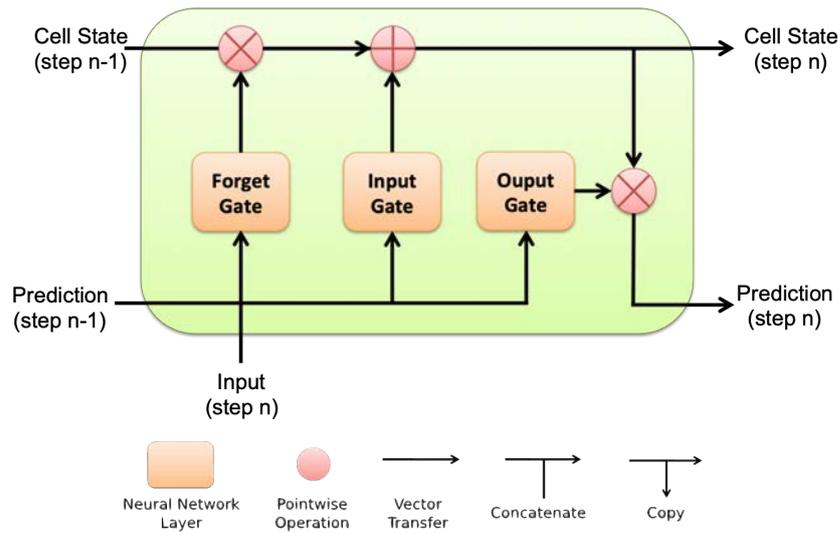


Fig. 4.10 Simplified structure of a LSTM cell. It is composed of three gates that enable the network to learn which features of the time series is important to keep or forget

(MLP). LSTM networks were designed as an enhancement of Recurrent Neural Network to perform better supervised learning tasks on time data series [140, 141, 142]. LSTMs are capable of learning long-term dependencies, while simple RNN only learn short term dependencies. LSTMs use a cell state that keeps information from the past, and three gates that update the cell state and compute the prediction. First, the forget gate enables updating the cell state to forget information that is no longer relevant based on the current input. Then, the input gate enables saving in the cell state relevant information from the current input. Finally, the output gate computes the prediction using the updated cell state and the current input. A simplified illustration of an LSTM structure is depicted in Figure G.2.

From data to features The data set is composed of 14807 A320 approaches and landing Flight Data Records from two French airlines. This study focuses on predicting on-board parameters such as flaps configuration, fuel flow rate, or landing gear position, knowing the last minute (with a four-second sampling rate) of radar equivalent available parameters: ground speed, vertical speed, and altitude. Additional features are extracted from these parameters. First, the height is computed with the local QNH. Energy features such as specific energies per unit of mass (potential, kinetic, total) are then computed. In summary, the inputs of the neural network are one-minute time series of ground speed, vertical speed, altitude, height, specific potential energy, specific kinetic energy, and specific total energy.

The data set is first divided into two subsets: 10% for the testing set and 90% for

the training set. Regarding the training set, 80% is used for actual training, and 20% for the validation.

Additionally, 200 A330 and 2000 B737 approach and landing Flight Data Records are used to assess the generalization of the model to other aircraft types.

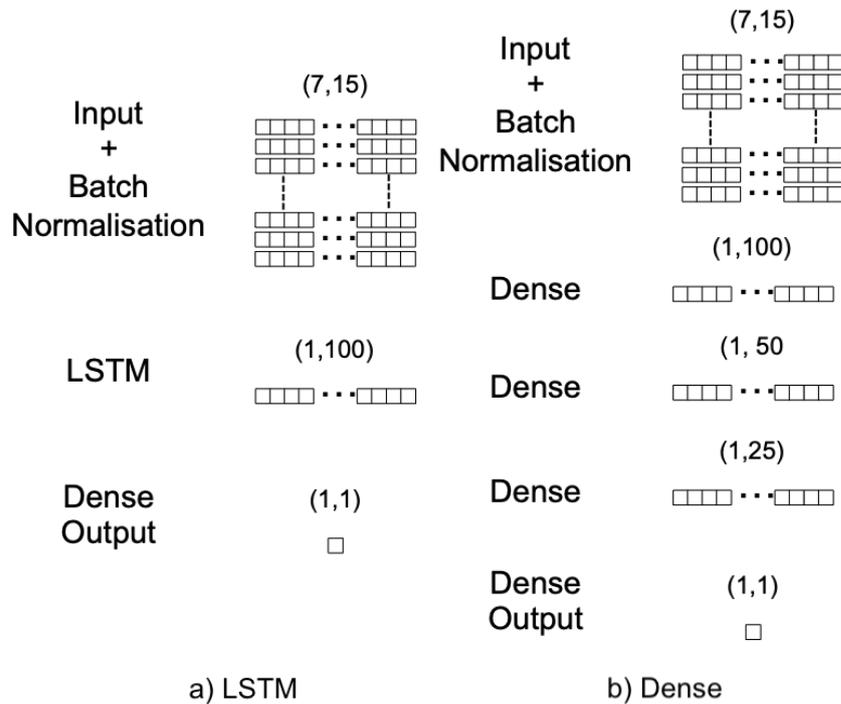


Fig. 4.11 Neural network architectures used to predict aircraft on-board parameters. On the left, is represented the LSTM architecture, and on the right the MLP Dense architecture.

Network Architectures and Learning This paragraph describes the neural network architectures used in the experiments. In the following, two neural network architectures are presented, their performance will be detailed in the following subsection. First, the LSTM neural network is composed of two layers and a regression output layer. The first layer is a batch normalization. The second layer is an LSTM layer with 100 units and a Rectified Linear Unit activation function. The output layer is a single neuron dense layer. This architecture will be referred to as LSTM. Second, the dense MLP is composed of one input batch normalization layer and three dense hidden layers with 100, 50 and 25 neurons respectively with ReLU activation. The output layer is a single neuron dense layer. This architecture will be referred to as Dense. Figure G.3 illustrates the networks' architecture.

The learning task is made using the Adam optimizer [143] with decay and the loss function used is the least square error. The learning rate is 10^{-3} and the decay is 10^{-9} . Each model is trained during 30 epochs several times on a multi-GPU cluster. The

best network over the validation set is kept and analyzed in this paper. The cluster is composed of a dual ship Bi-Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz (80 Core) - with 8 GPU GeForce RTX 2080 Ti.

4.2.2 Results

Performance metrics In order to compare the performance of the different models, four different indicators were used: the Mean Error (ME), the Mean Absolute Error (MAE), the Normalized Root Mean Squared Error (NRMSE), and the Pearson correlation score. Let \mathcal{D} be a data set and h a function.

The mean error **ME** or mean absolute relative prediction error of h over \mathcal{D} is computed using the following formula with $|\mathcal{D}|$ being the cardinal of \mathcal{D} , and $h(x)$ the predicted value of input x , while y the true output:

$$\mathbf{ME}(h, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \frac{|h(x) - y|}{y} \quad (4.2)$$

The mean-absolute error **MAE** is calculated using the following formula:

$$\mathbf{MAE}(h, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} |h(x) - y| \quad (4.3)$$

As for the **ME**, the smaller its value is, the more accurate the prediction is.

Regarding the Normalized Root Mean Squared Error **NRMSE**, similarly the smaller its value is, the more accurate the prediction is. It is calculated using the following formula

$$\mathbf{NRMSE}(h, \mathcal{D}) = \frac{1}{\sigma_{\mathcal{D}}(h)} \times \sqrt{\frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} (h(x) - y)^2} \quad (4.4)$$

where $\sigma_{\mathcal{D}}(h)$ is the standard deviation of the predicted value over \mathcal{D}

The Pearson correlation score indicates how well the curve of the predicted parameters follows the actual curve. The closer the score is to 1, the better the prediction is. It is calculated using the following equations:

$$r(h, \mathcal{D}) = \frac{\sum_{(x,y) \in \mathcal{D}} (h(x) - \bar{h})(y - \bar{y})}{\sqrt{\sum_{(x,y) \in \mathcal{D}} (h(x) - \bar{h})^2} \sqrt{\sum_{(x,y) \in \mathcal{D}} (y - \bar{y})^2}} \quad (4.5)$$

where \bar{h} (resp. \bar{y}) is the average of $h(x)$ (resp. y) over \mathcal{D} .

In practice, to evaluate the fuel flow rate estimation, the **ME**, the **NRMSE** and the Pearson score are computed per trajectory and a global average score is given. Besides, the fuel consumption (in kg) is computed per trajectory and a **ME** score is given for all the trajectories. For the landing gear position and the flap setting, the remaining distance to the runway threshold is computed per trajectory when gear down (resp. flaps 1, 2,

etc.) action is applied for both actual and predicted parameter. For example, if real flaps 1 appeared at 11NM and the model predict 10.5NM, the MAE error is 0.5NM. Then, the MAE score is averaged for all the trajectories to give an aggregated score. Therefore, it gives the average absolute distance difference between real and predicted values.

Airbus A320 In this paragraph are presented the performance metrics on the A320 test set of 1410 flights. Both LSTM and MLP models show high scores on the fuel flow rate estimation. The average scores for all the 1410 flights are summarised in Table 4.1. The high Pearson scores of 0.938 for LSTM model, and 0.927 for Dense model, indicate a high correlation between the real fuel flow and the prediction. This behavior is well illustrated in Figure 4.12. Indeed, the prediction seems to follow properly each variation of the real fuel flow rate. Even though the proposed models do not use any mass or take-off mass information, the ME and the NRMSE scores of the proposed models are similar to those described by Chati et Al. in [81, 82] for the descent and approach phases. The LSTM ME score is, on average, around 13.59% with a standard deviation of 3.26%, and the NRMSE is around 26.52% with a standard deviation of 8.41%. The fuel consumption ME for the test data set is around 4.2% (LSTM) and 4.5% (Dense), corresponding to an average absolute error of 8 kg. It seems to indicate that a good estimation of the fuel flow rate can be given without any mass information. Furthermore, the LSTM network outperforms the Dense network.

Metric	LSTM	Dense
ME (%)	13.59 (3.26)	14.55 (3.97)
NRMSE (%)	26.52 (8.41)	27.97 (8.70)
Pearson	0.938 (0.046)	0.927 (0.065)
Fuel Consumption - ME (%)	4.2 (3.28)	4.52 (3.56)

Table 4.1 Predictive performance of the LSTM and the Dense models on A320 test data for the fuel flow rate. Each entry shows the mean and the standard deviation (within parentheses) of the evaluation metric across all the flights in the test data set.

In the following, the performance results of the landing gear and flap setting predictions are analyzed. Only the LSTM model performance metrics are presented here since it outperforms the Dense model. The average MAE performance and its standard deviation (within parentheses) are summarized in Table 4.3. The landing gear down action is well localized with an average MAE of 0.99NM. Regarding the flap setting, the average distance error goes from 2.27NM for flaps 1 to 0.88NM for flaps FULL. Examples of flap and landing gear setting prediction are illustrated in Figures 4.13 and 4.14. The configuration prediction might be enhanced if extra information such as the air speed (available with radar mode S) are added. Indeed, these parameters are directly linked to the aircraft flight dynamics and the airspeed.

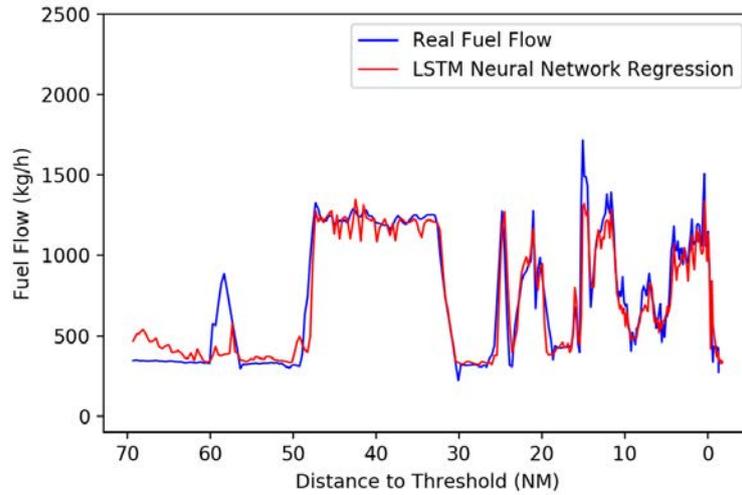


Fig. 4.12 Fuel flow rate prediction on a flight of the test data set

Parameter	MAE (%)
Landing Gear	0.99 (2.38)
Flaps 1	2.27 (4.03)
Flaps 2	1.67 (2.35)
Flaps 3	0.97 (1.15)
Flaps FULL	0.88 (1.03)

Table 4.2 Predictive performance of the LSTM models on A320 test data for the flap and landing gear setting

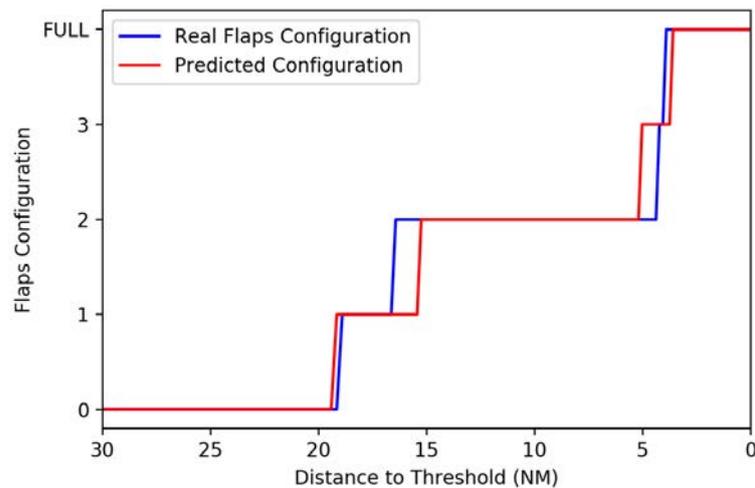


Fig. 4.13 Flap configuration prediction on a flight of the test data set

4.2.3 Generalization

In this section, the generalization of the fuel flow rate and the landing gear configuration learning model to other aircraft types is investigated. Indeed, the model is built with

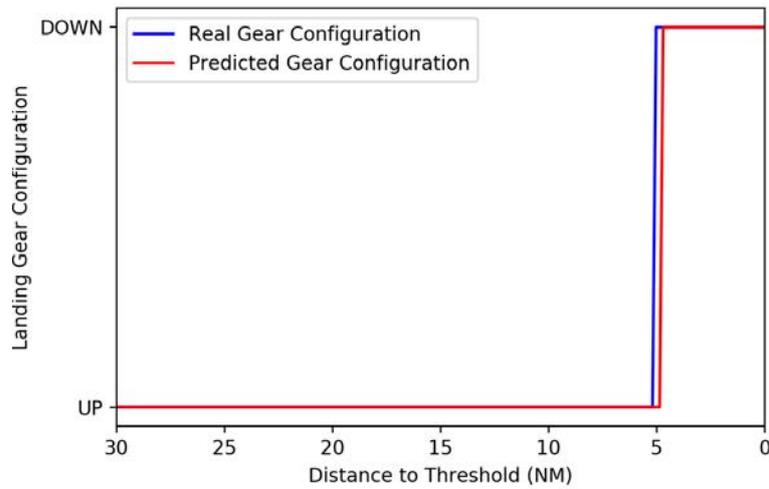


Fig. 4.14 Landing gear configuration prediction on a flight of the test data set

A320 flight data records. The access to all the aircraft types might be very difficult. Therefore, the performance of the learning model on two other aircraft types (B737 and A330) is analyzed. The flap setting model generalization is not investigated since the flap configurations are different depending on the aircraft type.

Boeing B737 The B737 data set is composed of 2000 FDR approach and landing trajectories equally divided into two subsets. The first one is located at Paris-Orly Airport (LFPO), and the second at Madrid-Barajas Airport (GMAD).

For both airports the fuel flow rate model is still performing well with a correlation score of 0.917 (LFPO) and 0.921 (GMAD) as summarized in Table 4.3. The ME (15.57% and 15.29%) and NRMSE metric 32.94% (LFPO) and 31.7% (GMAD) are not as good as the A320 test data set. Indeed, like in the example shown in Figure 4.15, there is a small offset between the real and the predicted fuel flow rate. Even though the A320 training set did not contain any flight at GMAD airport, the fuel flow rate model seems to perform well on this new platform. Moreover, the fuel consumption ME is on average of 4.35% (LFPO) and 4.85% (GMAD), which is similar to the mean error for the A320 test set. It confirms that the model is generic and can be used for any approach at any airport. Furthermore, the generalization to B737 seems to be relevant without any calibration. It is less precise than the A320 but depending on the use, the model might be efficient enough.

Regarding the landing gear configuration, the distance MAE is higher than for the A320 with 1.23NM and 1.92NM. The generalization process for the GMAD data set seems to be even more difficult.

Metric	B737 - LFPO	B737 - GMAD
ME (%)	15.57 (3.7)	15.29 (3.51)
NRMSE (%)	32.94 (10.25)	31.7 (8.87)
Pearson	0.917 (0.045)	0.921 (0.042)
Landing Gear - MAE (NM)	1.23 (2.21)	1.92 (3.52)
Fuel Consumption - ME (%)	4.35 (3.36)	4.86 (3.81)

Table 4.3 Predictive performance of the LSTM model on B737 test data for the fuel flow rate and landing gear at LFPO and GMAD airport

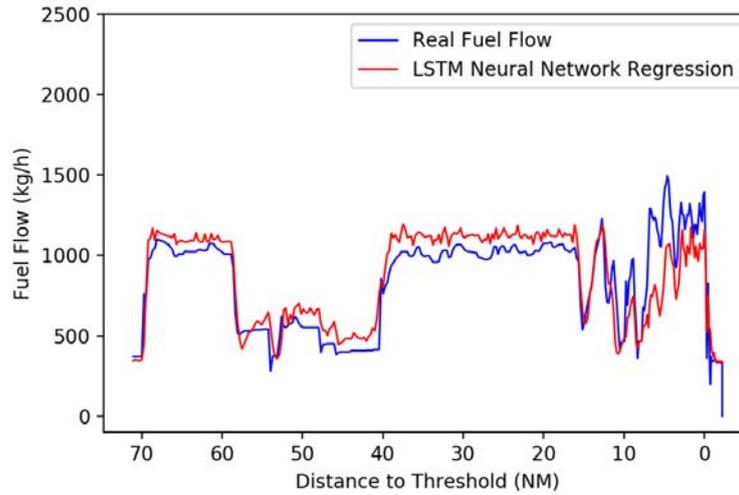


Fig. 4.15 Fuel flow rate prediction on a flight of the B737 test data set

Airbus A330 The A330 data set is composed of 200 FDR approach and landing trajectories at Paris Orly Airport (LFPO).

The generalization of the fuel flow rate model is not as simple as the B737 since the fuel flow rate values are not in the same range. Therefore, a very simple optimization problem is solved to define a scaling coefficient: c . It consists in minimizing the sum of square errors of the scaled model:

$$\min_{c \in \mathbb{R}^+} \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} (c \cdot h(x) - y)^2 \quad (4.6)$$

This is a polynomial function in c which optimal solution c^* is obtained by canceling out the c derivative:

$$c^* = \frac{\sum_{(x,y) \in \mathcal{D}} y \cdot h(x)}{\sum_{(x,y) \in \mathcal{D}} h(x)^2} \quad (4.7)$$

This method could have also been applied to the B737. With the A330, $c = 2$ is selected and the result of the prediction is illustrated in Figure 4.16. “

Besides, another solution could have been to use the values given by the BADA model. Indeed, it is possible to define a ratio of the fuel flow coefficient of the A330 over the A320. The coefficient is given per altitude and summarized in Table 4.4. The coefficients are similar (around 2 and increasing up to 2.5) to the coefficient obtained with the optimization problem.

It is possible to enhance the optimization problem by computing a scaling coefficient function of the flight level. The problem will estimate different values minimizing the error. On the other hand, it gives a possible solution without on-board real fuel flow rate for a particular aircraft type. One can refer to the coefficient ratio given by the BADA model to scale the A320 model and apply it to estimate the fuel flow rate to a new aircraft type.

Altitude (FL)	Coefficient ratio A330 / A320
0	1,9678
5	2,0333
10	2,0774
15	2,1309
20	2,1864
30	2,3018
40	2,4246
60	2,5184
80	2,5416
100	2,5422

Table 4.4 Fuel flow BADA coefficient ratio between A320 and A330 depending on the flight level

Moreover, the result of the performance metrics on the scaled LSTM model is summarized in Table 4.5. The ME is worse than the B737 with on average 20,77%. However, the Pearson correlation score is high with 0.93 and the NRMSE is around 33.18 similar to the B737. The fuel consumption ME is on average 4.84%, it is still acceptable with a small increase compared to the A320 test set. The generalization to the A330 seems to be relevant with the scaled model and acceptable depending on the use.

Regarding the landing gear setting, the distance MAE is higher than for the A320 and similar to the B737 data set with 1.63NM.

4.2.4 Use cases

Analysis of atypical flight parameters These models can be used to enhance the analysis of atypical approaches. As an example, an atypical A320 approach observed during the COVID-19 period from chapter 3 is illustrated in Figure 4.17. On top of the Figure is represented both the longitudinal trajectory and the altitude profile. At the bottom is illustrated the speed profile, and the prediction of the different models for

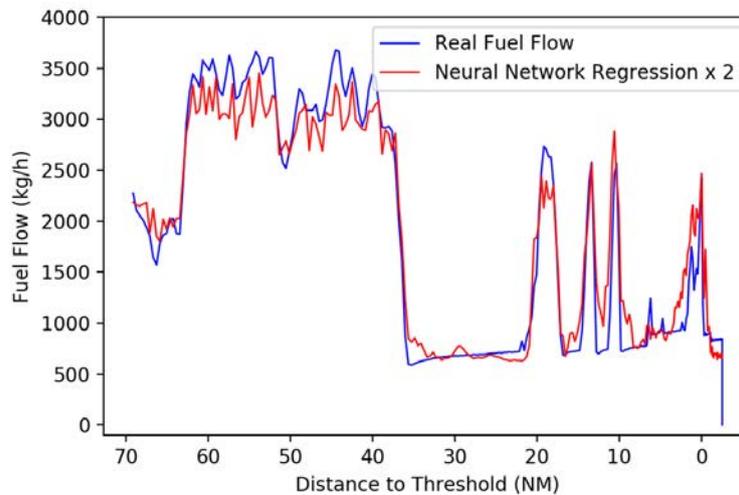


Fig. 4.16 Fuel flow rate prediction on a flight of the A330 test data set

Metric	A330 - LFPO
ME (%)	20.77 (4.4)
NRMSE (%)	33.18 (9.94)
Pearson	0.930 (0.041)
Landing Gear - MAE (NM)	1.63 (1.85)
Fuel Consumption - ME (%)	4.84 (3.53)

Table 4.5 Predictive performance of the scaled LSTM model on A330 test data for the fuel flow rate at LFPO airport

this A320 approach. The estimated fuel flow is represented in blue. The gear setting is illustrated in red. The flap configuration is depicted in green. Besides, for this analysis a speed brakes model was also created and represented in cyan.

The different models estimate that the aircraft was probably in idle mode during all the approach associated with a low consumption. It anticipated its aerodynamic configuration with gear and flaps down at around 10NM from the runway threshold, just before getting back onto the glide path. Besides, the speed brakes model estimates a use of air-brakes at around 20NM and 10NM.

Evaluation of ATM system environmental performance These models can also be used in the evaluation of the environmental performance of aircraft by Air Navigation Service Providers. The applications are multiple, in post-ops analysis. The estimation of parameters such as the fuel flow or the aerodynamic configuration allows to evaluate the fuel consumption but also to estimate the noise impact via noise models.

In addition, the great flexibility of the models makes it possible to foresee a real time use allowing to give an information of the environmental impact to the air traffic controllers. As example, a replay of an A320 approach, landing at Roissy Charles

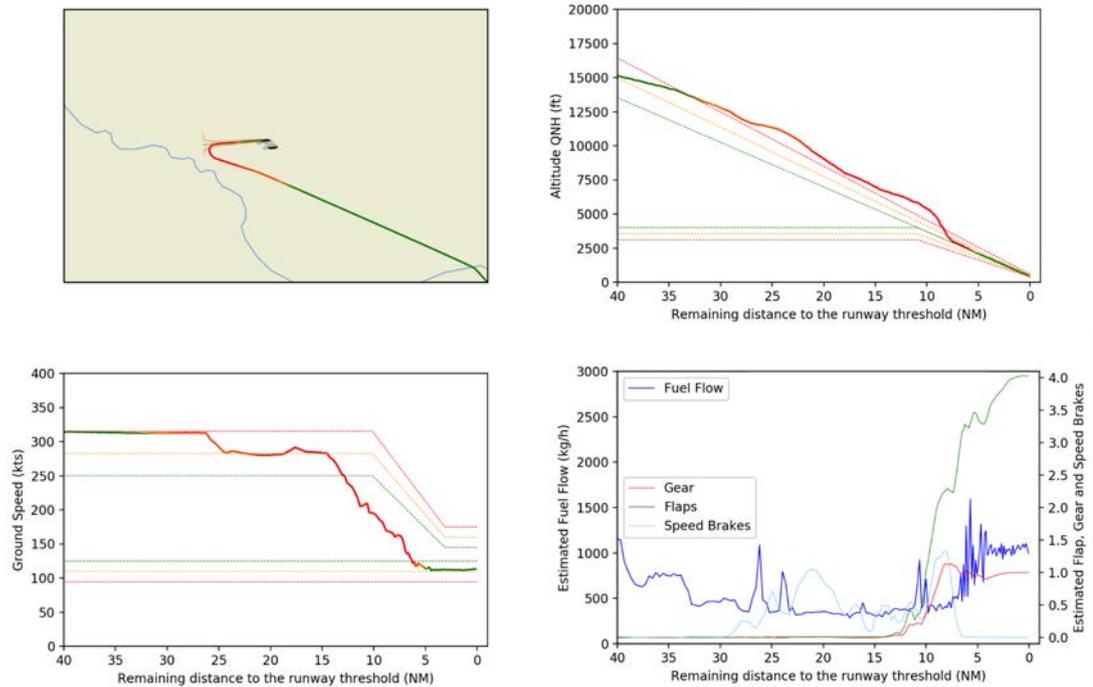


Fig. 4.17 Lateral, altitude, ground speed profiles and predicted parameters of an A320 glide interception from above at Roissy Charles de Gaulle airport, right y-axis information are the following Gear (0, 1 : up, down), Speed brakes (0, 1 : not commanded, commanded), Flaps (0, 1, 2, 3, 4 : Flaps 0, 1, 2, 3, Full)

de Gaulle airport was simulated and illustrated in Figure 4.18. This flight presents a holding pattern.

The fuel and noise efficiency proof of concept was computed between FL100 and 2000ft by defining a score between 0 and 1 from a very simple statistical regression models. First, the fuel flow score is given by a linear interpolation on the estimated fuel flow with two reference values. A null fuel flow is given a score of 0, and a fuel consumption greater than 3000kg/h is given a score of 1. Second, the noise on the ground at the vertical of the aircraft is first estimated with a regression based on noise measurements from the environmental office taken into account the altitude of the aircraft, the estimated flap configuration, and the estimated engine power. Second a linear interpolation is applied between two reference values. A noise of 60 decibels is given a score of 0, and a noise greater than 110 decibels is attributed as score of 1

The altitude profile is illustrated on top of the figure. The linear color map below the graph represents the noise efficiency (green for a score of 0 and red for a score of 1). The second graph represents the ground speed profile and the estimated flaps and gear configuration. Finally, the bottom graph, illustrates the estimated fuel flow. The linear color map below the graph represent the fuel efficiency (green for a score of 0 and red for a score of 1)

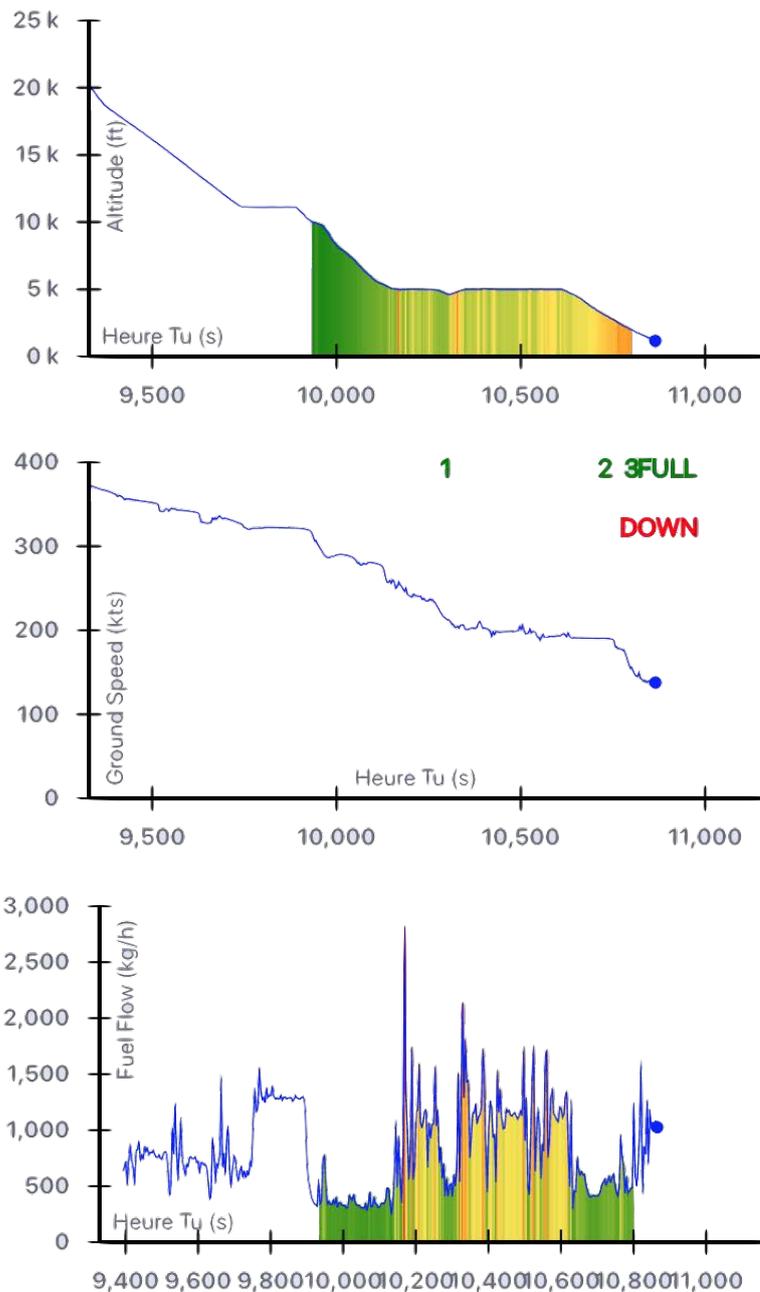


Fig. 4.18 A320 approach replay landing at Roissy Charles de Gaulle airport with extended parameters and environmental efficiency. On top of the figure is represented the altitude profile. The color below the curve is a linear mapping of the noise efficiency score. In the middle are represented both the ground speed and the estimated flaps and gear configuration. At the bottom is represented the estimated fuel flow with the color below the curve being the linear mapping of the fuel efficiency score

4.2.5 Discussion

This section investigates the uses of LSTM neural networks to predict aircraft flight deck parameters with ground-based surveillance system features. Three parameters are estimated: the fuel flow rate, and the flap and landing gear configurations.

The study shows that even without any mass information, satisfying estimations of fuel flow rate can be computed. Estimation curves present high correlations with the real curves and the error seems to be non-significant. Moreover, the trained models can be generalized to different airport approaches. Furthermore, even if they are trained with only one aircraft type, generalization to other aircraft can be made with a scaling coefficient. The residual error seems to be acceptable. This paves the way for new environmental performance assessment systems, which could provide controllers with real-time information on the environmental impact of aircraft on approach, for example. Additionally it can be added in the post-ops analysis framework to give a better understanding of atypical flights.

Finally, these models could be enhanced by integrating airspeed information available with radar mode S data. However, all aircraft are not yet equipped with the type of technology. This might also imply using one of the two models depending on the available parameters.

4.3 Conclusions

In this chapter, supervised learning techniques have been explored to improve the analysis, and generation of radar data.

First of all, Generative Adversarial Networks have been developed to generate radar trajectories and detect anomalies. The results obtained for anomaly detection corroborate those obtained in chapter 3. However, the complex convergence of the GANs due to the resolution of a min max problem makes them less flexible and non-deterministic. Two independent learning phases will probably lead to similar but different results.

Finally, the estimation of flight deck aircraft parameters was investigated with the use of LSTM networks. The models were built without the use of the mass parameter and have shown satisfying results. Although trained on a specific aircraft type, the generalization to other aircraft type has also been positive. These models can therefore bring an additional dimension to the analysis of atypical flights but also open the door to the development of new tools for the evaluation of the ATM system environmental efficiency.

This chapter was dedicated to the enhancement of radar data analysis and manipulation supervised learning techniques. The following chapter will focus on the use of Functional Principal Component Analysis in order to propose new interactions and generation process of radar data

CHAPTER 5

Toward novel interactive manipulation and generation techniques using Functional Principal Component Analysis

”Non, moi j’crois qu’il faut qu’vous arrêtez d’essayer d’dire des trucs. Ça vous fatigue, déjà, et pour les autres, vous vous rendez pas compte de c’que c’est. Moi quand vous faites ça, ça me fout une angoisse... j’pourrais vous tuer, j’crois. De chagrin, hein ! J’vous jure c’est pas bien. Il faut plus que vous parliez avec des gens.”

Arthur - Kaamelott, Livre I, Tel un chevalier

Contents

5.1	Interactive Trajectory Modification and Generation with Functional Principal Component Analysis	89
5.1.1	Related Work	89
5.1.2	Methodology and Pipeline	91
5.1.3	Use cases	101
5.1.4	Discussion	105
5.2	Interactive Shape Based Brushing Technique for Trail Sets	107
5.2.1	Related Work	109
5.2.2	Interaction Pipeline	111
5.2.3	Design space	115
5.2.4	Use cases	118
5.2.5	Discussion	124
5.3	Conclusions	126

This chapter proposes to explore novel interactive manipulation and generation techniques using Functional Principal Component Analysis. It is divided into two parts resulting from different publications. It presents the results obtained in collaboration with the ENAC colleagues working on Information Visualization (InfoVis). The first section, presents the use of Functional Principal Component Analysis to manipulate and generate data. This methodology was published in a conference [144]. It received

a best paper award and was selected for a special issue journal paper publication (in going). The second section investigates interactive brushing techniques to manipulate trajectory based data sets. This methodology uses the Functional Principal Component Analysis decomposition as a metric for shape similarity and was published in conference [145].

5.1 Interactive Trajectory Modification and Generation with Functional Principal Component Analysis

This section presents the work done in collaboration with ENAC's colleagues working on Interactive Data Visualization (InfoVis). It addresses the problem of generating and modifying temporal trajectory sets. Solving the problem relies on the use of the FPCA. It provides, on the one hand, a low-dimensional trajectory representation space that facilitates the learning of the trajectory distribution, and, on the other hand, a mean function and principal component functions that can be modified in order to infer the modified properties to the whole data set.

This section is divided into four parts. First, the related work is presented. Second, the modification pipeline is illustrated and detailed. Third, the pipeline is applied to several use cases. Finally, the obtained results are discussed. This work has also led to complementary research on novel brushing techniques presented in the next section 5.2.

Functional Principal Component Analysis allows a low-dimensional representation of a functional data set. This representation can be coupled with other techniques such as unsupervised learning to manipulate and generate data. It has already been used to support visual simplification (i.e. Edge Bundling [146]) and visual flow modeling in a previous work [147]. However, the proposed generation process that consists in estimating the distribution of the principal component analysis coefficient with a centered independent simple Gaussian distribution hypothesis is not really efficient since the PCA score observed seemed to follow multi-Gaussian distributions. In addition, the overall framework can be enhanced.

Therefore, this section investigates a novel analytic pipeline and the underlying system that would enable trajectory processing and generation using Functional Principal Component Analysis. The pipeline handles trajectory clustering, data cleaning, flow simplification, flow generation and flow transformation.

5.1.1 Related Work

There is abundant literature concerning the analysis of moving object trajectories. Even if it is a well-explored topic, it remains a popular area of research where functional

data analysis has barely been used [147]. This subsection presents the main research challenges encountered in the fields of trajectory analysis, trajectory deformation and interactive exploration of trajectories.

Functional modeling As explained in chapter 2, when manipulating objects that have a functional nature or are raised from a functional model, it is advised to preserve this model by using appropriate tools. Functional Data Analysis (FDA), [106] is a tool that aims to precisely preserve the functional nature of data by expanding it into an appropriate finite functional basis. The input object is transformed into a coefficient vector, which can then be used in a multivariate framework. This enables the use of traditional multivariate statistics but with the insurance of keeping the functional behavior of the underlying objects.

Trajectory clustering and simplification Trajectory analysis often relies on clustering algorithms. Clustering can be performed on the geographical space [148] with density maps [149], with pattern similarities [150] or with time clustering [151]. It is also possible to define distances between trajectories to enable clustering [26], or to use dimensional reduction processes [152]. Using the Functional Principal Component Analysis for trajectory clustering has barely been investigated yet, which makes this study a precursor in the area.

Trail-set simplification and manipulation Introduced by Holten [153], visual simplification helps to remove clutter in dense graphs or trail-set visualization such as road traffic or air traffic visualization. Thanks to this interactive visual aggregation, several goals can be achieved supporting *select*, *navigate*, *filter* and *arrange* tasks [154, 146]. [146] provides a review of existing edge bundling techniques and details the existing algorithms and their usages. A more recent work shows how to use FPCA tools to support visual simplification [147] but this work only operates on the visual perspective and no analytic processing is provided.

Other trajectory manipulation can be carried out, some based on the user's interaction [155], with variant layout deformation such as fisheye and bring and go techniques [156, 157] and edge plucking [158]. The fisheye deformation is illustrated in Figure 5.1 on air traffic trajectories over France and the bring and go deformation is explained in Figure 5.2. Such techniques are refined in more recent papers [159, 160]. Transmogrification also interactively transforms graphs or trail-sets with users' input [161], but again such transformation only operated on the visualization side and no further data processing can be performed. The work presented in the following sections is inspired by these techniques to perform trail-set deformation while allowing its analytic processing.

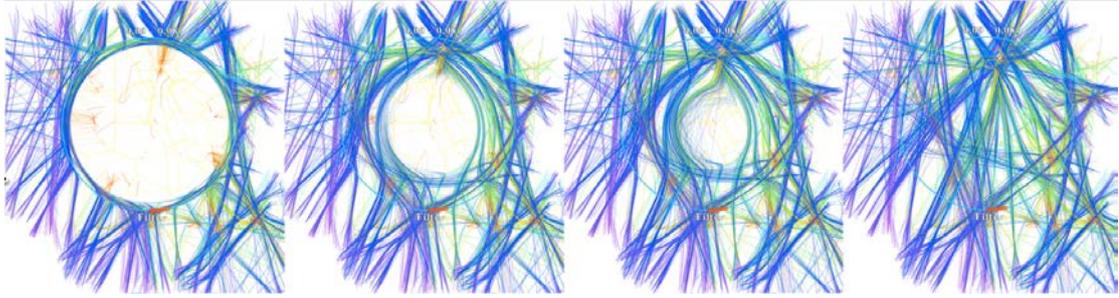


Fig. 5.1 Fisheye deformation applied to air traffic trajectories over France

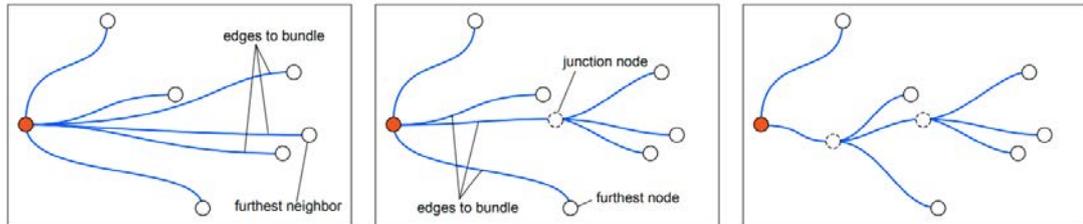


Fig. 5.2 Bring and go algorithm. Multiple links leaving a node in the same compass direction are collected into bundles by routing through a dummy junction node. The process is repeated until the number of nodes of bundles leaving all nodes or junctions in the same compass falls below a given threshold

Trajectory Exploration tools Exploring, analyzing and visualizing temporal data such as trajectories has a long history. Time series analysis [162] helps the extraction of relevant information. Frameworks [163] are available to gain a better understanding of such complex time-varying data sets thanks to aggregation techniques [164]. Specific events can be retrieved [165] thanks to data exploration paradigms [166]. A recent visualization framework has been provided to structure efficient temporal data representations [167].

