NNT : 2017SACLX082

THESE DE DOCTORAT
DE L'UNIVERSITE PARIS-SACLAY

préparée à

L'ÉCOLE POLYTECHNIQUE

ÉCOLE DOCTORALE N°574
École doctorale de mathématiques Hadamard (EDMH)

Spécialité de doctorat : Mathématiques

par

Cédric Rommel

Exploration de données pour l'optimisation de trajectoires aériennes

Thèse présentée et soutenue à Palaiseau, le – octobre 2018.

Après avis des rapporteurs:

Prénom NOM (Affil.)
Prénom NOM (Affil.)

Composition du jury:

Prénom NOM	(Affil.)	Président du jury
Prénom NOM	(Affil.)	Rapporteur
Prénom NOM	(Affil.)	Examinatrice
Baptiste GREGORUTTI	(Safety Line)	Examinateur
Frédéric BONNANS	(CMAP - Ecole Polytechnique, INRIA)	Directeur de thèse
Pierre MARTINON	(CMAP - Ecole Polytechnique, INRIA)	Co-directeur de thèse

Contents

1 INTRODUCTION **5**

1.1 Enjeux environnementaux et économiques 5
1.2 Le contexte industriel: OptiClimb 6
1.3 Données de vol 8
1.4 Contexte mathématique 11
1.4.1 L'optimisation de trajectoires 11
1.4.2 L'apprentissage statistique 14
1.5 Contributions et organisation du manuscrit 16
1.5.1 Axes d'étude 16
1.5.2 Contributions de la première partie 16
1.5.3 Contributions de la deuxième partie 17

I AIRCRAFT DYNAMICS IDENTIFICATION **19**

2 DATA PREPROCESSING **21**

2.1 Introduction 21
2.2 Statistical learning prerequisites 23
2.2.1 Regularized regression framework 23
2.2.2 Model selection using cross-validation 24
2.3 Smoothing splines 25
2.4 Preprocessing steps 26

3 AIRCRAFT DYNAMICS IDENTIFICATION **29**

3.1 Introduction 30
3.2 Dynamics modeling 32
3.2.1 Possible assumptions 32
3.2.2 Flat Earth dynamics in vertical plane with no wind 32
3.2.3 Dynamics with wind 34
3.2.4 2D flight with non-zero bank angle 35
3.2.5 3D dynamics 36
3.2.6 Spheric rotating earth dynamics 38

3.2.7 Projection of the wind speed onto the ground frame of reference . . 42
3.3 Description of our identification problem . . . 43
3.4 Parametric hidden models . . . 45
3.4.1 Restriction to parametric models . . . 45
3.4.2 Flight mechanics models . . . 45
3.4.3 Remark on monomials selection . . . 46
3.5 Aircraft systems identification state-of-the-art . . . 47
3.5.1 The Output-Error Method . . . 47
3.5.2 The Filter-Error Method . . . 49
3.5.3 The Equation-Error Method . . . 50
3.6 Nonlinear multi-task regression . . . 52
3.6.1 Baseline regression method: Ordinary Least-Squares . . . 52
3.6.2 Multi-task framework . . . 54
3.6.3 Maximum Likelihood Estimators . . . 56
3.7 Quality criteria design . . . 59
3.7.1 Static criterion . . . 59
3.7.2 Dynamic criterion . . . 59
3.8 Numerical benchmark . . . 61
3.9 Conclusion . . . 63

4 STRUCTURED MULTI-TASK FEATURE SELECTION 67
4.1 Introduction . . . 68
4.2 Feature selection . . . 68
4.2.1 A feature selection categorization . . . 68
4.2.2 The Lasso . . . 69
4.2.3 Inconsistency in high correlation setting . . . 70
4.2.4 A resampling adaptation: the Bolasso . . . 71
4.2.5 An efficient implementation: LARS . . . 72
4.3 Application to aircraft dynamics identification in a single-task setting . . . 73
4.4 Block-sparse estimators . . . 76
4.4.1 Structured feature selection . . . 76
4.4.2 Linear multi-task regression framework . . . 77
4.4.3 Hidden functions parametric models . . . 78
4.4.4 Block-sparse Lasso . . . 79
4.4.5 Bootstrap implementation . . . 81
4.5 Identifiability enhancement . . . 81
4.5.1 The issue with the predicted hidden functions . . . 81
4.5.2 Additional L2 Regularization . . . 82
4.5.3 Block coordinate descent . . . 83
4.6 Numerical results . . . 83
4.6.1 Experiments design . . . 83

4.6.2 Results . . . 84
4.7 Conclusion . . . 84

II SIMULATED TRAJECTORIES ASSESSMENT 91

5 PROBABILISTIC TRAJECTORY ASSESSMENT 93
5.1 Motivation . . . 94
5.2 The trajectory acceptability problem . . . 94
5.3 Marginal Likelihood estimators . . . 95
5.3.1 Mean Marginal Likelihood . . . 95
5.3.2 Empirical Version . . . 97
5.3.3 Consistency of the marginal density estimations . . . 98
5.3.4 Proof of the consistency of the marginal density estimation (theorem 5.3.6) . . . 100
5.4 Choice of the Marginal Density Estimator . . . 105
5.4.1 Parametric or nonparametric ? . . . 105
5.4.2 Kernel estimator . . . 106
5.4.3 Self-consistent kernel estimator . . . 113
5.5 Application to the Assessment of Optimized Aircraft Trajectories . . . 120
5.5.1 Experiments Motivation . . . 120
5.5.2 Experiments Design . . . 121
5.5.3 Alternate Approaches Based on Standard Methods . . . 121
5.5.4 Algorithms Settings . . . 122
5.5.5 Results and Comments . . . 125
5.6 Conclusions . . . 127

6 OPTIMIZATION OF ACCEPTABLE TRAJECTORIES 133
6.1 Introduction . . . 134
6.2 Aircraft Trajectory Optimization as an Identified Optimal Control Problem 134
6.3 A Parametric Marginal Density Estimator . . . 135
6.3.1 The Gaussian Mixture Model . . . 135
6.3.2 The Expectation-Maximization Algorithm . . . 137
6.3.3 More details on the derivation of the EM algorithm . . . 138
6.4 Application to an Aircraft Minimal-Consumption Problem . . . 139
6.4.1 Experiments Description . . . 139
6.4.2 Data Description . . . 139
6.4.3 Choice of the “time” variable . . . 140
6.4.4 Algorithm Settings . . . 140
6.4.5 Numerical Comparison Between MML using Gaussian Mixture and Self-Consistent Kernel Estimators . . . 141
6.4.6 Penalized Optimization Results . . . 142

6.5 Conclusion 143

Chapter 1

INTRODUCTION

Résumé: *Dans ce chapitre nous introduisons les objectifs de la thèse en présentant ses enjeux ainsi que les divers éléments décrivant le cadre applicatif et théorique dans lequel elle s'inscrit. Nous y présentons les deux axes d'étude qui ont été abordés: l'identification de modèles d'avions pour l'optimisation de trajectoires et l'évaluation de la validité des trajectoires optimisées. Nous proposons enfin un résumé détaillé des contributions contenues dans chaque chapitre.*

Contents

1.1 Enjeux environnementaux et économiques 5
1.2 Le contexte industriel: OptiClimb 6
1.3 Données de vol 8
1.4 Contexte mathématique 11
1.4.1 L'optimisation de trajectoires 11
1.4.2 L'apprentissage statistique 14
1.5 Contributions et organisation du manuscrit 16
1.5.1 Axes d'étude 16
1.5.2 Contributions de la première partie 16
1.5.3 Contributions de la deuxième partie 17

1.1 Enjeux environnementaux et économiques

D'après le rapport de 1999 du Groupe d'experts intergouvernemental sur l'évolution du climat (GIEC) [1], le transport aérien était déjà responsable en 1992 de 13% du CO_2 émis par des moyens de transports, participant à hauteur de 2% des émissions totales cette année. Cela fait de l'avion le moyen de transport le plus polluant en termes de masse de CO_2

1. www.ipcc.ch/pdf/special-reports/spm/av-fr.pdf

émise par passager et kilomètre parcouru, à savoir 223 g CO_2éq/passager-km [2], à comparer à 120 g CO_2/passager-km en moyenne pour une voiture [3] et à 4 g CO_2éq/passager-km pour un TGV [2]. Il est estimé par le GIEC que ces émissions seront multipliées par 10 d'ici 2050 [1], ceci étant largement dû à la forte croissance du nombre de voyageurs, qui est prévu de doublé dans les 20 prochaines années [4].

Ce sujet est d'autant plus important pour les acteurs du secteur, que l'Organisation de l'aviation civile internationale (OACI) a voté en octobre 2016 la mise en place d'un mécanisme mondial de compensation des émissions. À partir de 2020, les états signataires seront ainsi dans l'obligation de réduire en 80% les émissions de CO_2 issues du transport aérien international à horizon 2035 [5]. Par ailleurs, notons que la quantité de CO_2 émise par un avion est fortement liée à sa consommation de carburant, alors que celle-ci a représenté entre 17 et 36% du coût d'exploitation des compagnies aériennes entre 2004 et 2018 [6]. De ce fait, la réduction de la consommation et des émissions de CO_2 sont aujourd'hui de vrais enjeux économiques et environnementaux pour les compagnies aériennes.

La conception de nouveaux avions et moteurs plus performants par les constructeurs et équipementiers constitue une façon d'attaquer le problème. Cette stratégie est par contre très coûteuse pour les avionneurs et plutôt longtermiste, d'autant plus que les cycles de développement en aéronautique sont de l'ordre de la décennie. Dans cette thèse nous ne nous intéressons pas à ces aspects de la réduction de la consommation, mais plutôt aux économies qui peuvent être faites au niveau de l'exploitation par les avionneurs de leur flotte déjà existante. En effet, les mesures de ce type semblent plus simples et moins coûteuses à mettre en place que la modernisation du matériel. Dans cette catégorie d'approche, nous pouvons citer l'optimisation du remplissage des vols, ainsi que l'instauration de bonnes pratiques auprès des pilotes. Un autre levier d'action possible est l'optimisation du plan de vol et de la trajectoire suivie par les avions. C'est dans cette optique que Safety Line a conçu le produit OptiClimb, qui est au cœur de nos travaux.

1.2 Le contexte industriel: OptiClimb

Depuis 2014, Safety Line propose aux compagnies aériennes le service OptiClimb, permettant d'optimiser les profils de vol pendant la montée. En effet, la montée est une des phases de vol les plus consommatrices en carburant, car c'est à ce moment qu'on exige le plus des moteurs. De plus, alors que les compagnies low-cost continuent d'afficher une forte

2. www.bilans-ges.ademe.fr/fr/basecarbone/donnees-consulter/choix-categorie/categorie/7
3. "Les comptes des transports en 2010", RéférenceS, commissariat général au développement durable, Service de l'observation et des statistiques, ministère de l'écologie, du développement durable et des transports, juillet 2011, p.137
4. www.iata.org/pressroom/pr/Pages/2017-10-24-01.aspx
5. www.lemonde.fr/climat/article/2016/10/06/climat-l-aviation-civile-adopte-le-gel-des-emissions-carbonne_5009519_1652612.html
6. www.iata.org/pressroom/facts_figures/fact_sheets/Documents/fact-sheet-fuel.pdf

croissance [7], elles proposent surtout des vols court et moyen-courrier pour lesquelles les potentielles économies réalisées pendant la montée ont un fort impact. OptiClimb permet à ces compagnies d'économiser entre 70 et 90 kg de carburant par vol, ce qui approche les 10% de la consommation totale au cours d'une montée. Ainsi, les économies quotidiennes peuvent s'élever à 7-9 tonnes de carburant pour une compagnie effectuant 100 vols par jours [8].

Figure 1.1 – Logo OptiClimb.

Par ailleurs, des quantités faramineuses de données sont enregistrées lors de chaque vol et celles-ci sont encore peu exploitées par les opérateurs. Les nouvelles générations d'appareils sont équipés d'une multitude de capteurs qui permettent de recueillir 30 fois plus de données que la génération précédente. Il est ainsi prévu que, d'ici 2026, 5 à 8 teraoctets seront acquis à chaque vol [9]. Dans ce contexte, OptiClimb valorise les données de vols historiques pour estimer des modèles mathématiques individuels à chaque avion, de sorte à personnaliser l'optimisation des futures trajectoires. En effet, bien que des modèles généraux réalisés par les constructeur soient disponibles pour chaque type d'avion, des modèles individuels plus précis peuvent être estimés à partir des données de vol. Ceci est intéressant car le processus de fabrication des avions est encore en grande partie manuel, des aéronefs de même type étant donc assez différents dés la sortie de l'usine. De plus, deux avions de même type n'ont pas forcément le même âge, ni la même histoire en termes de maintenances et de vols réalisés, et ne vont donc pas se comporter de la même façon. Les modèles constructeurs considèrent des avions neufs, sans prendre en compte le vieillissement des appareils, et ne semblent donc pas être les mieux adaptés à ce type d'utilisation.
Les étapes de l'utilisation d'OptiClimb sont donc les suivantes:

1. La compagnie aérienne fournie les données de vol historiques de sa flotte à Safety Line, à partir desquels des modèles individuels de chaque avion sont réalisés;

2. Quelques minutes avant chaque nouveau vol effectué par la compagnie, celle-ci communique à Safety Line les informations nécessaires à l'optimisation, comme l'avion qui effectuera le vol, l'origine, la destination, la vitesse et l'altitude de croisière, etc.;

7. www.lesechos.fr/13/10/2017/LesEchos/22550-078-ECH_les-low-cost-continuent-la-conquete-du-ciel-francais.htm
8. www.safety-line.fr/des-solutions-adaptees/opticlimb/
9. www.forbes.com/sites/oliverwyman/2017/06/16/the-data-science-revolution-transforming-aviation/#3a48ba5a7f6c

3. Le problème d'optimisation de trajectoire est résolu en utilisant les modèles individuels préalablement construits;

4. La compagnie aérienne reçoit une liste de consignes simples que le pilote devra entrer dans le *Flight Management System* (FMS) pendant le vol, de sorte à suivre la trajectoire de montée optimale;

5. Une fois le vol effectué, les données recueillies pendant le vol intégrent la base de données de Safety Line, qui continue à mettre à jour les modèles des avions.

Cette thèse s'insère dans ce contexte industriel en visant à proposer des méthodes d'analyse de données et d'apprentissage statistique qui pourront être intégrées à OptiClimb au niveau des étapes 1 et 5, ainsi qu'entre les étapes 3 et 4.

AJOUTER IMAGE APPLI OPTICLIMB

1.3 Données de vol

Dans un avion de ligne, les données de vol sont recueillies au sein de deux types d'enregistreurs : le *Flight Data Recorder* (FDR), qui rassemble les réglages et mesures effectuées par l'avion au cours du vol, et le *Cockpit Voice Recorder* (CVR), qui enregistre les conversations au sein du cockpit. Ce sont ces deux dispositifs qui sont plus couramment connus sous le nom de *boites noires*. Elles ont originellement été conçues dans l'intention de servir pour les enquêtes lors d'incidents majeurs, mais l'utilité des données recueillies dans le FDR a été récemment étendue au domaine de la gestion de la sécurité.

Figure 1.2 – Flight Data Recorder (boîte noire).

En effet, avant d'être stockées dans ce dispositif, qui ne peut être accédé que par des enquêteurs, les enregistrements passent par un système appelé *Quick Access Recorder* (QAR) depuis lequel ils peuvent être téléchargés par la compagnie aérienne. Ces données comprennent plus de 1000 paramètres qui peuvent être des variables physiques, des alertes

ou des réglages réalisés par les pilotes. Les enregistrements sur lesquels nous nous sommes concentrés dans notre étude sont les signaux physiques disponibles. Ceux-ci sont souvent échantillonnés avec une période de 1 seconde, bien que des fréquences plus élevées soient observées pour les appareils plus récents. Notons aussi que certains de ces paramètres sont directement issues d'instruments de mesures de l'avion, comme par exemple le nombre de Mach mesuré par les sondes Pitot ou l'altitude barométrique mesurée par les altimètres. Ces enregistrements sont donc potentiellement perturbés par des erreurs de mesure, qui peuvent être de nature systématique ou stochastique. D'autres signaux ne sont pas directement mesurés, mais sont plutôt issus de calculateurs internes à l'avion, les modèles utilisés à cette étape étant inconnus. Par ailleurs, il est aussi important de noter que les données brutes issues du QAR sont codées en binaire sur plusieurs trames et que les processus mêmes d'enregistrement et de décodage de celles-ci sont aussi soumis à de potentielles erreurs. Nous avons essayé de corriger ces erreurs de mesure lors de la phase de prétraitement des données, décrite au chapitre 2.

Les signaux mesurés auxquelles nous nous intéressons particulièrement dans cette thèse correspondent à l'altitude h, le nombre de Mach M, la consommation de carburant C_f, la masse de l'avion m, le régime moteur N_1, la température statique SAT, l'assiette θ, le cap ψ, a vitesse du vent W et sa direction ψ_w. Nous donnons dans la suite un bref descriptif de ces grandeurs physiques.

- En ce qui concerne l'altitude, nous nous sommes intéressés à l'altitude barométrique uniquement, dont le calcul repose sur la pression ambiante captée par les baromètres de l'avion. Elle ne correspond pas à l'altitude radio, qui est mesurée par des émetteurs-récepteurs de radiofréquences situés sous l'avion. Alors que cette dernière indique la distance géométrique séparant l'avion du sol, l'autre utilise comme référence le niveau de la mer. Cette grandeur est souvent indiquée en pieds, bien que les signaux aient été convertis en mètres pour nos analyses.

- Le nombre de Mach correspond au rapport entre la vitesse du son dans l'air ambiant et la vitesse relative de l'avion par rapport à la masse d'air l'entourant. C'est donc une grandeur adimensionnelle. Il faut remarquer que même lorsque l'avion monte à une vitesse constante, son nombre de Mach augmente car le son se propage plus lentement dans l'air moins dense des plus hautes couches de l'atmosphère. Il est mesuré par les sondes Pitot, à partir de la différence entre la pression dynamique du flux d'air la percutant et la pression statique de l'air ambiant (voir figure 1.3).