Many interactive tools and systems for trail-set exploration and manipulation exist. Selection boxes help to filter objects of interest [168] [169], particle systems help to understand flow directions [170]. A specific type of movement events can be located in moving object selection [171] or in comparison with defined trajectories [172] [173]. More recently, image based techniques [174] have been applied for trajectory analysis [175]. Boolean operation can be performed to combine selections of trajectories [176] on a 2D screen or in virtual reality [177].

Overall, no previous system used the FPCA tools in a unified framework for trajectory analysis and this section provides the first of the kind.

5.1.2 Methodology and Pipeline

The proposed methodology mainly relies on learning the probability distribution of the functional principal component coefficients applied to the whole trajectory. Then, new

samples are drawn from this estimated distribution. In addition, manipulation processes are proposed to be applied to the mean and the principal component vectors in order to modify the global flow behavior. In the following paragraphs, the different components of the pipeline are detailed, and then the pipeline is presented.

Clustering Clustering is a very important initial step before applying the FPCA process. To be efficient, FPCA must operate on clusters with a representative mean curve. A two steps clustering process was derived for this study. A cutting down clustering, which aims at reducing large data-sets, is applied in a first step. For example, one can use a k-means clustering (or other simple literature algorithms) on arrival or departure trajectory locations. For the study of aircraft landing trajectories, the initial clustering is here done on the destination runways. The second step is to apply a refinement clustering based on the FPCA decomposition score. Displaying first dimension coefficients, the user is able to apply another clustering algorithm in order to group together similar trajectories. The choice of the Esperance-Maximization (EM) algorithm [178] is made here since the hypothesis of gaussian distribution is assumed, but other algorithms such as k-mean [179], or HDBSCAN [23] are also applicable. The choice of the algorithm and/or the number of clusters should be guided by the visualization of the FPCA score and by expert knowledge of the investigated data-set. In addition, the user is able to select the number of dimensions of the principal component score to use for the clustering and visualize the clustering result on trajectories to decide which clustering method produces the most representative clusters (i.e. mean curve dissimilarity).

On the other hand, the distribution of the principal component score can be used to cluster data. Indeed, the finite dimension representation enables the computation of distance. Besides, the euclidean norm of the principal component score is equal to the L2-norm in the Sobolev Space [106]. In a situation where the behavior of the group of trajectories to classify is known, this knowledge can be used to define a classification process using unsupervised learning techniques. First, trajectories are decomposed using the FPCA process. Then, the HDBSCAN [23] clustering algorithm is applied to all the trajectory principal component scores. Since the FPCA process clusters together similar data, it means that similar trajectories will be grouped together. HDBSCAN is really highly efficient in determining density-based clusters with irregular shapes, i.e. clusters that are generated from the same distribution with no assumptions on the type of distribution. In addition, the HDBSCAN algorithm gives the probability of being in a cluster. Knowing the behavior of the group to be detected, it is possible to identify to which cluster it corresponds. Finally, the user defines a probability value above which the trajectory is attributed to a cluster. This enables the user to choose the characteristics of their classification algorithms in terms of accuracy or specificity.

Data Cleaning To improve generation efficiency, the user must remove outliers that are not considered as representative members of a class of trajectory. This Data cleaning process is also an important step for clustering refinement. Since curves are already clustered, it is possible to define the probability that a sample is in a cluster. This probability is usually defined using the distance between a sample and a representative sample of the cluster. Therefore, outliers can be considered as samples whose probability of being in a cluster is lower than a threshold value or using the α -coefficient presented in chapter 3. The user should be able to select this likelihood threshold and therefore select outliers to remove. It is also possible to define an interaction to perform outlier cleaning by visualizing the principal coefficient scores and selecting the samples to remove. This interaction is similar to a brushing interaction but is carried out in the principal component score space.

In figure 5.3, the process of outlier detection is illustrated. Each ellipse represents a likelihood value level, which is user defined.

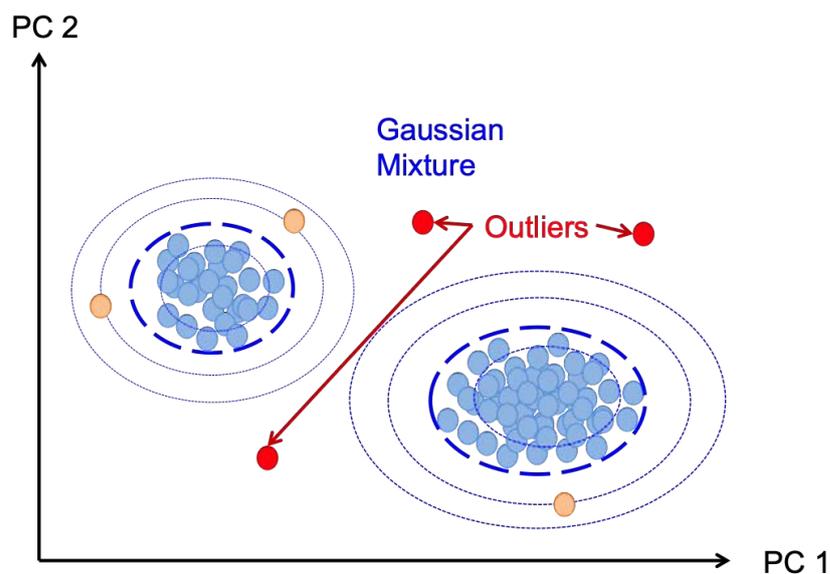


Fig. 5.3 Data cleaning or outlier removal process illustration

Curve shape modification While simple trajectory deformations can be performed with Cartesian dimensions, it become more complex with additional data dimensions such as altitude. Furthermore, deformation becomes cumbersome when it has to be applied to many trajectories. FPCA can help solely with the deformation of the cluster mean curve and its principal components to modify every trajectory of the investigated cluster. Hurter et al. [147] only modified the mean curve to perform trajectory modifications, which leads to many visual artifacts. Indeed, the mean curve modification is not sufficient, the principal components also have to be modified to correctly model the temporal behavior which was embedded in the undistorted original FPCA model.

Modifying the trajectory behavior implies being sure that the principal components, and therefore their underlying variation on the mean curve behaviour, are applied at the right time-stamp. Modifying the mean curve without insuring that the role of the principal components was not modified, resulted in most cases, with aberrant curve behaviors.

In figure 5.4 good and poor usage of curve modifications are illustrated in the context of the landing procedure interception altitude modifications. The interception altitudes were increased by 1000ft for noise sustainability issues. This use case will be further detailed in section 5.1.3.

The bottom figure 5.4, shows the result of the solely mean curve modification. In this case, only the mean curve has been modified to give an interception altitude 1000ft higher. However, the principal component functions have not been modified. As a result, the principal component functions are no longer temporally aligned with the mean curve. This implies that modifications that would normally occur at 5000ft start at 4000ft inducing visual artifacts around the level-off flight (red circle at the bottom of figure 5.4). The top figure 5.4 shows the correct results where both the mean and the principal components functions were modified.

The mean curve modification and its principal component modifications are not an easy task. First, a curve registration is needed to align curve landmarks. Then, the key idea is to apply the same temporal modifications to both, the mean and the principal components. This ensures that the variation induced by the principal component functions is correctly located. The next paragraphs propose modification operators.

Curve Interpolation : The interpolation operator of two curves γ_A and γ_B between t_1 and t_2 is defined for any $t \in [0, 1]$ such that it ensures the interpolation operator derivative continuity as:

$$\text{ITP}_{\gamma_A, \gamma_B}(t_1, t_2)(t) = \begin{cases} \gamma_A(t), & \text{if } t < t_1 \\ c(t) \cdot \gamma_A(t) + (1 - c(t)) \cdot \gamma_B(t), & \text{if } t_1 \leq t < t_2 \\ \gamma_B(t) & \text{else} \end{cases} \quad (5.1)$$

where c the interpolation coefficient is:

$$c(t) = \frac{1 - \cos\left(\frac{t_2-t}{t_2-t_1}\pi\right)}{2} \quad (5.2)$$

A simple illustration is given in Figure 5.5.

Curve translation: The translation operator of a curve γ is defined as v the translation vector for any $t \in [0, 1]$ as :

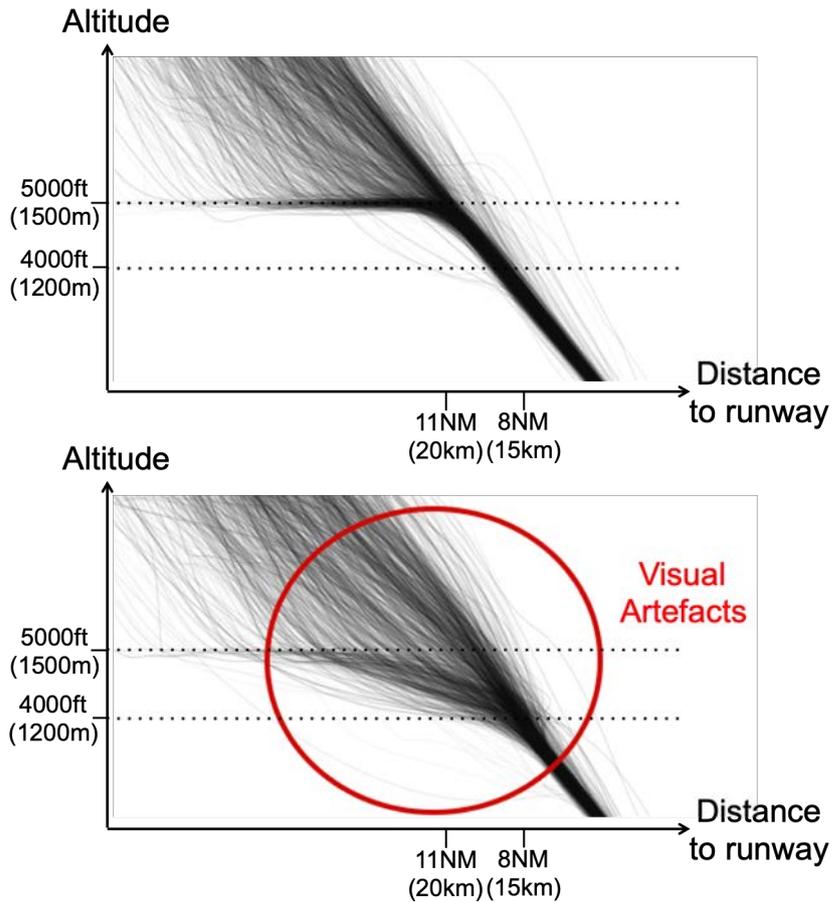


Fig. 5.4 Good and bad usage of curve modifications for aircraft approach trajectory altitude profile. At the top, both the mean and the principal components were modified. At the bottom, only the mean was modified. The behavior observed while modifying only the mean curve presents artifacts. Level-off flight was expected, but the trajectories present descent phases. These behaviors are not nominal and underline that the process was not executed properly.

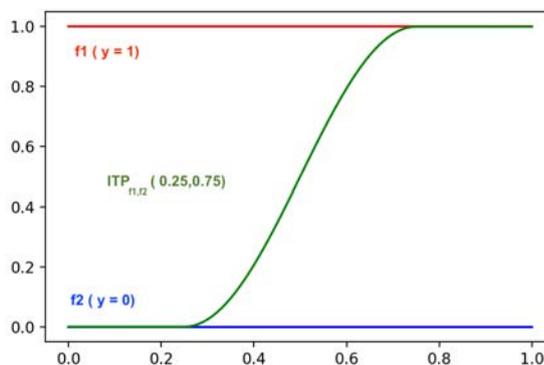


Fig. 5.5 Interpolation operator between the curves of equation $y(t)=0$ and $y(t)=1$, for $t_1=0.25$ and $t_2=0.75$

$$\text{Translation}_{\gamma}(v)(t) = \gamma(t) + v \quad (5.3)$$

This operator must be applied only to the mean curve since it does not affect landmark time position. A simple illustration is given in Figure 5.6

Curve 2D rotation: The 2D rotation of a curve $\gamma = (\gamma_x, \gamma_y)$ at time t_1 with angle θ for any $t \in [0, 1]$ is defined as:

$$\text{Rotation}_\gamma(t_1, \theta) = \begin{cases} \gamma(t), & \text{if } t < t_1 \\ \gamma(t_1) + \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} (\gamma(t) - \gamma(t_1)) & \text{else} \end{cases} \quad (5.4)$$

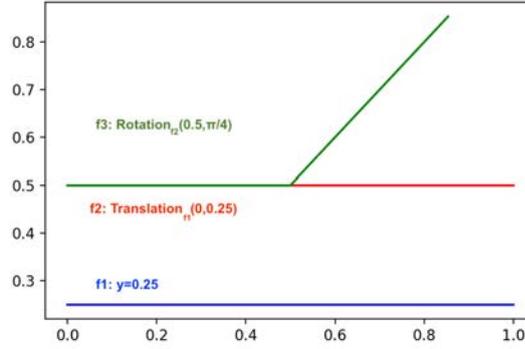


Fig. 5.6 Translation and Rotation operators illustration on 2D curves. The curve of coordinates $(t, y(t)=0.25)$ is first translated with vector $(0, 0.25)$, finally the rotation operator a is applied at $t=0.5$ and with angle $\theta = \frac{\pi}{4}$

Temporal compression and dilatation: The temporal compression/dilatation operator of a curve γ between t_1 and t_2 with the compression coefficient $\alpha \in \mathbb{R}^+$ is defined for any $t \in [0, 1]$ as :

$$\text{TCD}_\gamma(t_1, t_2, \alpha)(t) = \begin{cases} \gamma(t), & \text{if } t < t_1 \\ \gamma(\alpha \cdot t + (1 - \alpha) \cdot t_1), & \text{if } t_1 \leq t < t_{CD} \\ \gamma(t - (t_{CD} - t_2)), & \text{if } t \geq t_{CD} \end{cases} \quad (5.5)$$

with $t_{CD} = \frac{t_2 - (1 - \alpha) \cdot t_1}{\alpha}$. A simple illustration is given in Figure 5.7

Curve temporal cut or extension: The cutting or extending operator of a curve γ at t_1 with width $\delta_t \in [-t_1, 1 - t_1]$, for any $t \in [0, 1 - \delta_t]$ is defined as :

$$\text{CE}_\gamma(t_1, \delta_t)(t) = \begin{cases} \gamma(t), & \text{if } t < t_1 \\ \gamma(t + \delta_t) + \gamma(t_1) - \gamma(t_1 + \delta_t), & \text{if } t_1 \leq t < 1 - \delta_t \end{cases} \quad (5.6)$$

A simple illustration is given in Figure 5.7. In addition, this operator modifies the definition interval of the curve. A potential correction consists in applying the TCD (Eqn. 5.5) operator between 0 and $1 - \delta_t$ with compression coefficient $\alpha = 1 - \delta_t$.

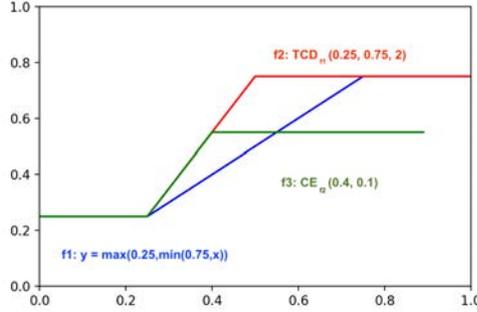


Fig. 5.7 Temporal compression and cut operators illustration. The curve of equation $y=\max(0.25,\min(0.75,x))$ is compressed between $t1=0.25$ and $t2=0.75$ with $\alpha = 2$. Second the curve is cut at $t1=0.4$ by $\delta t = 0.1$

Curve generation In [147], Hurter et al. generated curves with a random selection of principal coefficient scores with a centered independent simple Gaussian distribution hypothesis. Usually, coefficients are not simply Gaussian. Consequently, curves generated with this model do not present realistic behavior. Two alternatives are proposed.

Firstly, a neighborhood generation is proposed to ensure realistic aspect of the generated trajectories. The curve generation process needs to take into account a large number of principal coefficients (typically more than 60% of the total principal components). The process is as follows: first, each principal coefficient dimension variance is computed using a Gaussian centered model. Then, a curve is randomly selected and its principal component scores are kept. Finally, a new score is finally drawn in the ellipsoid $\mathcal{E}(b_k, \alpha \cdot \mu)$ centered on the selected sample and with semi-major axes the vector of variances. The range of the neighborhood is defined with the variance among each dimension. In addition, the user is able to tune an α coefficient between 0 and 1 that is multiplied to the range of the neighborhood. This coefficient enables the user to modify the similarity between the original and the generated trajectory. The following Algorithm 1 illustrates an implementation of the neighborhood algorithm. A simple example with two dimension is illustrated in Figure 5.8.

Secondly, an alternative to the neighborhood generation model consists in applying a multivariate Gaussian Mixture model, i.e. an EM algorithm [178], on the principal component scores that concentrate more than a given percentage of the explained variance. This process does not assume the independence of the principal component scores and enables a richer representation with a Gaussian Mixture instead of a simple Gaussian Distribution. With this generation, it is usually more difficult or even impossible

Data: $b_{ji} \in \mathbb{R}^d$, $1 \leq j \leq N$, and $1 \leq i \leq d$, matrix of the N principal scores,
with d the dimension of the principal component scores

Var is the variance operator

$Random(a, b)$ gives a random int between a and b

Result: Generate a new principal component score

```

begin
  for  $i = 1, \dots, d$  do
    |  $\mu_i \leftarrow Var\{b_{1i}, \dots, b_{Ni}\}$ 
  end
   $\mu \leftarrow \{\mu_1, \dots, \mu_N\}$ ;
   $k \leftarrow Random(1, N)$ ;
  return  $s \in \mathcal{E}(b_k, \alpha \cdot \mu)$ 

```

end

Algorithm 1: Neighborhood generation, α coefficient between 0 and 1 defines the neighbor threshold.

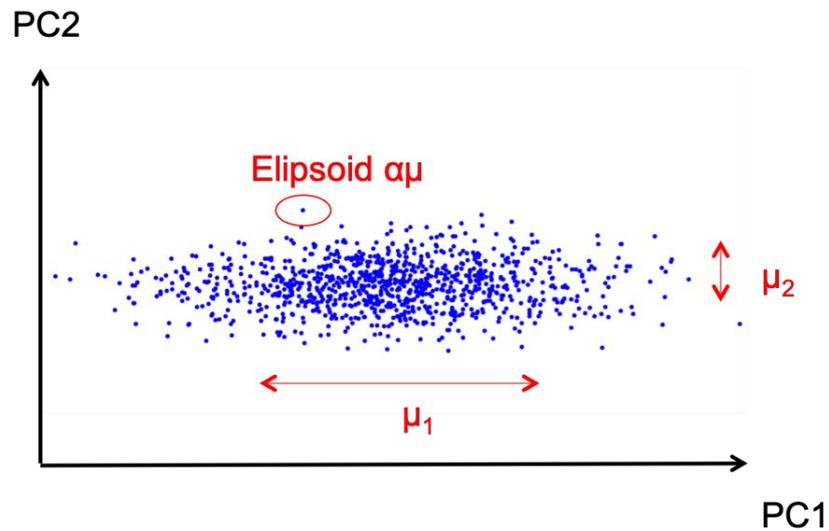


Fig. 5.8 Neighborhood generation illustration. The variance is computed along each dimension. A sample is generated by drawing a vector in a variance based ellipsoid neighborhood of an existing sample

to properly estimate the distribution for a large number of components. This is the well known problem referred to as the curse of dimensionality [180]. In high dimensional space, the volume of space increases rapidly and samples are usually isolated. The choice was made in this study to estimate only the distribution of the first components that explain most of the variance with the dependence hypothesis. The last components, which mostly correspond to the noise, are then assumed to be independent.

Figure 5.9 illustrates on a 2D representation the three models for curve generation. The red circle corresponds to the Simple Gaussian model, the blue ones to the Gaussian Mixture Model, and the green ones to the neighborhood model. Applying a methodology consists approximately to draw a sample in one of the corresponding circles.

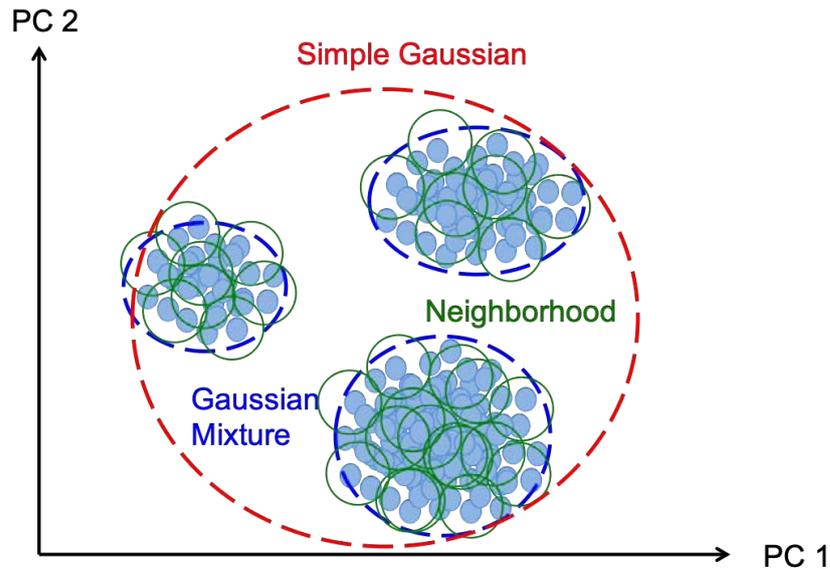


Fig. 5.9 Illustration of the different generation processes for the principal component score distribution. In red, a Simple Gaussian model is illustrated. In blue, the Mixture Gaussian model is represented. Finally, in green, the neighbourhood model is shown. Applying a methodology consists approximately to draw a sample in one of the corresponding circles

Visual simplification (Edge bundling) The edge bundling is performed thanks to the Inverse FPCA process. This reconstruction method uses all principal component functions. However, one can choose to reconstruct the trajectories with only a certain number of principal components. In [147], Hurter et Al. suggested removing the component by a percentage of the explained variance.

Usually, 99% of the variance is explained by a small number of components, less than 10% of the total number of principal components. A bundling coefficient between 0 and 1 was defined as representing the percentage of explained variance kept, a value of 0 means that only the mean curve is kept, while a value of 1 means that the original trajectory is fully reconstructed. For computational purposes, the transition between 0.99 and 1 is achieved by adding 90% of the principal components. In terms of total data set variance this is negligible. However, this transition may induce very large modifications to the behavior of certain trajectories.

In order to avoid this type of visual artifacts, the current study favors a piece-wise linear mapping for the bundling coefficient. For example, from 0 to 0.90, add the number of components equivalent to 0 to 99% of the variance. For the last 0.9 to 1 add the remaining components linearly. Figure 5.10 shows the bundling result with a set of landing trajectory and increasing explained variance (0%, 25%, 50%, 75%, 100%).

Pipeline Figure 5.11 represents the proposed pipeline with a trajectory data-set as input data. The first step performs an initial clustering to reduce the data-set into clusters

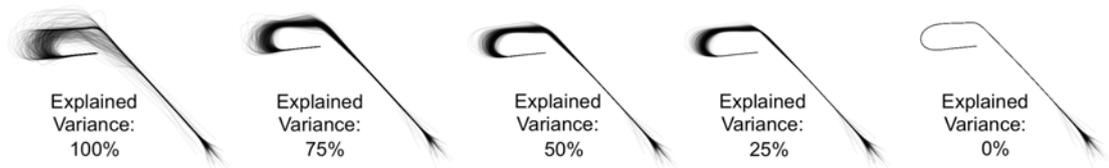


Fig. 5.10 Illustration of the Edge Bundling (i.e. trail visual simplification) process for landing trajectories at Charles de Gaulle Airport

with similar trajectories. This initial clustering is data-set dependant. For instance, with aircraft trajectories, it can be performed on departure or arrival airport or runway QFU. Then, for each cluster, an optional landmark registration can be applied to produce a more accurate functional decomposition. The FPCA process can then be applied to each cluster to compute the mean curve, the principal component functions and the principal component scores.

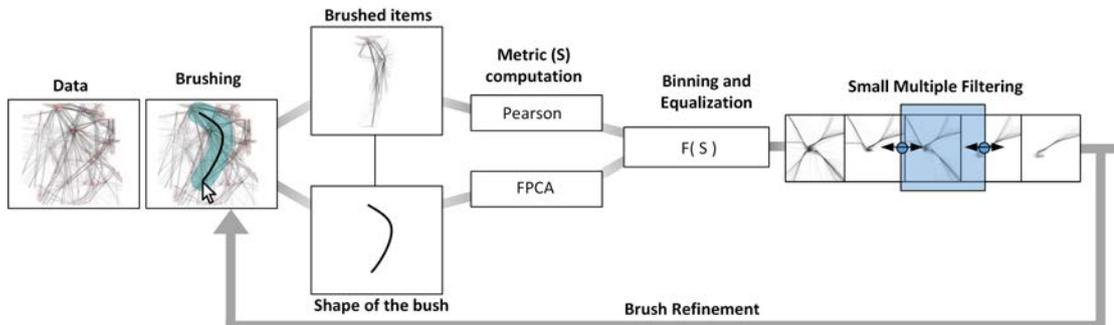


Fig. 5.11 Modification and generation pipeline. On the left, the input curve data-set passes through an initial clustering step. After this, trajectory registration based on landmarks is applied. Then, the FPCA process is computed for the first time for each cluster. This FPCA process gives two pieces/elements of information: the principal component scores (top figure) and the mean curve with the principal component functions (bottom figure). The principal component scores are used in two situations: First for clustering refinement and data cleaning, second for the generation process. Generation consists in estimating the principal component score distribution and in generating new samples following the estimated distribution. During the modification operations, the mean and the principal component functions are processed. It consists in applying modification operators (rotation, translation, dilatation) in order to obtain the desired distortion. Finally, the Inverse FPCA process enables the trajectory to be reconstructed with the new distribution (i.e. increase or decrease in trajectory number) and behavior. The user chooses if they wish to add a visual simplification, otherwise, the trajectories are fully reconstructed.

Then, a clustering refinement and a data cleaning step can be computed based on the principal component scores as previously explained. User input is needed at this clustering step: with the suggested EM algorithm, the user has to define the number of clusters and the number of principal components to use. User input is also needed at this data cleaning step with the probability value (i.e. threshold) below which samples

are considered as outliers.

The following step shows different possible trajectory processing. The Inverse FPCA processes data with the amount of requested variance (section 5.1.2) produces cluster simplification (i.e. edge bundling). The Inverse FPCA produces size varying trajectory with respect to their shape and statistical properties. Three different types of trajectory generation are available (5.1.2). The neighborhood process produces trajectories close to the original one (tuning the variability around it), the Gaussian mixture allows more variability. Finally, trajectory deformations can be applied to distort the final result. Thanks to the modifications operators (5.1.2) applied to the mean and to the principal component functions, the shape of the trajectory can be adjusted. These final data processing techniques (curves generation, deformation and simplification) can be combined to adjust end user final results.

5.1.3 Use cases

This subsection is divided into three parts. Each part illustrates a specific feature of the pipeline (5.11). The first part illustrates the clustering process, the second one the trajectory modification operator through concrete examples, and the last one demonstrates the generation task.

Landing sequence Classification In this paragraph is applied the unsupervised classification process defined in section 5.1.2 to the identification of the landing procedure at Bordeaux Merignac airport (one of the major airports in France). When landing on runway 05, aircraft follow four kinds of trajectory. The conventional approach, the VOR approach, and the two so called GNSS paths. GNSS approaches are very characteristic since they follow a path from defined way-points (geographical points on a map). It also means that this type of approach will be very similar in the FPCA space. The data set is composed of 2597 radar landing trajectories, during a three month period in 2018 (summer time).

Figure 5.12 shows the result of the clustering algorithm where classes of landing sequence are separated within four clusters. The detection seems to be relevant with the expected behaviours. Figure 5.12 bottom figures show the 4 main principal components with the identified clusters.

Interception Altitude modification After The Grenelle de l'environnement (annual meeting to discuss sustainability issues), landing procedures at Charles de Gaulle airports were raised by 1000ft to reduce noise emission. In this regard, this study case was led in collaboration with the Environmental Office of the French Air Navigation Service Provider. In this section, are reported the simulation results where the traffic before the rise is modified with the Grenelle 1000ft rise. The resulting noise emission

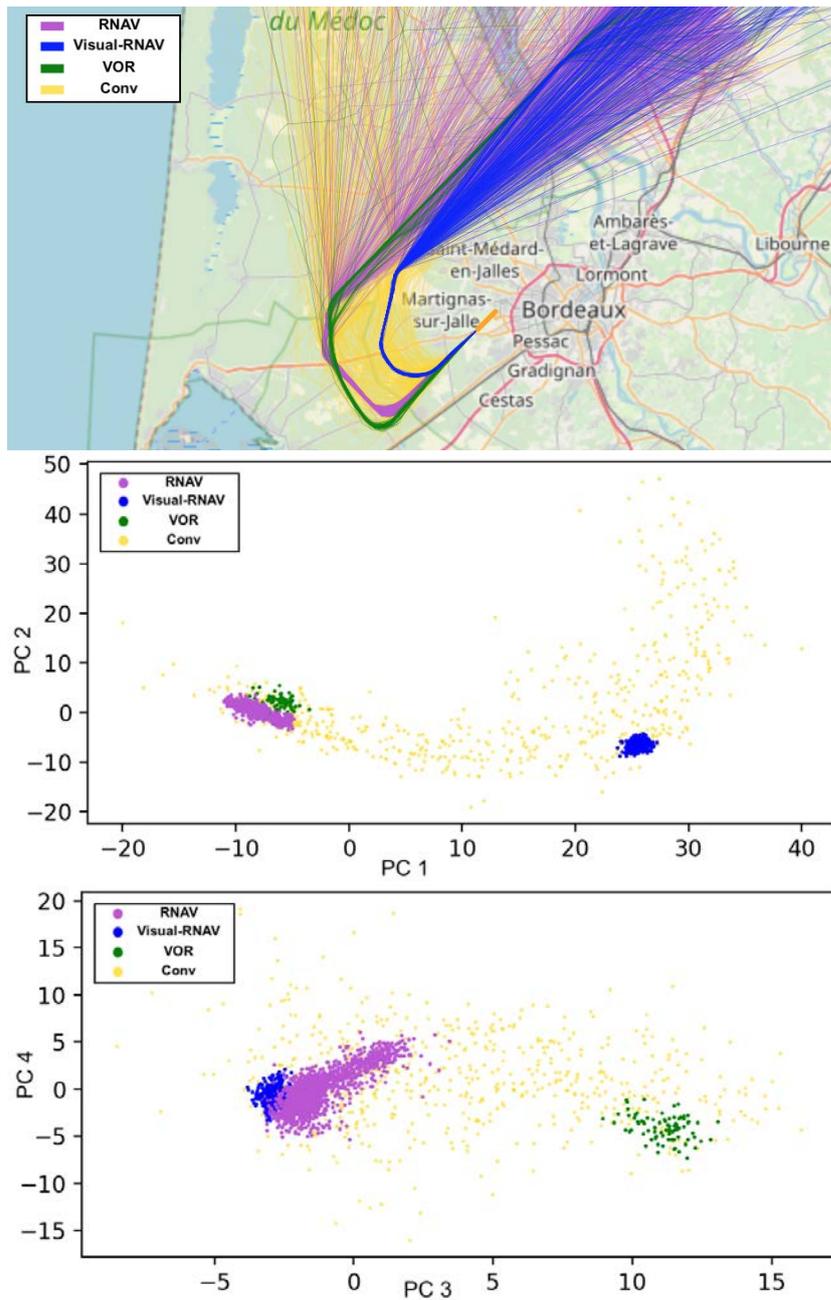


Fig. 5.12 Clustering task on landing trajectories at Bordeaux airport. On the top image is illustrated the result of the classification of the Bordeaux airport approaches from the 2017 data records. The two bottom graphs represent the first four principal component score distributions.

was computed and compared with the actual trajectories after the rise. This comparison provides a good assessment of the accuracy of the trajectory generation and modification pipeline.

The process is the following. First, trajectories were registered considering they are function of the remaining distance to the runway threshold and keeping only the last 75NM of each trajectory. Second, the FPCA decomposition and a clustering refinement

of the trajectories with the EM algorithm [178] were applied. Then, for each cluster, trajectories were modified to follow the 1000ft pull-up. The modification consists in applying the extension operator between t_1 , the distance corresponding to 1000ft on a three degree glide path, and t_2 , the distance corresponding to 2000ft on a three degree glide path, and with $\delta_t = t_2 - t_1$.

Figure 5.13 top illustrates the noise level for the real traffic (after the altitude rise), and at the bottom, the noise generated from the pipeline with the modified traffic. The noise indicator is the NA62 indicator which is computed over one day of traffic. This indicator is mainly used by the environmental office. It corresponds to the number of aircraft emitting noise above 62dB during the period. The area for 5 to 20 events above this threshold is represented here.

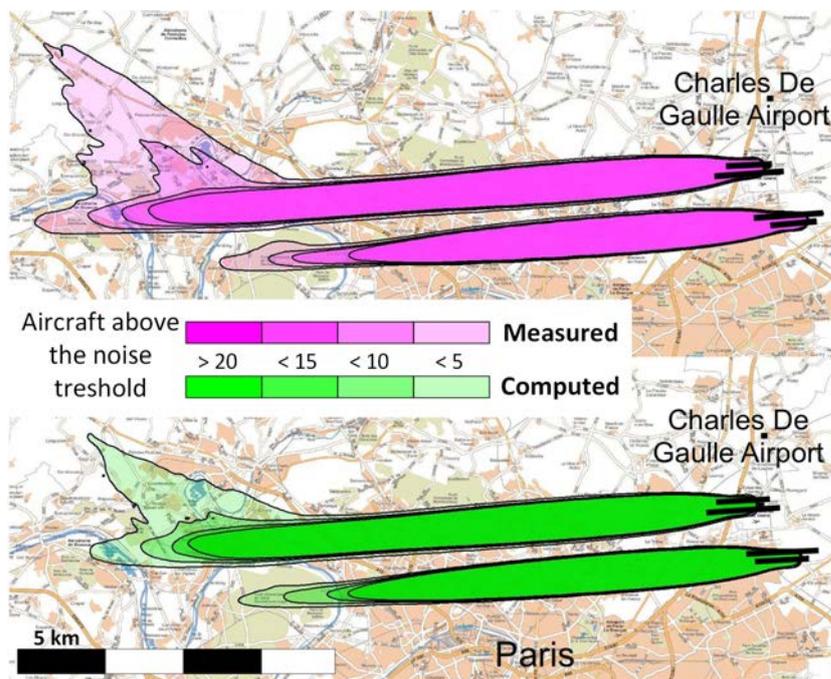


Fig. 5.13 Noise NA62 indicator map of the real aircraft traffic and simulated traffic obtained with the modification pipeline. At the bottom, the same indicator map for the simulated traffic obtained with the pipeline. The modification consists in raising by 1000ft the level-off flight of landing aircraft before landing. This modification was applied following the Grenelle de l'environnement for noise reduction purposes.

The result shows that the noise obtained with the simulation is close to the actual recorded noise, in particular for the parts aligned with the runway center line. However, few differences are observed. The simulation covers 92% of the area because the real noise map is slightly more extended on the left side. This is due to the fact that in the real context, approaches tend to have a longer level-off flight before starting the final descent. Nevertheless, this shows that the pipeline can produce valuable simulated trajectories and can be used for realistic flow simulation.

New departure flow investigation Another use case is the building a new aircraft departure procedure. Indeed, the pipeline enable a simulation of departure flow on this new procedure and its noise impact assessment. To test such a possibility, a novel departure air flow at Nantes-Atlantic Airport (one of the major airports in France) is considered using an existing flow at Bordeaux-Merignac Airport as a reference model. Such flow duplication is not straightforward since the original air flow (i.e. Bordeaux) has to be modified to follow mandatory way points at the destination airport (i.e. Nantes).

As a result, figure 5.14 represents at the top, the original flow of trajectories at Bordeaux and at the bottom, the generated and distorted air flow with the pipeline after its modification to fit Nantes airport landing procedure. A rotation and translation were applied to align the runway approach with the runway in Nantes. The new procedure follows different aeronautical way-points and is illustrated in red in figure 5.14.

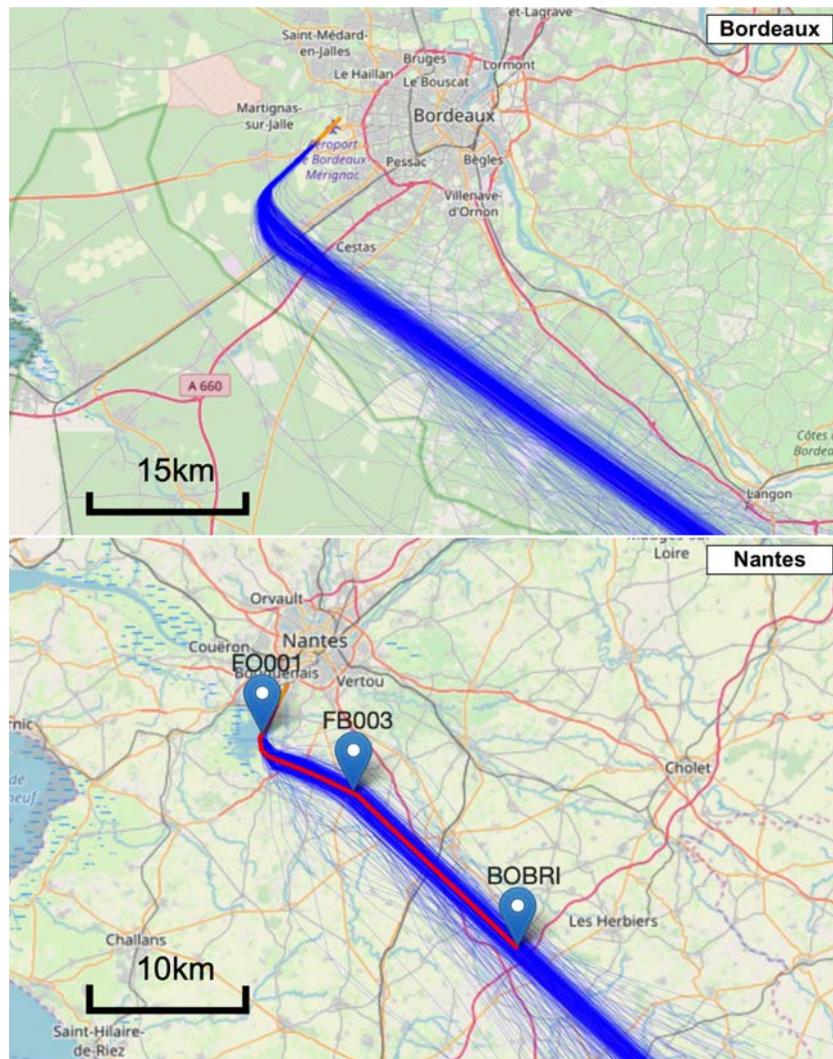


Fig. 5.14 Departure procedure deformation at Nantes airport using Bordeaux Departure data set. On top, the original trajectories at Bordeaux Airport are represented. At the bottom, the modified trajectories at Nantes Airport. In addition, the expected procedure following different aeronautical way points is shown in red.

Traffic Growth The simulation of traffic growth has already been studied by Hurter et al. [147]. This pipeline also supports such features but was enhanced thanks to a better control of the generated trajectories. As mentioned in subsection 5.1.2, a more realistic distribution estimation is used, using the neighborhood generation and the mixture Gaussian model. Being able to generate new trajectories, opens the possibility of predicting and simulating future traffic growth.

5.1.4 Discussion

This section discusses the methodology outcomes (clustering, trajectory distortion, trajectory generation and trajectory simplification) and provides a summary of FPCA good practice.

The objective of clustering refinement is to compute cluster with consistent mean curve regarding the Gaussian distributions. As previously explained, the FPCA process is efficient when groups of curves have similar shapes to correctly capture their variability around a mean curve. In addition to the clustering refinement, it is also important to remove outliers which may impact the mean curve and potentially induce weak results in the next FPCA processing steps.

The system, derived from the pipeline and illustrated in Appendix F, contains a set of specific tools for trajectory modifications. Modifying the mean curve without keeping the principal components aligned will generate artifacts. Users need to visually assess the modification results and fine tune the regenerated curves. Currently, the system has predefined modification presets, but could be enhanced by giving to the user the ability to choose the modifications and to interact on the mean and on the principal components to directly see the effect on the regenerated curves.