- La consommation de carburant correspond au débit massique de carburant injecté dans les moteurs et est souvent indiquée en kilogrammes par secondes kg/s. Bien que les moteurs d'avion accélèrent et rejettent l'air capté par les turbines, ce ne sont pas des systèmes fermés, dans la mesure où la masse de carburant consommé intègre

10. Source: https://fr.wikipedia.org/wiki/Tube_de_Pitot#/media/File:Pitot_tube_with_Bernoullis_law_fr.svg

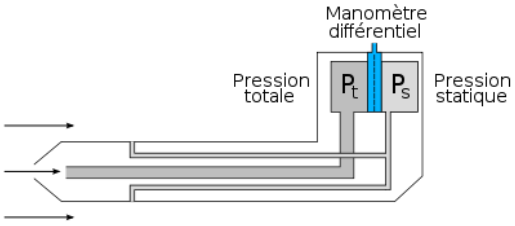


Figure 1.3 – Fonctionnement d'une sonde de Pitot. [10]

les émissions des moteurs. Cette grandeur est donc l'élément principal à l'origine de la variation de la masse de l'avion au cours du vol.

- Le régime moteur N_1 est une grandeur adimensionnelle comprise entre 0 et 1 pour chaque moteur. Il correspond au rapport entre la vitesse de rotation effective du turbofan et sa vitesse maximale. Plus intuitivement, il est équivalant au pourcentage de poussée demandée au moteur, par rapport à la poussée maximale que celui-ci peut fournir vis-à-vis des conditions de vols. En ce sens, il est réglé par le pilote à travers la position de la manette des gaz.

- La température statique est la température de l'air ambiant, indiquée en degrés Celcius $°C$ ou Farenheit $°F$ et convertie en Kelvins K pour certaines de nos analyses. L'adjectif "statique" permet de la distinguer de la température dynamique de l'air comprimé par le déplacement de l'avion, qui est plus élevée.

- L'assiette, aussi appelée tangage ou *pitch* en anglais, désigne l'angle entre l'axe de l'avion et l'horizontale (voir figure 1.4). Il est mesuré, entre autres, par les centrales inertielles embarquées dans l'avion. Celles-ci contiennent par exemple des gyroscopes et des accéléromètres, qui mesurent respectivement les vitesses angulaires de l'avion et les accélérations qu'il subit. En intégrant les vitesses angulaires et en utilisant les mesure de l'accéléromètre pour retrouver la verticale, il est donc possible de déduire les angles d'Euler décrivant l'orientation de l'avion. En pratique, ces mesures sont fusionnés avec bien d'autres informations, comme par exemple les signaux issus des magnétomètres et des capteurs GPS, de sorte à réduire l'imprécision des calculs.

- Le cap, en anglais *heading* ou *yaw*, correspond à l'angle séparant le nord magnétique de la direction horizontale vers laquelle se dirige l'avion (voir figure 1.4).

- Quant à la vitesse du vent, celle-ci est calculée par l'ordinateur de bord à partir de la différence entre la vitesse relative de l'avion et sa vitesse horizontale par rapport au sol. Le première est déduite, entre autres, du nombre de Mach et de l'altitude barométrique, alors que la deuxième est principalement issue de la fusion des don-

11. Source: https://en.wikipedia.org/w/index.php?curid=39309443

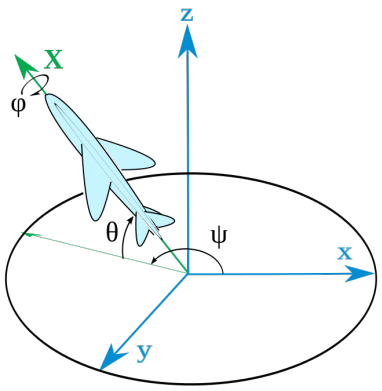


Figure 1.4 – Angles d'Euler avec certaines conventions. [11]

nées GPS et des signaux intégrés des accéléromètres. On n'a ainsi accès qu'à la composante horizontale de la vitesse du vent.

- Les mesures de direction du vent indiquent l'angle entre le nord magnétique et la direction d'où *arrive* le vent. Cet angle est mesuré par de nombreux capteurs d'incidence horizontaux sur le fuselage de l'aéronef (voir figure 1.5).

Figure 1.5 – Capteur d'incidence du vent. [12]

1.4 Contexte mathématique

1.4.1 L'optimisation de trajectoires

Le premier problème d'optimisation de trajectoire remonte à l'introduction par Bernoulli du problème de la courbe *brachistochrone* en 1696. Bien que les outils mathématiques nécessaires pour la résolution de cette classe de problèmes aient été développés bien plus tôt dans l'histoire, c'est surtout à l'époque de la conquête spatiale que des approches

12. Source: https://commons.wikimedia.org/wiki/File:JASDF_C-2(68-1203)_AOA_sensor_at_Miho_Air_Base _May_28,_2017.jpg

pratiques ont été conçues pour des contextes appliqués. En effet, l'arrivée des premiers ordinateurs a permis le développement des première méthodes de résolution numériques utilisées encore aujourd'hui dans des domaines très variés.

D'un point de vue mathématique, l'optimisation de trajectoire prend la forme d'un problème de commande optimale, où l'on modélise le système étudié par un système dynamique décrit par un certain nombre fini de variables $\boldsymbol{x}$, dites d'*état*, et contrôlé par d'autres variables $\boldsymbol{u}$, dites de *contrôle*. Celles-ci sont des fonctions d'une certaine variable indépendante, souvent appelée *temps*. On suppose qu'elles respectent la *dynamique* qui régit le système:

$$\dot{\boldsymbol{x}} = g(t, \boldsymbol{u}, \boldsymbol{x}). \tag{1.4.1}$$

Résoudre le problème d'optimisation de trajectoire consiste à trouver les fonctions d'état et de contrôle permettant de minimiser (ou maximiser) un certain critère

$$\min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{X}\times\mathbb{U}} \int_0^{t_f} C(t, \boldsymbol{u}(t), \boldsymbol{x}(t))dt, \tag{1.4.2}$$

tout en respectant à la fois la dynamique (1.4.1) et certaines autres contraintes:

$$\begin{cases} \boldsymbol{u}(t) \in U_{ad}, \quad \boldsymbol{x}(t) \in X_{ad}, \quad \text{for a.e. } t \in [0, t_f], \\ \Phi(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) \in K_\Phi, \\ c(t, \boldsymbol{u}(t), \boldsymbol{x}(t)) \leq 0, \quad \text{for all } t \in [0, t_f]. \end{cases} \tag{1.4.3}$$

Les fonctions C, g et c sont supposées continûment dérivables. Les ensembles X_{ad}, U_{ad} et K_Φ sont des ouverts d'espaces Euclidiens réels.

Dans notre cas précis, le critère d'optimisation dans (1.4.2) englobe la consommation de l'avion, sa vitesse horizontale et le temps mis pour effectuer la montée. En effet, notre but est de minimiser la consommation de carburant sans pour autant trop impacter la durée et la distance parcourue au cours de la montée. L'avion est modélisé pendant la phase de montée en choisissant comme variables d'état l'altitude h, la vitesse relative par rapport à la masse d'air, appelée *true airspeed* V, l'angle entre le déplacement de l'avion et l'horizontale, appelé *pente* γ (c.f. figure 1.6) et sa masse m, qui décroît au fur-et-à-mesure que du carburant est consommé: $\boldsymbol{x} = (h, V, \gamma, m)$. Le système est contrôlé par le régime moteur N_1 et l'angle d'attaque α, qui est l'angle entre les ailes de l'avion et la vitesse relative : $\boldsymbol{u} = (\alpha, N_1)$.

Concernant les contraintes (1.4.3), les ensembles X_{ad} et U_{ad} servent à définir le domaine opérationnel de vol. La fonction Φ et l'ensemble K_Φ modélisent une condition initiale bien précise correspondant à la fin du décollage, ainsi qu'une région acceptable pour la fin de la montée. En effet, ces conditions nous permettent de garantir que l'altitude en fin de montée correspond à l'altitude de croisière, et que le nombre de Mach final est proche

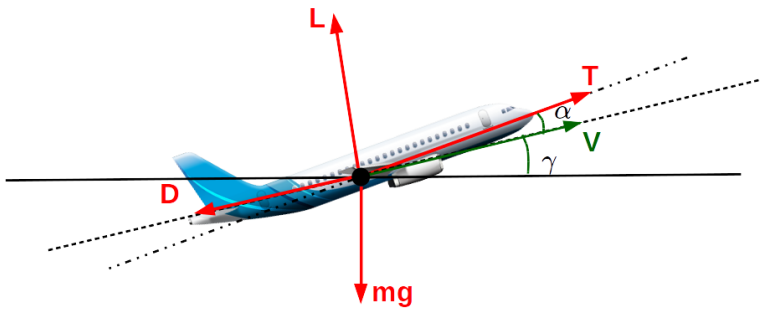


Figure 1.6 – Illustration de la dynamique d'un avion en montée.

du Mach de croisière. Quant à la fonctions c, elle définit des contraintes courantes qui permettent, par example, de fixer une vitesse maximale sur certaines plages d'altitude.

Étant donné les ambitions très pratiques de ces travaux, nous nous sommes concentrés sur des méthodes de résolution numériques. Celles-ci s'organisent en trois grandes classes de méthodes. Les méthodes *indirectes* reposent sur l'application du principe du maximum de Pontryagin et convertissent le problème d'optimisation en un *problème aux limites*, où l'on cherche à résoudre un système d'équations différentielles (souvent fortement non-linéaires) avec des contraintes aux limites. Ce sont souvent des problèmes difficiles dont la résolution peut à chaque fois conduire à des considérations différentes. Contrairement, les méthodes dites *directes* approximent le problème d'optimisation continu (1.4.1)-(1.4.3), qui est de dimension infinie, par un problème discrétisé. Ces méthodes sont souvent moins précises que les méthodes indirectes, mais elles sont réputées pour être plus robustes par rapport à l'initialisation et peuvent s'appliquer directement, sans étude théorique préalable. Pour ces raisons, les méthodes directes ont connu un franc succès depuis les années 80 pour des applications industrielles non critiques, alors que les méthodes indirectes sont encore privilégiées dans le secteur spatial. L'autre type de méthodes est la programmation dynamique, qui repose sur la résolution de l'équation d'Hamilton-Jacobi-Bellmann. Contrairement aux deux autres, celle-ci garanti de fournir une solution globale au problème d'optimisation. Cependant, c'est une méthode coûteuse pour laquelle un compromis doit être trouvé entre temps de calcul et précision du résultat. Le lecteur est renvoyé à Betts [1998, 2010] pour une description plus détaillée de ces méthodes numériques.

Tous les problèmes de contrôles résolus dans cette thèse on fait l'usage de l'outil d'optimisation Bocop [Bonnans et al., 2017], qui implémente la méthode de *transcription directe*. En tant que méthode directe, elle repose sur le principe de discrétisation du problème de commande optimale sur une grille homogène de temps. Les valeurs des états et des contrôles à chaque pas de discrétisation deviennent alors les variables d'un problème

d'optimisation nonlinéaire (NLP) à dimension finie, mais très grande. Malgré la grande dimension de ce problème, il est parcimonieux, ce qui rend sa résolution possible en un temps raisonnable. L'outil IPOPT [Wächter and Biegler, 2006], implémentant l'algorithme des points intérieurs primal-dual, est utilisé pour résoudre le problème transcrit. L'outil de différentiation automatique ADOL-C [Walther and Griewank, 2009] sert à calculer les dérivés de la dynamique, de l'objectif et des contraintes.

Bien que le problème de contrôle décrit dans cette section reste central dans nos travaux, nous ne discutons pas de sa résolution en tant que telle dans cette thèse, mais plutôt de l'utilisation des données de vol en amont et en aval du problème. La section suivante donne un aperçu des notions d'apprentissage statistique qui ont été utiles pour concevoir les outils traitant les données de vol, tandis que la section 1.5 donne plus de détails concrets sur les deux axes d'étude explorés dans cette thèse.

1.4.2 L'apprentissage statistique

L'apprentissage statistique, ou *machine learning*, est la discipline au carrefour des statistiques, de l'informatique et de l'optimisation, visant à construire des modèles mathématiques et algorithmiques à partir d'un ensemble de données. Les modèles créés peuvent être prévus pour répondre à des objectifs variés, comme la prédiction d'un événement futur, l'inférence d'un signal ou d'une caractéristique inobservable, ou encore la compréhension d'un phénomène ou d'une population. Il existe deux grandes classes de problèmes d'apprentissage: l'*apprentissage supervisé* et l'*apprentissage non-supervisé*.

Dans l'apprentissage supervisé, on s'intéresse à des systèmes ou phénomènes où des variables d'entrée et des variables de sortie ont été identifiées et on cherche à approximer une fonction les reliant. Les entrées sont souvent appelées *covariables* ou *features*, alors que les sorties sont parfois appelées *targets*. Toutes ces grandeurs sont considérées comme des variables aléatoires. L'adjectif supervisé vient du fait qu'on suppose avoir accès à un jeu de données *étiquetées*, c'est-à-dire contenant des observations à la fois des entrées et des sorties. On appelle ce jeu de données le *jeu d'entraînement* ou de *training*, vu qu'il est utilisé pour régler le modèle. Un bon algorithme ou *estimateur* sera celui capable de bien *généraliser* ses prédictions à des données issues du même phénomène auxquelles il n'a pas encore été exposé. À l'inverse, un mauvais estimateur restera trop proche des observations d'entraînement, sans bien se généraliser, ce qui est couramment appelé le *sur-apprentissage* ou *overfitting* en anglais. Par conséquent, l'évaluation de la capacité d'un algorithme d'apprentissage supervisé à se généraliser est souvent évalué par l'occultation d'une partie de l'échantillon lors de l'apprentissage. Le sous-échantillon mis de coté pour l'évaluation est souvent appelé *jeu de test*.

Dans la catégorie de l'apprentissage supervisé, nous pouvons cité deux grands types de problèmes qui sont la *régression* et la *classification*. Alors que la classification concerne les problèmes où les sorties sont à valeur dans un ensemble fini de catégories, la régression

traite les sorties à valeur dans un espace continu. Dans la première partie du manuscrit nous formulons plusieurs problèmes de régression où l'on cherche une fonction $f : \mathbb{R}^{d_X} \to \mathbb{R}^{d_Y}$ permettant de prédire les sorties Y, à valeurs dans $\mathbb{R}^{d_Y}$, à partir des entrées X, à valeur dans $\mathbb{R}^{d_X}$, sous l'hypothèse d'une perturbation aléatoire additive ε:

$$Y = f(X) + \varepsilon. \tag{1.4.4}$$

La recherche se fait dans un certain espace fonctionnel $f \in \mathcal{H}$, parfois appelé *espace hypothèse*, celui-ci pouvant être paramétré (cas par exemple des fonctions linéaires ou des polynômes de degré d) ou pas (e.g. l'espace des fonctions de carré sommable). L'hypothèse de cet espace fonctionnel est souvent implicite dans le choix du modèle ou de l'algorithme d'apprentissage. Une fois ces choix réalisés, l'entraînement et le test du modèle se feront selon une certaine métrique $\mathcal{L}$, aussi appelée *fonction de perte*. La fonction de perte la plus courante en régression est la perte quadratique:

$$\mathcal{L}(Y, f(X)) := \|Y - f(X)\|_2^2. \tag{1.4.5}$$

Dans ce cas, on peut montrer que la meilleur approximation possible de f sera donnée par l'espérance conditionnelle des sorties [c.f. Hastie et al., 2009, p.18]:

$$f(x) = \mathbb{E}\left[Y|X = x\right]. \tag{1.4.6}$$

Par ailleurs, dans certains cas on peut vouloir restreindre le nombre de covariables intégrant le modèle. Les motivations pour cela peuvent être que l'on sait que le phénomène à modéliser ne fait intervenir qu'une faible fraction des variables d'entrées, ou bien qu'on souhaite construire un modèle léger et facilement interpétable d'un phénomène complexe, ou encore que nous avons plus de covariables que d'observations. C'est ce qu'on appelle un problème de *sélection de variables*, pouvant être abordé avec un point de vue supervisé, comme nous l'avons fait dans la première partie du manuscrit, ou non-supervisé.

Au contraire du contexte supervisé, dans l'apprentissage non-supervisé les données ne sont pas étiquetés ni partagées en tant qu'entrées ou sortie. L'objectif est de concevoir des algorithmes capables de retrouver des structures sous-jacentes dans les données. L'analyse par composantes principales, par exemple, permet de retrouver les axes de plus forte variation. Les algorithmes de clustering sont un autre exemple d'outil d'apprentissage non-supervisé, vu qu'ils permettent de regrouper les observations qui se ressemblent selon un certain critère. Toujours dans le cadre de l'apprentissage non-supervisé, nous nous intéressons dans la deuxième partie du manuscrit à des problèmes d'estimation de densité. Cette classe de problèmes vise à estimer la densité de probabilité de la distribution ayant générée les données observées.

Une présentation plus détaillée de l'apprentissage statistique pourra être trouvé dans Hastie et al. [2009] par exemple. Dans la section suivante nous précisons les objectifs

de la thèse ainsi que nos contributions, en précisant dans quelle mesure nous avons fait intervenir les concepts d'apprentissage que nous venons de présenter brièvement.

1.5 Contributions et organisation du manuscrit

1.5.1 Axes d'étude

Deux questions principales ont été abordées au cours de cette thèse. La première concerne l'identification de modèles de la dynamique de l'avion. En effet, avant de résoudre le problème d'optimisation décrit dans la section 1.4.1, il est important de le poser avec un modèle décrivant fidèlement le comportement réel de l'avion. Avec la même optique adoptée dans OptiClimb, notre premier objectif à donc été de développer une méthodologie pour estimer de tels modèles de façon individuelle à chaque avion, à partir de leur données historiques. Le deuxième axe d'étude exploré concerne le domaine de validité de ces modèles de dynamique. Notre objectif a donc été de définir un indicateur d'évaluation des trajectoires optimisées au vue de ces régions de validité. Nous voulions ainsi prévenir que nos solutions ne diffèrent trop des trajectoires réelles utilisées pour construire les modèles, ce qui devra contribuer à ce qu'elles soient acceptées par les compagnies aériennes.