For the generation process, the users have to select which kind of generation process they wish to apply and define how many curves to generate. The size of the original data-set can be adjusted while keeping a consistent distribution or extending the data-set in number in order to simulate traffic growth or decrease. The choices in the generation process and in the parameters are guided by the visualizations of the principal component score and the desired proximity in shape to the original trajectories. Besides, the user can also adapt the generation process with the visualization of the reconstructed trajectories and ensure that the generated curves have the shape expected.

As previously explained, the pipeline supports visual simplification (i.e. edge bundling). While previous work already explained that FPCA processing can produce such visual transformation, this process was extended with a quantitative measurement. As such, a smooth transition between the original cluster and its simplified version can only operate when taking into account the amount of score variance around the mean curve. Initial works only considered adding the principal components step by step which leads

to notable visual artifacts. Even if the process still suffers from other artifacts (mainly due to the last principal component which captures specific variability such as straight lines or common sharp turns), its visual simplification operates in a smoother way.

Regarding the generation process, it is interesting to highlight that results from FPCA generation by keeping a reduced amount of the variance are similar to those obtained with the GAN model (Section 4.1). For example, level-off flights in altitude profiles are not captured either by GAN or by a truncated FPCA generation.

As a summary, recommendations for efficient usage of FPCA tools for trajectory analysis are provided here:

- trajectory registration: this initial step is important to efficiently capture the variance around the mean curve of the considered clusters,
- initial clustering: an initial clustering is mandatory to have a meaningful mean trajectory,
- data cleaning: outlier removal can help to capture the variance around the mean curve in the most effective manner,
- trajectory simplifications: smooth visual simplification only operates with the incremental removal of the principal components and with a continuous decrease of the global trajectory variance,
- trajectory deformation: trajectory deformation only operates with the mean curve deformation associated with the principal component function modifications,
- trajectory generation: many possible methods exist to increase or decrease the number of trajectories. We proposed three methods taking into account the global, local and neighbor variance. The user needs to choose the appropriate method with regard to the task to be achieved (number of data dimensions to generate, closeness to existing trajectories).

In this section, a new approach to analyze trajectories from a functional decomposition perspective for the underlying data-set was presented. While previous works applied such a technique to visualize and simplify data-sets, it has been proposed to use this statistical tool to quantitatively process them. Thus, a functionally based pipeline was developed to support the following trajectory processing: clustering, visual simplification, trajectory deformation and trajectory generation.

Thanks to the pipeline, trajectories can be clustered taking into account trajectory curvature and their variability around the mean curve. This provides another clustering tool which mainly considers trajectory shapes as a grouping parameter. Visual

simplification, i.e. edge bundling, can be computed, and the pipeline refines previous works with better control of the visual aggregation with respect to the trajectory variance around the mean cluster curve.

Through one concrete example, an aircraft path deformation and the corresponding noise computation, it underlines that the pipeline can produce reliable solutions for trajectory simulation. The results were validated by air traffic practitioners and this work opens new application perspectives. Rather than processing every trajectory to deform it and make it compliant with new air traffic flow constraint, the pipeline enables the deformation of a single mean curve to produce an equivalent result. Furthermore, this pipeline is more flexible, since the user can also increase or decrease the number of trajectories while keeping a coherent distribution around the mean curve. Finally, a prototype interactive system was developed (See appendix F), where end users can apply the pipeline to trajectory data-sets and process them for analytic purposes.

While quantitative and accurate results was shown with this pipeline, many improvements can be considered. Firstly, FPCA tools need some fine tuning and the underlying parameters require some prior knowledge in statistical tool manipulations. An additional abstraction layer could be considered to ease user understanding of these complex parameters such as cluster scores and principal components. Secondly, the pipeline provides trajectory deformations applied to the mean curve and to the principal component of the considered cluster. Simple transformation like rotation, stretching and bending are currently provided, some additional work is needed to make this transformation applicable to any kind of trajectory. The methodology was applied to landing trajectories where their end parts are mainly straight lines. If this is not the case, deformation can have a significant impact on the end part of the trajectories and insert artifacts which are too large. Finally, this system and its pipeline show great potential to ease the manipulation of large sets of trajectories by practitioners. While we developed the system with the end users, much additional work remains to make it compliant with industrial constraints which go beyond the scope of this paper.

In this section, a pipeline for generating and modifying functional data set was presented. The underlying FPCA decomposition space enables several uses. In particular, it can be used to provide novel interactive manipulations of the data such as new Brushing techniques. This is the aim of the following section.

5.2 Interactive Shape Based Brushing Technique for Trail Sets

This section presents the second contribution done in collaboration with ENAC's colleagues working on Interactive Data Visualization (InfoVis). The contribution of this thesis consists on the development of the Functional Principal Component analysis

framework to give a similarity metric between trajectories to enhance a brushing interaction.

Brushing techniques [181], which are part of the standard InfoVis pipeline for data visualization and exploration [182], already have a long history. They are now standard interaction techniques in visualization systems [183] and toolkits [184] [185]. Such techniques help to visually select items of interest with interactive paradigms (i.e. lasso, boxes, brush) in a view. A simple example of a brushing interaction is presented in Figure 5.15. When the user visually detects a relevant pattern (i.e. a specific curve or a trend), the brushing technique can then be applied to select it. While this selection can be seamlessly performed, the user may still face issues when the view becomes cluttered with many tangled items. In such dense visualization, existing brushing techniques also select items in the vicinity of the target and thus capture part of the clutter. To address such an issue, the user can adjust the brushing parameters by changing the brush size or the selection box locations. However, this may take time and requires many iterations or trials.

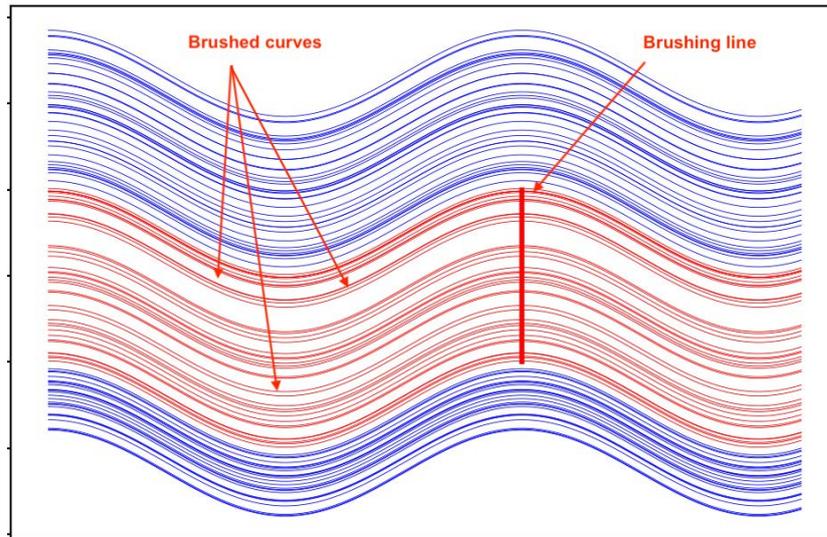


Fig. 5.15 Brushing interaction on simple curves. The curves brushed by the red line are displayed in red. The remaining curves are colored in blue

This section proposes a novel brushing technique that filters trajectories taking into account the shape of the brush in addition to the brush area. This dual input is provided at the same time and opens novel opportunities for brushing techniques. The cornerstone of such a technique relies on a shape comparison algorithm. This algorithm must provide a numerical similarity measurement which is ordered (low value for unrelated shapes, and high value for correlated shapes), continuous (no steps in the computed metric) and with a semantic so that the user can partially understand the logic behind this similarity measurement. Thus, to build such a dual filtering technique, the follow-

ing design requirements (DR) must be fulfilled:

- DR1: the technique enables users to select occluded trajectories in dense or cluttered view.
- DR2: the shape comparison metric is flexible with continuous, ordered and meaningful values.
- DR3: the technique enables incremental selection refinement.
- DR4: the technique is interactive.

Taking into account the identified requirements (DR1–DR4), this section presents a novel shape-based brushing tool. Such a combination of brushing and shape comparison technique has not yet been explored in trajectory analysis and this research fills this gap. The section is divided as follow. First, previous works in the domain of brushing and shape comparison are provided. Second, the brushing pipeline is detailed with explanations of the comparison metric data processing. Next, the use of such a technique through different use-cases is demonstrated. The brushing technique is discussed in terms of usefulness, specific application and possible limitations. Finally, the section concludes with a summary of the contributions and provides future research directions.

5.2.1 Related Work

There are various domain-specific techniques targeting trail exploration and analysis. In this sub-section, three major components of selection techniques for trail-set exploration and analysis relevant to this research are explored: brushing, query-by-content, and similarity measurement.

Brushing in Trajectory Visualization Trail-set exploration relies on pattern discovery [186] where relevant trails need to be selected for further analysis. Brushing is a selection technique for information visualization, where the user interactively highlights a subset of the data by defining an area of interest. This technique has been shown to be a powerful and generic interaction technique for information retrieval [181]. The selection can be further refined using interactive filtering techniques [187]. The approach presented here is based on dynamic queries [188] and direct manipulation [189].

Systems designed specifically for spatio-temporal visualization and in particular in trajectory visualizations are very complex because of their 3D and time varying nature. Due to this, several systems and frameworks have been especially designed to visualize them [176, 177, 170, 171, 164, 167]. Most of these systems include selection techniques based on brushing, and some of them enable further query refinement through boolean operations [176, 177].

These techniques do not take into account the shape of the trails, so selecting a specific one with a particular shape requires many manipulations and iterations to fine-tune the selection.

Query-by-Content While this research attempts to suggest a shape-based brushing technique for trail sets, researchers have explored shape-based selection techniques in different contexts, both using arbitrary shapes and sketch-based queries.

Sketch-based querying presents several advantages over traditional selection [190]. It has been used for volumetric data sets, [191] and neural pathway selection [192]. This last work is the closest to the current study. However, the authors presented a domain-specific application and they based their algorithms on the Euclidean distance. This is not a robust metric for similarity detection since it is hard to provide a value indicating a high similarity and varies strongly according to the domain and the data considered. In addition, this metric does not support direction and orientation matching neither the combination of brushing with filtering.

In addition, user-sketched pattern matching plays an important role in searching and localizing time series patterns of interest [193, 194]. For example, Holz and Feiner [194] defined a relaxed selection technique in which the users draw a query to select the relevant part of a displayed time-series. Correl et al. [190] propose a sketch-based query system to retrieve time series using dynamic time wrapping, mean square error or the Hough transform [195]. They present all matches individually in a small multiple, arranged according to the similarity measurement. These techniques, as the one proposed here, also take advantage of sketches to manipulate data. However, they are designed for querying rather than selecting, and specifically for 2D data. Other approaches use boxes and spheres to specify the regions of interest [196, 197, 198], and the desired trails are obtained if they intersect these regions of interest. However, many parameters must be changed by the analysts in order to achieve a simple single selection. The regions of interest must be re-scaled appropriately, and then re-positioned back and forth multiple times for each operation. Additionally, many selection box modifications are required to refine the selection and thus hinder and alter the selection efficiency [196].

Similarity measures Given a set of trajectories, the objective is to retrieve the most similar subset of trajectories with a user-sketched query. Common approaches include selecting the K-nearest-neighbors (KNN) based on the Euclidean distance (ED) or elastic matching metrics (e.g, Dynamic Time Warping - DTWs). Affinity cues have also been used to group objects. For example, objects of identical color are given a high similarity coefficient for the color affinity [199].

The Euclidean distance is the most simple to calculate, but, unlike mathematical *similarity* measurements [200] which are usually bound between 0 and 1 or -1 and 1,

ED is unbounded and task-specific. A number of works have suggested transforming the raw data into a lower-dimensional representation (e.g., SAX [201, 202], PAA [203, 204]). However, they require adjusting many abstract parameters which are dataset-dependant and thus reduce their flexibility. In [199], global and local proximity of 2D sketches are presented. The second measure is used for similarity detection where an object is contained within another one, and is not relevant to this work. While the first measure refers to the distance between two objects (mostly circles and lines), there is no guarantee that the approach could be generalized to large datasets such as eye tracking, GPS or aircraft trajectories. In contrast, Dynamic Time Warping has been considered as the best measurement [205] in various applications[206] to select shapes by matching their representations. It has been used in gestures recognition [207], eye movements [208] and shapes [209]. An overview of existing metrics is available in [206].

The k-Nearest Neighbors (*KNN*) approach has also long been studied for trail similarity detection [210, 211]. However, using this metric, two trails may provide a good accurate connection (i.e, a small difference measure as above) even if they have very different shapes. Other measurements to calculate trajectory segment similarity are the Minimum Bounding Rectangles (MBR) [212] or Fréchet Distance [213] which leverage the perpendicular distance, the parallel distance and the angular distance in order to compute the distance between two trajectories.

In order to address the aforementioned issues, it is proposed to investigate two different approaches. The first approach (pearson) is based on directly calculating the correlations on x -axis and y -axis independently between the shape of the brush and the trails (section 5.2.2). The second approach FPCA in section 5.2.2 is based on the geometrical information of the trails, i.e, the trails are transformed into a new space (using eigenvectors of the co-variance matrix) which is more suitable for similarity detection. This research's approach leverages the potential of these two metrics to foster efficient shape-based brushing for large cluttered datasets. As such, it allows targeting detailed motif discovery performed interactively.

5.2.2 Interaction Pipeline

This subsection presents the interactive pipeline (Figure 5.16) which fulfills the identified design requirements (DR1–DR4). As for any interactive system, user input plays the main role and will operate at every stage of the data processing. First, the input data (i.e. trail set) is given. Next, the user inputs a brush where the pipeline extracts the brushed items, the brush area and its shape. Then, two comparison metrics are computed between every brushed item and the shape of the brush (similarity measurement). Followed by the binning process and its filtering, the resulting data is presented to the user. The user can then refine the brush area and choose another comparison metric

until the desired items are selected.

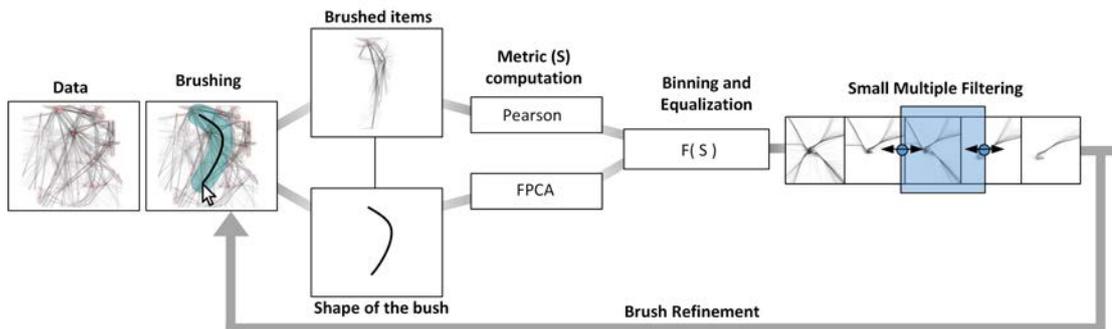


Fig. 5.16 Interaction pipeline to filtered items thanks to the user brush input. The pipeline extracts the brushed items but also the shape of the brush which will be then used as a comparison with the brushed items (metrics stage). Then, brushed items are stored in bins displayed with small multiples where the user can interactively refine the selection (i.e. binning and filtering stage). Finally, the user can adjust the selection with additional brushing interactions. Note that both the PC and FPCA can be used for the binning process, but separately. Each small multiple comes from one metric exclusively.

As previously detailed in the related word section, many comparison metrics exist. While the pipeline can use any metric that fulfill the design requirement DR2 (continuous, ordered and meaningful comparison), the presented pipeline contains only two complementary algorithms: Pearson and FPCA. The first one focuses on a shape comparison basis with correlation between their representative vertices, while the latter focuses on curvature comparison. As shown in Figure 5.17, each metric produces different results. The user can use either of them depending on the type of filtering to be performed. During the initial development of this technique, the Euclidean distance (ED) and DTW was first considered, but their limitations were rapidly observed and PC and FPCA proved to be much more suitable to trajectory datasets. First, PC values are easier to threshold. A PC value greater than 0.8 provides a clear indication of the similarity of 2 shapes.

Moreover, to accurately discriminate complex trajectories, it is needed to go beyond the performance of ED. Furthermore, the direction of the trajectories, which is essential for the brushing technique, is not supported by ED and DTW similarity measures. Another disadvantage of using ED is domain- and task-specific threshold that can drastically vary depending on the context. PC, that is used in this approach, on the other hand, uses the same threshold independently of the type of datasets.

The two following paragraphs detail the two proposed algorithms.

Pearson’s Correlation (PC) Pearson’s Correlation (PC) is a statistical tool that measures the correlation between two datasets and produces a continuous measurement within $[-1, 1]$ with 1 indicating a high degree of similarity, and -1 an anti-correlation

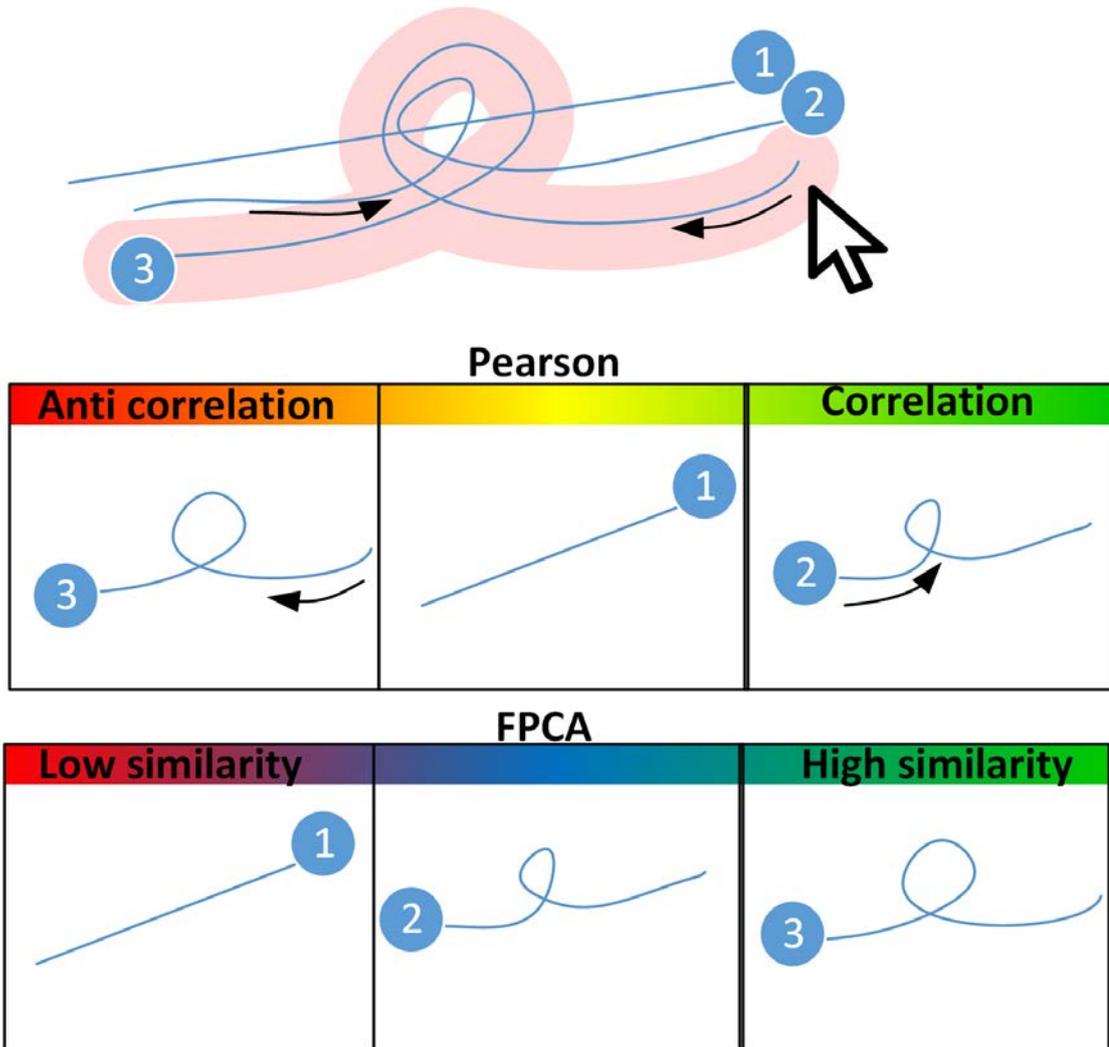


Fig. 5.17 Example of the two metrics for shape comparison usage. The user brushed around curve 3 and thus also selected curves 1 and 2. Thanks to the Pearson computation, the associated small multiples show that only curve 2 is correlated to the shape of the brush. Curve 3 is anti correlated since it is in an opposite direction to the shape of the brush. The FPCA computation does not take into account the direction but rather the curvature similarity. As such, only shape 3 is considered as highly similar to the brush shape input.

indicating an opposite trend [214]. This metric is well suited (DR2) for measuring dataset similarity (i.e. in trajectory points) [215, 216].

Pearson's Correlation PC between two trails T_i and T_j on the x -axis can be defined as follows:

$$r_x = \frac{COV(T_{i_x}, T_{j_x})}{\sigma_{T_{i_x}} \sigma_{T_{j_x}}}, \quad COV(T_{i_x}, T_{j_x}) = \mathbb{E}[(T_{i_x} - \overline{T_{i_x}})(T_{j_x} - \overline{T_{j_x}})] \quad (5.7)$$

Where $\overline{T_{i_x}}$ and $\overline{T_{j_x}}$ are the means, \mathbb{E} the expectation and $\sigma_{T_{i_x}}, \sigma_{T_{j_x}}$ the standard deviations. The correlation is computed on the y -axis and the x -axis for two dimensional points.

This metric is invariant in point translation and trajectory scale but it does not take into account the order of points along a trajectory. Therefore, the pipeline also considers the FPCA metric that is more appropriate to trajectory shape but that does not take into account negative correlation.

Functional Principal Component Analysis As presented in Chapter 2.3 knowing the mean curve $\bar{\gamma}$ and the principal component functions ϕ_j , a group of curves can be described and reconstructed (Inverse FPCA) with the matrix of the principal component score b_j of each curve. Usually, a finite vector (with fixed dimension d) of b_j scores is selected such that the explained variance is more than a defined percentile.

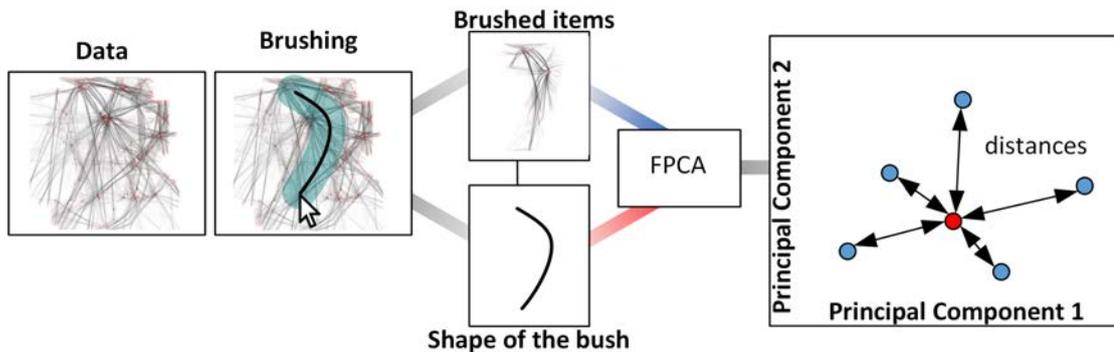


Fig. 5.18 Diagram of the transformation from the trail space to the 2D points space using FPCA

To compute a continuous and meaningful metric (DR2), the metric computation uses the two first Principal Components (PC) to define the representative point of a considered trajectory. Then, the metric is computed by the euclidean distance between the shape of the brush and each brushed trajectory in the Cartesian scatterplot PC1/PC2 (Figure 5.18). Each distance is then normalized between $[0, 1]$ with 1 corresponding to the largest difference in shape between the considered shape of the brush and the corresponding trajectory.

Binning and small multiple filtering Taking into account the computed comparison metrics, the pipeline stores the resulting values into bins. Items can then be sorted in continuous ways from the less similar to the most similar ones. While the Pearson measurements $\in [-1, 1]$ and the FPCA $\in [0, 1]$, this binning process operates in the same way. Each bin is then used to visually show the trajectories it contains through small multiples (5 small multiples which gives a good compromise between visualization compactness and trajectory visibility were used). The user can then interactively filter the selected items (DR4) with a range slider on top of the small multiple visualization. The user is thus able to decide whether to remove uncorrelated items or refine the correlated one with a more restricted criteria (DR3).

5.2.3 Design space

Interaction Paradigm by example This technique is designed to enable flexible and rapid brushing of trajectories, by both the location and the shape of the brush. The technique’s interaction paradigm is now described and illustrated in a scenario where an air traffic management expert studies the flight data depicted in Figure 5.20.

Scenario Introduction Aircraft trajectories can be visually represented as connected line segments that form a path on a map. Given the flight level (altitude) of the aircraft, the trajectories can be presented in 3D and visualized by varying their appearances [217] or changing their representation to basic geometry types [218]. Since the visualization considers a large number of trajectories that compete for the visual space, these visualizations often present occlusion and visual clutter issues, rendering exploration difficult. Edge bundling techniques [146] have been used to reduce clutter and occlusion but they come at the cost of distorting the trajectory shapes which might not always be desirable.

Analysts need to explore this kind of datasets in order to perform diverse tasks. Some of these tasks compare expected aircraft trajectories with the actual trajectories. Other tasks detect unexpected patterns and perform out traffic analysis in complex areas with dense traffic [217, 177]. To this end, various trajectory properties such as aircraft direction, flight level and shape are examined. However, most systems only support selection techniques that rely on starting and end points, or predefined regions. The interactive shape brush technique would be helpful for these kinds of tasks, as they require the visual inspection of the data, the detection of the specific patterns and then their selection for further examination. As these specific patterns might differ from the rest of the data precisely because of their shape, a technique that enables their selection through this characteristic will make their manipulation easier, as detailed in the example scenario. The considered dataset includes 4320 aircraft trajectories of variable lengths from one day of flight traffic over the French airspace.

Brushing A *trail* T is defined as a set of real-valued consecutive points:

$$T = [(Tx_1, Ty_1))^T, (Tx_2, Ty_2))^T, \dots, (Tx_n, Ty_n))^T] \quad (5.8)$$

where n is the number of points and $(Tx_i, Ty_i))^T$ corresponds to the $i - th$ coordinate of the *trail*. Figure 5.20 depicts an example of 4133 *trails* (aircraft in French airspace). The brush *Shape* consists of a set of real-valued consecutive points:

$$S = [(Sx_1, Sy_1))^T, (Sx_2, Sy_2))^T, \dots, (Sx_m, Sy_m))^T] \quad (5.9)$$

where m is the number of points. Note that while the length n of each trail is fixed, the

length m of the *Shape* depends on the length of the user brush. The *Shape* is smoothed using a 1-euro filter¹ [219] and then resampled to facilitate the trail comparison. The similarity metrics are then used in subsequences of the *shape* of approximately the same length as the brush *Shape*. In order to do this, each trial is first resampled so that each pair of consecutive vertices on the trail has the same distance $l_{vertices}$ [220].

The user starts by exploring the data using pan and zoom operations as depicted in Figure 5.19. He is interested in trajectories from the south-east of France to Paris. The user can choose if they are looking for a subsequence match or an exact match. A subsequence match involves the detection of trajectories having a subsequence similar to the *Shape* locally. Exact match comparison also takes into account the length of the trajectory and the *Shape* in order to select a trajectory, i.e., its length must be approximately similar to the length of the *Shape* (general measurements). This option is especially useful to select a trajectory by its start and end points (e.g., finding trajectories taking off from an airport A and landing at an airport B).

The exact matching is supported by analyzing the length of the trail and the *Shape* before applying the similarity metric algorithm. The analyst in the scenario activates the subsequence match where the Pearson's Correlation metric is selected by default and starts brushing in the vicinity of the target trajectories following the trajectory shape with the mouse. This will define both (1) the brush region and (2) the brush shape, that captures also the brush direction. Once the brushing has been completed, the selected trajectories are highlighted in green, as depicted in Figure 5.19-(b).

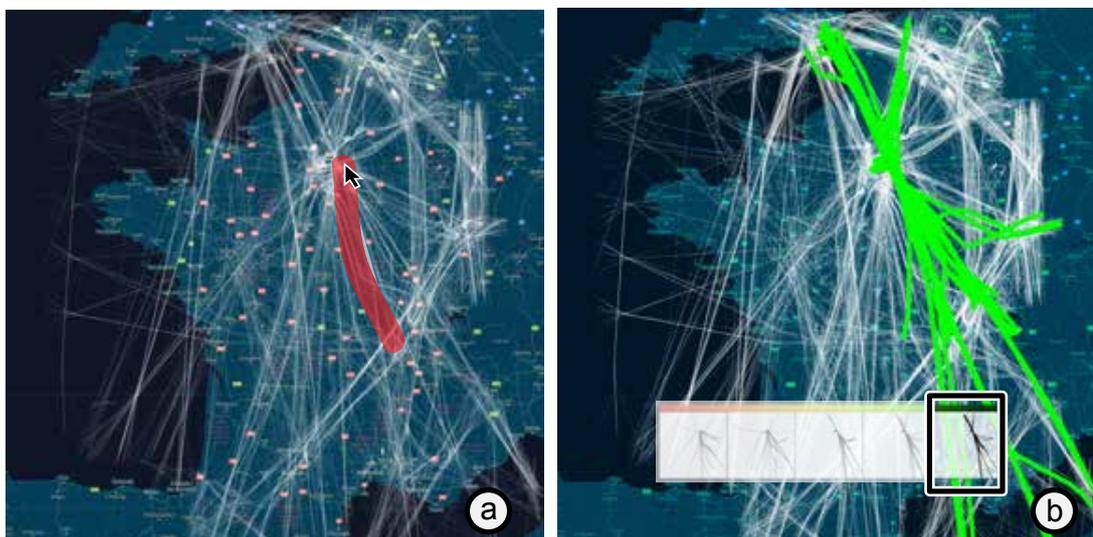


Fig. 5.19 (a) The user brushes the trajectories in order to select those from the south-east of France to Paris. (b) He selects the most correlated value to take into account the direction of the trails

¹1-euro-filter is an algorithm for filtering noisy signals with high precision and responsiveness. It consists of a first-order low-pass filter with an adaptive cut-off frequency

Small Multiples and User filtering The similarity calculation between the *Shape* and the brushed *region* will produce a *similarity* value for each trail contained in the region and the trails are distributed into small multiples as detailed in Section 5.2.2.

Once the user has brushed, he can adjust the selection by selecting one of the bins displayed in the small multiples and using its range slider. The range slider position controls the similarity level, and its size determines the number of trajectories selected at each slider position: the smallest size selects one trajectory at a time. The range slider size and position are adjusted by direct manipulation using the mouse. This enables a fine control over the final selection and makes the algorithm thresholding easier to understand as the user controls both the granularity of the exploration and the chosen similarity level.

As the bins are equally sized, the distribution of the similarity might not be linear across the small multiples. This makes navigation easier since the trajectories distribution in the small multiples is continuous. However, this also entails that not every bin corresponds to the same similarity value interval. To keep this information available to the user, a colored heatmap (from red to green) displays the actual distribution, as depicted in Figure 5.17.

In the current scenario, the expert, as he wishes to select only the flights *to* Paris and not *from* Paris, selects the trajectories that are correlated with the original brush, as the correlation takes into account the brush direction. These trajectories are on the right side of the small multiple, highlighted in green as depicted in Figure 5.19-(b).

The expert is then interested in exploring the flights that land on the north landing strip but that are not coming from the east. For this, he performs a new shape brush that will consider only the previously selected trajectories to identify the planes that do come from the east, and distinguishable by the "C" shape in the trajectories, as depicted in Figure 5.21. To be able to select the geometry precisely, the expert shifts to the FPCA metric, using the keyboard shortcut. In this case the small multiple arranges the trajectories from less similar to more similar. This entails that the small multiple based on FPCA also enables the selection of all the trajectories that *do not* match the specified *Shape* but which are contained in the brushing region. As all trajectories passing through the north landing strip are contained in the brushing region, the most similar trajectories will correspond to the ones that have a "C" shape, in the same orientation as the *Shape*, and thus come from the east. The less similar will be the ones that interest the analyst, so he can select them by choosing the most dissimilar small multiple as depicted in Figure 5.21-(b).

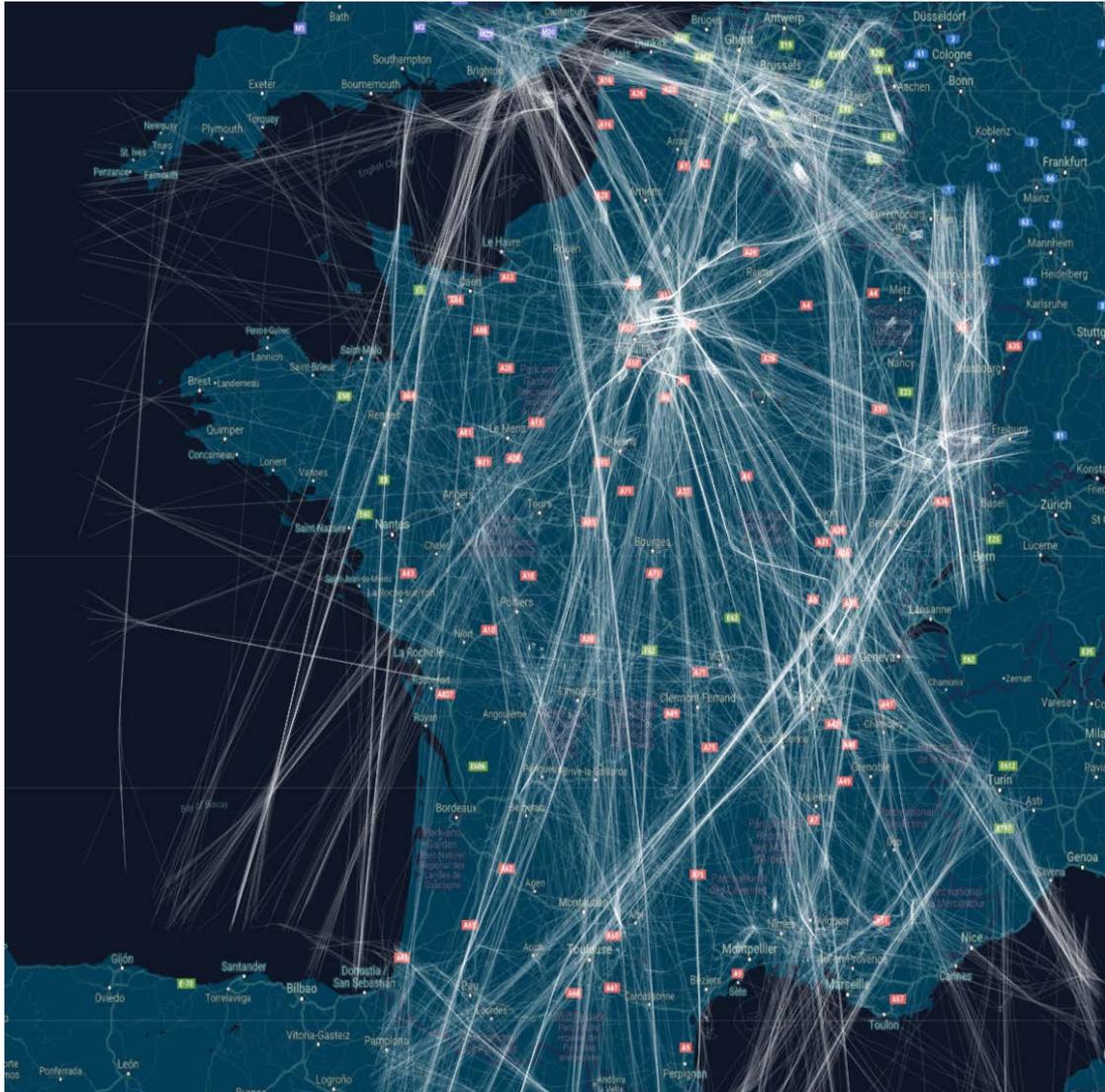


Fig. 5.20 One day’s aircraft trajectories in French airspace, including taxiing, taking-off, cruise, final approach and landing phases, selecting specific trajectories using the standard brushing technique will yield inaccurate results due to the large number of trajectories, occlusions, and closeness in the spatial representation.

5.2.4 Use cases

There is a strong demand for targeted brushing to select motifs in datasets. In various domains, including aircraft trajectories, eye tracking, GPS trajectories or brain fiber analysis, there is a substantial need to be able to discover hidden motifs in large datasets. Undoubtedly, retrieving desired trails in such datasets would help analysts to focus on the most interesting parts of the data. The system was built using C# and OpenTK on a 64bit² XPS 15 Dell Laptop. Although, both PC and FPCA provide different but valuable results, the running performance was 10 times faster with PC compared to FPCA .

²Intel(R) Core(TM) I7-4712HQ CPU @ 2.30GHz,2301 MHz, 4 core, 8 threads

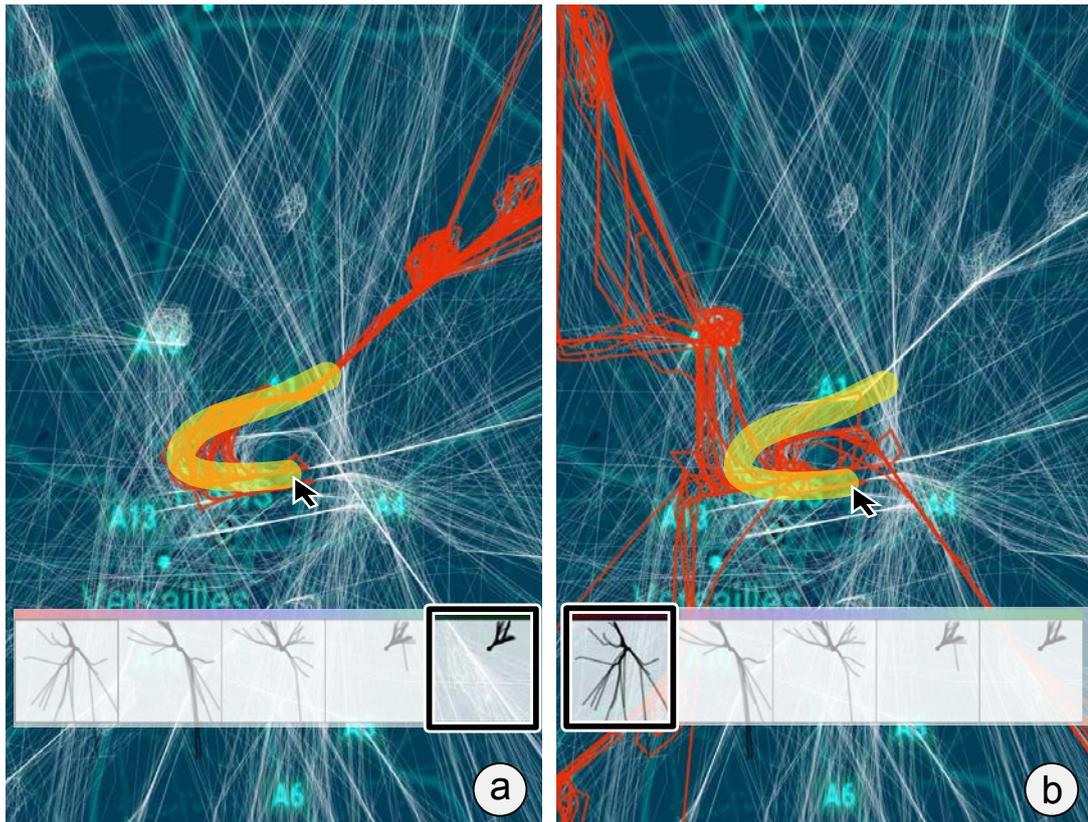


Fig. 5.21 (a) The user filters the trajectories that land on the north runway in Paris by brushing following the "C" shape. This retrieves the flights that come from the east. (b) He changes the selection on the small multiples, to select all the dissimilar Shapes, resulting in the trajectories that land on the north landing strip strip but that do not come from the east.

The technique was first tested informally with experts from aerospace domain with more than 10 years of experience in trajectories analysis. While the collected feedback was largely positive, some limitations regarding the misunderstanding of the filtering parameters were observed. Given the novelty of the interaction technique, users needed a small training period to better understand the semantic of the small multiples. Nevertheless, the technique provides initial good selection result without any parameter adjustment.

Because the presented technique is not designed to replace standard brushing but rather to complement it, the informal user study with an evaluation based on real use cases is extended. These use cases show how the proposed technique facilitates trajectories selection in dense areas, where standard brushing would require multiple user actions (panning, zooming, brushing).