Au vue de ces deux axes d'étude, le manuscrit est organisée en deux parties. Nos contributions dans chacune de ces deux directions sont décrites avec plus de détails dans les sections qui suivent.

1.5.2 Contributions de la première partie

Le chapitre 2 présente plus en détails les données et prétraitements réalisés préalablement à toutes nos simulations.

Dans le chapitre 3, nous décrivons plusieurs formes de dynamique possibles pour des avions de ligne en phase de montée et introduisons le problème d'identification de la dynamique. Nous expliquons ensuite comment nous avons formuler l'identification en tant que problème de régression et situons cette approche dans l'état de l'art des méthodes d'identification de systèmes en aéronautique. Après avoir motivé et décrit les structures paramétriques choisies pour les différentes fonctions impliquées dans le problème, nous proposons plusieurs variantes d'une approche d'estimation par maximum de vraisemblance. Ces méthodes sont alors comparées numériquement avec des données réelles, tout en gardant à l'esprit l'objectif final d'optimisation de trajectoire. En ce sens, des critères d'évaluation adaptés à cette utilisation sont conçus et présentés.

Une des principales sources de difficulté dans les approches du chapitre 3 repose sur la nonlinéarité de la dynamique relativement aux paramètres du modèle. Dans le quatrième chapitre, une formulation linéarisée est proposée. Nous suggérons également dans ce contexte d'employer des méthodes de sélection de variable pour construire une structure de modèle de régression polynomiale dépendant des données. L'approche proposée est une

extension à un contexte multi-tâche structuré du *bootstrap lasso* Bach [2008]. Elle nous permet en effet de sélectionner les variables du modèle dans un contexte à fortes corrélations, tout en conservant la structure du problème inhérente à nos connaissances métier. Nous illustrons le comportement de notre méthode à l'aide de données de 25 avions différents. Bien que de bons résultats en termes de sélection de variables soit obtenus, nous mettons en évidence dans nos simulations des problèmes d'identifiabilité de nos modèles et proposons deux types de solution. Les méthodes obtenues sont alors comparées à celles du premier chapitre.

1.5.3 Contributions de la deuxième partie

Le chapitre 5 traite la caractérisation des solutions du problème de contrôle relativement au domaine de validité des modèles identifiés. Dans cette optique, nous présentons une approche probabiliste générale pour quantifier la proximité entre une courbe arbitraire dans un espace Euclidien et un ensemble de trajectoires supposées échantillonnées à partir d'un même processus stochastique. Nous définissons avec ce formalisme un critère appelée *mean marginal likelihood* (vraisemblance marginale moyenne) construit en intégrant en temps les densités de probabilité marginales du processus sous-jacent. Nous proposons une classe d'estimateurs de cet quantité, construit en partitionant finement l'espace des temps pour y construire un "histogramme" d'estimateurs de densité locaux. Nous prouvons que les estimateurs de densité construits à partir des données de cette partition approximent de façon consistance les densité marginales du processus ayant généré les données, un résultat plus fort étant montré pour le cas spécifique d'estimateurs à noyaux. D'un point de vue pratique, une implémentation particulièrement flexible et adaptative de l'estimateur à noyau est étendue à ce contexte. Nous comparons le comportement de notre approche à d'autres méthodes plus classiques d'analyse de données fonctionnelles et d'estimation de densité conditionnelle.

Dans le chapitre 6 nous proposons une version paramétrique de l'estimateur de *mean marginal likelihood*, construit à partir de modèles de mélange Gaussiens. Cette approche est motivée par la volonté d'utiliser ce critère comme terme de pénalité dans le problème d'optimisation de trajectoire, dans l'intention d'obtenir directement des trajectoires consommant peu sans trop s'éloigner des régions de validité. Des modèles paramétriques sont en effet nécessaires dans ce contexte lorsque l'on souhaite résoudre le problème de commande optimale avec des méthodes numériques utilisant des outils de différentiation automatique. Dans un premier temps, nous montrons numériquement que cette version paramétrique est aussi précise que la version nonparamétrique du chapitre précédent. Dans un deuxième temps, nous illustrons à travers de nombreuses simulations l'impact de telle pénalité sur les solutions du problème.

Chapter references

F. Bach. Bolasso: model consistent Lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40, 2008.

J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control and dynamics*, 21(2):193–207, 1998.

J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2010.

J. F. Bonnans, D. Giorgi, V. Grelard, B. Heymann, S. Maindrault, P. Martinon, O. Tissot, and J. Liu. Bocop – A collection of examples. Technical report, INRIA, 2017. URL http://www.bocop.org.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1): 25–57, 2006.

A. Walther and A. Griewank. Getting started with adol-c. In *Combinatorial scientific computing*, pages 181–202, 2009.

Part I

AIRCRAFT DYNAMICS IDENTIFICATION

Chapter 2

DATA PREPROCESSING

Abstract: *In this chapter we give more details concerning the data used in this thesis and provide an overview of the preprocessing steps which were carried before all the analysis presented herein. We also recall briefly some basic statistical learning concepts and vocabulary which are useful throughout the manuscript.*

Contents

2.1 Introduction **21**
2.2 Statistical learning prerequisites **23**
2.2.1 Regularized regression framework 23
2.2.2 Model selection using cross-validation 24
2.3 Smoothing splines **25**
2.4 Preprocessing steps **26**

2.1 Introduction

During all commercial flights, thousands of variables are recorded and stored in the *Quick Access Recorder* (QAR), whose data is commonly used for flight safety and efficiency purposes. Its data is easily accessible and we suppose it may be used for inferring the dynamic behaviour of an aircraft.

Although many alerts and pilot settings are also contained in this type of dataset, we limited the scope of our study to signals corresponding to physical quantities, which seem more adequate for the task. More precisely, only the raw data of the following variables were considered: the pressure altitude h, the Mach number M, the fuel consumption C_f, the aircraft mass m, the engines turbofan speed N_1 (as a percentage of the maximum speed, corresponding to full throttling), the static air temperature SAT, the pitch angle θ, the heading angle ψ, the wind speed W and the wind direction ψ_w. The time-sampling

here was of 1 observation per second. Figure 2.1 shows what these raw data signals look like.

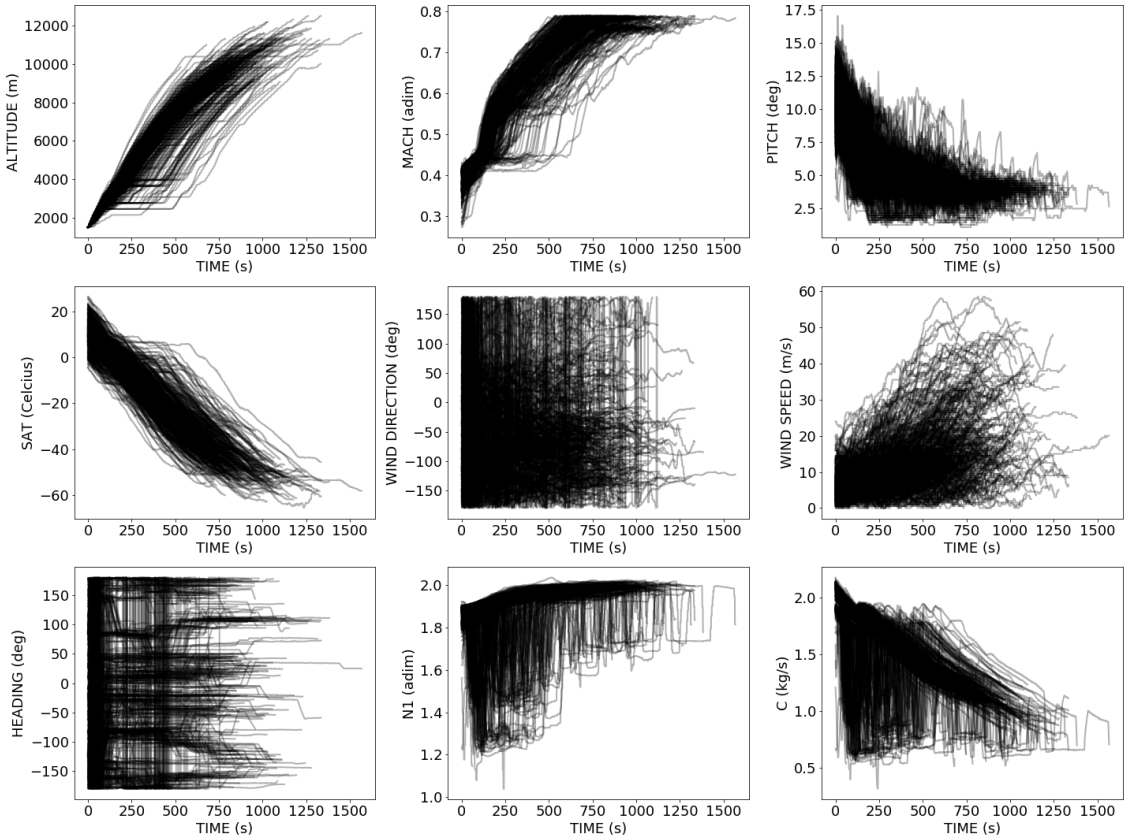


Figure 2.1 – Raw signals of 424 different flights extracted from the QAR of the same B737-800.