Eye tracking Data Eye-tracking technologies are gaining popularity for analyzing human behaviour, in visualization analysis, human factors, human-computer interaction, neuroscience, psychology and training. The principle consists in finding the likely

objects of interest by tracking the movements of the user's eyes [221]. Using a camera, the pupil center position is detected and the gaze, i.e, the point in the scene the user is fixating on, is computed using a prior calibration procedure [222, 223, 224, 225]. Therefore, the gaze data consist of sampled trails representing the movements of the user's eye gaze while completing a given task.

Two important types of recorded movements characterize eye behaviour: the fixations and saccades [226]. Fixations are the eye positions the user fixates for a certain amount of time, in other words, they describe the locations that captured the attention of the user. The saccades connect the different fixations, i.e, they represent the rapid eye movements from one location to another. The combination of these eye movements are called the scanpath (Figure 5.22A). The scanpath is subject to overplotting. This challenge may be addressed through precise brushing techniques to select specific trails. Usually, fixation events are studied to create an attention map which shows the salient elements in the scene. The salient elements are located at high density fixation areas. However, the temporal connections of the different fixations provide additional information. The saccades enable the links between the fixations to be maintained and the temporal meaning of the eye movement to be held. Discovering patterns in the raw scanpath data is difficult since, in contrast to aircraft trajectories, eye movements are sparser and less regular (Figure 5.22). To address this, different kinds of visualizations for scanpaths have been proposed in the literature. For example, edge bundling techniques [227] minimize visual clutter of large and occluded graphs. However, these techniques either alter trail properties such as shape and geometric information, or are otherwise computationally expensive, which makes them unsuitable for precise exploration and mining of large trail datasets. Moreover, it is possible to animate eye movements in order to have an insight of the different fixations and saccades. However, given the large datasets of eye movements retrieved from lengthy experiments containing thousands of saccades, this approach is unnecessarily time-consuming and expensive.

In the following, is described how this study's approach supports proper and more efficient motif discovery on such eye tracking datasets. The tested dataset is adapted from Peysakhovich et al. [228], where a continuous recording of eye movement in a cockpit was performed. The gaze data was recorded at 50 Hz. Sequential points located in a square of 20×20 pixels and separated by at least 200 ms were stored as a fixation event and replaced by their average in order to reduce noise coming from the micro-saccades and the tracking device.

In order to illustrate some examples, a domain expert who wishes to explore the movements of the pilot's eyes in a cockpit could be considered. When performing a task, the pilot scans the different instruments in the cockpit, focuses more on certain instruments or interacts with them. Especially in this context, the order of pilot attention is important since checking a parameter in one instrument may give an indication of the

information displayed in another instrument. For example, the priority of the Primary Flight Display (PFD) instrument compared to Flight Control Unit (FCU) will differ for the cruise phase as compared to the final landing approach [228]. As an example of analysis, the user wishes to explore the movement of the eye from the Primary Flight Display (PFD) to the Navigation Display (ND). Selecting these scanpaths using traditional brushing techniques would be challenging because of the clutter, selecting those scanpaths would introduce additional accidental selections. Therefore, he brushes these scanpaths using a shape that extends from the PFD to the ND, applying the Pearson metric to consider the direction. Figure 5.22(a) depicts the brushed eye movements that correspond to the most correlated trails in the small multiple. There are several saccades between those two devices, and this is inline with the fact that saccadic movements between the PFD and the ND are typically caused by parameter checking routines.

However, when the user changes the selection and brushes the scanpath between the ND and the FCU, it is surprising to see that there is only one saccade between them. Brushing now with a shape that goes between the PFD and the FCU (Figure 5.22-(c)) reveals only one scanpath. This is difficult to visualize in the raw data or using the standard brushing technique. A final *Shape* searching for an eye movement from the PFD to the LAI and passing by the FCU, results in only one saccade (Figure 5.22-(d)). To determine the meaning of this behavior, the tool also enables the expert to exploit a continuous transition to increase the visibility and gain insight on when these saccadic movements occurred (temporal view). The user can change the visual mapping from the (x,y) gaze location to the (time,y) temporal view. This smooth transition avoids abrupt change to the visualization [176] (figure 5.23). As an extension, it would thus be possible to propose experiments to evaluate the ocular behaviors of the pilots within the framework of atypical approaches analyzed in this thesis. The proposed brushing interactions would allow a finer analysis and a simpler selection of relevant patterns. This human factor analysis could provide a better understanding of the mechanisms at stake and the behavior of pilots in the management of atypical events.

GPS Data GPS trajectories consist of sequential spatial locations recorded by a measurement instrument. Subjects such as people, wheeled vehicles, transportation modes and devices may be tracked by analyzing the spatial positions provided by these instruments. Analysts may need to explore and analyze different paths followed by the users. The advances in position-acquisition and ubiquitous devices have granted extremely large location data, which indicate the mobility of different moving targets such as autonomous vehicles, pedestrians, natural phenomena, etc. The commonness of these datasets calls for novel approaches in order to discover information and mine the data [229].

Traditionally, researchers analyse GPS logs by defining a distance function (e.g,

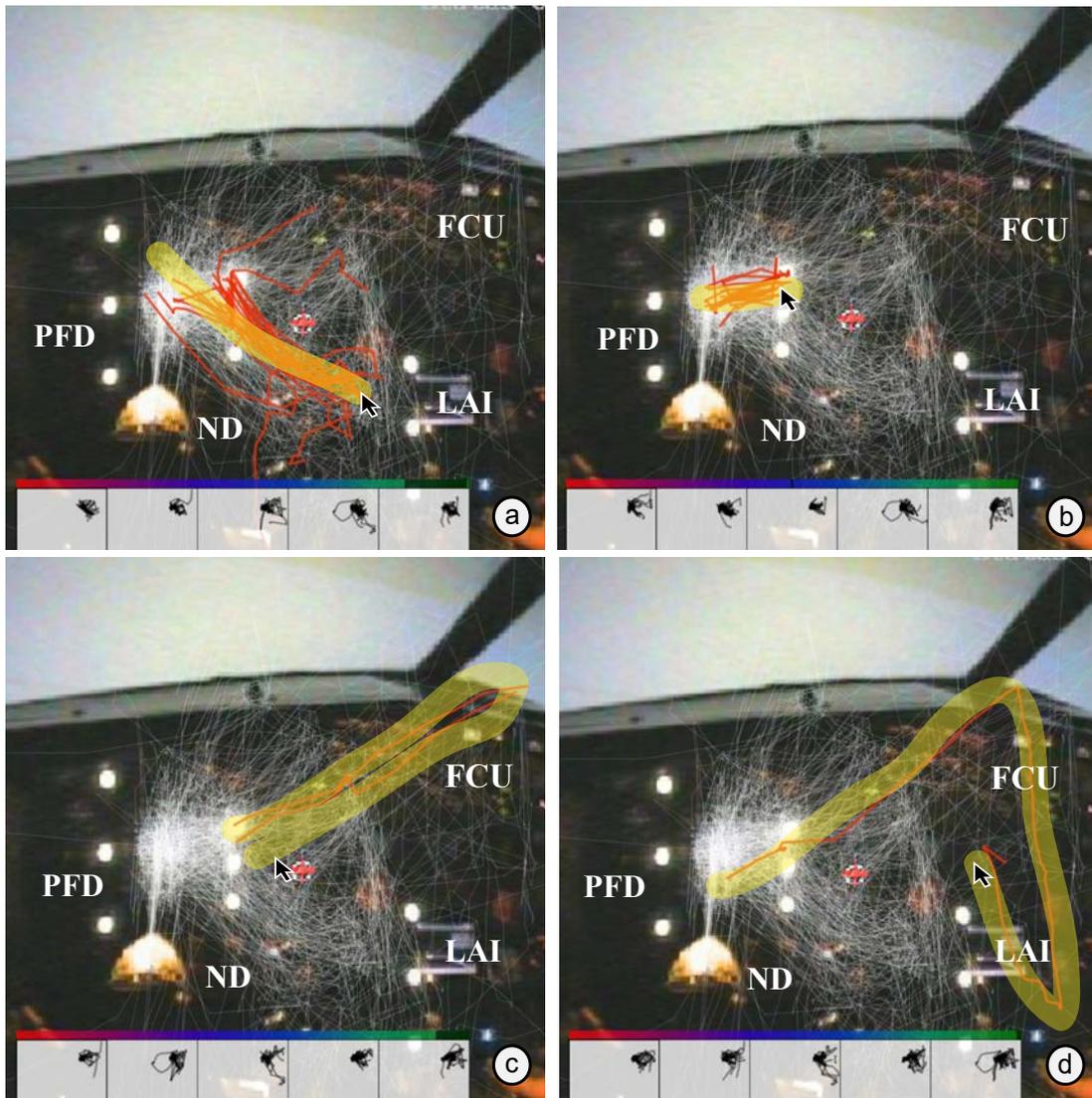


Fig. 5.22 (a) Selected eye movements between the PFD and ND, (b) Selected eye movements in the vicinity of the PFD , (c) Only one saccade between the ND and the FCU, (d) Eye movement from the PFD to the LAI passing by the FCU.

KNN) between two trajectories and then applying expensive processing algorithms to address the similarity detection. For example, they first convert the trajectories into a set of road segments by leveraging map-matching algorithms. Afterwards, the relationship between trajectories is managed using indexing structures [230, 229]. Using the data provided by Zheng et al. [2], the aim is to investigate different locations followed by the users in Beijing. The data consists of GPS trajectories collected for the Geolife project by 182 users during a period of over five years (from April 2007 to August 2012) [2]. Each trajectory is represented by a 3D latitude, longitude and altitude point. A range of users' outdoor movements were recorded, including life routines such as travelling to work, sports, shopping, etc.

As the quantity of GPS data is becoming increasingly large and complex, proper

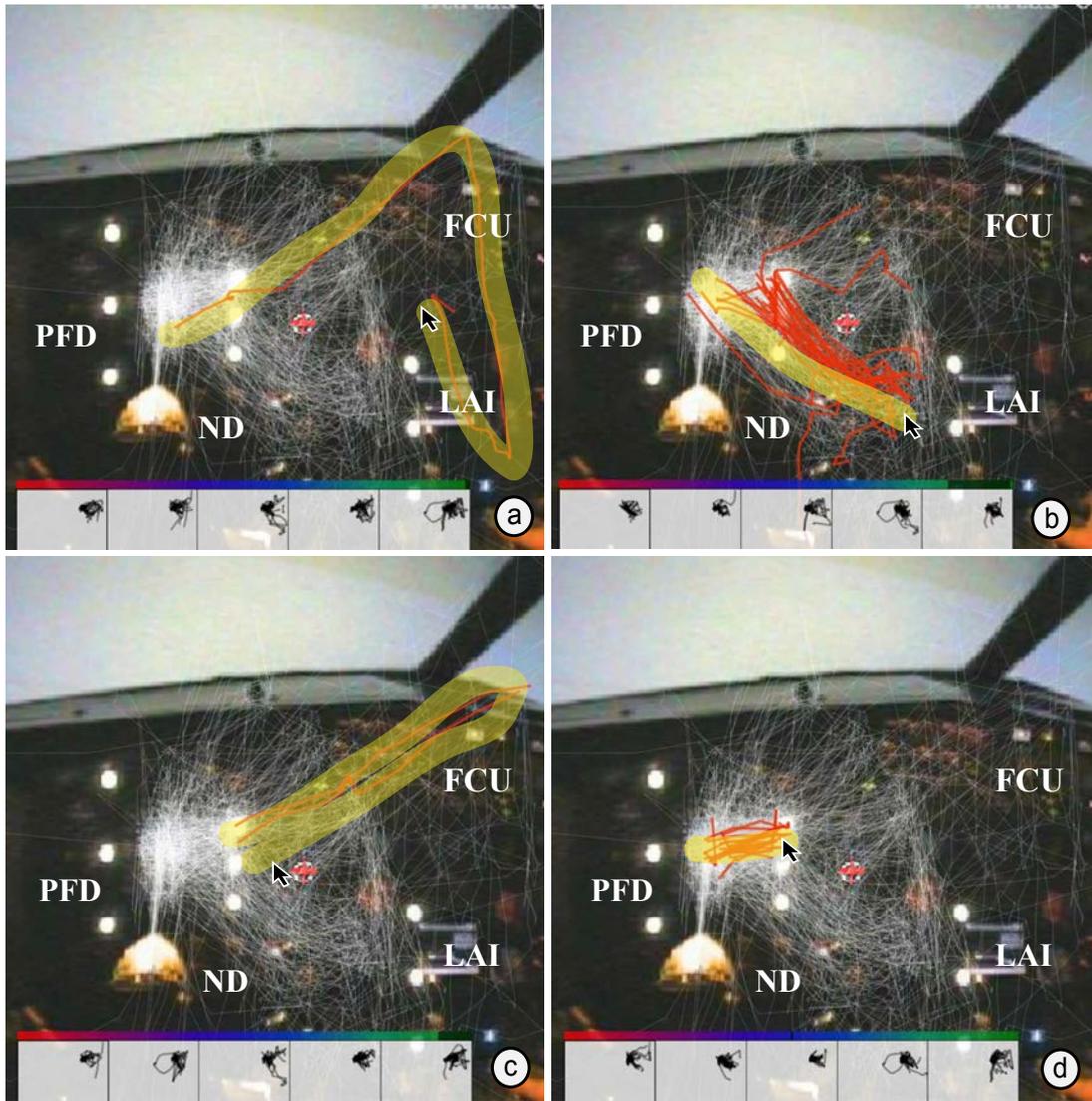


Fig. 5.23 Animated transition between the X/Y gaze view to the temporal view. This helps to detected how the selected eye movement occurred over time.

brushing is challenging. Using bounding boxes somewhat alleviate this difficulty by setting the key of interest on the major corners. However, many boxes must be placed carefully for one single selection. The boxes can help the analysts to select all the trajectories that pass through a specific location, but do not simplify the analysis of overlapping and directional trajectories. This study's approach intuitively supports path differentiation for both overlapping trajectories and takes direction into account. For example, it is interesting to answer questions about the activities people perform and their sequential order [2]. For this dataset, the authors were interested in finding event sequences that could inform tourists. The shape based brushing could serve as a tool to further explore their results. For example, if they find an interesting *classical sequence* that passes through locations A and B, they can further explore if this sequence corresponds to a larger sequence and what other locations are visited before or after. A

first brushing and refinement using the FPCA metric and small multiples enables them to select all the trajectories that include a precise *event sequence* passing through a set of locations, as depicted in Figure 5.24. A second brushing using the Pearson metric enables further explorations that also take into account the direction of the trajectories. Switching between the correlated trajectories and the anti-correlated ones, the user can gain insight about the visitation order of the selected locations.

The use on GPS data shows that these techniques could be used to manually retrieve atypical approaches by exploring the data. This approach is different from the approach developed in chapter 3 since it is not automatic and implies that the user has an a priori knowledge of the shape or a particular pattern of the trajectories he is looking for.



Fig. 5.24 Three different trajectories containing three different event sequences from [2]. The user is able to select all trajectories passing through particular locations

5.2.5 Discussion

The proposed brushing technique leverages existing methods with the novel usage of the shape of the brush as an additional filtering parameter. The interaction pipeline shows different data processing steps where the comparison algorithm between the brushed items and the shape of the brush plays a central role. While the presented pipeline contains two specific and complementary comparison metric computations, another one can be used as long as it fulfills the continuity and metric semantic requirements (DR2). There are indeed many standard approaches (ED, DTW, Discrete Fréchet distance) that are largely used by the community and could be used to extend the technique when faced with different datasets. Furthermore, the contribution of this research is a novel shape-based brushing technique and not simply a shape similarity measure. In this work, two reasonable similarity measures that fulfill the shape-based brushing method are found: the FPCA distance comparison provides an accurate curve similarity mea-

surement while the Pearson metric provides a complementary criteria with the direction of the trajectory.

In terms of visualization, the binning process provides a valuable overview of the order of the trajectory shapes. This important step eases the filtering and adjustment of the selected items. It is important to mention that this filtering operates in a continuous manner as such trajectories are added or removed one by one when adjusting this filtering parameter. This practice helps to fine tune the selected items with accurate filtering parameters. The presented scenario shows how small multiple interactions can provide flexibility. This is especially the case when the user brushes specific trajectories to be then removed when setting the compatibility metrics to uncorrelated. This operation performs a brush removal. The proposed filtering method can also consider other types of binning and allows different possible representations (i.e. various visual mapping solutions).

This section illustrates the shape based brushing technique with three application domains (air traffic, eye tracking, gps data), but it can be extended to any moving object dataset. However, the evaluation is limited by the number of studied application domains. Furthermore, even if various users and practitioners participated in the design of the technique, and assessed the simplicity and intuitiveness of the method, a more formal evaluation was not conducted. The shape based brush is aimed at complementing the traditional brush, and in no way is said to be more efficient or effective than the original technique for all cases. The scenarios are examples of how this technique enables the selection of trails that would be otherwise difficult to manipulate, and how the usage of the brush area and its shape to perform comparison opens novel brushing perspectives.

The technique also presents limitations in its selection flexibility, as it is not yet possible to combine selections. Many extensions can be applied to the last step of the pipeline to support this. This step mainly addresses the DR4 where the selection can be refined thanks to user inputs. As such, multiple selections can be envisaged and finally be composed. Boolean operations can be considered with the standard And, Or, Not. While this composition is easy to model, it remains difficult for an end user to master the operations when there are more than two subset operations [231][176]. As a solution, Hurter et al. proposed an implicit item composition with a simple drag and drop technique [176]. The pipeline can be extended with the same paradigm where a place holder can store filtered items and then be composed to produce the final result. The user can then refine the selection by adding, removing or merging multiple selections.

In this section, a novel sketch-based brushing technique for trail selection was proposed and investigated. This approach facilitates user selection in occluded and cluttered data visualization where the selection is performed on a standard brush basis while taking into account the shape of the brush area as a filtering tool. This brushing tool

works as follows. Firstly, the user brushes the trajectory of interest trying to follow its shape as closely as possible. Then, the system pre-selects every trajectory which touches the brush area. Next, the algorithm computes a distance between every brushed shape and the shape of the brushed area. Comparison scores are then sorted and the system displays visual bins presenting trajectories from the lowest scores (unrelated - or dissimilar trajectories) to the highest values/scores (highly correlated or similar trajectories). The user can then adjust a filtering parameter to refine the actual selected trajectories that touch the brushed area and which have a suitable correlation with the shape of the brushed area. The cornerstone of this shape-based technique relies on the shape comparison method. Therefore, two algorithms which provide enough flexibility to adjust the set of selected trajectories were chosen. One algorithm relies on functional decomposition analysis which insures a shape curvature comparison, while the other method insures an accurate geometric based comparison (Pearson algorithm). To validate the efficiency of this method, three examples of usage with various types of trail datasets were shown.

This work can be extended in many directions. First, it can be extended with additional application domains and other types of dataset such as car or animal movements or any type of time varying data. Then, other types of input to extend the mouse pointer usage can be considered. Virtual Reality data exploration with the so called immersive analytic domain gives a relevant work extension which will be investigated in the near future. Finally, it can also be considered adding machine learning to help users brush relevant trajectories. For instance, in a very dense area, where the relevant trajectories or even a part of the trajectories are not visible due to the occlusion, additional visual processing may be useful to guide the user during the brushing process.

5.3 Conclusions

In this chapter, novel manipulation and generation techniques using Functional Principal Component Analysis have been explored.

First a dedicated pipeline to support, clustering, generation and modification of trajectory data was presented. It was applied to the modification of aircraft interception altitudes after the Grenelle de l'Environnement at Charles de Gaulle airport. The results are promising and underlined a potential use in the simulation of noise impact while procedure are changed.

Second, a novel brushing technique using Functional Principal Component Analysis as a similarity metric between shape was shown. It enhance the manipulation of trail set by adding an extra feature in order to select the desired samples.

This chapter was dedicated to development of novel manipulation and generation techniques using Functional Principal Component Analysis. The next Chapter will focus on extending to real time the atypical detection methodology presented in Chapter 3

CHAPTER 6

Directions to real time atypical approach detection using Dubins Path

"Y a pas à dire, dès qu'il y a du dessert, le repas est tout de suite plus chaleureux !"

Arthur - Kaamelott, Livre I, La tarte aux myrtilles

Contents

6.1	Methodology and Background	129
6.1.1	Dubin curves problem for trajectory generation	129
6.1.2	Real-time atypical scoring methodology	130
6.1.3	Suggested trajectory methodology	132
6.2	Study	132
6.2.1	Asiana flight 214	133
6.2.2	Pegasus Airlines flight 2193	136
6.2.3	Hermes Airlines flight 7817	139
6.2.4	Air India Express flight 1344	141
6.3	Discussion	142
6.4	Conclusions	144

This chapter presents two major contributions. First, it extends to real time the post-operational methodology presented in Chapter 3 using Dubins path as an estimator of the remaining distance. Second, a methodology to generate a typical 2D energy management trajectory is proposed while the aircraft is flying downwind or on base leg. At the end, a complete real-time atypical trajectory detection tool for Air Traffic Controllers is presented with two modes depending on the position of the aircraft. Before intercepting the localizer, the aircraft is usually radar vectored. At this stage, the tool gives two types of information. First, the current status of the aircraft considering the shortest path to intercept the localizer is given. Then, if this trajectory presents an atypicality in the energy management, a suggested trajectory is calculated. Finally, when

the aircraft has intercepted the localizer, the tool switches to the back-up net mode, giving the current status of the aircraft.

This chapter is divided into three parts. First, the mathematical background around Dubins' paths and optimal control is presented. In addition, the real-time extension and the 2D generation process using Dubins paths are detailed. Second, the methodology and results are illustrated on different case studies, in particular the Asiana Airline Flight 214 at San Francisco airport on July 6, 2013, the Pegasus Airlines Flight 2193, which overran Istanbul runway on February 5, 2020, the Hermes Airline Flight 7817, which overran Lyon Saint-Exupéry runway on March 29, 2013, and the Air India Express Flight 1344, which overran Calicut airport on August 07, 2020. Finally the methodology is discussed. This research is currently under review in a journal [232]

6.1 Methodology and Background

6.1.1 Dubin curves problem for trajectory generation

The real-time extension of the atypicality score triggers a major problem. To calculate the atypicality score, the remaining distance to the runway threshold must be known. This distance is well known if the aircraft is aligned with the runway extension axis. However, if the aircraft is still flying downwind or on base leg, the distance remaining to the threshold is unknown. A simple estimate of the remaining distance can be obtained using Dubins curves.

Dubin curves solve the well-known problem of the shortest path between two oriented points under the constraint of maximum curvature. The general solution was published by Dubins in 1958 [233]. It was also proved by Boissonnat et al. [234] by applying the principle of Pontryagin maximum [235].

The problem can be formulated as an optimal control problem, where the state variables are x, y, θ , the control variable is u , R is the minimum turning radius, and t is the curvilinear abscissa :

$$\begin{aligned} \min_u \quad & t_f = \int_0^{t_f} 1 \, dt \\ \text{s.t.} \quad & \dot{x}(t) = \cos \theta(t), \quad x(0) = x_0, \quad x(t_f) = x_f \quad (1), \\ & \dot{y}(t) = \sin \theta(t), \quad y(0) = y_0, \quad y(t_f) = y_f \quad (2), \\ & \dot{\theta}(t) = u(t), \quad \theta(0) = \theta_0, \quad \theta(t_f) = \theta_f \quad (3), \\ & |u(t)| \leq \frac{1}{R}, \quad t \in [0, t_f] \quad (4) \end{aligned}$$

The problem parameters are illustrated in Figure 6.1. The speed is assumed to be constant. The minimum turning radius R induce constraint (4) over the norm of the

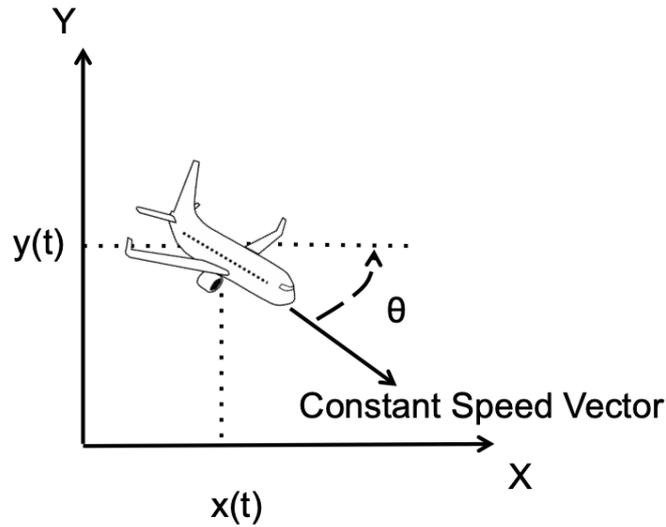


Fig. 6.1 Control model for Dubins path

turning angle. The angle control command is given in constraint (3). The position derivatives are given in constraints (1) and (2) and depends on the turning angle θ . Initial and final states are known and depict the position and orientation.

The solution is one of the following six combinations: RSR, RSL, LSR, LSL, RLR, LRL. Where R and L represent respectively a right and left turning arc at maximum curvature, and S a straight line segment. A simple illustration is given in Figure 6.2, where the solution is RSL : a right turn, then a straight line, and finally a left turn. It has been applied to various fields such as robotics for example. Buil et al. applied Dubins' method to find the shortest path for non-holonomical robots [236]. Furthermore, it has been extended to 3D by Chitsaz et al. referred as to Dubins' plane [237].

6.1.2 Real-time atypical scoring methodology

Returning to the real-time problem extension, the goal is to estimate the remaining distance to the runway threshold, consequently a 2D path is sufficient for this evaluation. The following assumptions will be applied. First, the aircraft has a constant ground speed of 180kts and turns with a bank angle of 25° , which implies a turning radius of 1.01NM. Three situations, illustrated in Figure 6.3, will be considered to calculate the remaining distance:

1. The aircraft is aligned with the runway extended centerline. The aircraft will be considered aligned with the runway extended centreline if the distance between the current position and its projection above the runway extended centreline is less than the minimum turning radius (blue area in Figure 6.3).
2. The aircraft is not aligned with the runway extended centreline and its orthog-

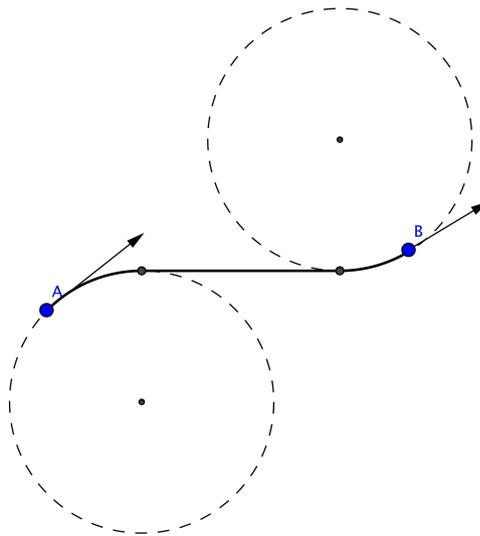


Fig. 6.2 Dubins path between two points A and B

onal projection on the runway extended centreline is located after the intercept chevrons of the minimum altitude FAP (green area in Figure 6.3).

3. The aircraft is not aligned with the runway extended centreline, and its orthogonal projection on the runway extended centreline is located before the FAP minimum altitude intercept chevrons (red area in Figure 6.3).

In situation 1), the remaining distance to the runway threshold is assumed to be the distance to the orthogonal projection plus the distance from the projection to the runway threshold. In situation 2), the length of the Dubins curve from the current position with the current heading to the orthogonal projection with the runway heading is calculated and added to the distance from the projection to the runway threshold. Finally, in situation 3), the length of the Dubins curve from the current position with the current heading to the FAP chevrons with the runway heading is calculated and added to the distance between the FAP chevrons and the runway threshold.

When the estimated remaining distance is calculated, it is then easy to apply the scoring on the appropriate window model, as proposed in the post-operational methodology developed in Chapter 3.

In summary, there are two ways to determine the remaining distance depending on the aircraft situation. When the aircraft has not yet intercepted the runway centreline, the distance is estimated by using Dubins curves. Otherwise, the remaining distance correspond to a straight line to the runway threshold. Finally, this distance is used to compute the atypical coefficient by applying the corresponding sliding window atypical energy model.

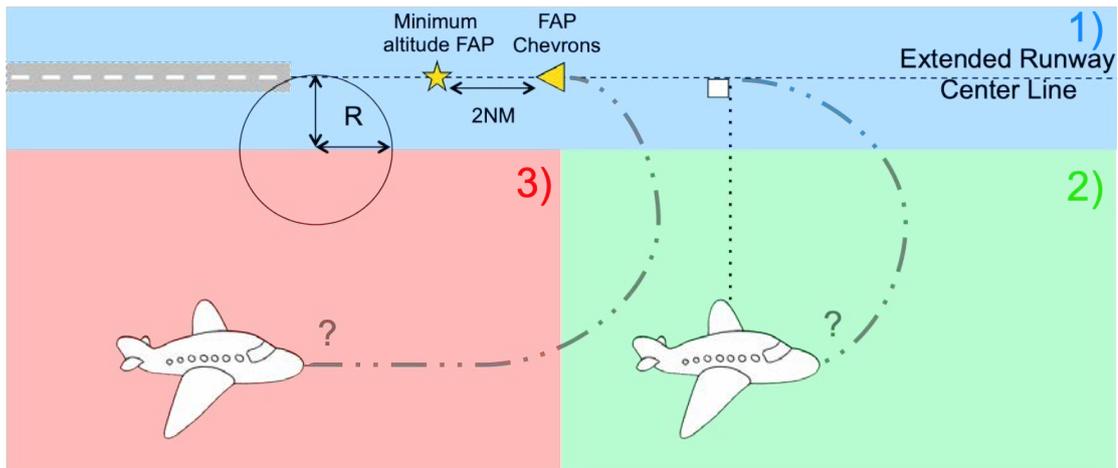


Fig. 6.3 The different cases of the methodology

6.1.3 Suggested trajectory methodology

The trajectory generation process only addresses high-energy cases where the aircraft has not intercepted the extended runway centreline. Low-energy cases are not studied here. Indeed, when the atypicality is due to high energy, the idea is to give a longer trajectory in order to allow easier dissipation of the excess of energy. Conversely, when a low energy is detected, the idea is not to shorten the trajectory, so only the atypicality information is given to allow a better situation awareness.

The generation process is simple; the point of interception of the localizer is gradually moved away from the runway threshold until a suitable atypicality level is reached. At each stage, the current energy state is evaluated using Dubins curves. The generation process is carried out if the coefficient of the current state is greater than 0.2 and the process is stopped if such a coefficient on the generated trajectory is less than 0.05.

The tool therefore gives two pieces of information to the air traffic controller. Firstly, it indicates the current energy state in the event of a direct trajectory to the intercepting chevrons. Finally, if this energy state is high energy. The tool proposes a trajectory where the interception point has been pushed back. This extended trajectory should enable the aircraft to manage this over-energy in a more nominal way.

6.2 Study

The purpose of this section is to illustrate the use of the tool and its functionalities. The analysis of four air crashes in replay mode is presented : the Asiana Airline Flight 214 at San Francisco airport on July 6, 2013, the Pegasus Airlines Flight 2193, which overran Istanbul runway on February 5, 2020, the Hermes Airline Flight 7817, which overran Lyon Saint-Exupéry runway on March 29, 2013, and the Air India Express Flight 1344,

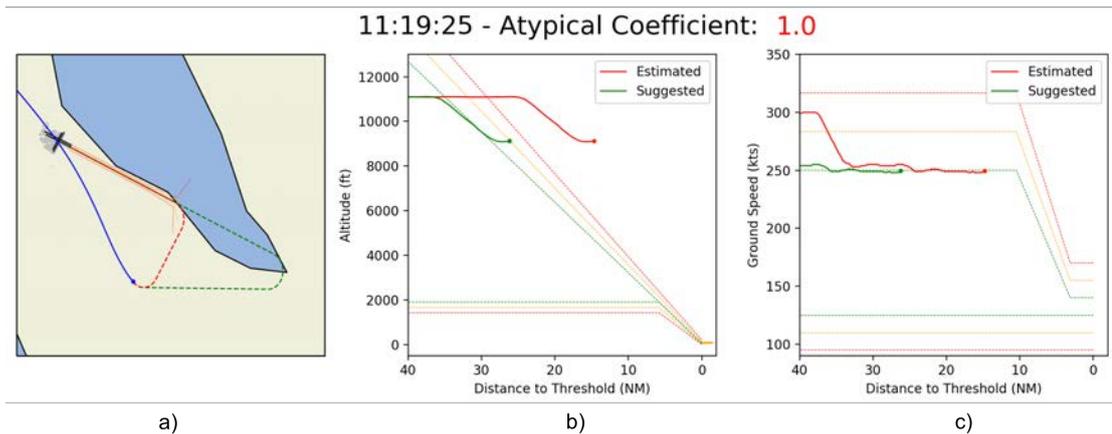


Fig. 6.4 Asiana Airline Flight 214 lateral, altitude and speed profiles, when the aircraft was in downwind leg. The tool indicates that a direct to the interception chevrons is critical (in red) and would cause the plane to be in over-energy due to high potential energy figure (a). The tool suggests a trajectory extending the tailwind leg (in green)

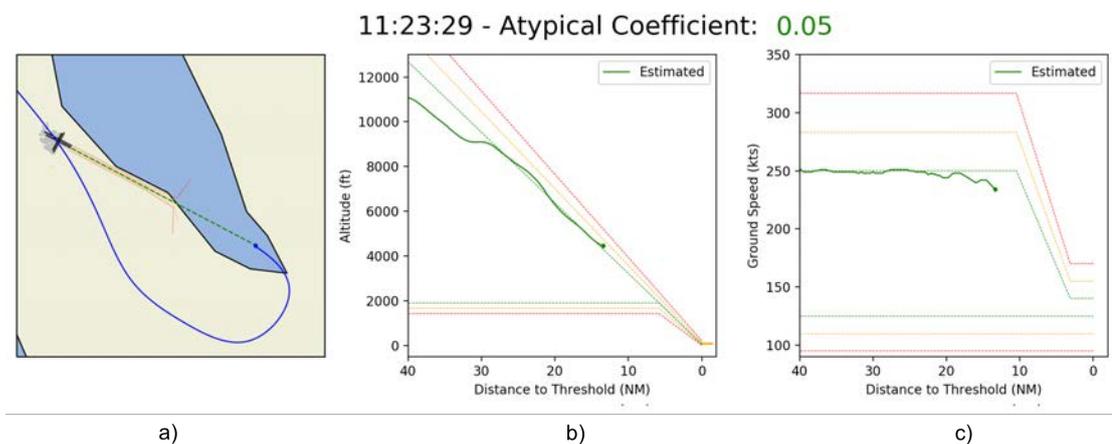


Fig. 6.5 Asiana Airline Flight 214 lateral, altitude and speed profiles. The plane has just intercepted the runway centerline (a) and is in good condition, it is on the glide path (b) and presents a nominal speed (c).

which overran Calicut airport on August 07, 2020.

6.2.1 Asiana flight 214

On July 6, 2013, a Boeing 777-200ER operating Asiana Airlines Flight 214 struck a sea-wall at San Francisco International Airport (SFO) in San Francisco, California. Three passengers were fatally injured; 40 passengers were seriously injured. The aircraft was destroyed by the force of the impact and a post-crash fire. [6].

The flight was vectored for a visual approach to Runway 28L and intercepted the final approach track at an altitude slightly above the desired 3° glide path. After accepting air traffic control's instruction to maintain a speed of 180 knots at 5 nm from the runway, the crew mismanaged the aircraft's descent and the aircraft was well above the 3° glide path. In an attempt to increase the aircraft's rate of descent and capture the

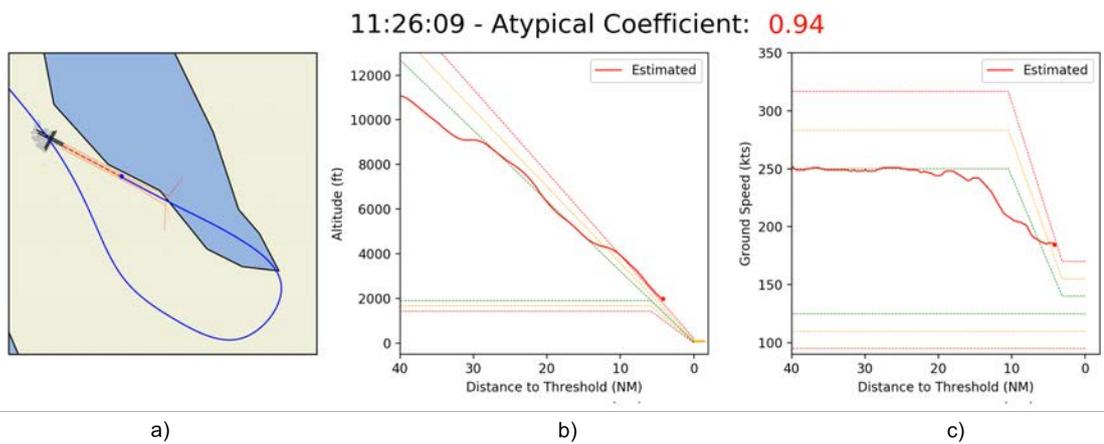


Fig. 6.6 Asiana Airline Flight 214 lateral, altitude and speed profiles. The aircraft is 5NM from the runway threshold (a), above the glide path (b) with a ground speed of 180kts due to air traffic constraints (c). The plane is in over-energy due to both high potential and kinetic energy.

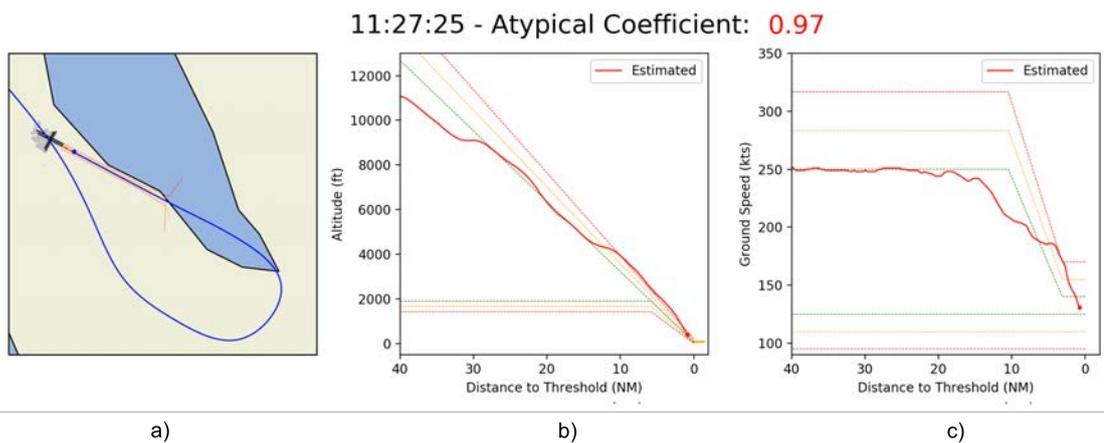


Fig. 6.7 Asiana Airline Flight 214 lateral, altitude and speed profiles. The plane is at 500ft ground level, its speed (c) and altitude (b) have returned to nominal values but its energy variation is atypical (with a coefficient of 0.97) and directs it towards an under-energy

desired glide path, the pilot flying (PF) selected an autopilot (A/P) mode (speed of flight level change (FLCH SPD)) that, instead, caused the autopilot system to initiate a climb because the aircraft was below the selected altitude. The PF disconnected the A/P and moved the thrust levers to idle, causing the autothrottle (A/T) to go into HOLD mode, a mode in which the A/T does not control airspeed.

At 500 feet above the airport elevation, the stabilization altitude, the precision approach path indicator (PAPI) would have shown that the aircraft was slightly above the desired glide path. The airspeed had just reached the correct approach speed of 137 knots. However, the thrust levers were still at idle, and the rate of descent was about 1200 feet per minute, well above the theoretical descent rate of about 700 fpm. The aircraft subsequently descended below the desired glide path and the airspeed continued to decrease. At about 200 feet, the flight crew became aware of the low airspeed and the low trajectory but did not initiate a go-around until the aircraft was below 100 feet, at which point the aircraft did not have the ability to perform a go-around.

On-board data, available on the National Transportation Safety Board website, was used to study this crash. Only the parameters available on the ground (ground speed, vertical speed and altitude) are used by the algorithm. The study of this crash allows to highlight two key points of the algorithm. First, the use when the aircraft is downwind, allows to alert of the aircraft status in case of a radar vectoring for example. Second, the ability to detect atypical variations within acceptable limits.