All these signals are most likely corrupted with different errors, some of which are stochastic and others deterministic or systematic. They include for example measurement errors, calculations errors and encoding-decoding errors. As an illustration of a systematic error, we may note the jumps between -180 and 180 degrees of the heading and wind direction signals in figure 2.1. In order to correct the outlying data to some extent and prepare it for the identification methods described in chapter 3, several preprocessing procedures were performed:

- systematic errors correction;
- signal smoothing;
- construction of new variables;
- signal differentiations.

Before describing the details of such preprocessing steps, some statistical learning concepts and vocabulary used throughout the manuscript must be recalled.

2.2 Statistical learning prerequisites

2.2.1 Regularized regression framework

Let Y and X be random variables valued respectively on $\mathbb{R}^{d_Y}$ and $\mathbb{R}^{d_X}$. We say that a *regression model* exists between Y and X if there is a certain function f and a random variable ε called *noise* such that

$$Y = f(X) + \varepsilon. \tag{2.2.1}$$

In a regression task, a dataset $\{(x^i, y^i)\}_{i=1}^N$ of observations of (X, Y), usually called *training set*, is used to build an estimator $\hat{f}$ of the function f. The objective usually sought is to be able to predict a response $\hat{f}(x^{N+1})$ from a new input observation x^{N+1} as close as possible to the new (unknown) output y^{N+1}. Closeness can be defined according to several possible metrics $\mathcal{L}$ called *loss functions*. The most commonly used in the case of real valued variables is the *squared-loss*:

$$\mathcal{L}\left(\hat{f}(x^{N+1}), y^{N+1}\right) \triangleq \left\| y^{N+1} - \hat{f}(x^{N+1}) \right\|_2^2. \tag{2.2.2}$$

For assessing an estimator, it seems natural to want to generalize such metric to any possible new observation drawn from the joint distribution of (X, Y). This is the idea behind the *Expected Prediction Error*, also called *Generalization Error* or *Risk*:

$$EPE(\hat{f}) = \mathbb{E}\left[\mathcal{L}(\hat{f}(X), Y)\right]. \tag{2.2.3}$$

However, the distribution of (X, Y) needed to evaluate this expectation is usually unknown, which means that we cannot minimize this quantity in practice. This is why many regression methods are originally defined through an optimization problem where one minimizes the empirical average of the loss across the training set over some predefined subspace $\mathcal{F}$ of functions from $\mathbb{R}^{d_X}$ to $\mathbb{R}^{d_Y}$:

$$\hat{f} \in \underset{f \in \mathcal{F}}{\arg\min}\ TE(f) := \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x^i), y_i). \tag{2.2.4}$$

The criterion TE was originally called the *empirical risk* in the statistical learning literature, being also known as the *training error* among the machine learning community.

If the functional search space $\mathcal{F}$ is not restrictive enough, minimizing the training error will lead to fitting to the training sample noise $\{\varepsilon_i\}_{i=1}^N$ instead of filtering it, which is commonly called *overfitting*. This is why assumptions are usually made on f leading to particular choices of $\mathcal{F}$. Another common way of fighting overfitting is by adding to the *empirical risk minimization* problem (2.2.4) a regularization terms $R(f)$ promoting some

hypothesized property of f:

$$\hat{f}_\lambda \in \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x^i), y_i) + \lambda R(f), \tag{2.2.5}$$

where $\lambda > 0$ is called the *regularization parameter*. An example of regularized empirical risk minimization problem are the *smoothing splines* presented in section 2.3. More details on the subject of empirical risk minimization can be found in [Hastie et al., 2009, Chapter 7] or Vapnik [2013].

2.2.2 Model selection using cross-validation

In the regularized problem (2.2.5), the regularization parameter λ determines the desired trade-off between minimizing the loss and enforcing the properties promoted by the regularizer $R(f)$. The solution of this problem depends on λ and is hence denoted by $\hat{f}_\lambda$. The task of choosing the parameter λ is commonly called *model selection* in the statistical learning community. The objective of this task is usually to look for a value of λ for which $\hat{f}_\lambda$ would minimize the *expected prediction error*. As previously explained, such a quantity is intractable and needs thus to be approximated. The training error being too optimistic, one way of assessing the quality of $\hat{f}_\lambda$ for some value of λ is to evaluate the averaged loss on a separate set of observations drawn from the distribution of (X, Y), called *validation set*. However, the results may vary a lot depending on the way the original dataset is split into *training* and *validation*. Thus, a common solution to approximate the EPE is to use *K-fold cross-validation*, a technique originally introduced by Stone [1974]. It consists in randomly partitioning the observations in $K < N$ subsets of same size m and, at each cross-validation step $k = 1, \ldots, K$, the k^{th} subset is used for *validation*, while the union of the other $K - 1$ subsets are used to *train* the estimator $\hat{f}^k_\lambda$, i.e. to solve problem (2.2.5). The *cross-validation criteria* for the parameter value λ writes then as the average of all the validation errors:

$$CV(\lambda) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{m} \sum_{i=1}^{m} \left(y_i - \hat{f}^k_\lambda(t_i) \right)^2. \tag{2.2.6}$$

Practitioners often compute the cross-validation scores for a finite number of values of λ and choose the one giving the minimum score. More generally speaking, cross-validation is a technique used to assess and compare the predictive power of statistical models. It is useful for choosing not only regularization parameters, but also any model parameter which is supposed to be fixed prior to training, often called *hyperparameters*. More details concerning cross-validation can be found in [Hastie et al., 2009, Chapter 7.10].

2.3 Smoothing splines

Among the regression techniques where a regularized empirical minimization problem is solved, we can cite the *smoothing splines*. *Splines* usually denote piecewise polynomial functions, which were originally used in applied mathematics for interpolation purposes. *Smoothing splines* [Schoenberg, 1964], however, allow to use this class of functions to fit a smooth curve to a set of noisy observations, filtering the noise and keeping only the "real" signal.

Let $[a;b]$ be an interval of $\mathbb{R}$. We consider here that we want to smooth N data points $\{(t_i, y_i)\}_{i=1}^N$ in $[a;b] \times \mathbb{R}$. We suppose that the times $(t_1, \dots, t_N)$ are known (i.e. deterministic), while the outputs $(y_1, \dots, y_N)$ are corrupted by some random noise $(\varepsilon_1, \dots, \varepsilon_N)$ of unknown distribution. We also assume that a *regression function* f exists such that:

$$y_i = f(t_i) + \varepsilon_i, \qquad \forall i = 1, \dots, N. \tag{2.3.1}$$

Smoothing splines are defined, for a given fixed $\lambda \geq 0$, as a solution of the following optimization problem:

$$\hat{f}_\lambda \in \underset{f \in H^2(a,b)}{\arg\min} \sum_{i=1}^{N} (y_i - f(t_i))^2 + \lambda \int_a^b f''(t)^2 dt, \tag{2.3.2}$$

where $H^2(a,b)$ denotes the Sobolev space of square-integrable functions over $(a;b)$ having derivatives up to order 2 also square-integrable over $(a;b)$.

We see that (2.3.2) is a regularized empirical risk minimization problem as (2.2.5), where the loss function is the squared-loss and the search space is $H^2(a,b)$. The regularization parameter λ is often called *smoothing parameter* in this context, since it determines the trade-off between how much curvature is allowed for the solution and how close to the data we want it to be. This idea can be illustrated by examining the two extreme possible values for λ:

- when $\lambda = 0$, the set of solutions of (2.3.2) contains any function f in $H^2(a,b)$ which interpolates the data;
- when $\lambda \to \infty$, no curvature is allowed and problem (2.3.2) corresponds to linear least-squares.

One popular way of setting λ is by cross-validation, as described in section 2.2.2, which has originally been suggested in the splines context by Wahba [1975].

Smoothing splines get their name from the fact that problem (2.3.2) has a unique solution, which is a *natural cubic spline*[1] with *knots*[2] at the sample points $\{t_i, i = 1, \dots, N\}$

1. *Cubic* comes from the fact that the piecewise polynomial is of order 3 between the knots and C^2 everywhere ; *natural* means that the extrapolation before the first data point a and after the last data point b is affine.
2. Define the boundaries of the intervals inside which the spline is a polynomial.

(see proof in Eubank [1999] for example). This means that, despite the infinite-dimensional search space $H^2(a;b)$, the solution has a finite-dimensional parametrization of the form

$$\hat{f}_\lambda(t) = \sum_{i=1}^{N} \hat{\beta}_i B_i(t), \tag{2.3.3}$$

where $\{B_i, i = 1, \ldots, N\}$ is a basis of natural splines (such as *B-splines*) and $\{\hat{\beta}_i, i = 1, \ldots, N\}$ are coefficients to be estimated. For more details on this matter the reader is referred to [Hastie et al., 2009, Chapter 5.4] and [Andrieu, 2013, Chapter 3.2].

2.4 Preprocessing steps

The first errors that had to be corrected were systematic ones, which include offsets and time-shifts. Those were identified by visualizing the trajectories and corrected one by one. Once systematic errors have been addressed, we smoothed the signals flight by flight using smoothing splines described in section 2.3, setting the smoothing parameter through cross-validation. We hoped to correct most part of the remaining problematic points. As we will show in the following chapter 3, some useful physical quantities for our problem include variables which are not directly available in QAR data. Some of this quantities describe atmospheric conditions, such as the air pressure, the density and the sound speed. They were derived from the measured signals using formulas from the International Standard Atmosphere model of the troposphere (altitudes lower than 1100 m) [International Organization for Standardization, 1975]:

- the air pressure in Pascals:

$$P = P_0 \left(1 + \frac{\kappa_1 h}{T_0}\right)^{-\frac{g}{\kappa_1 R_s}}, \tag{2.4.1}$$

- the air density in kg/m^3:

$$\rho = \frac{P}{R_s SAT}, \tag{2.4.2}$$

- and the sound speed in m/s:

$$V_{sound} = \sqrt{\kappa_2 R_s SAT}, \tag{2.4.3}$$

where the physics constants used are listed in table 2.1.

Other quantities of interest describe the aircraft state at a given time. The following standard flight mechanics formulas were used for computing them:

- the true air speed of the aircraft in m/s:

$$V = M V_{sound}, \tag{2.4.4}$$

Table 2.1 – Atmospheric constants

Constant	Meaning	Value
T_0	Standard air temperature at sea level (OACI)	288.15 K
P_0	Standard air pressure at sea level (OACI)	101325 Pa
κ_1	Troposphere Temperature gradient (OACI)	-0.0065 $K.m^{-1}$
κ_2	Adiabatic index for a diatomic ideal gas	1.4
g	Gravity acceleration	9.80665 $m.s^{-2}$
R_s	Air specific constant	287.053 $J.kg^{-1}.K^{-1}$

- the flight path angle in radians (when there is no wind - see section 3.2 for further details):

$$\gamma = \arcsin\left(\frac{\dot{h}}{V}\right), \tag{2.4.5}$$

- and the angle of attack in radians:

$$\alpha = \theta - \gamma. \tag{2.4.6}$$

Concerning the aircraft mass, the measured signals extract from the QAR were undersampled and inaccurate. As we had access to better quality signals for the fuel consumption, we derived new mass data from it, only using the first mass measurement as an offset:

$$m(t) = m_0 - \int_0^t C(t)dt. \tag{2.4.7}$$

Chapter references

C. Andrieu. *Modélisation fonctionnelle de profils de vitesse en lien avec l'infrastructure et méthodologie de construction d'un profil agrégé.* PhD thesis, Université Paul Sabatier-Toulouse III, 2013.

R. Eubank. *Nonparametric regression and spline smoothing.* Marcel Dekker, 1999.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, 2nd edition, 2009.

International Organization for Standardization. Standard Atmosphere, 1975.

I. J. Schoenberg. Spline functions and the problem of graduation. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 52, page 947. National Academy of Sciences, 1964.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

V. Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

G. Wahba. A completely automatic french curve: Fitting spline functions by cross validation. *Communications in Statistics: Theory and Methods*, 4(1):1–17, 1975.

Chapter 3

AIRCRAFT DYNAMICS IDENTIFICATION

Abstract: *This chapter lays the groundwork for the aircraft dynamics identification methods proposed in this thesis. It recalls the many different aircraft dynamics models and existing identification techniques, before motivating the choice for 2D parametric flight mechanics and for the Equation-Error Method. We interpret this static formulation of the identification problem adopting a structured multi-task regression viewpoint and propose four maximum likelihood fitting approaches, which are compared using real flight data from more than 400 flights.*

Contents

3.1 Introduction 30
3.2 Dynamics modeling 32
3.2.1 Possible assumptions 32
3.2.2 Flat Earth dynamics in vertical plane with no wind 32
3.2.3 Dynamics with wind 34
3.2.4 2D flight with non-zero bank angle 35
3.2.5 3D dynamics 36
3.2.6 Spheric rotating earth dynamics 38
3.2.7 Projection of the wind speed onto the ground frame of reference 42
3.3 Description of our identification problem 43
3.4 Parametric hidden models 45
3.4.1 Restriction to parametric models 45
3.4.2 Flight mechanics models 45
3.4.3 Remark on monomials selection 46
3.5 Aircraft systems identification state-of-the-art 47
3.5.1 The Output-Error Method 47

3.5.2 The Filter-Error Method 49
3.5.3 The Equation-Error Method 50
3.6 Nonlinear multi-task regression 52
3.6.1 Baseline regression method: Ordinary Least-Squares 52
3.6.2 Multi-task framework 54
3.6.3 Maximum Likelihood Estimators 56
3.7 Quality criteria design 59
3.7.1 Static criterion 59
3.7.2 Dynamic criterion 59
3.8 Numerical benchmark 61
3.9 Conclusion 63

3.1 Introduction

We consider an aircraft in climb phase, modeled as a dynamical system of state variables $\boldsymbol{x} = (h, V, \gamma, m) \in \mathbb{X}$ and control variables $\boldsymbol{u} = (\alpha, N_1) \in \mathbb{U}$ (see table 3.1 for the notations), where $\mathbb{U} = L^\infty(0, t_f; \mathbb{R}^2)$ and $\mathbb{X} = W^{1,\infty}(0, t_f; \mathbb{R}^4)$ (i.e. the set of primitives of functions in $L^\infty(0, t_f; \mathbb{R}^4)$). In such a context, the problem of optimizing the trajectory during a certain horizon $t_f > 0$ may be seen as a nonlinear constrained optimal control problem of the following form:

$$\min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{X}\times\mathbb{U}} \int_0^{t_f} C(t, \boldsymbol{u}(t), \boldsymbol{x}(t))dt,$$
$$\text{s.t.} \begin{cases} \dot{\boldsymbol{x}} = g(t, \boldsymbol{u}, \boldsymbol{x}), \quad \text{for a.e. } t \in [0, t_f]; \\ \boldsymbol{u}(t) \in U_{ad}, \quad \boldsymbol{x}(t) \in X_{ad}, \quad \text{for a.e. } t \in [0, t_f], \\ \Phi(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) \in K_\Phi, \\ c_j(t, \boldsymbol{x}(t)) \leq 0, \quad j = 1, \ldots, n_c, \quad \text{for all } t \in [0, t_f]. \end{cases} \tag{3.1.1}$$

All the mappings from (6.2.1), i.e. the *running cost* C, the *dynamics* function g, the *initial-final state constraint* function Φ and *state path constraints* c_j for $j = 1, \ldots, n_c$, are continuously differentiable. The sets X_{ad} and U_{ad} are assumed to be closed subsets of $\mathbb{R}^4$ and $\mathbb{R}^2$, K_Φ is assumed to be a nonempty closed convex set of $\mathbb{R}^{n_\Phi}$ and $n_\Phi, n_c \in \mathbb{N}$.

The function g from (6.2.1) reflects the fact that we want the optimized trajectory to respect some model of the aircraft dynamics. This constraint has a huge impact on the results of the optimization, which motivates the search for accurate dynamical systems identification techniques, based on historic flight data.

We limit our scope to civil flights, and more specifically to the climb phase. We also limit our study to the case where the control problem (6.2.1) is solved numerically

Table 3.1 – Variables nomenclature

Notation	Meaning
h	Aircraft altitude
V	Aircraft true airspeed (TAS)
γ	Path angle
m	Aircraft mass
α	Angle of attack (AOA)
N_1	Engines turbofan speed
ρ	Air density
M	Aircraft Mach number
SAT	Static air temperature
θ	Pitch angle
C	Total fuel flow
T	Total thrust force
D, L	Drag and lift forces
C_{sp}	Specific consumption
ψ	Heading angle
μ	Bank angle (around TAS)
W, ξ	Wind speed norm and heading angle
W_x, W_y, W_z	Horizontal and vertical wind speed components

using *direct transcription* methods (see e.g. in [Betts, 1998]). Moreover, the techniques presented hereafter are suited for data extracted from the Quick Access Recorder (QAR) (see sections 1.3 and 2.1).

According to the literature [Jategaonkar, 2006, Maine and Iliff, 1986], two widely used approaches for aircraft dynamics estimation are the *Output-Error Method* and *Filter-Error Method*, based on the main ideas of minimizing measurement error and integrating the dynamics several times. Recent advances include using neural networks for the integration of the dynamics under uncertainty [Peyada and Ghosh, 2009]. On the other hand, renewed interest for the older *Equation-Error Method* has also been observed [Morelli, 2006]. We propose in this chapter variations of the latter.

In the next section, we provide an overview of the possible flight mechanics models that can be used to design the dynamics structure. A description of the identification problem in our precise case will be given in section 3.3, while further details concerning the state-of-the-art methods listed above can be found in section 3.5. After clarifying how the dynamics were parametrized (section 3.4), several *regression* formulations of our problem are stated and solved using *maximum likelihood* based techniques (section 3.6). In section 3.7 we define the criteria devised to assess these methods in the context of trajectory optimization, before illustrating our methods in sections 3.8 with numerical results based on real data from 10 471 flights.