At 11:19:25 the aircraft is downwind, perpendicular to the intercepting chevrons. The atypicality score for a direct hit to the chevrons is maximum (1.0). The aircraft has a ground speed of 250kts, and is very high on the plane. The suggested trajectory mode proposes to extend the trajectory to intercept the runway centerline further (see Figure 6.4). At 11:23:29, the aircraft intercepted the runway centreline after an extended downwind leg. The atypical coefficient returned to nominal values. The plane is very slightly above the glide path with a speed of 240 kts ground speed (see Figure 6.5) At 11:26:09, the aircraft is 5NM from the runway threshold, at 180kts ground speed (Air Traffic Control constraint) and above the theoretical glide path. The atypicality is high (0.94), and the aircraft is showing an over-energetic condition (see Figure 6.6). At 11:27:25, the aircraft is at 500ft ground level on glide path, its ground speed falls in nominal range. However its energy variation is atypical and it has an atypical score of 0.97 which will not decrease for the rest of the final (see Figure 6.7).

This case is very interesting because the aircraft has switched from high energy to low energy. Traditional methods with a high and a low bound do not take into account the sudden and atypical variations. Thus, during a certain period of time the aircraft could be considered in a nominal phase when it is not. Its abrupt energy variation is atypical and could have been detected with this type of tool.

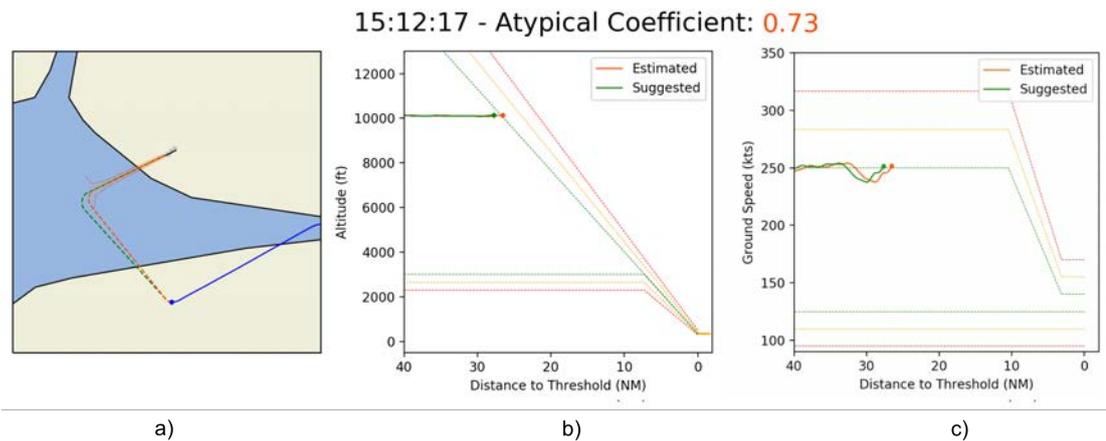


Fig. 6.8 Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The plane turns in base leg (a). Its ground speed increases (c) and it passes over the glide path (b). The tool suggests to slightly lengthen the track (in green).

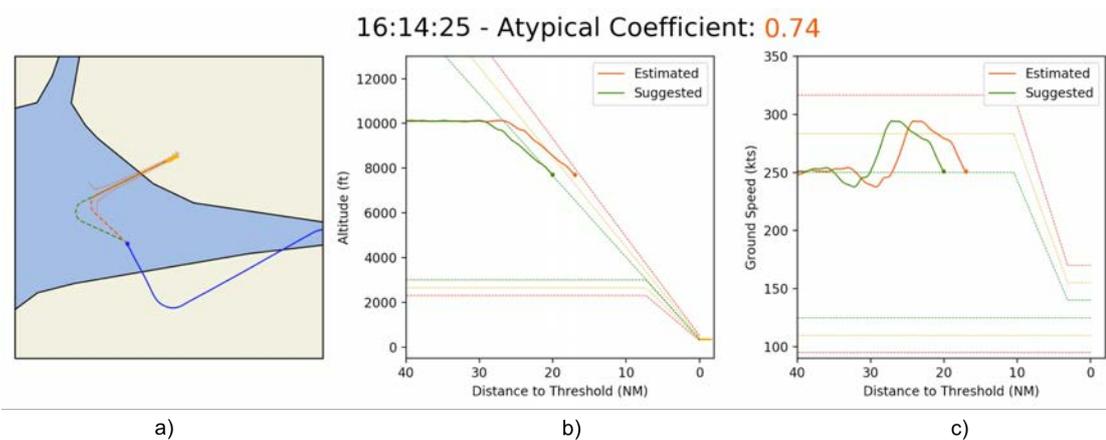


Fig. 6.9 Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The plane is on the base segment (a). Its ground speed is back to normal (c) but it is high on the glide path (b). The tool still suggests extending the trajectory (in green).

6.2.2 Pegasus Airlines flight 2193

On February 5, 2020, the Boeing 737-800 of Pegasus Airlines flight 2193 from Izmir suffered a runway overrun on landing at Sabiha-Gökçen Airport in Istanbul, Turkey and broke into three parts [116, 117]. Three people were killed, 179 injured and the plane was destroyed. It comes less than a month after another Boeing 737 from the same company, which overran the runway at the same airport.

The journey from Adnan-Menderes Airport in Izmir to Istanbul went without a hitch. Around 18:30 local time (15:30 UTC), the plane attempted to land at Istanbul-Sabiha Gökçen in heavy rain and with a strong tail wind. A thunderstorm with strong gusts of wind was crossing the area at the time of the accident [116, 117].

After what the Turkish Minister of Transport and Infrastructure described as a "hard landing", the plane failed to decelerate in time. After skidding at the eastern end of the

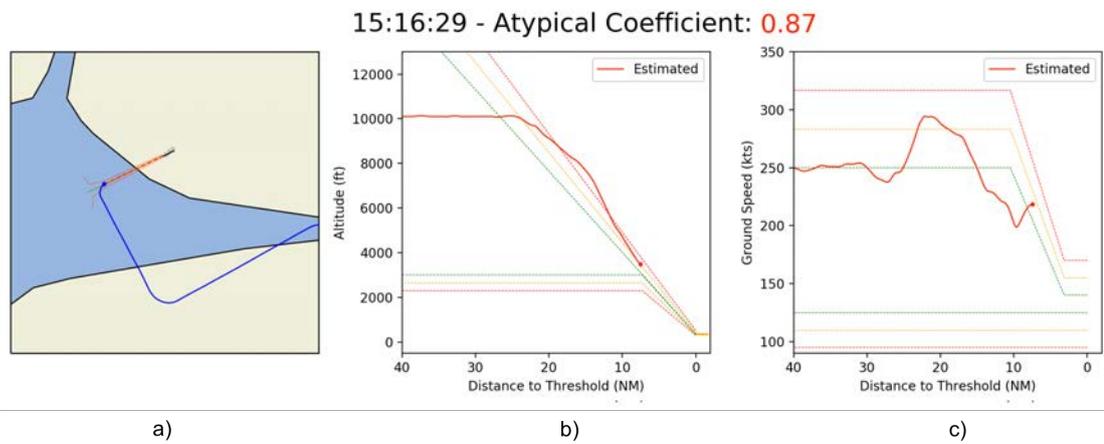


Fig. 6.10 Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The plane is on final approach (a), it is high on the glide path (b). Its ground speed is increasing (c) and its atypical score is 0.87.

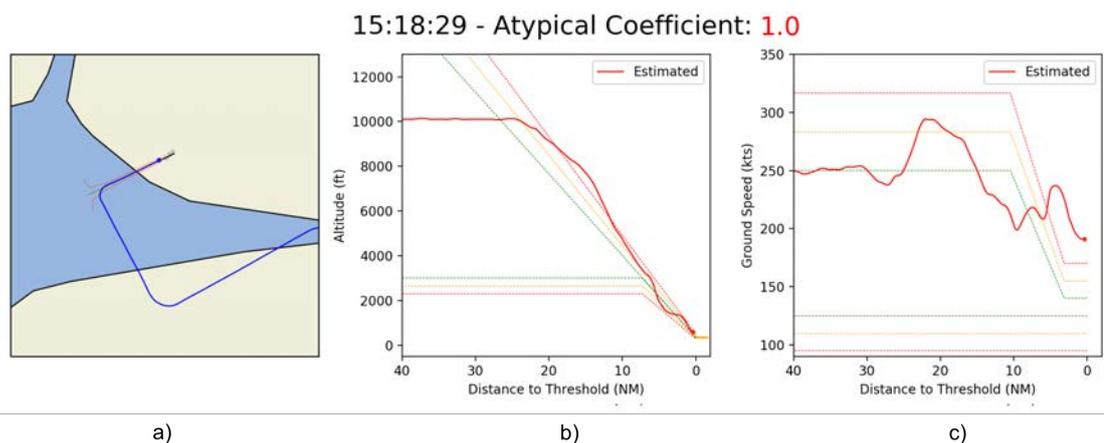


Fig. 6.11 Pegasus Airline Flight 2193 lateral, altitude and speed profiles. The aircraft is 1NM from the runway threshold. The glide path management presents large variations, its ground speed is 190kts the atypicality did not decrease throughout the final.

runway, it slid about 60 metres and fell from a 30-40 metre high embankment and split into three sections. The forward part of the fuselage was particularly damaged during the incident. The passengers escaped from the aircraft through holes between the fuselage sections. A fire broke out and was quickly extinguished by the firefighters.

This flight presents an over-energy, and raises the question of energy management upstream. To what extent could it be detected that the flight presented excess energy and could remedial trajectories be proposed, or could a go-around be suggested ? To answer this question, the algorithm presented in the previous sections has been applied to the ADS-B trajectory of this flight from flightradar24.

The flight analysis is now presented. For context, the aircraft fly over the runway threshold at 15:18:30 UTC. At 15:12:17, six minutes before, our algorithm indicates an atypical coefficient of 0.73 materialized by the red dashed trajectory as shown in Figure 6.8. The aircraft turns in base leg and passes above the standard 3° glide path with a ground speed of 250kts. The algorithm proposes a suggested trajectory (in green dashed line) by slightly moving back the point of interception of the localizer.

At 15:14:25, two minutes later, the plane is still high on the glide path, the ground speed momentarily increased before coming back to 250kts, probably due to a gust of wind. The coefficient of atypicality is still high (0.74), in the same way the algorithm proposes to lengthen the trajectory to dissipate the excess of energy as shown in Figure 6.9.

At 15:16:29, the plane is on the localizer, the algorithm switches to backup net mode. The algorithm gives an atypicality coefficient of 0.83. The plane is still a bit high on the glide path and its ground speed has just slightly increased to 215kts as shown in Figure 6.10.

Throughout the final approach, the atypicality coefficient will not fall below 0.8 and will cap at 1.0 from 1500ft to the runway threshold. It is also noted that the vertical profile shows strong variations, the aircraft momentarily passes below and then above the glide path. The ground speed is also very fluctuating up to 240kts, and finally 190kts when flying over the runway threshold as shown in Figure 6.11.

In addition, the full post-operational study of the trajectory shows a non-conformity of the approach, the trajectory has been shortened and the localizer intercepted downstream of the interception chevrons. The on-board parameters are not available but the management of the vertical profile brings the question of flight stabilisation. In any case, the flight had been showing non-negligible signs of high energy for 6 minutes. The extreme weather conditions, the wet runway, could have prompted the controllers, if they had had access to such a tool, to suggest to the pilots a go-around and thus perhaps avoid the accident.

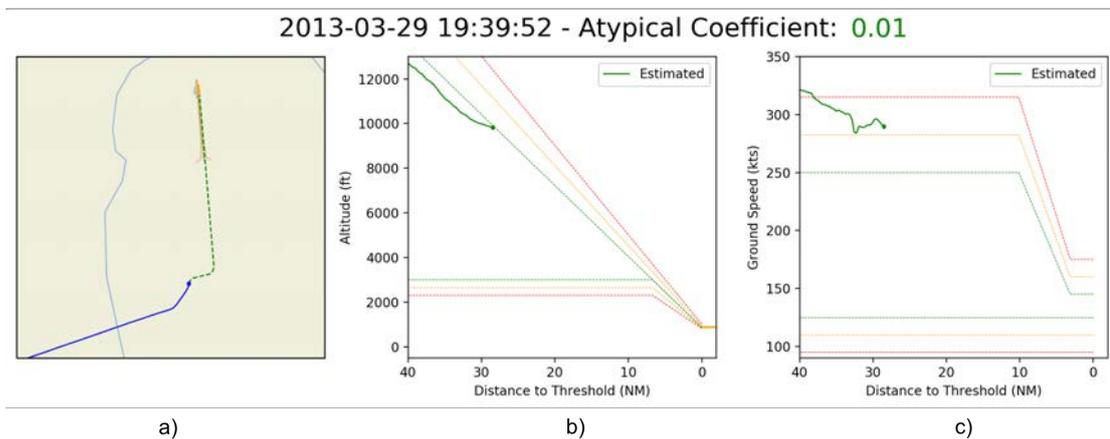


Fig. 6.12 Hermes Airline Flight 7817 lateral, altitude and speed profiles. The crew follows the localizer interception heading given by the Air Traffic Controller and requests permission to deviate its trajectory by 10° left to avoid a cloud a). The aircraft's ground speed is slightly high (c) and it is still under the glide path (b). The tool indicates that the aircraft is in a typical situation.

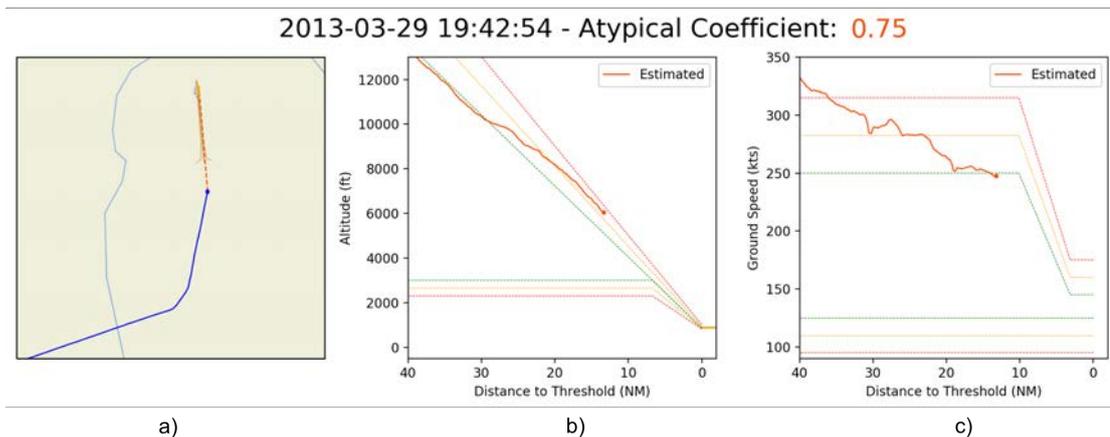


Fig. 6.13 Hermes Airline Flight 7817 lateral, altitude and speed profiles. The aircraft intercepts the runway extended centerline. The deviation of 10° right, shortened the track a), the plane is now above the glide path b). The aircraft's ground speed is around 250kts (c). The tool indicates an atypical situation due to an excess of potential energy.

6.2.3 Hermes Airlines flight 7817

The report of the French Bureau d'Enquêtes et d'Analyse (BEA) [115], in charge of investigating this crash describes the accident as follows.

The crew flew a category 1 (CAT I) ILS approach on runway 36R at Lyon Saint-Exupéry aerodrome. The weather conditions were such that low visibility operating procedures (LVP) prevailed.

As the aircraft crossed the stabilization height at 1 000 ft, the speed of the aircraft was 57 kt higher than the approach speed. At 140 ft, an inappropriate increase in thrust by autothrust keeps the aircraft at a high speed.

At 19:42:54, the plane intercepts the runway extended centerline. The deviation of 10° right, shortened the track, the plane is now above the glide path. Its ground speed is 250kts. The atypicality is 0.75 due to an excess of potential energy (see Figure 6.13).

At 19:44:22, the plane has just recovered the glide path. However, the potential energy has been transformed into kinetic energy and there is a slight increase of the ground speed towards 260kts. The atypicality increases around 0.88 due to an excess of kinetic energy (see Figure 6.14).

At 19:45:59, the aircraft is about 3NM from the runway threshold. Its atypicality peaks at 1.0 because of a very high ground speed of 210 kts, although decreasing. The aircraft is not stabilized with a very high speed and nevertheless continues its approach (see Figure 6.15).

The study based on the algorithm shows that an energy atypicality due to excessive energy appeared as early as 13NM before the runway threshold and continued throughout the final approach. The tool therefore shows the possibilities of anticipation and can be seen as a situation awareness alerting system.

6.2.4 Air India Express flight 1344

On August, 7 2002, a B737 Air India Express Flight 1344 was intended to land at Calicut International Airport [238]. The approach was for runway 28, but two landings were aborted due to tailwind, and the aircraft circled, awaiting clearance, before landing on runway 10. Due to the monsoon and flooding in Kerala at the time, poor weather conditions reduced the visibility at the time of landing to 2 000 m (6 600 ft). Runway 28 was in use, and on the first attempt to land, the pilot could not see the runway and requested runway 10. On the second attempt on Runway 10 at 2 860 m (9 380 ft), the aircraft landed near Taxiway "C", which is approximately 1 000 m (3 300 ft) beyond the runway threshold. The aircraft did not stop short of the end of the runway and plunged 9 to 10.5 m (30-35 ft) into a gorge, which split the fuselage in two at impact. A total of 184 passengers, four cabin crew and two cockpit crew were on board. Eighteen people died in the crash (16 passengers and both pilots) and more than 100 people were injured. Additionally, bad weather conditions with tail wind in final approach and a wet runway were observed.

Using the FlightRadar24 ADS-B data, the replay methodology will be applied to analyse the final attempt.

At 14:04:42, the aircraft intercepts the runway extended center line, the ground speed, the altitude profile and the atypical coefficient are nominal (see Figure 6.16).

At 14:08:02, the aircraft intercepts the glide path, the altitude profile is nominal however, its ground speed is around 200kts with no sign of reduction. The atypical coefficient has increased up to 0.97 underlying an atypical variation leading the aircraft

to high energy (see Figure 6.17).

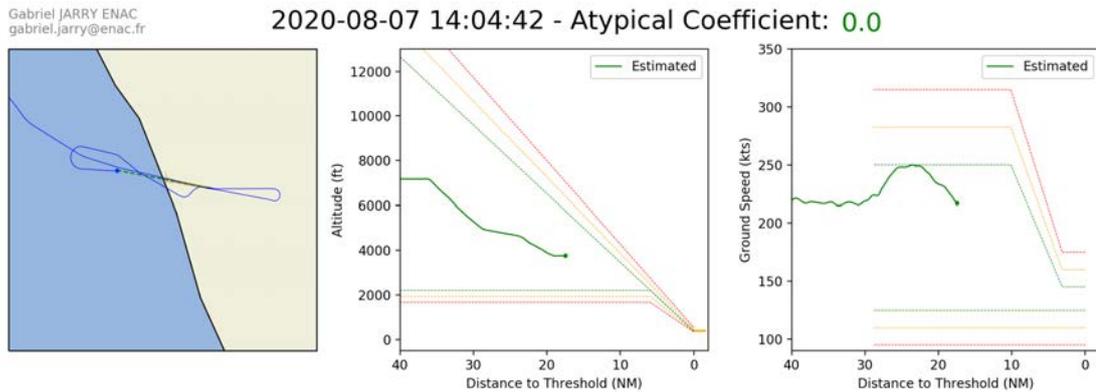


Fig. 6.16 Air India Express Flight 1344 lateral, altitude and speed profiles. The aircraft intercepts the runway extended center line (a). The aircraft's ground speed and altitude profile are nominal (a and c). The tool indicates that the aircraft is in a typical situation.

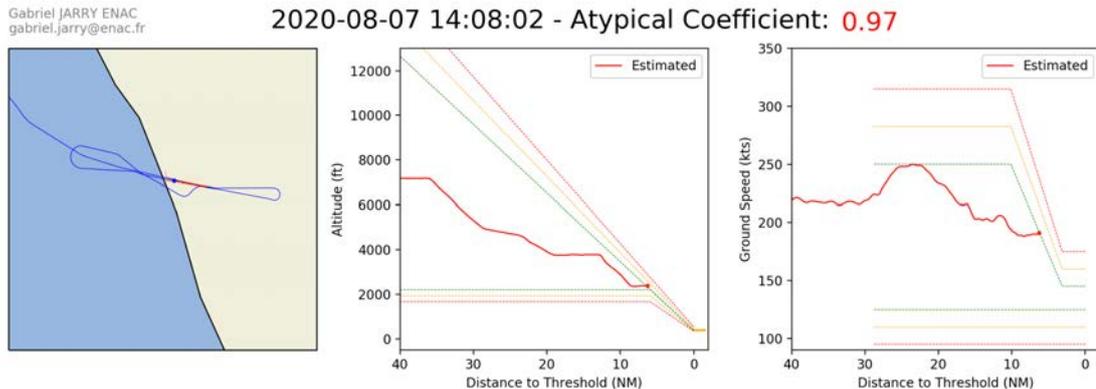


Fig. 6.17 Air India Express Flight 1344 lateral, altitude and speed profiles. The altitude profile is nominal however, its ground speed is around 200kts with no sign of reduction. The atypical coefficient has increased up to 0.97 underlying an atypical variation leading the aircraft to high energy.

At 14:08:54, the aircraft descends on the glide path, the ground speed has started to reduce down to 175kts. The atypical coefficient is back to green areas(0.09). The reduction of the energy returns the aircraft to a less atypical state (see Figure 6.18).

At 14:09:30, the ADS-B data stopped at 1.6NM to the runway threshold. The aircraft is still on the glide path. However, the ground speed is not reducing anymore and stagnates at 175kts. The atypical coefficient increased up to 0.96 again underlying a high kinetic energy (see Figure 6.18).

6.3 Discussion

With this proposal, different questions immediately come to mind and have been raised by the different operators (ATC and airlines) to whom the model has been presented.

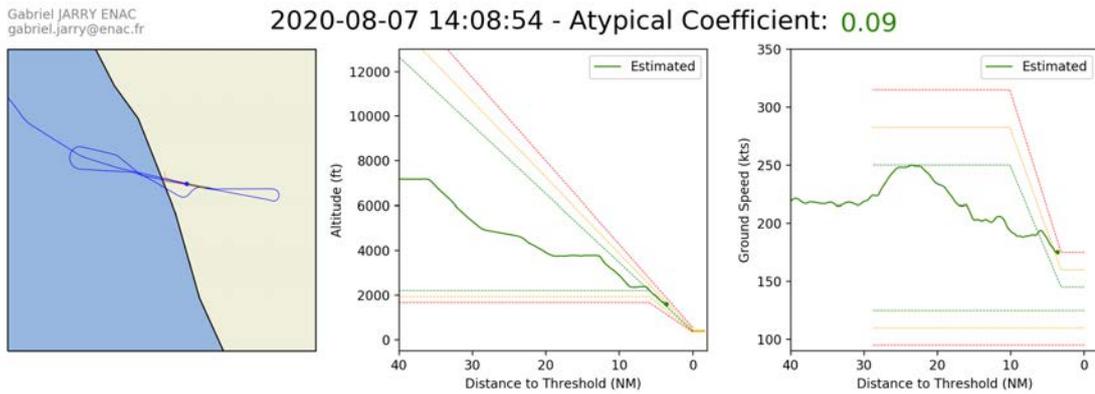


Fig. 6.18 Air India Express Flight 1344 lateral, altitude and speed profiles. The aircraft descends on the glide path, the ground speed has started to reduce down to 175kts. The atypical coefficient is back to green areas(0.09). The reduction of the energy returns the aircraft to a less atypical state

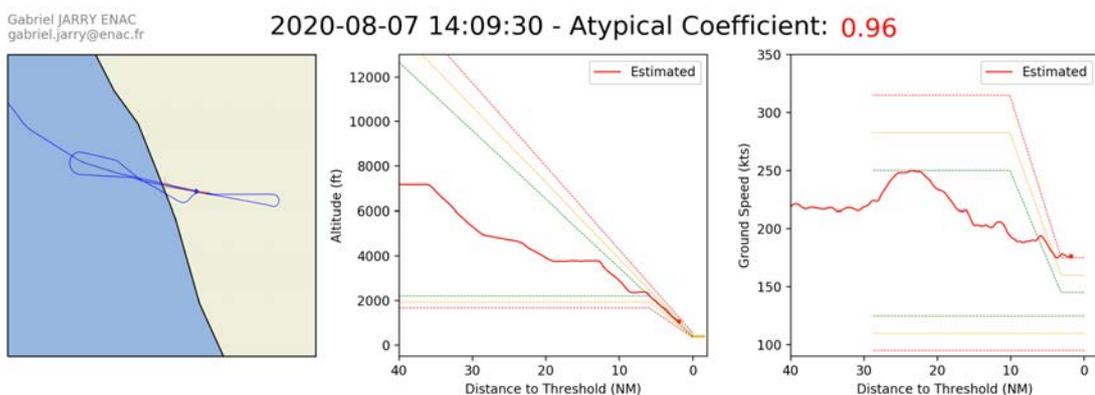


Fig. 6.19 Air India Express Flight 1344 lateral, altitude and speed profiles. The ADS-B data stopped at 1.6NM to the runway threshold. The aircraft is still on the glide path. However, the ground speed is not reducing anymore and stagnates at 175kts. The atypical coefficient increased up to 0.96 again underlying a high kinetic energy

First of all, the notion of false positives is crucial. Indeed, in the case of real time use by air traffic controllers, it is not possible for too many warnings to appear. This implies an appropriate calibration which could be done by coupling with other statistical methods. It is nevertheless important to stress that 100% of the high atypicality scores are truly atypical with respect to the learning set (low density areas). In other words, if the atypicality score is high, it implies that only very few flights in the learning set have performed the same energy management.

Concerning the energy management on approach, the work carried out previously in Chapter 3, has shown that one out of two atypical flights between 5NM and the runway threshold are unstabilized. Calibration could therefore be considered by phase of flight and be more severe when approaching the runway threshold.

Secondly, the tool is placed in the framework of artificial intelligence, which implies an elaboration of the model from a training set. On this subject, a road map has just

been published by EASA : [8]. It seems obvious that such a tool, if used in real time, must follow the guidelines of the road map to insure the trustworthiness of Artificial Intelligence. In addition, such tools could lead to legal liabilities and therefore imply the implementation of detailed operational manuals and procedures.

Nevertheless, a less critical and simpler to implement use is plebiscited by airlines in the framework of flight data analysis. Indeed, this type of tool presents a major advantage for the training and analysis of flights in replay mode. It allows operators to highlight potentially unmonitored flight events and above all is a pedagogical means of making crews aware of the stakes of energy management during approach and landing.

6.4 Conclusions

In this Chapter a tool for detecting atypical aircraft approaches was presented. The major contribution is the real-time extension of a post-operational algorithm using Dubins paths. Two underlying uses have been presented and applied to aircraft crashes. First, an estimation of the current state of the aircraft by considering that the aircraft makes a direct to the interception chevrons. Second, in the case where the previous estimate presents a high energy state, a trajectory is suggested to bring the aircraft back to a nominal energy state.

This methodology presents a major advance, giving directions to new situation awareness alerting systems. The use in real time must obviously be subject to adequate calibration and the development of operational procedures. However, a use in the framework of flight data analysis and flight safety training as well as a use by safety authorities for prevention and safety improvement is envisaged in the very short term.

Finally, future works should focus on the real time calibration of the methodology, for instance, the development of an appropriate learning process and the use of complementary methods in order to obtain a robust model meeting the safety requirement set by the EASA road-map. Other extensions could be made to improve the trajectory generation process, by extending a 3D trajectory, or by integrating the atypicality coefficient in a dedicated control model. Finally, it is also possible to consider the use of reinforcement learning to suggest an on-board trajectory that takes into account the aircraft configurations.

CHAPTER 7

Thesis Conclusions and Perspectives

7.1 Research and Operational Contributions

The research work resulting from this thesis has led to both research and operational contributions.

In Chapter 3, a post-operational methodology for aircraft atypical energy approach detection was developed. It is based on an outlier scoring from the density estimation of the Functional Principal Component Decomposition score. The whole process is applied on a sliding window to enable local and global analysis and detection of atypical energy managements. A study was led to assess the link between atypical energy managements and safety events monitored by airlines. The study underlines positive correlations and links with safety events such as non-stabilization. The methodology was applied to monitor atypical energy managements before and during the COVID-19 period, underlying an increasing ratio of atypical flights, owing to an increasing number of glide interceptions from above and high speed approaches. These results corroborate the latest figures published by International Air Transport Association (IATA) on the sharp increase in the rate of unstabilized approaches for the period of COVID-19 [119].

In Chapter 4, supervised learning techniques were investigated to enhance the analysis of ground data. The use of Generative Adversarial Networks to generate new trajectories or detect anomalies from radar data was first investigated. The anomaly detection process was compared with the methodology of Chapter 3, showing similar results. However, the GAN methodology appeared to be less flexible due to convergence problems. Besides, supervised learning techniques based on LSTM neural network were used to predict aircraft flight deck parameters from basic radar data information. Even without any mass information, the models showed accurate results. Investigation around the generalisation of these models to other aircraft types was led, underlying promising properties. The application of such kind of algorithm paved the way to new ATM system performance evaluation tools.

In Chapter 5, research studies were conducted in collaboration with ENAC colleagues working on information visualisation (InfoVis). It resulted in a pipeline using Functional Principal Component Analysis to support functional trajectory clustering,

generation, modification and visual simplification (edge bundling). This pipeline was applied to provide simulation in the context of novel aircraft procedure noise estimation. Additionally, a similarity shape based metric using FPCA was proposed and integrated into a novel trail set brushing interaction. The technique provide a refinement of the classical brushing techniques enabling a better selection of desired trajectory trail set.

Finally, In Chapter 6, a real time extension of the atypical detection methodology of Chapter 3 was proposed using Dubins Path as an estimator of the remaining distance to the runway threshold. The methodology was tested on several crashes and runway excursions. The tool provides a novel point of view in the analysis and replay of aircraft trajectory, which focuses on the energy management of the aircraft. It paves the way to new situation awareness alerting systems.

The results of this thesis were presented and promoted to different operational working group. In particular, to the European Action Plan for the Prevention of Runway Excursions group (EAPPRE). It contributed to novel recommendations regarding the energy management in approach that are to be published in October 2020. Besides, the work was presented to different Collaborative Analysis Groups (CAG) of EASA such as the Human Factors (FH CAG), Air Traffic Management (ATM CAG), Commercial Air Transport (CAT CAG), to promote situation awareness, anticipation and threat management. The last version of the EASA annual safety review [239] includes the approach path management as safety issue to key risk areas such as runway excursion or terrain collision. Finally, the EASA has shown interest to integrate and try the post-operational methodology in the context of D4S program.

7.2 Perspectives

7.2.1 Research Perspectives

Although this work has led to many advances, it also presents many opportunities and perspectives. In the following are listed the research perspectives of this work:

- The atypical detection algorithm should be conjugated with the aircraft on-board parameter estimation and classical statistical models to provide an automatic finer detection and analysis of atypical flight presenting operational risks
- GAN methodology could be extended to Wasserstein GAN to enhance the generation process.
- Flight-deck parameters prediction using LSTM network should be enhanced by integrating radar mode S parameter such as airspeed.
- The FPCA pipeline should be extended by providing user interactions to support the modifications

- The novel brushing interaction could be extended to the 3D and virtual reality framework.
- The real time methodology should be calibrated to support real time operations constraints.
- Reinforcement learning framework could be used to provide flight-deck trajectory that ensure a typical energy management considering aerodynamic configuration deployment.
- Research should be led on appropriate learning procedures in order to fit with the EASA road map toward certification of AI algorithms.

7.2.2 Operational Perspectives

Additionally, this thesis led to operational perspectives and participation.

- An initial participation in EASA Safe 360° 2020 as panel speaker in approach path management was scheduled for May 2020, but finally postponed to the fourth quarter of 2020 due to COVID-19 crisis.
- A proposal for a presentation was accepted in the Flight Safety Foundation International Air Summit Summit 2020 originally scheduled in Paris but due to COVID-19 crisis, will take place virtually on October 2020.
- Discussions for an operational development of the post-operational analysis software (TAPAS) and its real time extension have started with the DGAC.
- A project of a software (ACROPOLE) for the analysis and the evaluation of the post-operational or real time environmental impact of trajectory has started with the environmental office of the DSNA.

LIST OF PUBLICATIONS

Journal Papers

1. Gabriel Jarry, Daniel Delahaye, Florence Nicol, Eric Féron. Aircraft atypical approach detection using functional principal component analysis. *Journal of Air Transport Management*, Elsevier, 2020, 84, pp.101787
2. Gabriel Jarry, Daniel Delahaye, Eric Féron. Flight safety during Covid-19: A study of Charles de Gaulle airport atypical energy approaches. *Submitted (under revisions) to TRIPS : Transportation Impacts and Responses to the Coronavirus (COVID- 19)*
3. Gabriel Jarry, Daniel Delahaye, Stéphanne Puechmorel, Eric Féron. Real Time Aircraft Atypical Approach Detection for Air Traffic Control. *Submitted (under reviews) to Transportation Research Part C: Emerging Technologies*
4. Gabriel Jarry, Hassoumi Almoctar, Daniel Delahaye, Christophe Hurter. Interactive Trajectory Modification and Generation with FPCA. *CEAS Aeronautical Journal Special Issue to be submitted*

Conference Proceedings

1. Gabriel Jarry, Daniel Delahaye, Florence Nicol, Eric Féron. Aircraft Atypical Approach Detection using Functional Principal Component Analysis. *SID 2018, 8th SESAR Innovations Days, Dec 2018, Salzburg, Austria, Selected for special issue*
2. Gabriel Jarry, Nicolas Couellan, Daniel Delahaye. On the Use of Generative Adversarial Networks for Aircraft Trajectory Generation and Atypical Approach Detection. *EIWAC 2019:, 6th ENRI International Workshop on ATM/CNS, Oct 2019, Tokyo, Japan, Selected for book publication*

3. Gabriel Jarry, Daniel Delahaye, Eric Féron. Trajectory Approach Analysis: A Post-operational Aircraft Approach Analysis Tool. SID 2019, 9th SESAR Innovation Days, Dec 2019, Athenes, Greece
4. Almoctar Hassoumi, María Jesús Lobo, Gabriel Jarry, et al. Interactive Shape Based Brushing Technique for Trail Sets. Graphics Interface 2020 Conference Blind Submission
5. Gabriel Jarry, Daniel Delahaye, Eric Féron. Approach and landing aircraft on-board parameters estimation with LSTM networks. AIDA-AT 2020, 1st conference on Artificial Intelligence and Data Analytics in Air Transportation, Feb 2020, Singapore, Singapore. pp.ISBN: 978-1-7281-5381-0
6. Philippe Monmousseau, Gabriel Jarry, Florian Bertosio, Daniel Delahaye, Marc Houalla. Predicting Passenger Flow at Charles De Gaulle Airport Security Checkpoints. AIDA-AT 2020, 1st International Conference on Artificial Intelligence and Data Analytics for Air Transportation, Feb 2020, Singapore, Singapore.
7. Gabriel Jarry, Hassoumi Almoctar, Daniel Delahaye, Christophe Hurter. Interactive Trajectory Modification and Generation with FPCA. ICRAAT 2020, 9th International Conference for Research in Air Transportation, Jun 2020, Tampa, United States, *Best Paper Award, selected for special issue*

REFERENCES

- [1] Boeing, “Statistical summary of commercial jet airplane accidents, worldwide operations 1959 – 2018, 50th edition,” 2018. Available at https://www.boeing.com/resources/boeingdotcom/company/about_bca/pdf/statsum.pdf.
- [2] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining interesting locations and travel sequences from gps trajectories,” in *Proceedings of the 18th International Conference on World Wide Web, WWW '09*, (New York, NY, USA), pp. 791–800, ACM, 2009.
- [3] DGAC, “Safety state program, horizon 2023,” 2019. Available at https://www.ecologique-solidaire.gouv.fr/sites/default/files/DSAC_PlanHorizon_2023_FR.pdf.
- [4] F. Jackman, “Nearly Half of Commercial Jet Accidents Occur During Final Approach, Landing,” Nov. 2014.
- [5] Centro de Publicaciones, Ministerio de Fomento, “Report A-029/2011, Accident Involving a Bombardier CL-600-2b19 (CRJ200), Registration EC-ITU, Operated by Air Nostrum, at the Barcelona Airport, on 30 July 2011,” tech. rep., Comisión de Investigación de Accidentes e Incidentes de Aviación Civil, Madrid, 2013. Available at https://www.fomento.gob.es/NR/rdonlyres/0E877F5B-5703-4AF3-95D9-3C6B1B8A2899/118577/2011_029_A_ENG.pdf.
- [6] C. A. Hart, R. L. Sumwalt, M. R. Rosekind, and E. F. Weener, “Descent Below Visual Glidepath and Impact With Seawall Asiana Airlines Flight 214 Boeing 777-200er, HL7742 San Francisco, California July 6, 2013,” tech. rep., National Transportation Safety Board, 2014. Available at <https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR1401.pdf>.
- [7] F. S. Foundation, “Foundation Releases Go-Around Execution and Decision-Making Project Final Report,” Mar. 2017. Available at <https://flightsafety.org/go-around-final-report-public/>.

- [8] EASA, “EASA Artificial Intelligence Roadmap 1.0 published — A human-centric approach to AI in aviation,” 2020. Available at <https://www.easa.europa.eu/newsroom-and-events/news/easa-artificial-intelligence-roadmap-10-published>.
- [9] European Aviation Artificial Intelligence High-Level Group, “FLY AI REPORT,” tech. rep., Jan. 2020. Available at <https://www.eurocontrol.int/sites/default/files/2020-03/eurocontrol-fly-ai-report-032020.pdf>.
- [10] ICAO, “Exploitation Technique Des Aéronefs, Volume II: Construction Des Procédure de Vol à Vue et de Vol Aux Instruments,” 2014.
- [11] ICAO, “Annexe 13 à La Convention Relative à l’aviation Civile Internationale, Enquête Sur Les Accidents et Incidents d’aviation,” 2010. Available at https://www.icao.int/WACAF/Documents/Meetings/2015/ICAO-BEA/an13_cons_fr.pdf.
- [12] A. Vernay, “Defining a Compliant Approach (CA): A joint response to enhance the safety level of approach and landing,” *Hindsight 17*, pp. 44–46, 2013. Available at <https://skybrary.aero/bookshelf/books/2348.pdf>.
- [13] SKYbrary, “Short Vectoring and Glideslope Interception from Above: Guidance for Controllers - SKYbrary Aviation Safety,” 2019. Available at https://www.skybrary.aero/index.php/Short_Vectoring_and_Glideslope_Interception_from_Above:_Guidance_for_Controllers.
- [14] SKYbrary, “Energy Management during Approach - SKYbrary Aviation Safety,” 2019. Available at https://www.skybrary.aero/index.php/Energy_Management_during_Approach.
- [15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, Springer New York.
- [16] V. Vapnik, *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control, Wiley.
- [17] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer New York.
- [18] B. W. Silverman, *Density estimation for statistics and data analysis*, vol. 26. CRC press, 1986.

- [19] T. K. Moon, “The expectation-maximization algorithm,” *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.
- [20] C. Hurter, O. Ersoy, and A. Telea, “Graph bundling by kernel density estimation,” in *Computer Graphics Forum*, vol. 31, pp. 865–874, Wiley Online Library, 2012.
- [21] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” tech. rep., Stanford, 2006.
- [22] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, vol. 96, pp. 226–231, 1996.
- [23] R. J. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172, Springer, 2013.
- [24] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: ordering points to identify the clustering structure,” *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.
- [25] G. E. Hinton, “Connectionist learning procedures,” in *Machine learning*, pp. 555–610, Elsevier, 1990.
- [26] D. Delahaye, S. Puechmorel, S. Alam, and E. Féron, “Trajectory mathematical distance applied to airspace major flows extraction,” in *ENRI International Workshop on ATM/CNS*, pp. 51–66, Springer, 2017.
- [27] G. Andrienko, N. Andrienko, G. Fuchs, and J. M. C. Garcia, “Clustering trajectories by relevant parts for air traffic analysis,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 34–44, 2017.
- [28] M. Gariel, A. N. Srivastava, and E. Feron, “Trajectory clustering and an application to airspace monitoring,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1511–1524, 2011.
- [29] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [30] I. Jolliffe, *Principal component analysis*. Springer, 2011.
- [31] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” in *Linear Algebra*, pp. 134–151, Springer, 1971.

- [32] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.
- [33] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [34] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [35] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
- [36] S. Agrawal and J. Agrawal, “Survey on anomaly detection using data mining techniques,” *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.
- [37] M. Goldstein and S. Uchida, “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,” *PloS one*, vol. 11, no. 4, p. e0152173, 2016.
- [38] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, “Informal identification of outliers in medical data,” in *Fifth international workshop on intelligent data analysis in medicine and pharmacology*, vol. 1, pp. 20–24, 2000.
- [39] F. E. Grubbs, “Procedures for detecting outlying observations in samples,” *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.
- [40] W. Stefansky, “Rejecting outliers in factorial designs,” *Technometrics*, vol. 14, no. 2, pp. 469–479, 1972.
- [41] B. Rosner, “Percentage points for a generalized esd many-outlier procedure,” *Technometrics*, vol. 25, no. 2, pp. 165–172, 1983.
- [42] E. Eskin, “Anomaly detection over noisy data using learned probability distributions,” 2000.
- [43] S. Byers and A. E. Raftery, “Nearest-neighbor clutter removal for estimating features in spatial point processes,” *Journal of the American Statistical Association*, vol. 93, no. 442, pp. 577–584, 1998.
- [44] A. Kind, M. P. Stoecklin, and X. Dimitropoulos, “Histogram-based traffic anomaly detection,” *IEEE Transactions on Network and Service Management*, vol. 6, no. 2, pp. 110–121, 2009.

- [45] D. Endler, “Intrusion detection. applying machine learning to solaris audit data,” in *Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217)*, pp. 268–279, IEEE, 1998.
- [46] R. Laxhammar, G. Falkman, and E. Sviestins, “Anomaly detection in sea traffic—a comparison of the gaussian mixture model and the kernel density estimator,” in *2009 12th International Conference on Information Fusion*, pp. 756–763, IEEE, 2009.
- [47] E. Schubert, A. Zimek, and H.-P. Kriegel, “Generalized outlier detection with flexible kernel density estimates,” in *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 542–550, SIAM, 2014.
- [48] M. Desforges, P. Jacob, and J. Cooper, “Applications of probability density estimation to the detection of abnormal conditions in engineering,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 212, no. 8, pp. 687–703, 1998.
- [49] S. D. Bay and M. Schwabacher, “Mining distance-based outliers in near linear time with randomization and a simple pruning rule,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 29–38, 2003.
- [50] G. Münz, S. Li, and G. Carle, “Traffic anomaly detection using k-means clustering,” in *GI/ITG Workshop MMBnet*, pp. 13–14, 2007.
- [51] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, “K-means+ id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods,” *IEEE transactions on knowledge and data engineering*, vol. 19, no. 3, pp. 345–354, 2007.
- [52] R. J. Bolton, D. J. Hand, *et al.*, “Unsupervised profiling methods for fraud detection,” *Credit scoring and credit control VII*, pp. 235–255, 2001.
- [53] L. D. Baker, T. Hofmann, A. McCallum, and Y. Yang, “A hierarchical probabilistic model for novelty detection in text,” in *Proceedings of the International Conference on Machine Learning*, 1999.
- [54] L. Khan, M. Awad, and B. Thuraisingham, “A new intrusion detection system using support vector machines and hierarchical clustering,” *The VLDB journal*, vol. 16, no. 4, pp. 507–521, 2007.

- [55] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *IEEE International Conference on Image Processing 2005*, vol. 2, pp. II-602, IEEE, 2005.
- [56] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 109-120, 2007.
- [57] D. Brauckhoff, K. Salamatian, and M. May, "Applying pca for traffic anomaly detection: Problems and solutions," in *IEEE INFOCOM 2009*, pp. 2866-2870, IEEE, 2009.
- [58] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft, "In-network pca and anomaly detection," in *Advances in Neural Information Processing Systems*, pp. 617-624, 2007.
- [59] Y. Liu, L. Zhang, and Y. Guan, "Sketch-based streaming pca algorithm for network-wide traffic anomaly detection," in *2010 IEEE 30th International Conference on Distributed Computing Systems*, pp. 807-816, IEEE, 2010.
- [60] L. Li, S. Das, R. John Hansman, R. Palacios, and A. N. Srivastava, "Analysis of flight data using clustering techniques for detecting abnormal operations," *Journal of Aerospace information systems*, vol. 12, no. 9, pp. 587-598, 2015.
- [61] L. Li, R. J. Hansman, R. Palacios, and R. Welsch, "Anomaly detection via a gaussian mixture model for flight operation and safety monitoring," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 45-57, 2016.
- [62] L. Li, M. Gariel, R. J. Hansman, and R. Palacios, "Anomaly detection in onboard-recorded flight data using cluster analysis," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, pp. 4A4-1, IEEE, 2011.
- [63] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [64] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63-71, Springer, 2003.
- [65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097-1105, 2012.

- [66] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649, IEEE, 2013.
- [67] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.
- [68] X. J. Zhu, “Semi-supervised learning literature survey,” tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2005.
- [69] V. Roth, “Outlier detection with one-class kernel fisher discriminants,” in *Advances in Neural Information Processing Systems*, pp. 1169–1176, 2005.
- [70] M. Sakurada and T. Yairi, “Anomaly detection using autoencoders with nonlinear dimensionality reduction,” in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, p. 4, ACM, 2014.
- [71] C. Zhou and R. C. Paffenroth, “Anomaly detection with robust deep autoencoders,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, ACM, 2017.
- [72] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient gan-based anomaly detection,” *arXiv preprint arXiv:1802.06222*, 2018.
- [73] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “Ganomaly: Semi-supervised anomaly detection via adversarial training,” in *Asian conference on computer vision*, pp. 622–637, Springer, 2018.
- [74] B. J. B. Rani *et al.*, “Survey on applying gan for anomaly detection,” in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, IEEE, 2020.
- [75] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, “Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 47–56, ACM, 2010.
- [76] R. V. Jategaonkar and F. Thielecke, “Aircraft parameter estimation—a tool for development of aerodynamic databases,” *Sadhana*, vol. 25, no. 2, pp. 119–135, 2000.
- [77] B. P. COLLINS, “Estimation of aircraft fuel consumption,” *Journal of Aircraft*, vol. 19, no. 11, pp. 969–975, 1982.

- [78] D. Delahaye and S. Puechmorel, “Tas and wind estimation from radar data,” in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, pp. 2–B, IEEE, 2009.
- [79] D. Delahaye and S. Puechmorel, “Aircraft local wind estimation from radar tracker data,” in *2008 10th International Conference on Control, Automation, Robotics and Vision*, pp. 1033–1038, IEEE, 2008.
- [80] J. Sun, J. Ellerbroek, and J. M. Hoekstra, “Aircraft initial mass estimation using bayesian inference method,” *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 59–73, 2018.
- [81] Y. S. Chati and H. Balakrishnan, “A gaussian process regression approach to model aircraft engine fuel flow rate,” in *Proceedings of the 8th International Conference on Cyber-Physical Systems*, pp. 131–140, ACM, 2017.
- [82] Y. S. Chati and H. Balakrishnan, “Statistical modeling of aircraft engine fuel flow rate,” in *30th Congress of the International Council of the Aeronautical Science*, 2016.
- [83] A. A. Trani, F. Wing-Ho, G. Schilling, H. Baik, and A. Seshadri, “A Neural Network Model to Estimate Aircraft Fuel Consumption,” 2004.
- [84] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [85] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge, 1998.
- [86] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [87] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [88] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, “Mastering the game of go with deep neural networks and tree search,” *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

- [89] O. Vinyals, I. Babuschkin, J. Chung, M. Mathieu, M. Jaderberg, W. M. Czarnecki, A. Dudzik, A. Huang, P. Georgiev, R. Powell, *et al.*, “Alphastar: Mastering the real-time strategy game starcraft ii,” *DeepMind blog*, p. 2, 2019.
- [90] A. L. Bazzan, “Opportunities for multiagent systems and multiagent reinforcement learning in traffic control,” *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 3, p. 342, 2009.
- [91] A. M. F. Crespo, L. Weigang, and A. G. de Barros, “Reinforcement learning agents to tactical air traffic flow management,” *International Journal of Aviation Management*, vol. 1, no. 3, pp. 145–161, 2012.
- [92] L. L. Cruciol, A. C. de Arruda Jr, L. Weigang, L. Li, and A. M. Crespo, “Reward functions for learning to control in air traffic flow management,” *Transportation Research Part C: Emerging Technologies*, vol. 35, pp. 141–155, 2013.
- [93] M. Brittain and P. Wei, “Autonomous air traffic controller: A deep multi-agent reinforcement learning approach,” *arXiv preprint arXiv:1905.01303*, 2019.
- [94] J. O. Ramsay, “Functional data analysis,” *Encyclopedia of Statistical Sciences*, vol. 4, 2004.
- [95] J.-C. Deville, “Méthodes statistiques et numériques de l’analyse harmonique,” in *Annales de l’INSEE*, pp. 3–101, JSTOR, 1974.
- [96] J. Dauxois, A. Pousse, and Y. Romain, “Asymptotic theory for the principal component analysis of a vector random function: some applications to statistical inference,” *Journal of multivariate analysis*, vol. 12, no. 1, pp. 136–154, 1982.
- [97] J. Dauxois and A. Pousse, “Les analyses factorielles en calcul des probabilités et en statistique: essai d’étude synthétique,” *These d’état, Université Paul Sabatier, Toulouse*, 1976.
- [98] J. O. Ramsay and B. W. Silverman, *Applied functional data analysis: methods and case studies*. Springer, 2007.
- [99] J. O. Ramsay, G. Hooker, and S. Graves, *Functional Data Analysis with R and MATLAB. Use R!*, Springer-Verlag.
- [100] F. Ferraty and P. Vieu, *Nonparametric functional data analysis: theory and practice*. Springer Science and Business Media, 2006.
- [101] F. Ferraty, *Recent advances in functional data analysis and related topics*. Springer Science and Business Media, 2011.

- [102] S. Ullah and C. F. Finch, “Applications of functional data analysis: A systematic review,” vol. 13, no. 1.
- [103] B. Gregorutti, *Forêts aléatoires et sélection de variables: analyse des données des enregistreurs de vol pour la sécurité aérienne*. PhD thesis, 2015.
- [104] R. Suyundikov, S. Puechmorel, and L. Ferré, “Multivariate functional data clusterization by pca in sobolev space using wavelets,” 2010.
- [105] K. Tastambekov, *Aircraft trajectory prediction by local functional regression*. PhD thesis, Toulouse, INSA, 2012.
- [106] J. Ramsay and B. Silverman, *Functional Data Analysis*. Springer Series in Statistics, Springer, 2005.
- [107] C. Hurter, S. Puechmorel, F. Nicol, and A. Telea, “Functional decomposition for bundled simplification of trail sets,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 500–510, 2017.
- [108] G. Jarry, D. Delahaye, F. Nicol, and E. Féron, “Aircraft atypical approach detection using functional principal component analysis,” in *SESAR Innovations Days 2018*, 2018.
- [109] G. Jarry, D. Delahaye, F. Nicol, and E. Feron, “Aircraft atypical approach detection using functional principal component analysis,” *Journal of Air Transport Management*, vol. 84, p. 101787, 2020.
- [110] G. Jarry, D. Delahaye, and E. Feron, “Trajectory approach analysis: A post-operational aircraft approach analysis tool,” 2019.
- [111] G. Jarry, D. Delahaye, and E. Feron, “Flight safety during covid-19: A study of charles de gaulle airport atypical energy approaches,” *Transportation research interdisciplinary perspectives*, vol. 9, p. 100327, 2021.
- [112] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.
- [113] L. A. Goodman and W. H. Kruskal, “Measures of association for cross classifications,” in *Measures of association for cross classifications*, pp. 2–34, Springer, 1979.
- [114] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

- [115] B. d’Enquêtes et Analyses, “Rapport accident survenu le 29 mars 2013 sur l’aéroport de lyon saint-exupéry (69) à l’airbus a321 immatriculé sx-bhs exploité par hermes airlines affrété par air méditerranée,” tech. rep., Bureau d’Enquêtes et d’Analyses, 2015. Available at <https://www.bea.aero/fileadmin/documents/docspa/2013/sx-s130329/pdf/sx-s130329.pdf>.
- [116] “Three dead, 180 hurt as plane skids off runway,” *BBC News*, Feb. 2020. Available at <https://www.bbc.com/news/world-europe-51384667>.
- [117] “Pegasus Airlines flight 2193 overruns runway in Istanbul,” Feb. 2020. Available at : <https://www.flightradar24.com/blog/pegasus-airlines-flight-2193-overruns-runway-in-istanbul/>.
- [118] T. D. S. Board, “Crashed during approach, boeing 737-800, near amsterdam schiphol airport, 25 february 2009,” tech. rep., The Dutch Safety Board, 2009. Available at https://catsr.vse.gmu.edu/SYST460/TA1951_AccidentReport.pdf.
- [119] IATA, “Unstable approaches during reduced operation,” 2020. Available at https://www.iata.org/contentassets/e5bc94292b44433ba852925ee9ac47bb/on_002_20_ua_during_reduced_operations.pdf.
- [120] G. Jarry, N. Couellan, and D. Delahaye, “On the use of generative adversarial networks for aircraft trajectory generation and atypical approach detection,” 2019.
- [121] G. Jarry, D. Delahaye, and E. Feron, “Approach and landing aircraft on-board parameters estimation with lstm networks,” in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–6, IEEE, 2020.
- [122] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’14, (Cambridge, MA, USA), pp. 2672–2680, MIT Press, 2014.
- [123] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 613–621, Curran Associates, Inc., 2016.

- [124] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *International Conference on Learning Representations*, 2019.
- [125] E. Putin, A. Asadulaev, Q. Vanhaelen, Y. Ivanenkov, A. V. Aladinskaya, A. Aliper, and A. Zhavoronkov, “Adversarial threshold neural computer for molecular de novo design,” *Molecular Pharmaceutics*, vol. 15, no. 10, pp. 4386–4397, 2018. PMID: 29569445.
- [126] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 1060–1069, PMLR, 20–22 Jun 2016.
- [127] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” *CoRR*, vol. abs/1703.05921, 2017.
- [128] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [129] N. Hirose, A. Sadeghian, P. Goebel, and S. Savarese, “To go or not to go? a near unsupervised learning approach for robot navigation,” *CoRR*, vol. abs/1709.05439, 2017.
- [130] W. Ding, W. Wang, and D. Zhao, “Multi-Vehicle Trajectories Generation for Vehicle-to-Vehicle Encounters,” in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [131] I. J. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *CoRR*, vol. abs/1701.00160, 2017.
- [132] D. Poles, A. Nuic, and V. Mouillet, “Advanced aircraft performance modeling for atm: Analysis of bada model capabilities,” in *29th Digital Avionics Systems Conference*, pp. 1–D, IEEE, 2010.
- [133] A. Nuic, D. Poles, and V. Mouillet, “Bada: An advanced aircraft performance model for present and future atm systems,” *International journal of adaptive control and signal processing*, vol. 24, no. 10, pp. 850–866, 2010.
- [134] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980.

- [135] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” *ArXiv*, vol. abs/1701.07875, 2017.
- [136] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, “LSTM-based Flight Trajectory Prediction,” pp. 1–8, July 2018.
- [137] H. Zhang and T. Zhu, “Aircraft Hard Landing Prediction Using LSTM Neural Network,” in *Proceedings of the 2Nd International Symposium on Computer Science and Intelligent Control, ISCSIC '18*, (New York, NY, USA), pp. 28:1–28:5, ACM, 2018.
- [138] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, “The elements of statistical learning: data mining, inference and prediction,” *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.
- [139] V. Vapnik, *The nature of statistical learning theory*. Springer science and business media, 2013.
- [140] Y. Bengio, P. Simard, P. Frasconi, *et al.*, “Learning long-term dependencies with gradient descent is difficult,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [141] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” 1999.
- [142] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [143] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [144] G. Jarry, H. Almoctar, D. Delahaye, and C. Hurter, “Interactive trajectory modification and generation with fpca,” in *ICRAT 2020, 9th International Conference for Research in Air Transportation*, 2020.
- [145] A. Hassoumi, M. J. Lobo, G. Jarry, V. Peysakhovich, and C. Hurter, “Interactive shape based brushing technique for trail sets,” 2019.
- [146] A. Lhuillier, C. Hurter, and A. Telea, “State of the art in edge and trail bundling techniques,” *Comput. Graph. Forum*, vol. 36, pp. 619–645, June 2017.
- [147] C. Hurter, S. Puechmorel, F. Nicol, and A. Telea, “Functional decomposition for bundled simplification of trail sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 500–510, Jan 2018.

- [148] N. Adrienko and G. Adrienko, “Spatial generalization and aggregation of massive movement data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 205–219, Feb 2011.
- [149] R. Scheepens, N. Willems, H. van de Wetering, and J. J. van Wijk, “Interactive visualization of multivariate trajectory data with density maps,” in *2011 IEEE Pacific Visualization Symposium*, pp. 147–154, March 2011.
- [150] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, “Trajectory pattern mining,” in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’07, (New York, NY, USA), pp. 330–339, ACM, 2007.
- [151] P. Kalnis, N. Mamoulis, and S. Bakiras, “On discovering moving clusters in spatio-temporal data,” in *Advances in Spatial and Temporal Databases* (C. Bauzer Medeiros, M. J. Egenhofer, and E. Bertino, eds.), (Berlin, Heidelberg), pp. 364–381, Springer Berlin Heidelberg, 2005.
- [152] X. Olive and P. Bieber, “Quantitative assessments of runway excursion precursors using mode s data,” 06 2018.
- [153] D. Holten, “Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 741–748, 2006.
- [154] M. Brehmer and T. Munzner, “A multi-level typology of abstract visualization tasks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013.
- [155] C. Hurter, A. Telea, and O. Ersoy, “MoleView: An Attribute and Structure-Based Semantic Lens for Large Element-Based Plots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2600–2609, 2011.
- [156] C. Tominski, J. Abello, F. van Ham, and H. Schumann, “Fisheye tree views and lenses for graph visualization,” in *Proc. International Conference on Information Visualisation (IV)*, pp. 17–24, 2006.
- [157] A. Lambert, D. Auber, and G. Melancon, “Living flows: Enhanced exploration of edge-bundled graphs based on GPU-intensive edge rendering,” in *Proc. 14th International Conference on Information Visualisation*, pp. 523–530, 2010.
- [158] N. Wong and S. Carpendale, “Supporting interactive graph exploration using edge plucking,” in *Proc. SPIE Visualization and Data Analysis*, vol. 6495, pp. 235–246, 2007.

- [159] N. H. Riche, T. Dwyer, B. Lee, and S. Carpendale, “Exploring the design space of interactive link curvature in network diagrams,” in *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI)*, pp. 506–513, ACM, 2012.
- [160] S.-J. Luo, C.-L. Liu, B.-Y. Chen, and K.-L. Ma, “Ambiguity-free edge-bundling for interactive graph visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 810–821, 2012.
- [161] J. Brosz, M. A. Nacenta, R. Pusch, S. Carpendale, and C. Hurter, “Transmogrification: Causal manipulation of visualizations,” in *Proc. of the 26th ACM Symposium on User Interface Software and Technology (UIST)*, pp. 97–106, 2013.
- [162] J. D. Cryer, *Time Series Analysis*. Belmont, CA, USA: Wadsworth Publ. Co., 1st ed., 1986.
- [163] G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, and S. Wrobel, “A conceptual framework and taxonomy of techniques for analyzing movement,” *Journal of Visual Languages and Computing*, vol. 22, no. 3, pp. 213 – 232, 2011. Special Issue on Visual Analytics and Visual Semantics.
- [164] G. Andrienko and N. Andrienko, “A general framework for using aggregation in visual exploration of movement data,” *The Cartographic Journal*, vol. 47, no. 1, pp. 22–40, 2010.
- [165] K. Beard, H. Deese, and N. R. Pettigrew, “A framework for visualization and exploration of events,” *Information Visualization*, vol. 7, no. 2, pp. 133–151, 2008.
- [166] J. F. Roddick and M. Spiliopoulou, “A survey of temporal knowledge discovery paradigms and methods,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, pp. 750–767, July 2002.
- [167] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale, “A descriptive framework for temporal data visualizations based on generalized space-time cubes,” in *Computer Graphics Forum*, vol. 36, pp. 36–61, Wiley Online Library, 2017.
- [168] H. Hochheiser and B. Shneiderman, “Interactive exploration of time series data,” in *Proceedings of the 4th International Conference on Discovery Science, DS ’01*, (Berlin, Heidelberg), pp. 441–446, Springer-Verlag, 2001.
- [169] P. Buono, A. Aris, C. Plaisant, A. Khella, and B. Shneiderman, “Interactive pattern search in time series,” in *Visualization and Data Analysis*, 2005.

- [170] R. Scheepens, C. Hurter, H. Van De Wetering, and J. J. Van Wijk, “Visualization, selection, and analysis of traffic flows,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 379–388, Jan 2016.
- [171] G. Andrienko, N. Andrienko, and M. Heurich, “An event-based conceptual model for context-aware movement analysis,” *International Journal of Geographical Information Science*, vol. 25, no. 9, pp. 1347–1370, 2011.
- [172] P. L. C. author, S. Imfeld, and R. Weibel, “Discovering relative motion patterns in groups of moving point objects,” *International Journal of Geographical Information Science*, vol. 19, no. 6, pp. 639–668, 2005.
- [173] S. Dodge, R. Weibel, and E. Forootan, “Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects,” *Computers, Environment and Urban Systems*, vol. 33, no. 6, pp. 419 – 434, 2009. Spatial Data Mining–Methods and Applications.
- [174] C. Hurter, “Image-based visualization: Interactive multidimensional data exploration,” *Synthesis Lectures on Visualization*, vol. 3, no. 2, pp. 1–127, 2015.
- [175] C. Hurter, S. Conversy, D. Gianazza, and A. Telea, “Interactive image-based information visualization for aircraft trajectory analysis,” *Transportation Research Part C: Emerging Technologies*, vol. 47, pp. 207 – 227, 2014. Special Issue: Emerging Technologies Special Issue of ICTIS 2013 – Guest Editors: Liping Fu and Ming Zhong and Special Issue: Visualization and Visual Analytics in Transportation – Guest Editors: Patricia S. Hu and Michael L. Pack.
- [176] C. Hurter, B. Tissoires, and S. Conversy, “Fromdady: Spreading aircraft trajectories across views to support iterative queries,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1017–1024, Nov. 2009.
- [177] C. Hurter, N. H. Riche, S. M. Drucker, M. Cordeil, R. Alligier, and R. Vuillemot, “Fiberclay: Sculpting three dimensional trajectories to reveal structural insights,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, pp. 704–714, Jan 2019.
- [178] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley and Sons, 2007.
- [179] P. Bickel, P. Diggle, S. Fienberg, U. Gather, I. Olkin, and S. Zeger, “Springer series in statistics,” 2009.
- [180] A. Hinneburg and D. A. Keim, “Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering,” 1999.

- [181] R. A. Becker and W. S. Cleveland, “Brushing scatterplots,” *Technometrics*, vol. 29, pp. 127–142, May 1987.
- [182] S. K. Card, J. D. Mackinlay, and B. Shneiderman, eds., *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [183] M. O. Ward, “Xmdvtool: Integrating multiple methods for visualizing multivariate data,” in *Proceedings of the Conference on Visualization '94, VIS '94*, (Los Alamitos, CA, USA), pp. 326–333, IEEE Computer Society Press, 1994.
- [184] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, “Vega-lite: A grammar of interactive graphics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, pp. 341–350, Jan. 2017.
- [185] M. Bostock, V. Ogievetsky, and J. Heer, “D3 data-driven documents,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 2301–2309, Dec. 2011.
- [186] S. K. Card, J. D. Mackinlay, and B. Shneiderman, eds., *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [187] A. Gennady, A. Natalia, B. Peter, K. Daniel, and W. Stefan, “Visual analytics of movement,” (Berlin, Heidelberg), Springer, 2013.
- [188] B. Shneiderman, “Dynamic queries for visual information seeking,” *IEEE Softw.*, vol. 11, pp. 70–77, Nov. 1994.
- [189] B. Shneiderman, “Human-computer interaction,” ch. Direct Manipulation: A Step Beyond Programming Languages, pp. 461–467, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.
- [190] M. Correll and M. Gleicher, “The semantics of sketch: Flexibility in visual query systems for time series data,” in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 131–140, Oct 2016.
- [191] S. Owada, F. Nielsen, and T. Igarashi, “Volume catcher,” in *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games, I3D '05*, (New York, NY, USA), pp. 111–116, ACM, 2005.
- [192] D. Akers, “Cinch: A cooperatively designed marking interface for 3d pathway selection,” in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, UIST '06*, (New York, NY, USA), pp. 33–42, ACM, 2006.

- [193] P. K. Muthumanickam, K. Vrotsou, M. Cooper, and J. Johansson, “Shape grammar extraction for efficient query-by-sketch pattern matching in long time series,” in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 121–130, Oct 2016.
- [194] C. Holz and S. Feiner, “Relaxed selection techniques for querying time-series graphs,” in *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology, UIST '09*, (New York, NY, USA), pp. 213–222, ACM, 2009.
- [195] J. Illingworth and J. Kittler, “A survey of the hough transform,” *Computer vision, graphics, and image processing*, vol. 44, no. 1, pp. 87–116, 1988.
- [196] D. Akers, A. Sherbondy, R. Mackenzie, R. Dougherty, and B. Wandell, “Exploration of the brain’s white matter pathways with dynamic queries,” in *Proceedings of the Conference on Visualization '04, VIS '04*, (Washington, DC, USA), pp. 377–384, IEEE Computer Society, 2004.
- [197] H. Hauser, F. Ledermann, and H. Doleisch, “Angular brushing of extended parallel coordinates,” in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pp. 127–130, Oct 2002.
- [198] M. Nielsen, N. Elmqvist, and K. Grønbaek, “Scribble query: Fluid touch brushing for multivariate data visualization,” in *Proceedings of the 28th Australian Conference on Computer-Human Interaction, OzCHI '16*, (New York, NY, USA), pp. 381–390, ACM, 2016.
- [199] D. Lindlbauer, M. Haller, M. Hancock, S. D. Scott, and W. Stuerzlinger, “Perceptual grouping: Selection assistance for digital sketching,” in *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces, ITS '13*, (New York, NY, USA), pp. 51–60, ACM, 2013.
- [200] C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, “Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1317–1322, Dec 2016.
- [201] J. Lin, E. Keogh, L. Wei, and S. Lonardi, “Experiencing sax: A novel symbolic representation of time series,” *Data Min. Knowl. Discov.*, vol. 15, pp. 107–144, Oct. 2007.

- [202] W. Luo, H. Tan, H. Mao, and L. M. Ni, “Efficient similarity joins on massive high-dimensional datasets using mapreduce,” in *2012 IEEE 13th International Conference on Mobile Data Management*, pp. 1–10, July 2012.
- [203] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, “A symbolic representation of time series, with implications for streaming algorithms,” in *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD ’03, (New York, NY, USA), pp. 2–11, ACM, 2003.
- [204] Y. Ma, X. Meng, and S. Wang, “Parallel similarity joins on massive high-dimensional data using mapreduce,” *Concurr. Comput. : Pract. Exper.*, vol. 28, pp. 166–183, Jan. 2016.
- [205] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, “Querying and mining of time series data: Experimental comparison of representations and distance measures,” *Proc. VLDB Endow.*, vol. 1, pp. 1542–1552, Aug. 2008.
- [206] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, “Searching and mining trillions of time series subsequences under dynamic time warping,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’12, (New York, NY, USA), pp. 262–270, ACM, 2012.
- [207] J. O. Wobbrock, A. D. Wilson, and Y. Li, “Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes,” in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST ’07, (New York, NY, USA), pp. 159–168, ACM, 2007.
- [208] H. Almoctar, P. Irani, V. Peysakhovich, and C. Hurter, “Path word: A multi-modal password entry method for ad-hoc authentication based on digits’ shape and smooth pursuit eye movements,” in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI ’18, (New York, NY, USA), pp. 268–277, ACM, 2018.
- [209] J. Zhao and L. Itti, “shapedtw: Shape dynamic time warping,” *Pattern Recognition*, vol. 74, pp. 171 – 184, 2018. .
- [210] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie, “Searching trajectories by locations: An efficiency study,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’10, (New York, NY, USA), pp. 255–266, ACM, 2010.

- [211] Y. Tao, D. Papadias, and Q. Shen, “Continuous nearest neighbor search,” in *Proceedings of the 28th International Conference on Very Large Data Bases, VLDB ’02*, pp. 287–298, VLDB Endowment, 2002.
- [212] H. Jeung, M. L. Yiu, and C. S. Jensen, *Trajectory Pattern Mining*, pp. 143–177. New York, NY: Springer New York, 2011.
- [213] T. Devogele, L. Etienne, M. Esnault, and F. Lardy, “Optimized discrete frÉchet distance between trajectories,” in *Proceedings of the 6th ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data, BigSpatial’17*, (New York, NY, USA), pp. 11–19, ACM, 2017.
- [214] P. Karl, “Note on regression and inheritance in the case of two parents,” *Proceedings of the Royal Society of London*, 1895.
- [215] M. Vidal, A. Bulling, and H. Gellersen, “Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets,” in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 439–448, ACM, 2013.
- [216] C. Clarke, A. Bellino, A. Esteves, and H. Gellersen, “Remote control by body movement in synchrony with orbiting widgets: An evaluation of tracematch,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, pp. 45:1–45:22, Sept. 2017.
- [217] N. Andrienko, G. Andrienko, J. M. C. Garcia, and D. Scarlatti, “Analysis of flight variability: a systematic approach,” *IEEE Transactions on Visualization and Computer Graphics*, August 2018. © 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
- [218] S. Buschmann, M. Trapp, and J. Döllner, “Animated visualization of spatial–temporal trajectory data for air-traffic analysis,” *The Visual Computer*, vol. 32, pp. 371–381, Mar 2016.
- [219] G. Casiez, N. Roussel, and D. Vogel, “1-euros filter: A simple speed-based low-pass filter for noisy input in interactive systems,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’12*, (New York, NY, USA), pp. 2527–2530, ACM, 2012.

- [220] M. H. Everts, E. Begue, H. Bekker, J. B. T. M. Roerdink, and T. Isenberg, “Exploration of the brain’s white matter structure through visual abstraction and multi-scale local fiber tract contraction,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 7, pp. 808–821, 2015.
- [221] H. Almoctar, P. Irani, V. Peysakhovich, and C. Hurter, “Path word: A multi-modal password entry method for ad-hoc authentication based on digits’ shape and smooth pursuit eye movements,” in *Proceedings of the 20th ACM International Conference on Multimodal Interaction, ICMI ’18*, (New York, NY, USA), pp. 268–277, ACM, 2018.
- [222] F. M. Celebi, E. S. Kim, Q. Wang, C. A. Wall, and F. Shic, “A smooth pursuit calibration technique,” in *Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA ’14*, (New York, NY, USA), pp. 377–378, ACM, 2014.
- [223] A. Hassoumi, V. Peysakhovich, and C. Hurter, “Uncertainty visualization of gaze estimation to support operator-controlled calibration,” *Journal of Eye Movement Research*, vol. 10, no. 5, 2018.
- [224] A. Hassoumi, V. Peysakhovich, and C. Hurter, “Improving eye-tracking calibration accuracy using symbolic regression,” *PLOS ONE*, vol. 14, pp. 1–22, 03 2019.
- [225] T. Santini, W. Fuhl, and E. Kasneci, “Calibme: Fast and unsupervised eye tracker calibration for gaze-based pervasive human-computer interaction,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI ’17*, (New York, NY, USA), pp. 2594–2605, ACM, 2017.
- [226] A. Duchowski, *Eye tracking methodology: Theory and practice*, vol. 373. Springer, 2007.
- [227] C. Hurter, S. Puechmorel, F. Nicol, and A. Telea, “Functional decomposition for bundled simplification of trail sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 500–510, Jan 2018.
- [228] V. Peysakhovich, C. Hurter, and A. Telea, “Attribute-driven edge bundling for general graphs with applications in trail analysis,” in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 39–46, April 2015.
- [229] Y. Zheng, “Trajectory data mining: An overview,” *ACM Transaction on Intelligent Systems and Technology*, September 2015.
- [230] Y. Wang, Y. Zheng, and Y. Xue, “Travel time estimation of a path using sparse trajectories,” in *Proceedings of the 20th ACM SIGKDD International Conference*

on *Knowledge Discovery and Data Mining*, KDD '14, (New York, NY, USA), pp. 25–34, ACM, 2014.

- [231] D. Young and B. Shneiderman, “A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation,” *J. Am. Soc. Inf. Sci.*, vol. 44, pp. 327–339, July 1993.
- [232] G. Jarry, D. Delahaye, S. Puechmorel, and E. Feron, “Real time aircraft atypical approach detection for air traffic control,” 2021.
- [233] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [234] J.-D. Boissonnat, A. Cérézo, and J. Leblond, “Shortest paths of bounded curvature in the plane,” *Journal of Intelligent and Robotic Systems*, vol. 11, no. 1-2, pp. 5–20, 1994.
- [235] L. S. Pontryagin, *Mathematical theory of optimal processes*. Routledge, 2018.
- [236] X.-N. Bui, J.-D. Boissonnat, P. Soueres, and J.-P. Laumond, “Shortest path synthesis for dubins non-holonomic robot,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 2–7, IEEE, 1994.
- [237] H. Chitsaz and S. M. LaValle, “Time-optimal paths for a dubins airplane,” in *2007 46th IEEE conference on decision and control*, pp. 2379–2384, IEEE, 2007.
- [238] “Air India Express Flight 1344,” *Wikipedia*, Aug. 2020. Page Version ID: 972120506.
- [239] EASA, “Easa annual safety review,” 2020. Available at https://www.easa.europa.eu/sites/default/files/dfu/easa_asr_2020.pdf.
- [240] P. Monmousseau, G. Jarry, F. Bertosio, D. Delahaye, and M. Houalla, “Predicting passenger flow at charles de gaulle airport security checkpoints,” in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–9, IEEE, 2020.
- [241] L. Liu and R.-C. Chen, “A novel passenger flow prediction model using deep learning methods,” *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 74–91, 2017.
- [242] Y. Sun, B. Leng, and W. Guan, “A novel wavelet-svm short-time passenger flow prediction in beijing subway system,” *Neurocomputing*, vol. 166, pp. 109–121, 2015.

- [243] Y. Wei and M.-C. Chen, “Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks,” *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 148–162, 2012.
- [244] G. Xie, S. Wang, and K. K. Lai, “Short-term forecasting of air passenger by using hybrid seasonal decomposition and least squares support vector regression approaches,” *Journal of Air Transport Management*, vol. 37, pp. 20–26, 2014.
- [245] C.-I. Hsu and Y.-H. Wen, “Improved grey prediction models for the trans-pacific air passenger market,” *Transportation planning and Technology*, vol. 22, no. 2, pp. 87–107, 1998.
- [246] S. V. Kumar, “Traffic flow prediction using kalman filtering technique,” *Procedia Engineering*, vol. 187, pp. 582–587, 2017.
- [247] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results,” *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.
- [248] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal arima model with limited input data,” *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.
- [249] D. Wilson, E. K. Roe, and S. A. So, “Security checkpoint optimizer (sco): an application for simulating the operations of airport security checkpoints,” in *Proceedings of the 38th conference on Winter simulation*, pp. 529–535, Winter Simulation Conference, 2006.
- [250] K. Leone and R. R. Liu, “Improving airport security screening checkpoint operations in the us via paced system design,” *Journal of Air Transport Management*, vol. 17, no. 2, pp. 62–67, 2011.
- [251] R. De Lange, I. Samoilovich, and B. Van Der Rhee, “Virtual queuing at airport security lanes,” *European Journal of Operational Research*, vol. 225, no. 1, pp. 153–165, 2013.
- [252] V. Vapnik, *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control, Wiley, 1998.
- [253] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, “Scikit-learn: Machine Learning in Python,” *Machine Learning in Python*, 2011.

Appendices

Appendix A

Operational boundaries for flight parameters

In this appendix, the different operational limits and categories used in the illustrations and studies are presented. In this thesis three operational parameters are mainly studied: the longitudinal trajectory (longitude, latitude), the altitude profile, and the ground speed profile.

Longitudinal Trajectory Regarding the longitudinal trajectory, the operational limits used correspond to the extension of the compliance criteria presented in chapter . The aircraft in approach are supposed to intercept the extended runway center line with an angle not exceeding 45° . Then, they are supposed to follow a levelled-off flight of 30s (around 2NM), and finally descend on the glide path.

Suppose the trajectory is followed backward, the proposed limits for lateral deviations are 150m (respectively 300m) at the runway threshold. They then increase linearly up to the FAP to reach 350m (respectively 700m). They increase again linearly up to the intercept chevrons located 2NM after the FAP and reach 700m (respectively 1400m). Finally, the limits deviate with an angle of 45° (respectively 60°). They are represented by the orange dashed lines (respectively red) in Figure A.1.

Altitude Profile The altitude limits include upper and lower limits. The lower limits are the following if the trajectory is followed backward. At the runway threshold the limit is -25ft under the nominal glide (respectively -50ft). Then it increases linearly until the FAP to reach 350ft (respectively 700ft) under the lowest published altitude of interception. Finally, it does not increase more since an aircraft is not supposed to intercept the runway extended center line with an altitude lower than the lowest published altitude. Regarding the upper limits, the deviation at the runway threshold is +75ft (respectively 150ft). It corresponds to a glide angle of 3.6° (respectively 4.0°). The limits are materialized by orange (respectively red) dashed lines in Figure A.2. A flight with a point outside the orange (respectively red) limits will be categorized as warning (respectively critical) altitude.

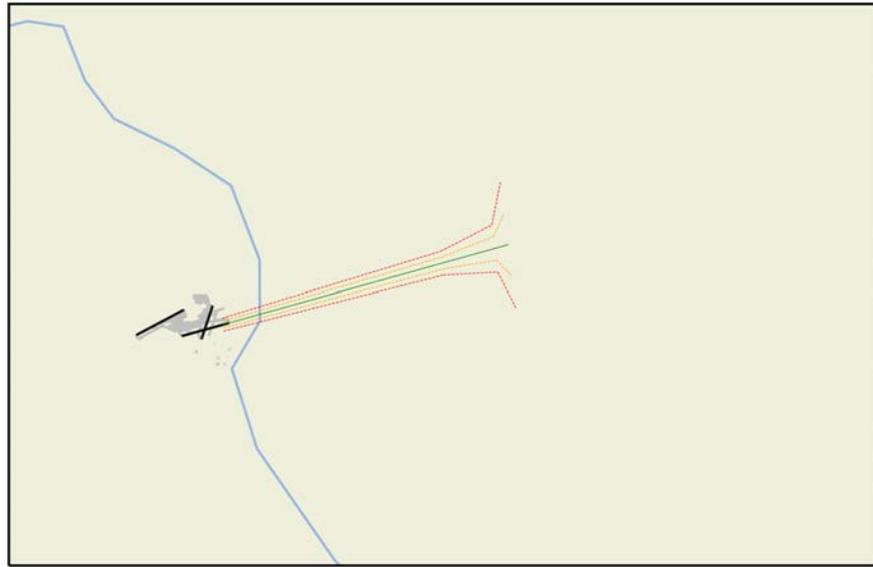


Fig. A.1 Longitudinal operational limits illustration. The green line corresponds to the extended runway center line. The orange and red line correspond to deviation from the runway center line and from the interception chevrons.

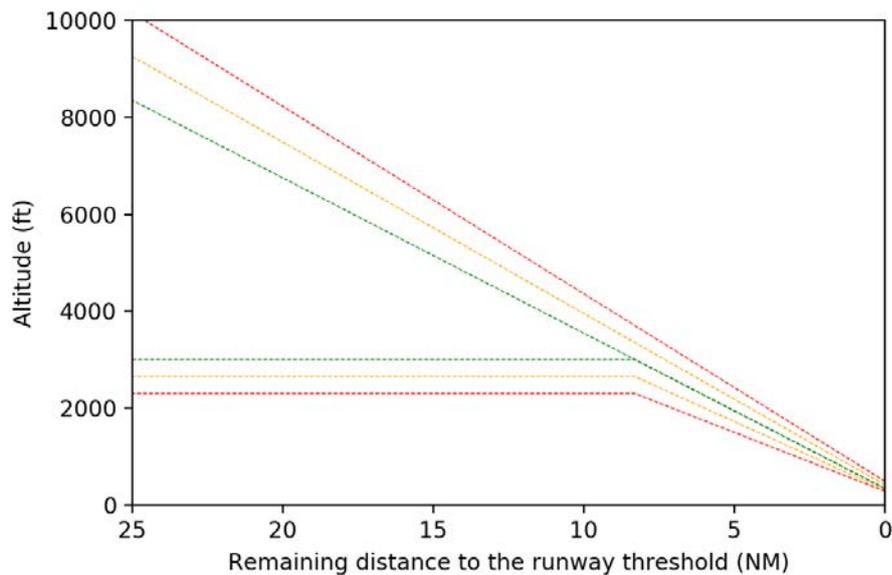


Fig. A.2 Altitude operational limits illustration. The green lines correspond to the glide path and the minimum published interception altitude. The orange and red line correspond to deviation from the glide path and the minimum published interception altitude.

Ground Speed Profile Similarly, the limits for ground speed are divided into upper and lower limits. The green nominal zone for ground speed will depend if one or several aircraft are studied. If only one aircraft is studied the nominal zone will be defined between the approach speed at the maximum landing weight $V_{app,MLW}$ and $V_{app,MLW}$ minus 30kts. Otherwise, the green nominal zone is selected between 120kts and 150kts. The lower warning (respectively critical) limits correspond to the lower nominal minus 15kts (respectively 30kts). The upper limits are divided into three segments. The first

segment correspond to the stabilisation between the runway threshold and the distance corresponding to 1000ft on a 3° glide path. The limits are defined as the upper nominal limit plus 15kts (respectively 30kts). The second segment corresponds to a deceleration segment, and is based on nominal deceleration on a 3° glide path [14]. The nominal upper limits shows a 15kts/NM deceleration until 250kts (the maximum speed below FL100), the warning limit a 17.5kts/NM deceleration until the same distance, and the critical a 20kts/NM deceleration. The last segment is a constant segment, which ends at the distance corresponding to FL100 on a 3° glide path. These limits are represented by the green, orange, and red dashed in Figure A.3.

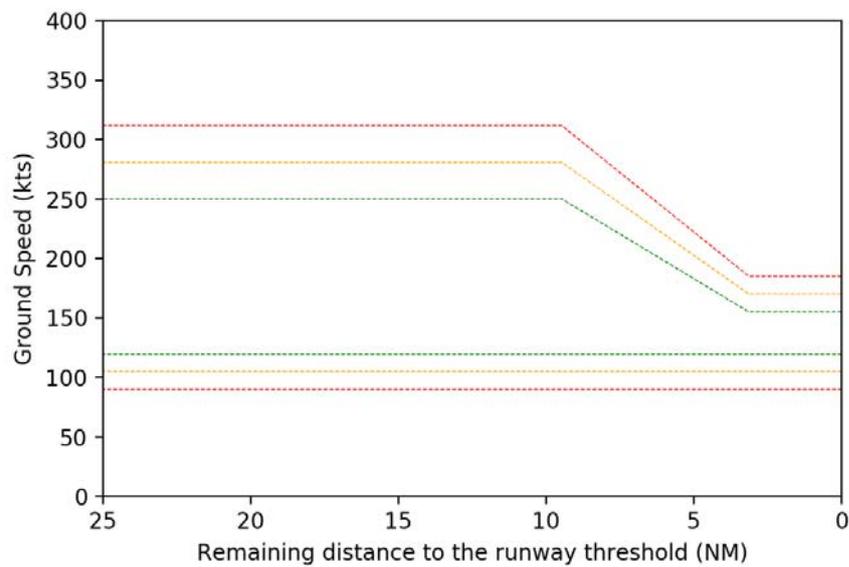


Fig. A.3 Ground speed operational limits illustration. The green lines correspond to nominal ground speed areas. The orange and red lines correspond to deviations from the green line.

Appendix B

Kernel Density Estimation

Kernel density estimation (or Parzen-Rosenblatt method) is a statistical non-parametric method for estimating the probability density of a random variable [18]. It is based on a sample of a statistical population and makes it possible to estimate the density at any point of the support. In this sense, this method cleverly generalizes the method of estimation by a histogram.

If $x_1, \dots, x_n \sim f$ is an i.i.d. sample of a random variable, then the non-parametric kernel density estimator is:

$$\hat{f}_h(x) = \frac{1}{N \cdot h} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (\text{B.1})$$

where K is a kernel and h a parameter named window, which governs the degree of smoothing of the estimation. Very often, K is chosen as the density of a standard Gaussian function (null expectation and unit variance):

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) \quad (\text{B.2})$$

The Parzen method is a generalization of the histogram estimation method. In a histogram, the density at a point x is estimated by the proportion of observations x_1, \dots, x_n which are close to x . To do this, a box is drawn in x for which width is governed by a smoothing parameter h ; the number of observations belonging to this box is then counted. This estimate, which depends on the smoothing parameter h , has good statistical properties but is non-continuous by construction.

The kernel method consists in recovering continuity: to do this, the box centered in x with width h is replaced by a Gaussian centered in x . The closer an observation is to the support point x , the more the bell curve will give it an important numerical value. Conversely, observations too far from x are assigned a negligible numerical value. The estimator is formed by the sum (or rather the average) of the bell curves.

The practical use of this method requires two things: the kernel K (usually the density of a statistical law) and the smoothing parameter h .

If the choice of the kernel is deemed to have little influence on the estimator, the

same is not true for the smoothing parameter. A parameter that is too weak causes artificial details to appear on the estimator's graph. If the value of h is too large, most of the characteristics are erased. The choice of h is thus a central issue in density estimation.

A common way to obtain a value of h is to assume that the sample is distributed according to a given parametric distribution, for example according to the normal distribution $\mathcal{N}(\mu, \sigma^2)$, then a possible value is:

$$h = 1.06 \hat{\sigma} n^{\frac{1}{5}} \tag{B.3}$$

Appendix C

Neural Networks

C.1 Introduction

An artificial neural network, is a system whose design was originally schematically inspired by the functioning of biological neurons, and which later came close to statistical methods.

A neural network is the association, in a more or less complex graph, of elementary objects, the formal neurons. The main networks are distinguished by the organization of the graph (layered, complete . . .), that is to say their architecture, their level of complexity (the number of neurons, presence or not of feedback loops in the network), by the type of neurons (their transition or activation functions) and finally by the objective: supervised or unsupervised learning, optimization, dynamic systems.

In a very simplistic way, a biological neuron is a cell that is characterized by:

- synapses, the points of connection with other neurons, nerve or muscle fibers,
- the dendrites or input of the neuron,
- axons, or output of the neuron to other neurons,
- the nucleus that activates the outputs according to the input stimulation.

By analogy, the formal neuron is a model that is characterized by an internal state $s \in \mathbf{S}$, input signals x_1, \dots, x_p and an activation function:

$$s = h(x_1, \dots, x_p) = g \left(w_0 + \sum_{i=1}^p w_i x_i \right) \quad (\text{C.1})$$

The g activation function performs a transformation of an affine combination of input signals. This affine combination is determined by a weight vector (w_0, \dots, w_p) associated with each neuron and whose values are estimated in the training phase. They constitute the memory or distributed knowledge of the network. The formal neuron structure is illustrated in Figure C.1.

The different types of neurons are distinguished by the g nature of their activation function. The main types are :

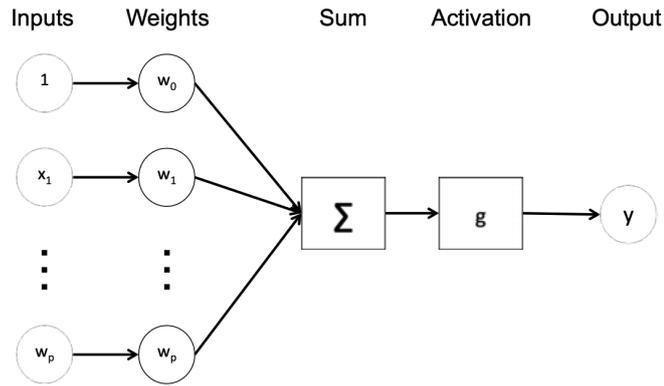


Fig. C.1 Formal neuron illustration. A weighted sum of the input goes through an activation function to compute the output.

- Linear g is the identity function.
- Sigmoide $g(x) = \frac{1}{1+e^x}$.
- Rectified Linear Unit (ReLU) $g(x) = \max(0, x)$.
- Softmax $g(x)_j = \frac{e^{x_k}}{\sum_{k=1}^{k=K} e^{x_k}} \forall k \in \{1, \dots, K\}$.

Linear, sigmoide, ReLU, softmax activation functions are well suited to learning algorithms involving gradient back-propagation because their activation function is differentiable; they are the most widely used.

C.2 Multilayer perceptron and learning

The multilayer perceptron (MCP) is a network composed of successive layers. A layer is a set of neurons with no connection between them. An input layer reads the incoming signals, one neuron per input, and an output layer provides the system response. One or more hidden layers participate in the transfer. In a perceptron, a neuron of a hidden layer is connected in input to each neuron of the previous layer and in output to each neuron of the next layer. The architecture of a multilayer perceptron is illustrated in Figure C.2.

Once the network architecture is defined, the weight w_j of the network must be estimated from training samples. The estimation of the parameters is done by minimizing a loss function usually via a gradient descent algorithm. In the particular context of neural networks, gradient back-propagation is usually used. This method computes the error gradient for each neuron in a neural network, from the last layer to the first and updates the weight in order to decrease the loss error. The purpose of the gradient algorithm is to iteratively converge towards an optimized configuration of synaptic weights. This state can be a local minimum of the function to be optimized and ideally, a global minimum of this function.

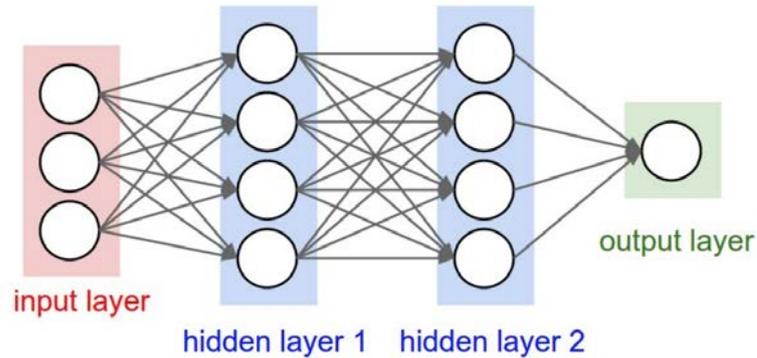


Fig. C.2 Multilayer perceptron architecture. Each neuron uses as input the outputs of the previous layer and computes the input for the following layer.

C.3 Deep Learning and complex architectures

Deep learning is a set of machine learning methods attempting to model with a high level of data abstraction thanks to complex articulated architectures of different non-linear transformations. These techniques have led to significant and rapid progress in the fields of sound and visual signal analysis, in particular facial recognition, voice recognition, computer vision, automated language processing, and so on.

A first example of these architecture are the convolutional neural networks (CNN or ConvNet). It is a type of acyclic (feed-forward) artificial neural network, in which the connection pattern between neurons is inspired by the visual cortex of animals. The neurons in this region of the brain are arranged so that they correspond to overlapping regions when the visual field is paved. Their functioning is inspired by biological processes, they consist of a multi-layered stack of perceptrons, whose purpose is to pre-process small amounts of information. Convolutional neural networks have wide applications in image and video recognition, recommendation systems and natural language processing.

A second example, are the recurrent neural networks. The architecture is composed of artificial neurons with recurrent connections. A recurrent neural network consists of interconnected units (neurons) interacting non-linearly and for which there is at least one cycle in the structure. The units are connected by arcs (synapses) that have a weight. The output of a neuron is a non-linear combination of its inputs. Recurrent neural networks are suitable for input data of variable size. They are particularly suitable for time series analysis. They are used in automatic speech or handwriting recognition - more generally in pattern recognition - or in machine translation.

Appendix D

Pearson's χ^2 Independence Test

The Pearson's χ^2 test is a statistical test that verifies the adequacy of a variable to a probability distribution law or the absence of a statistical relationship between two variables X and Y. In the context of the independence test, the two variables are said to be independent when there is no statistical link between them, in other words, knowledge of X does not in any way make it possible to make a statement about Y. The null hypothesis (H0) of this test is as follows: the two variables X and Y are independent.

For example, if X denotes a flight category (being atypical or typical) and Y denotes another flight event (being stabilized or unstabilized). The hypothesis to be tested is the independence between the atypical category to which the flight belongs X and Y the event category. The hypothesis therefore states that knowing the atypical category of an individual does not influence the value of the stabilization category.

Each observation is assigned to a cell in a two-dimensional table of cells (called a contingency table) based on the values of the two outcomes. If there are r rows and c columns in the table (2 rows and 2 columns in the binary case), the "theoretical frequency" for a cell, given the assumption of independence, is:

$$E_{i,j} = Np_{i.}p_{.j} \quad (\text{D.1})$$

Where N is the total sample size (the sum of all cells in the table), and $p_{i.}$ (respectively $p_{.j}$) is the fraction of observations of type i (resp j) ignoring the other attribute defined as follow:

$$p_{i.} = \sum_{j=1}^c \frac{O_{i,j}}{N}, \quad p_{.j} = \sum_{i=1}^r \frac{O_{i,j}}{N} \quad (\text{D.2})$$

with $O_{i,j}$ being the number of samples with the observations i , $0 \leq i \leq r$ and j , $0 \leq j \leq c$. The value of the χ^2 test-statistic is

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \quad (\text{D.3})$$

Finally, the number of degrees of freedom k to fit with the χ^2 distribution is equal

to $(r - 1)(c - 1)$. In terms of p-value, the null hypothesis is generally rejected when $p \leq 0,05$. In the case of two binary variables, which induce a degree of freedom of one, if the score is greater than 3,8415, it corresponds to a p-values lower than 0.05. Different p-value function of the chi^2 cumulative test statistic are displayed in Figure D.1 with several degrees of freedom k .

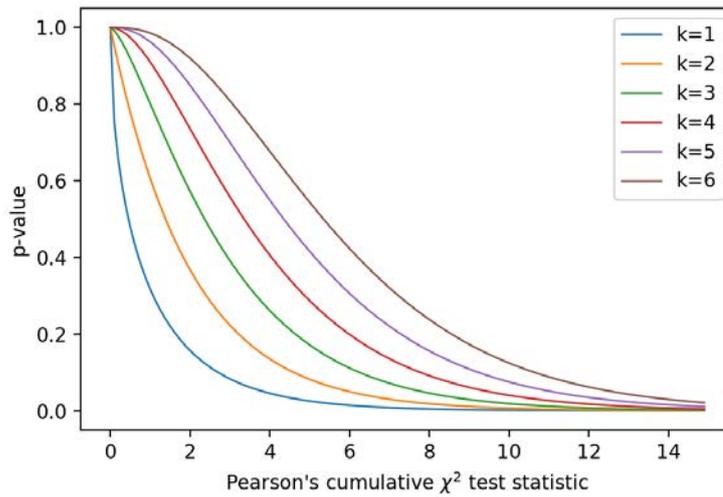


Fig. D.1 P-value function of the the chi^2 cumulative test statistic for several degree of freedom k .

Appendix E

Trajectory Approach Analysis (TAPAS): A post-operational aircraft approach analysis tool

This appendix presents Trajectory Approach Analysis (TAPAS). TAPAS is a prototype software that enables a post-operational analysis of aircraft approach trajectories with respect to an energy atypicality model. TAPAS software is divided into four sections. The first section called Data, gives general statistic on the analyzed trajectories such as the aircraft distribution, the airport distribution, and the runways QFU distributions. Figure E.1 gives an example of the section Data.

The second section, called Statistics, illustrated in Figure E.2, is dedicated to operational statistics of the analyzed trajectories. It gives the distribution of the analyzed trajectories in terms of compliance [108], operational limits [108], atypicality, event phase, event intensity level, and event number per flights.

The third section, called Events, represented in Figure E.3, displays for each type of event, its number of occurrences and its intensity. It provides a list of event types sorted by their number and intensity.

Finally, the last section, called Flight Study, represented in Figure E.4, enables analyzing individually each flight. It gives the different properties of the flight, the compliance, the operational limit, the atypicality per area, and the events notified for this flight. It also displays different parameters.

There are four graphs displayed in the Flight Study section. The first graph shows the longitudinal trajectory path and is illustrated in Figure E.5. Besides, the user is able to display compliance limits. For instance, the flight illustrated in figure E.5 presents a non-compliance corresponding to a lateral deviation for the 4000ft FAP procedure at Paris Orly runway 26 [108].

The second graph, represented in figure E.6, illustrates the altitude profile and the atypical coefficients represented by the colored dots. The atypical coefficient is bounded between 0 (corresponding to nominal situations and displayed in green), and 1 (corresponding to atypical situations displayed in red). The user is also able to display operational or compliance limits. For instance, the flight illustrated in figure E.6 presents a glide deviation associated with the event Glide Interception From Above [108]. The

TAPAS - Trajectory Approach Analysis

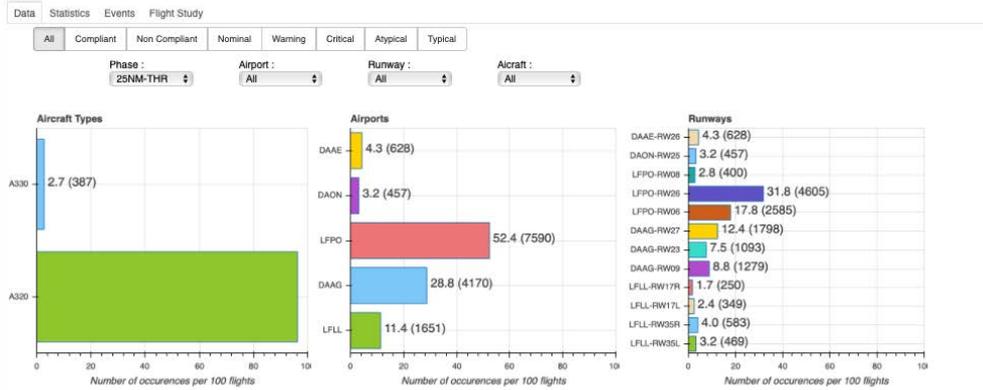


Fig. E.1 Data section of TAPAS prototype software. It provides general information on the data set such as the aircraft type or the airports studied.

TAPAS - Trajectory Approach Analysis



Fig. E.2 Statistics section of TAPAS prototype software. It provides information on operational statistics such as the compliance or the average number of monitoring events per 100 flights.

TAPAS - Trajectory Approach Analysis

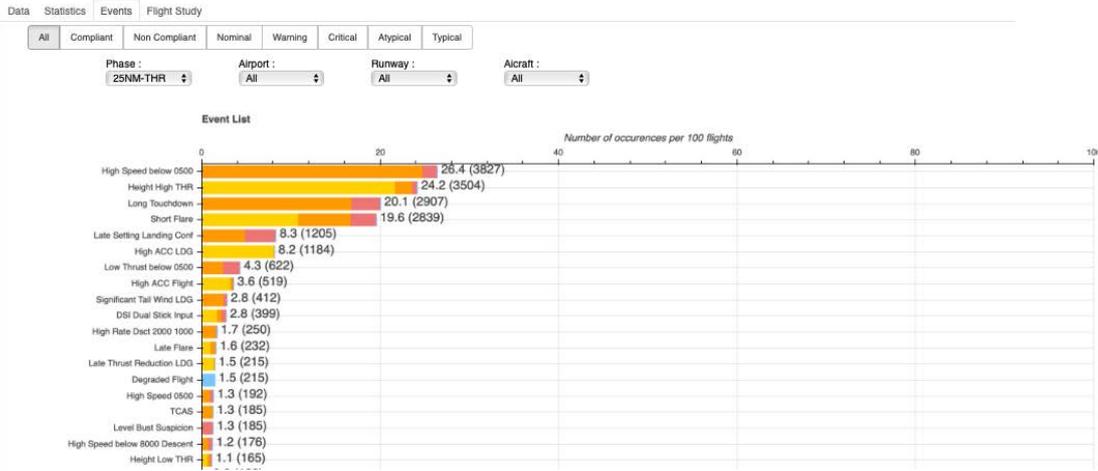


Fig. E.3 Event section of TAPAS prototype software. It provides further information and detail on the monitoring events occurrences.

TAPAS - Trajectory Approach Analysis

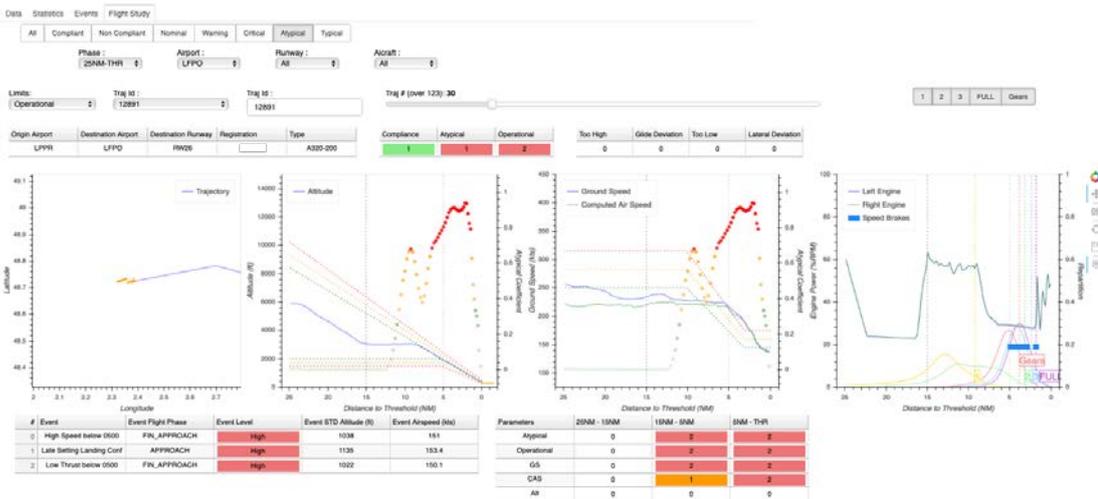


Fig. E.4 Flight Study section of TAPAS prototype software. It enables the analysis of a particular trajectory.

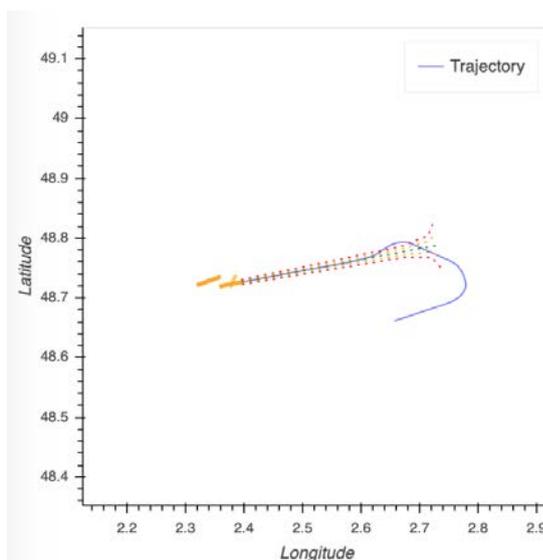


Fig. E.5 First graph of TAPAS section Flight Study, displaying the longitudinal path and lateral compliance criteria.

atypical coefficient highlights this atypical area.

The third graph, illustrated in Figure E.7, shows the ground speed profile, the computed airspeed profile, and the atypical coefficients. Operational limits are also available. The flight illustrated in figure E.7 presents a late speed reduction associated with an atypical area in red.

Finally, the fourth graph, represented in figure E.8, illustrates the aircraft engine power, and aircraft configuration elements. This graph is optional for ground data sources (radar, ADS-B) or can give the estimation given by the models presented in Chapter 4. The blue horizontal bar shows the use of speed brakes. The vertical dashed lines display the flaps and landing gear sequence. Several Gaussian curves are also represented. They illustrate the distribution of flaps and landing gear sequence over all the flights available for this QFU. This enables analyzing this particular flight flaps configuration policy compared with the others.

The user is able to use TAPAS interactions to group the trajectories regarding different parameters. This enables showing statistics over a particular trajectory set, highlighting correlations, and studying a particular trajectory group. The user is able to select compliant, non-compliant, nominal, warning, critical, typical and atypical trajectories [108]. In addition, the user can select a particular airport, runway QFU, aircraft type, and the phase of the atypicality (25NM-15NM, 15NM-5NM, 5NM-THR).

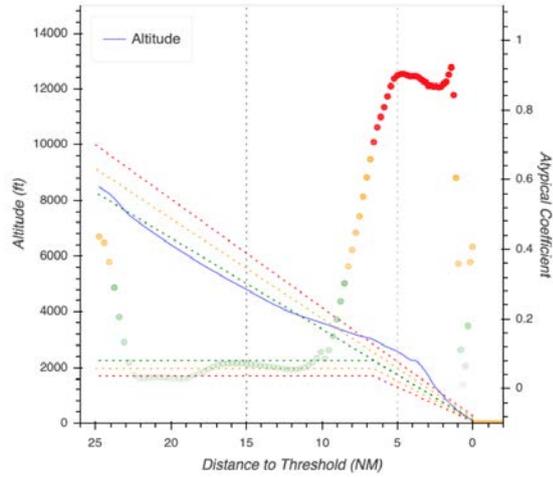


Fig. E.6 Second graph of TAPAS section Flight Study, displaying the altitude profile and the atypical coefficient.

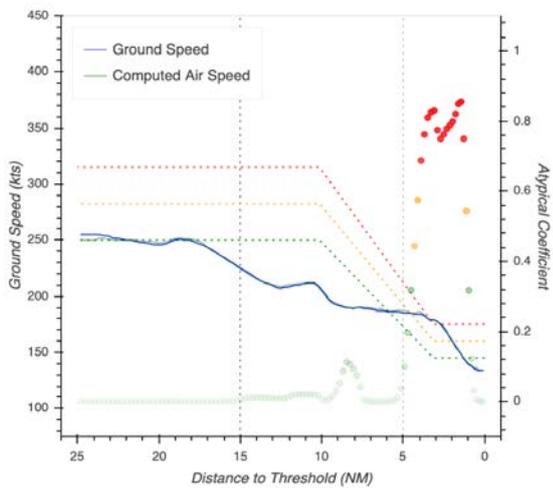


Fig. E.7 Third graph of TAPAS section Flight Study, displaying the ground speed profile, the computed air speed profile, and the atypical coefficient.

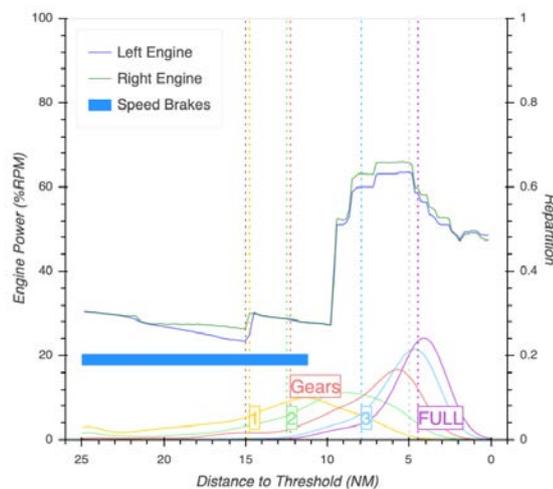


Fig. E.8 Fourth graph of TAPAS section Flight Study, displaying the engine power, the flaps settings, the landing gear setting, and the airbrake use.

Appendix F

Interactive Trajectory Modification and Generation with Functional Principal Component Analysis: User Interface

This appendix presents the user interface of the modification pipeline developed in Chapter 6. This pipeline enables trajectory processing and generation using Functional Principal Component Analysis. The pipeline handles trajectory clustering, data cleaning, flow simplification, flow generation and flow transformation.

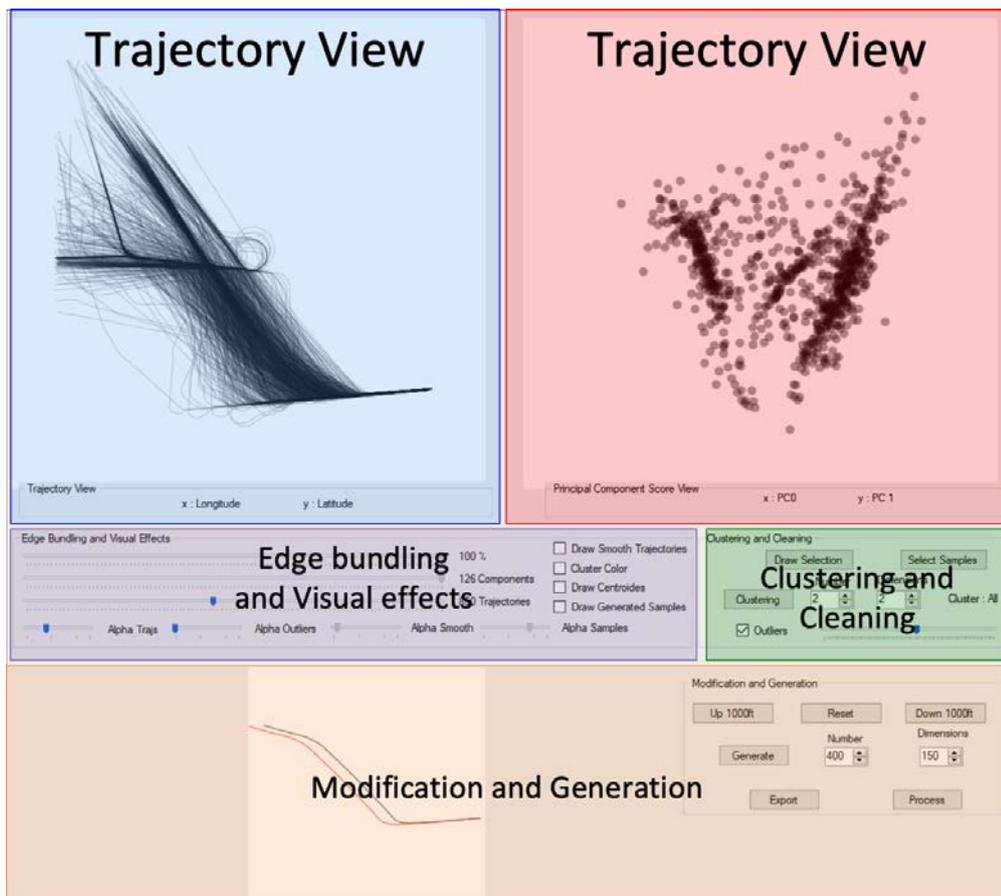


Fig. F.1 User interface of the Trajectory Modification and Generation with Functional Principal Component Analysis prototype software.

Figure F.1 illustrates the user interface of the system to support the different steps of the analytic pipeline. On top of the interface, there are two visualizations. One is the

original curve space visualization and the other is the Principal Component Score Visualization. The visualization of the scores will guide the user in the clustering refinement and the data cleaning processes.

In the center, there are two categories of interactions, those for the clustering and cleaning are located on the right side, and those for the visual effects on the left side. At the bottom, are located interactions for modifications and generation.

Appendix G

Predicting Passenger Flow at Charles De Gaulle Airport Security Checkpoints

This appendix presents the work done in collaboration with Charles De Gaulle Airport and another PhD Student. It aims at predicting passenger flow at security checkpoint with machine learning. The use of LSTM networks and the prediction philosophy is similar to the one used to predict on-board parameters in chapter 4. This research has therefore contributed to the deepening of recurrent neural network expertise. This research led to a conference publication [240].

Airport security checkpoints are critical areas in airport operations. Airports have to manage an important passenger flow at these checkpoints for security reason while maintaining service quality. The cost and quality of such an activity depend on the human resource management for these security operations. An appropriate human resource management can be obtained using an estimation of the passenger flow. This appendix investigates the prediction at a strategic level of the passenger flows at Paris Charles De Gaulle airport security checkpoints using machine learning techniques such as Long Short-Term Memory neural networks. The derived models are compared to the current prediction model using three different mathematical metrics. In addition, operational metrics are also designed to further analyze the performance of the obtained models.

G.1 Introduction

G.1.1 Motivation

Airport security checkpoints are key areas in airport operations. All passengers are checked at security checkpoint before entering the airside area. This continuous passenger flow implies an appropriate human resource management, which must satisfy two main objectives. A security checkpoint must be reliable in terms of security, while maintaining a predefined standard regarding passenger wait time. In addition, airports try to minimize their cost providing the best possible services.

At Charles De Gaulle airport, the human resources at security checkpoints are managed at two levels. The first level is a strategic level: passenger flows at security checkpoints are predicted 20 days upstream for the following month in order to determine the appropriate number of agents required. The second level is a tactical level: in real time, the agents are distributed at the security checkpoints to provide the service. This appendix investigates new learning methods such as neural networks in order to improve the prediction phase at the strategic level.

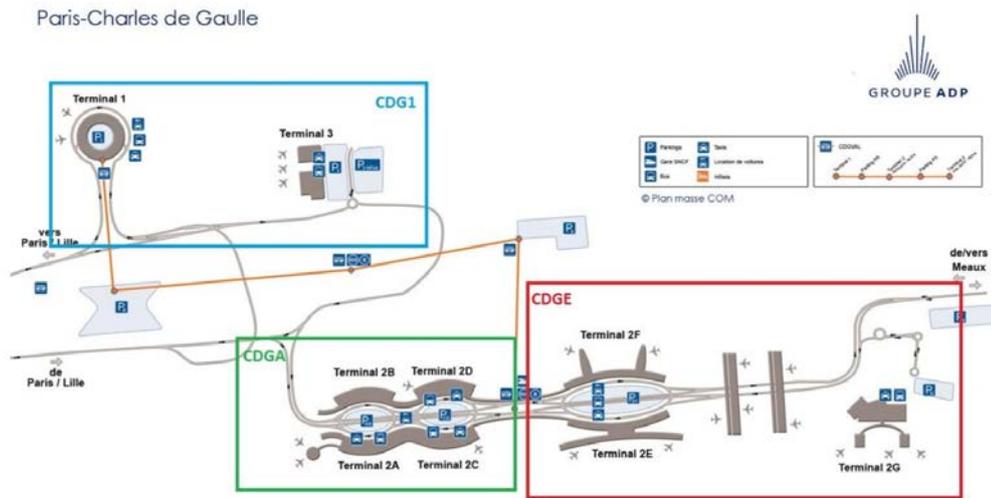


Fig. G.1 Overview map of Charles De Gaulle terminals

These learning methods are applied to the checkpoints within the zone of Charles De Gaulle airport corresponding to Air France’s hub and named CDGE (cf. Figure G.1). It contains eight security checkpoints, separated in three different categories, depending on the type of passengers going through:

- checkpoints handling only passengers with local flights: C2F-Centraux;
- checkpoints handling only connecting passengers: C2E-GalerieEF, C2E-Puits2E, C2E-PorteL-CNT;
- checkpoints handling passengers on both local and connecting flights: C2E-PorteK, C2E-PorteL, C2E-PorteM, C2G-Depart.

Checkpoints C2E-GalerieEF and C2E-PorteL-CNT have the added particularity of linking two different terminals (E and F). C2E-Puits2E has the specificity of handling connecting passengers arriving to and leaving from Terminal E.

G.1.2 State of the art

Passenger flow prediction has been investigated for a long time in transportation areas. An exhaustive review was done by Liu et al. [241]. Traffic flow prediction for public

transportation was studied in [242, 243], and for air transportation in [244, 245] using various prediction methods. Time series models were developed by Kumar [246] based on Kalman filtering, while Williams and Hoel [247] and then Kumar and Vanajakshi [248] worked on auto-regressive models. In the machine learning field, regression models such as Support Vector Machines [242, 244] or Neural Networks [241, 243] were used to forecast passenger flow. So far, the models derived try to predict the passenger flow using only historical data of the flow. Nevertheless, an airport passenger flow is a complex process. Extra features could be added in order to enhance the model performance. Indeed, a model which includes information relative to the arriving and departing flights should outperform basic time series models. This motivates the use of machine learning models, that can fit multidimensional inputs.

Optimization of security checkpoints at a tactical level has also been thoroughly investigated. The efficiency of security checkpoint systems and organizations is discussed by Wilson et al. [249] and by Leone and Liu [250]. De Lange et al. [251] suggested creating virtual queuing in order to decrease waiting time at peak periods. However, to the best of the authors' knowledge, no study has been conducted around the airport security checkpoint strategic passenger flow prediction. Usually, each airport has its own process. Yet, the methodology presented in this appendix is generic and could be applied everywhere. The only constraint is the availability of information regarding departing and arriving flight and their expected occupancy.

This appendix is organized as follows: Section G.2 describes the data considered, the features extracted from them and the learning models used. In Section G.3 the different models are compared using both theoretical and operational performance measures. An in-depth analysis is performed in Section G.3.2 for two chosen checkpoints. Section G.4 concludes this study and suggests some possible improvements and future steps.

G.2 Model building

This section presents the machine learning models chosen for the following experiments as well as the data considered.

G.2.1 Machine Learning and Long Short-Term Memory Neural Network

A learning process consists in using data analysis methods and artificial intelligence to predict the behavior of a system. The aim is to define a model that will fit as best as possible the considered system. Machine learning algorithms define learning models h_θ , with parameters θ , that approximate the system function. The learning process is done upon a finite training set \mathcal{D} , and aims at minimizing the error over the training set by tuning the parameters θ of the learning model [252, 138, 139].

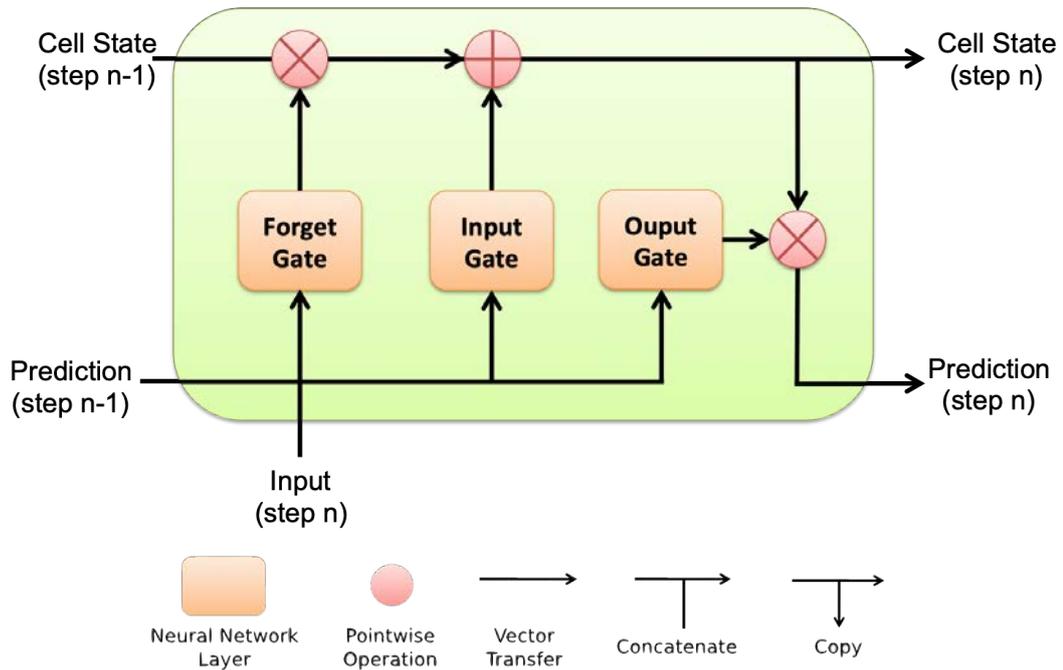


Fig. G.2 Simplified illustration of the structure of a LSTM cell.

Various learning models exist in the literature and for various real-world applications, and in this appendix the choice of a particular neural network named Long Short-Term Memory (LSTM) was made and compared to a Random Forest model. LSTM networks were designed as an enhancement of Recurrent Neural Networks (RNN) to perform better supervised learning task on time series data [140, 141, 142]. LSTM are capable of learning long-term dependencies, while simple RNN only learn short term dependencies. LSTM use a cell state that keeps information from the past, and three gates that update the cell state and compute the prediction. First, the forget gate enables updating the cell state in order to forget information that are no longer relevant based on the current input. Second, the input gate enables saving in the cell state relevant information from the current input. Finally, the output gate computes the prediction using the updated cell state and the current input. A simplified illustration of a LSTM structure is depicted in Figure G.2.

G.2.2 From data to features

In this study, the values to predict correspond to the real passenger count at each Safety Checkpoint per ten minute period. As explained in Section G.1, the original input dataset used by Charles De Gaulle operational experts is composed with information relative to the schedule and occupancy of arriving and departing flights aggregated per

five minute periods.

The dataset starts on February, 1st 2017 and ends on March, 31st 2019. Data from both 2017 and 2018 were used for the training phase, and the data from 2019 for the validation phase. For each flight, there are three passenger count expectations corresponding to:

- the expected number of connecting passengers
- the expected number of local passengers
- the expected total number of passengers

These passenger counts are given by the airlines to the airport. In addition, there are various information such as the date, the status of the flight (departing or arriving flight), the airport terminal, the airline, the origin airport, the aircraft type, the departure geographic area, the flight range, and the check-in terminal.

Categorical features were represented using one-hot encoding. Additional features were extracted to complete the passenger count expectations. Passenger count expectations were aggregated per terminal, per status, and per terminal and status to create new features. Besides, features relative to the date were created: the month of the year, the day of the month, the day of the week, the hour of the day and the minute of the hour, and categorical variables for weekends, aeronautical weekends (including Fridays), holidays, and public holidays. Additional categories were created to capture whether a day is just before or after a public holiday or is the first or last day of a holiday.

This feature extraction yields a vector of 371 features for every five minutes of data. This vector sums-up the information over all the flights during the corresponding five minute period. The LSTM neural network was then fed with a time series corresponding to the input feature vector ranging from three hours before to five hours after the output 10 minutes time period. This time range was chosen based on two real-world considerations in order to encompass all the relevant flight and passenger information. On the one hand, airlines and airports recommend passengers on international flights to arrive about three hours before their flight departure time. On the other hand, the transfer time between two flights seldom exceeds five hours.

G.2.3 Network Architecture and Learning

This section describes the neural network architecture used in the experiments. The neural network is composed of two layers and a regression output layer. The first layer is a batch normalization. The second layer is a LSTM layer with 200 units and a sigmoid activation function. The layer also contains a dropout to regularize the network. The output layer is a single neuron dense layer with a ReLU activation function. This

architecture will be referred to as LSTM200. Figure G.3 illustrates the network architecture.

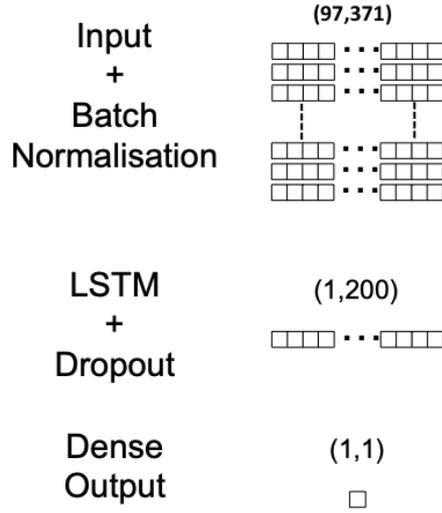


Fig. G.3 Description of the neural network LSTM200 architecture used in the experiments.

The learning task was made using Adam optimizer [143] with a decay. The learning rate is 10^{-3} and the decay is 10^{-9} . Networks were trained during 10 epochs over the training set on a multi-GPU cluster. The cluster is composed of a dual ship Intel Xeon E5-2640 v4 - Deca-core (10 Core) 2,40GHz - Socket LGA 2011-v3 with 8 GPU GF GTX 1080 Ti 11 Go GDDR5X PCIe 3.0.

G.2.4 Penalized Loss

In practice, passenger count overestimation is costly. Therefore, a custom loss was designed. The loss aims to minimize overestimation by penalizing the positive part of the mean square error (MSE). As a reminder, the mean square error is the usual loss for regression problems. Let \mathcal{D} be the training set, and h the learning model. The MSE of h over \mathcal{D} is detailed in equation (G.1):

$$\text{MSE}(h, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \cdot \sum_{(x,y) \in \mathcal{D}} (h(x) - y)^2 \quad (\text{G.1})$$

Let $E = h(x) - y$ be the error of a sample $(x, y) \in \mathcal{D}$. E_+ is the positive part of this error, and E_- the negative part. The α -Penalized MSE is defined in equation (G.2) with $\alpha \in \mathbb{R}$:

$$\alpha\text{-PMSE}(h, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \cdot \sum_{(x,y) \in \mathcal{D}} (E_- + (1 + \alpha) \cdot E_+)^2 \quad (\text{G.2})$$

G.2.5 Model Summary

For this study, three models were used. The first model is a LSTM200 architecture trained with the MSE loss. The second model is a LSTM200 architecture trained with a 0.5-PMSE loss. The last model is a Random Forest model trained with MSE loss using the scikit-learn library [253]. The hyper-parameters of the Random Forest models were set to 40 for the number of estimators, with a max depth of 10, and a minimum sample split of 2. The three models are summarized in Table G.1.

Table G.1 Summary of the three models used in the appendix

Model Name	Model Type	Loss
LSTM (MSE)	LSTM200	MSE
LSTM (0.5-PMSE)	LSTM200	0.5-PMSE
RF	Random Forest	MSE

Additionally, in order to assess the effect of the hour of the day on the robustness of the chosen models, these models were trained twice: a first time with the hour of the day as a feature, and a second time without that feature.

G.3 Model comparison

G.3.0.1 Performance metrics

Theoretical metrics

In order to compare the performance of the different models, three different indicators were used: the R^2 score, the mean-absolute error (MAE) and a daily Pearson correlation score (DPC).

The R^2 score, also known as the coefficient of determination, is defined as the unity minus the ratio of the residual sum of squares over the total sum of squares:

$$\mathbf{R}^2(h, \mathcal{D}) = 1 - \frac{\sum_{(x,y) \in \mathcal{D}} (y - h(x))^2}{\sum_{(x,y) \in \mathcal{D}} (y - \bar{y})^2} \quad (\text{G.3})$$

where y is the value to be predicted, \bar{y} its mean and $h(x)$ is the model prediction and \mathcal{D} the dataset. It ranges from $-\infty$ to 1, 1 being a perfect prediction and 0 meaning that the prediction does as well as constantly predicting the mean value for each occurrence. In the case of a negative R^2 , then the model has a worse prediction than if it were predicting the mean value for each occurrence and therefore yields no useful predictions.

Regarding the mean-absolute error, the smaller its value is, the more accurate the prediction is. It is calculated using the following formula:

$$\text{MAE}(h, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} |h(x) - y| \quad (\text{G.4})$$

The daily Pearson correlation score is an average of the usual Pearson correlation score applied to non-overlapping subsets \mathcal{D}_d of \mathcal{D} , with each subset \mathcal{D}_d containing the data from an entire day d and $\mathcal{D} = \bigcup_{d \in D} \mathcal{D}_d$. It gives an indication of how well the curve of the predicted number of arriving passenger follows the actual curve of arriving passengers. The closer the score is to 1, the better the prediction is. It is calculated using the following equations:

$$r(h, \mathcal{D}_d) = \frac{\sum_{(x,y) \in \mathcal{D}_d} (h(x) - \bar{h}_d)(y - \bar{y}_d)}{\sqrt{\sum_{(x,y) \in \mathcal{D}_d} (h(x) - \bar{h}_d)^2} \sqrt{\sum_{(x,y) \in \mathcal{D}_d} (y - \bar{y}_d)^2}} \quad (\text{G.5})$$

$$\text{DPC}(h, \mathcal{D}) = \frac{1}{|D|} \sum_{\mathcal{D}_d} r(h, \mathcal{D}_d) \quad (\text{G.6})$$

where \bar{h}_d (resp. \bar{y}_d) is the average of $h(x)$ (resp. y) over \mathcal{D}_d .

Operational metrics

Airport management being a balance between minimizing costs and maximizing the service given to passengers, two additional metrics were introduced based on these operational considerations. These metrics are simplified versions of reality since the security agent providers do not share their calculation processes and the actual staffing of checkpoints is decided at a tactical level.

From a cost perspective, the key figure is the number of security agents necessary for a smooth operation. Agents being paid per hour, the cost metric considered is the total number of agent-hours induced by the predicted passenger arrivals. A smooth operation is here defined as a nominal passenger flow f_N , which has a unit of passengers per line per ten minutes. These flows are specific to each security checkpoint and are determined by the airport management. Airports also define a peak-time passenger flow f_P that security agents should be able to cope with when needed. From these nominal flows and the number of expected passengers p_t at time step t , it is then possible to compute the number of lines n_t required to achieve this flow: $n_t = \frac{p_t}{f_N}$. Assuming that each line is staffed by five security agents yields the number of agents required at each time step t . Each time steps being of ten minutes, it is then necessary to divide the resulting cost by six to obtain the agent-hour cost. The total cost metric C_T can be resumed by the following equation:

$$C_T = \frac{5}{6} \sum_t \frac{p_t}{f_N} \quad (\text{G.7})$$

From a quality perspective, the key figure is the average waiting time at the security

checkpoints. In order to estimate it at each time step, the following simplified queuing model is considered. At time step t , y_t passengers arrive at the checkpoint SC adding to the r_{t-1} passengers not processed during the previous time step. Under nominal conditions, $n_t \cdot f_N$ passengers are processed during a ten minute time step, where n_t is the number of lines estimated for the cost calculation. Peak-time conditions were defined here as time steps where the remaining number of passengers r_{t-1} was greater than the nominal flow f_N . Under peak-time conditions, the number of processed passengers becomes $\max(n_{t-1}, n_t) \cdot f_P$, i.e. the number of lines kept open stays the same if it was initially supposed to become smaller. If the prediction indicated that no lines should be open and that there are in fact passengers, then either the lines open in the previous time step are kept open if any, or one line is opened.

The processed number of passengers π_t at time step t can therefore be calculated as followed:

$$\pi_t = \begin{cases} \max(n_{t-1}, n_t) \cdot f_P & \text{if } r_{t-1} > f_N \text{ and } n_t > 0 \\ n_{t-1} \cdot f_N & \text{if } n_t = 0 \text{ and } n_{t-1} > 0 \\ f_N & \text{if } n_t = 0 \\ n_t \cdot f_N & \text{otherwise} \end{cases} \quad (\text{G.8})$$

The average wait time τ_t during a time step t can be computed using the following equation:

$$\tau_t = \sum_{i=1}^{y_t+r_{t-1}} \frac{i}{\pi_t} = \frac{y_t + r_{t-1} + 1}{2\pi_t} \quad (\text{G.9})$$

The overall quality metric Q_T is then calculated by taking the average of all τ_t .

The passenger flow model at a checkpoint is represented as an automata in Figure G.4.

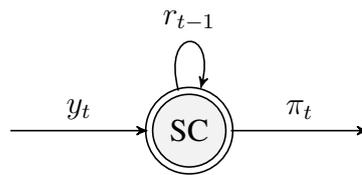


Fig. G.4 Model of the passenger flow at a security checkpoint.

G.3.1 First results

All three models presented in Section G.2 were trained using data from February 2017 to December 2018 and tested on the months of January to March 2019 using the performance metrics presented in Section G.3.0.1. These metrics were also applied to the current model in use at Charles De Gaulle airport for comparison. Based on operational observations, the output of the neural nets was forced to 0 when the hour of the day was between 00:00 and 04:00.

	LSTM (MSE)				LSTM (0.5-PMSE)				Random Forest			
	With Hour		Without Hour		With Hour		Without Hour		With Hour		Without Hour	
	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
C2G-Depart	0.736	4.02	0.732	4	0.768	3.75	0.754	3.85	0.721	4.27	0.712	4.68
C2E-PorteM	0.822	11.08	0.887	9.02	0.796	11.92	0.866	9.82	0.853	11.68	0.808	17.34
C2E-PorteL	0.674	13.96	0.641	14.58	0.578	15.8	0.636	14.67	0.684	14.21	0.558	19.58
C2F-Centraux	0.834	18.34	0.861	16.79	0.851	17.34	0.86	16.62	0.788	21.48	0.826	21.08
C2E-PorteL-CNT	0.436	16.54	0.474	16.14	0.426	16.6	0.425	16.49	0.471	16.36	0.475	17.62
C2E-GalerieEF	0.667	15.32	0.667	15.46	0.656	15.46	0.626	15.95	0.68	15.61	0.608	20.03
C2E-Puits2E	0.411	3.77	0.37	3.76	0.395	3.78	0.375	3.77	0.405	4.09	0.381	4.56
C2E-PorteK	0.688	18.37	0.758	15.63	0.662	15.67	0.769	15.26	0.726	16.73	0.647	23.53

Fig. G.5 Comparison of the models using or not the hour in the training set. Green color cells correspond to the model kept in the following study. Bold cells correspond to the best models.

Hour of the day

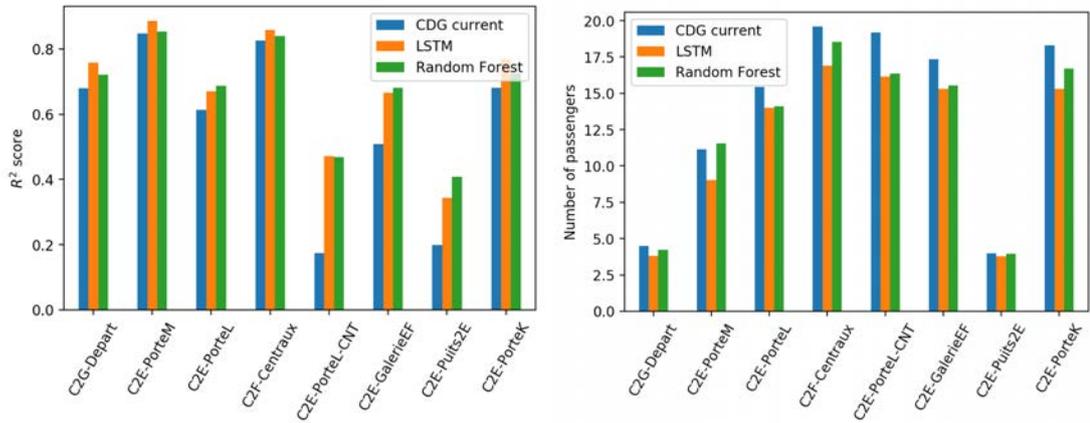
Table G.5 summarizes the performances of the three developed learning models based on two of the three mathematical metrics. This table enables a quick comparison of the use of the hour of the day as a feature. For the upcoming analysis, only one LSTM model and one Random Forest regressor were kept per checkpoint based on their MAE. The kept models have their performance cells highlighted in green, while the best of all models are also highlighted in bold. A first observation is that the influence of the hour of the day is not the same for Random Forests and for neural networks. For seven checkpoints over eight, using the hour of the day highly improves the Random Forest's performance. On the other side, six checkpoints over eight with LSTM (MSE and 0.5-PMSE) have better scores without the hour of the day.

Mathematical Performance Metrics

Figure G.6 presents the performance of the current model and the kept models from Section G.3.1 using the mathematical metrics introduced in Section G.3.0.1. From a R^2 score perspective, both the LSTM and Random Forest models outperform the current prediction model with improvements ranging from 0.01 for C2E-PorteM to 0.3 for C2E-PorteL-CNT. Regarding the mean-absolute error performance, the LSTM nets outperform once more the current model while the Random Forest regressors have higher errors for two of the checkpoints (C2E-PorteM and C2E-Puits2E). The LSTM reduces the mean-absolute errors from 5.6% (C2E-Puits2E) to 18.9% (C2E-PorteM) compared to the current model: LSTM net have a mean-absolute error of less than seventeen passengers per ten minutes for all checkpoints while the current model has an error greater than seventeen passengers per ten minutes for half of the checkpoints. Finally, regarding the daily Pearson correlation score, LSTM model outperforms the current prediction model at every checkpoint, while the Random Forest regressor outperforms it for seven checkpoints out of eight.

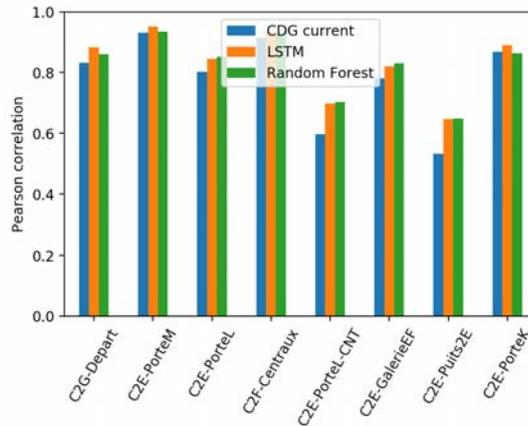
Operational Performance Metrics

Using the simplified operational metrics introduced in Section G.3.0.1, the difference in performance is less straightforward. Figure G.7 shows the comparison of the cost metric (i.e. the number of agent-hour over the three months) per checkpoint as



(a) Comparison of the R^2 score

(b) Comparison of mean absolute error



(c) Comparison of daily Pearson correlation score

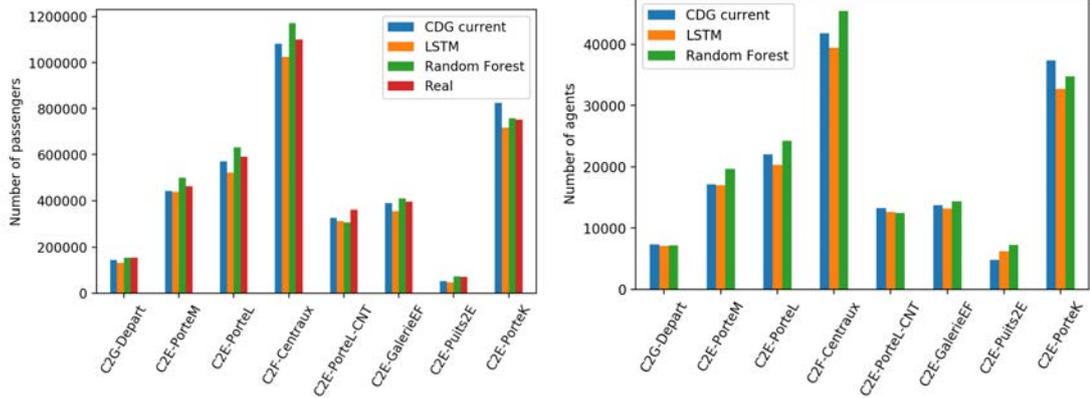
Fig. G.6 Comparison per checkpoints of different mathematical metrics for the three considered models.

well as the comparison of the quality metric. Figure G.7a presents the comparison of the total number of predicted passengers per checkpoints along with the actual number of passengers for comparison. A first observation is that the LSTM nets tend to underestimate the number of passengers regardless of the loss function considered, while the Random Forest regressors overestimate the number of passengers.

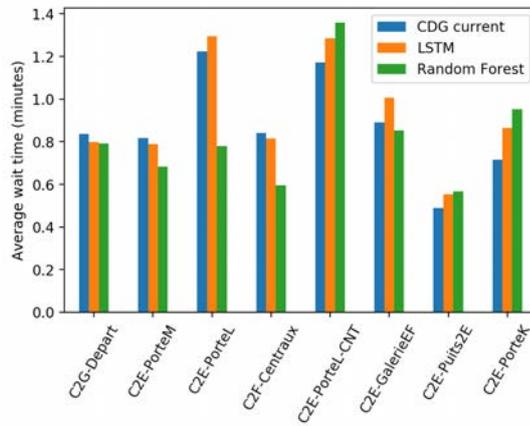
Since LSTM nets tend to underestimate the number of passengers more than the current model, it is also reflected from a cost perspective in Figure G.7b: For seven of the checkpoints, the number of agent-hours required based on the neural nets is less than the number required based on the current model. For C2E-Puits2E, the number of required agent-hours is greater than the current model, a paradox illustrating the specificity of that terminal and further analyzed in Section G.3.2.

Synthesis

Figure G.8 shows the performance difference between the neural networks and the current prediction model, for all the metrics, and all the security checkpoints. All the



(a) Comparison of the number of predicted passengers. (b) Comparison of the number of estimated hour agents.



(c) Comparison of the number of the estimated average wait times.

Fig. G.7 Comparison per checkpoints of different operational metrics for the three considered models.

metrics are normalized by the current prediction model value, except for the R^2 score, and the correlation score since they already have consistent magnitude and a norm lower than 1. The normalization enables comparison between security checkpoints. The difference is explained in percentage of improvement relative to the current model, except for the R^2 score and the correlation score where it is the improvement difference in percentage (norm lower than 1). In addition, the performance sign is selected such that a positive sign corresponds to a metric improvement. Finally, the performance difference is displayed with color from green when the improvement is greater than 20% to red when the best model deteriorates the performance more than 20%

For three security checkpoints over eight (C2G-Depart, C2E-PorteM, C2F-Centraux), LSTM models outperforms the current prediction model for all the performance metrics. For four over eight (C2E-PorteL, C2E-PorteL-CNT, C2E-GalerieEF, C2E-Portek) LSTM models outperform current model for all the metric excepted the waiting time metric, which is deteriorated more than 10% in half of the cases (C2E-GalerieEF, C2E-

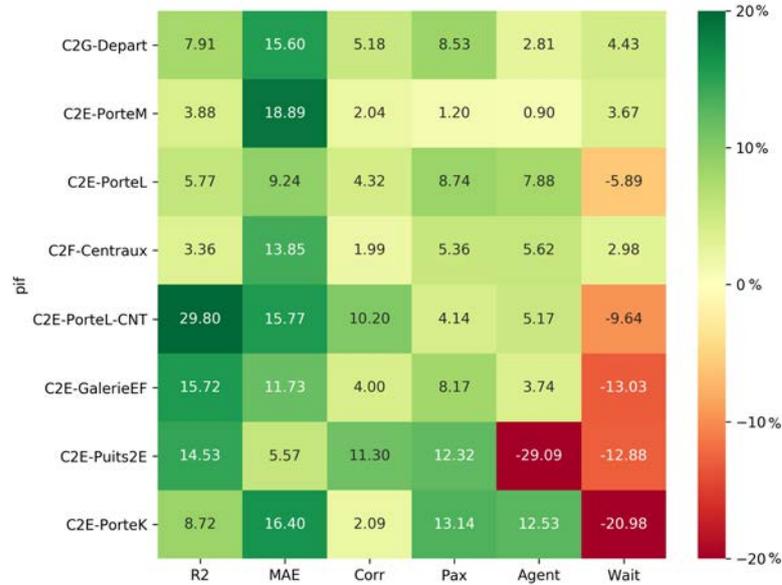


Fig. G.8 Heatmap visualization of the performance difference between the LSTM models and the current predictive model.

PorteK). Finally, at security checkpoint C2E-Puit2E, the performance metric is highly deteriorated for the agent number (-29%) and waiting time (-12.9%). This particular behavior will be explained in Section G.3.2.

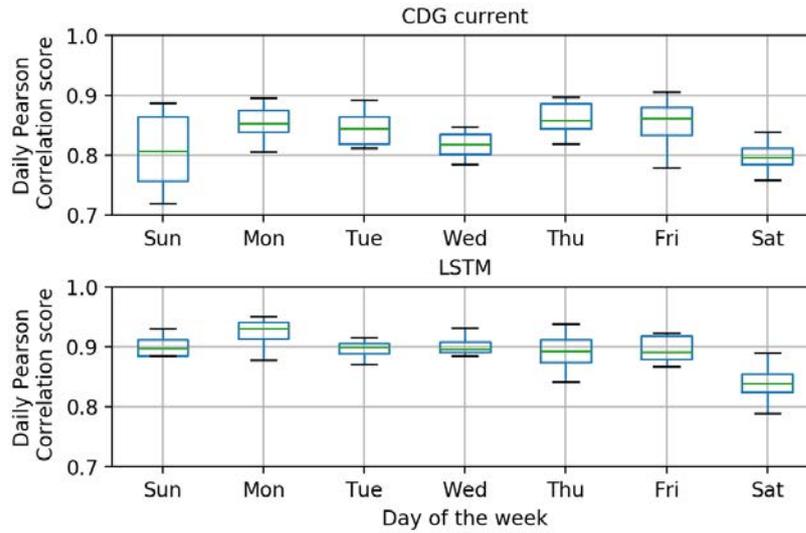
G.3.2 Case Study

In this section, two security checkpoints were selected with respect to their performances for a further analysis. C2G-Depart was chosen to illustrate the good results of the LSTM model while C2E-Puits2E was chosen to better understand why the LSTM model does not outperform the current model from an operational perspective.

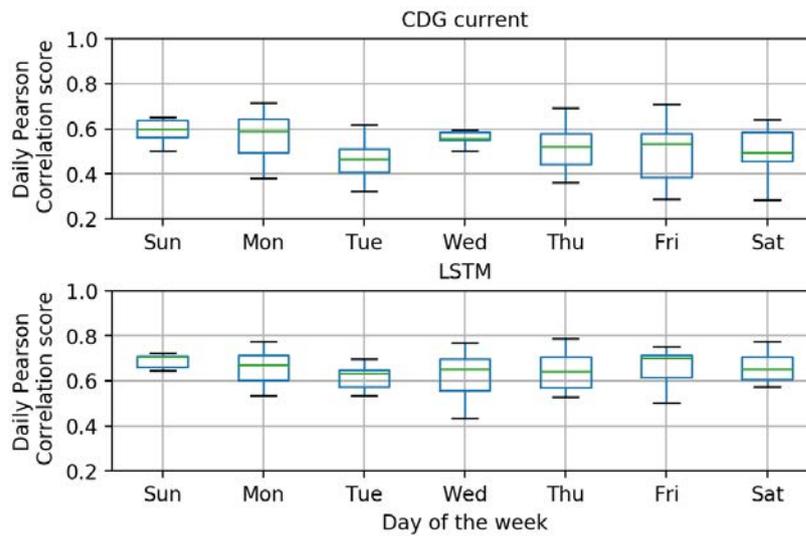
G.3.3 Daily analysis

A first step in understanding the differences in performance is to analyze the performances of the two models (LSTM and current) on a less aggregated level such as the different days of the week. Figure G.9 shows the distribution of the daily Pearson correlation score per day of the week for the two chosen security checkpoints. It confirms the previous observation that the LSTM model is overall better than the current model with this metric, while adding some information on how this improvement is structured.

Regarding C2G-Depart, Figure G.9a shows that both models are less precise on Saturdays compared to other days, though the LSTM model reduces the score variability on that day. An important improvement can be seen for Sundays: the current model has a lower score with a large variability, whereas the LSTM model reduces drastically that variability and improves the median score of 0.1.



(a) C2G-Depart



(b) C2E-Puits2E

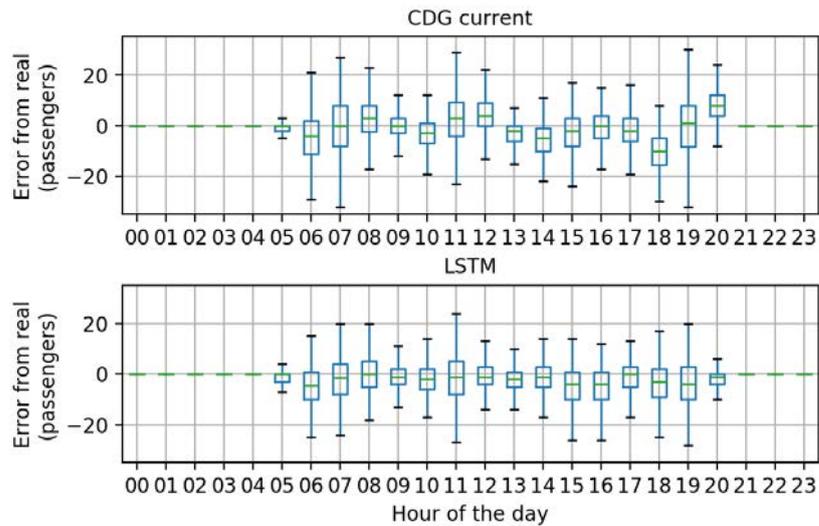
Fig. G.9 Daily correlation distribution per day of the week for C2G-Depart and C2E-Puits2E.

Regarding C2E-Puits2E, Figure G.9b shows that the LSTM model manages to reduce variability on most days, with an important reduction on Fridays. Wednesdays show an opposite behavior: though the LSTM model does increase the median correlation score, it also triples the score variability.

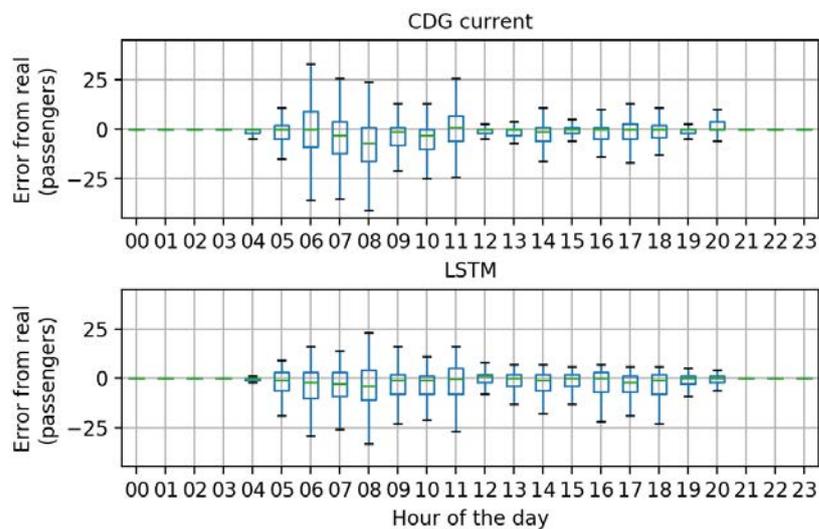
G.3.4 Hourly analysis

A similar analysis can be conducted by aggregating the performance metrics per hour of the day. Figure G.10 shows the hourly distribution of the error in predicting the number of arriving passengers for the current model and the LSTM model at the two chosen security checkpoints. It confirms the LSTM tendency to underestimate the number of

passengers: All medians are at or below zero for the LSTM while the current model tend to overestimate for five hours out of the sixteen considered hours for C2G-Depart. For C2E-Puits2E, both models have median errors at or below zero, however the LSTM model variations are shifted towards the negative with a smaller tendency to overestimation, which is indicated by smaller upper whiskers. This underestimation can be seen as a lower cost, since the predicted number of passengers determines the number of required agents.



(a) C2G-Depart

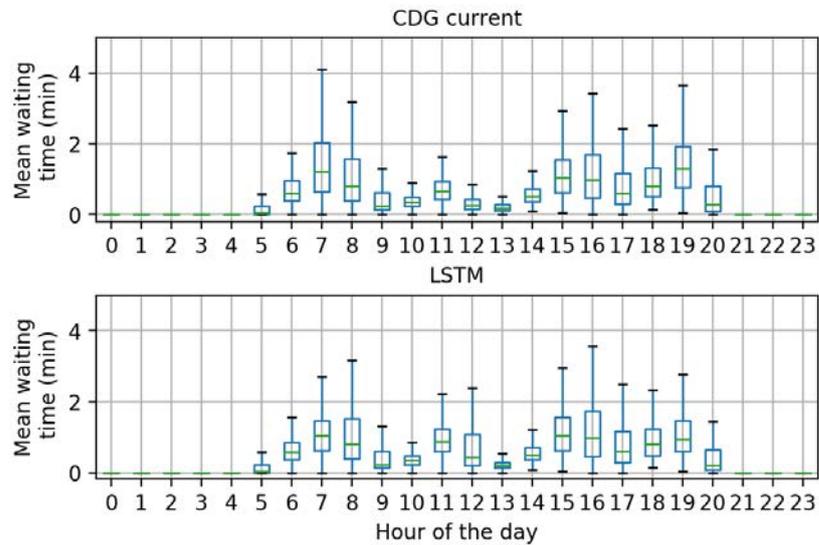


(b) C2E-Puits 2E

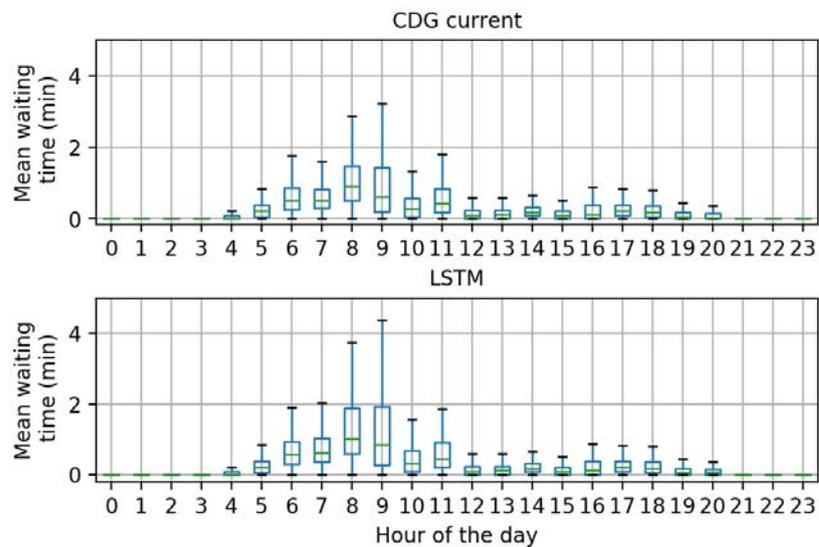
Fig. G.10 Hourly passenger error boxplots comparison between the current model and the neural net trained with a mean squared error loss function at two different checkpoints.

Figure G.11 shows the hourly distribution of the average wait time using the predictions from the current model and the LSTM model. Combining Figures G.11 and G.10 makes the impact of underestimation clearer on the quality of service. Underestima-

tions in the number of passengers is associated with a higher median average wait time, which is then propagated in the following hours. For C2G-Depart, the underestimations at 3pm and 7pm on Figure G.11a are clearly associated with a rise and propagation of the average wait time on Figure G.10a. For C2E-Puits2E, it is most visible for the underestimation at 7am for both models.



(a) C2G-Depart



(b) C2E-Puits 2E

Fig. G.11 Hourly average wait time boxplots comparison between the current model and the neural net trained with a mean squared error loss function at two different checkpoints.

This analysis could be used to further improve the derived models and the determination of the number of required agents. By highlighting hours of the days where the models are known to underestimate (resp. overestimate) the number of passengers, it should be possible to mitigate this underestimation (resp. overestimation) by adjusting

the predicted value or by adapting accordingly the number of required agents for these specific periods.

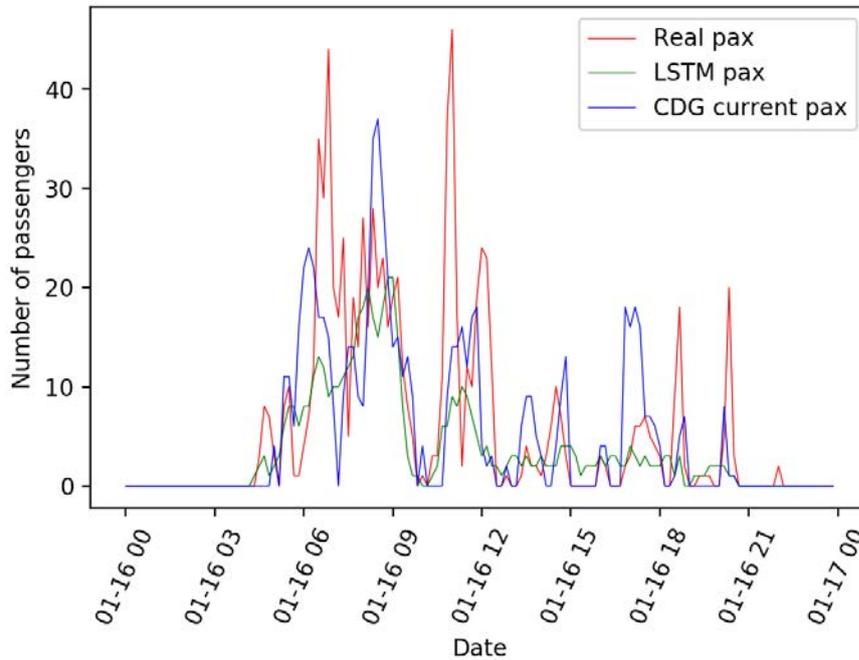
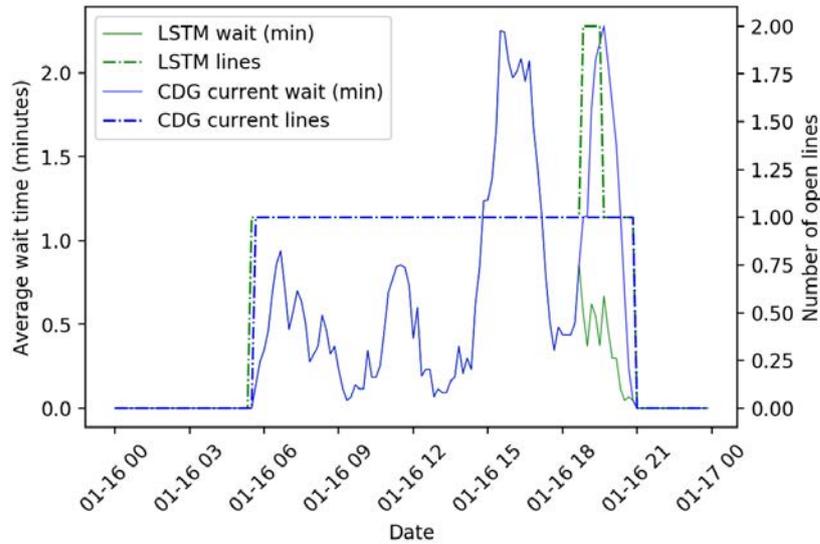


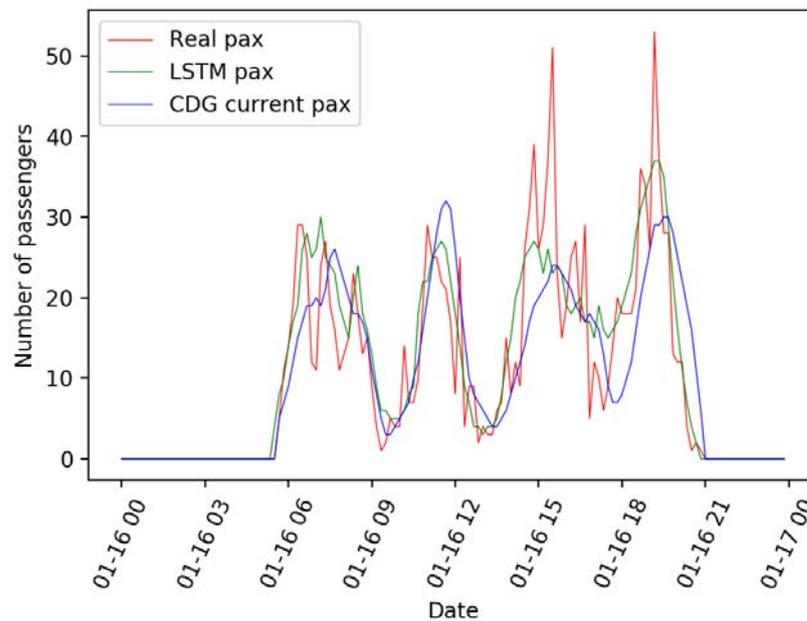
Fig. G.12 Hourly comparison of the predicted number of passengers between the current model and the neural net at C2E-Puits2E on January 16th 2019.

In order to better understand the differences in performance for these two checkpoints, the estimated number of passengers is plotted over a day (January 16th, 2019) in Figure G.12 for C2E-Puits2E and in Figure G.13b for C2G-Depart. Figure G.12 highlights the difficulty of predicting the number of passengers for C2E-Puits2E: There are irregular yet continuous arrival spikes in the early morning (5am-9am) and then the rest of the day is composed of arrival spikes of varying amplitudes with periods with no passengers at all. From a prediction performance perspective, Figure G.12 clearly illustrates the paradox of predicting less passengers while requiring more agents. The LSTM model underestimates more the passenger arrival spikes in the early morning than the current model, and estimates a low number of passengers for the rest of the day though never predicting zero arrivals. This means that agents are required all day long from the LSTM perspective, while the current model captures better the periods with no arrivals, enabling an economy of agents. A potential improvement of the LSTM model would be to hardcode the periods where operational expertise indicates that transfers within Terminal E are highly unlikely.

Regarding C2G-Depart, Figure G.13b is a good day example to understand the better performance of the LSTM model compared to the current model. There are four daily spikes in passenger arrival with varying amplitude, and though both models capture the number of spikes, the LSTM model yields a better estimation of the amplitude



(a) Comparison of the average wait time and number of open lines.



(b) Comparison of the predicted number of passengers.

Fig. G.13 Hourly comparison between the current model and the neural net trained with a mean squared error loss function at C2G-Depart on January 16th 2019

of each spike as well as their initial slope increase. This higher accuracy has a direct impact on the estimated wait time, as shown in Figure G.13a. The average wait time is identical for both model until the fourth spike, where the better estimation of the increase in passengers triggers the opening of a second line, which reduces the wait time by half compared to the current model.

G.4 Discussion and Conclusion

This appendix investigated predicting passenger flow at Paris Charles De Gaulle airport security checkpoints using LSTM neural networks. The models performance was evaluated over several theoretical and operational metrics. The overall results are promising since LSTM models outperform the current model for every checkpoints using the theoretical metrics and for three checkpoints out of eight, LSTM models outperform the current prediction model using all the considered metrics. Though the considered operational metrics were simplified, these results illustrate that implementing a better and accurate strategic passenger flow prediction would surely reduce operational cost while maintaining predefined standard regarding passengers waiting time.

The methodology presented in this study can still be enhanced and tuned to be efficient and dedicated on specific cases. Future works should investigate a more elaborated queuing model or simulation. In addition, the models could be validated with real experimentation in the operations. Further works could be done on the neural network architecture and learning, or with expert to tune the models bringing relevant information to improve the prediction (hybrid models).