3.2 Dynamics modeling

3.2.1 Possible assumptions

Many aircraft dynamics models have been proposed and studied by the flight mechanics community since the early ages of aviation. The analytic models proposed usually come from the application of Newtonian physics, such as the 2nd law and the principle of mass conservation. Hence, the different dynamics models available in the literature depend on different sets of physical assumptions. We may or may not, for example, assume that the atmosphere moves relatively to the ground, which incorporates the presence of wind in the dynamics model. We may also assume that the aircraft climb trajectory is bounded to a vertical plane or not. Another possible assumption concerns whether or not the roundness and rotation of the Earth has an impact on the aircraft trajectory. The dynamic models emerging from these assumptions are presented in the following subsections, with growing complexity. We only studied models with 3 degrees of freedom (DOF), but similar analysis could be carried in a 6 DOF framework (which would result in models too complex for the approaches that follows). Note also that many other assumptions made concerning the atmosphere structure, the aircraft velocity vector and forces directions will not be discussed.

3.2.2 Flat Earth dynamics in vertical plane with no wind

We consider an aircraft at an arbitrary time t of mass m and speed $\vec{V}$. The later is the speed of the aircraft's center of mass relative to an inertial frame of reference attached to the ground. Hence, we call $\vec{V}$ the *inertial speed* hereafter, its projection onto the ground being what is commonly called *ground speed*. At time $t+dt$, it has ejected a small mass of fuel $\delta m << m$ at speed $\vec{V_e}$ relative to the aircraft frame of reference. By speed composition, the ejected mass speed in the inertial frame of reference is $\vec{V} + \vec{V_e}$. The new mass of the aircraft is $m(t + dt) = m - \delta m$ and its speed also becomes $V(t + dt) = V + \delta\vec{V}$, with $\|\delta\vec{V}\| << \|\vec{V}\|$. Considering that during this time our system was subjected to external forces $\vec{F_{ext}}$, we can write the momentum conservation equation as follows:

$$(m - \delta m)(\vec{V} + \delta\vec{V}) + (\vec{V} + \vec{V_e})\delta m - mV = \vec{F_{ext}}dt. \tag{3.2.1}$$

After some simplifications and by neglecting the second order terms, we obtain the vector equation

$$m\delta\vec{V} + \vec{V_e}\delta m = \vec{F_{ext}}dt, \tag{3.2.2}$$

which leads to the following aircraft dynamics:

$$m\frac{d\vec{V}}{dt} = \vec{F_{ext}} - \vec{V_e}\frac{dm}{dt}, \tag{3.2.3}$$

where the derivatives are computed in the inertial frame of reference.

The term $-\vec{V}_e \frac{dm}{dt}$ corresponds to what is commonly called the aircrafts thrust, that we shall write $\vec{T}$ here. The external forces applied to the aircraft are typically its weight $m\vec{g}$ and the aerodynamic forces. We shall decompose the later into a drag force $\vec{D}$, which is collinear to the aircraft *true air speed* (speed in the air frame of reference), which is denoted $\vec{V}_r$ in this section, and a lift force $\vec{L}$ perpendicular to it, as depicted in figure 3.1. As we assume there is no wind in this subsection, the air frame of reference is immobile relative to the inertial frame of reference, which means that $\vec{V} = \vec{V}_r$ here. This brings our dynamics to the following form:

$$m\frac{d\vec{V}}{dt} = m\vec{g} + \vec{D} + \vec{L} + \vec{T}. \tag{3.2.4}$$

Note that equation (3.2.4) is valid in all the following subsections for all the assumptions considered. Only the inertial frame of reference in which we differentiate $\vec{V}$ may change, as we will see in subsections 3.2.5 and 3.2.6.

Let $\mathcal{R}_i = (O, \vec{x}_i, \vec{y}_i, \vec{z}_i)$ and $\mathcal{R}_v = (G, \vec{x}_v, \vec{y}_v, \vec{z}_v)$ be respectively the coordinate systems of the inertial and the aircraft frames of reference (see figure 3.1). We choose G to denote the aircraft center of mass, $(\vec{x}_i, \vec{y}_i)$ to define the horizontal plane, $\vec{x}_v$ to be aligned with $\vec{V}_r$ and $\vec{y}_v$ to be aligned with the aircraft's wing tips. We assume that the aircraft is not turning and is flying in the vertical plane $(\vec{x}_i, \vec{z}_i)$, which means that $\vec{y}_v = \vec{y}_i$. We assume that $\vec{T}$ is in the vertical plane $(\vec{x}_i, \vec{z}_i) = (\vec{x}_v, \vec{y}_v)$, making an angle α relative to $\vec{x}_v$ which is usually called the Angle of Attack. The lift $\vec{L}$ is also supposed to be contained in the vertical plane here. The aircraft weight $m\vec{g}$ is directed along $-\vec{z}_i$.

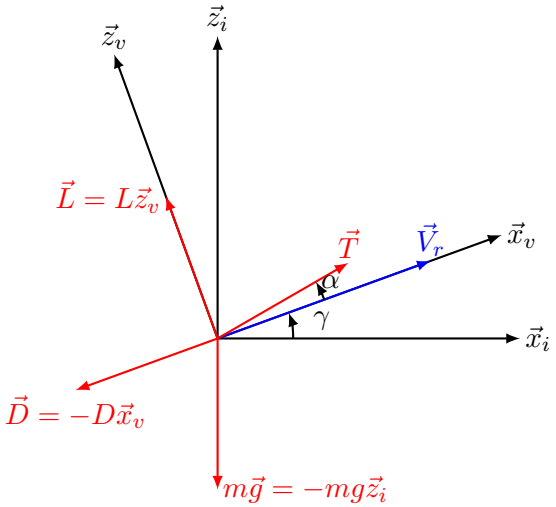


Figure 3.1 – $\mathcal{R}_v$ rotation relative to $\mathcal{R}_i$ in the case of a trajectory in a vertical plane with flat Earth assumption (all the vectors depicted are contained in the same plane).

As R_v is rotating at the speed $\vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_i} = -\dot{\gamma}\vec{y}_i$ relative to R_i, by *Bour*'s formula, we

have

$$\begin{aligned}\frac{d\vec{V_r}}{dt} = \left(\frac{d\vec{V_r}}{dt}\right)_{\mathcal{R}_i} &= \left(\frac{d\vec{V_r}}{dt}\right)_{\mathcal{R}_v} + \vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_i} \times \vec{V_r} \\ &= \dot{V_r}\vec{x}_v + V_r\dot{\gamma}\vec{z}_v,\end{aligned} \tag{3.2.5}$$

where $\times$ denotes the cross product and $V_r, \dot{V_r}$ denote the true airspeed and its derivative norms. Hence, by projecting equation (3.2.4) into axis $\vec{x}_v, \vec{z}_v$ we obtain the following equations:

$$\begin{cases} m\dot{V} \;\; = m\dot{V_r} \;\; = T\cos(\alpha) - D - mg\sin(\gamma) \\ mV\dot{\gamma} = mV_r\dot{\gamma} = T\sin(\alpha) + L - mg\cos(\gamma). \end{cases} \tag{3.2.6}$$

3.2.3 Dynamics with wind

If we consider that there is wind, the Inertial speed $\vec{V}$ is different from the True Airspeed $\vec{V_r}$ because the air frame of reference is moving relative to the inertial frame of reference with a speed $\vec{W}$, called the wind speed. By speed composition we have that

$$\vec{V} = \vec{V_r} + \vec{W} \tag{3.2.7}$$

and our dynamics become

$$m\frac{d\vec{V_r}}{dt} = m\vec{g} + \vec{D} + \vec{L} + \vec{T} - m\frac{d\vec{W}}{dt}. \tag{3.2.8}$$

The air frame of reference $\mathcal{R}_w$ is only translating at the speed $\vec{W}$ relative to $\mathcal{R}_i$, without rotating. We shall call the wind speed coordinates in the inertial frame of reference by $\vec{W} = W_x\vec{x}_i + W_y\vec{y}_i + W_z\vec{z}_i$. By injecting (3.2.5) in (3.2.8) and projecting on $\vec{x}_v$, $\vec{y}_v$ and $\vec{z}_v$, we obtain the system of equations with wind:

$$\begin{cases} m\dot{V_r} \;\; = T\cos(\alpha) - D - mg\sin(\gamma) - m\dot{W}_{xv} \\ 0 \qquad\;\; = -m\dot{W}_{yv} \\ mV_r\dot{\gamma} = T\sin(\alpha) + L - mg\cos(\gamma) - m\dot{W}_{zv} \end{cases} \tag{3.2.9}$$

where

$$\begin{cases} \dot{W}_{xv} = \dot{W}_x\cos(\gamma) - \dot{W}_z\sin(\gamma) \\ \dot{W}_{yv} = \dot{W}_y \\ \dot{W}_{zv} = \dot{W}_z\cos(\gamma) + \dot{W}_x\sin(\gamma) \end{cases} \tag{3.2.10}$$

are the components in $\mathcal{R}_v$ of the derivative of $\vec{W}$. We may notice that no lateral wind acceleration is possible with this hypothesis.

3.2.4 2D flight with non-zero bank angle

Now, we make the assumption that non-zero roll angles are allowed. We still keep the assumption of a trajectory contained in a vertical plane, which means that the aircraft rotations are only meant to counter the wind lateral accelerations from the last subsection.

We denote by μ the *velocity bank angle* defining the position of a new frame of reference $\mathcal{R}_L = (G, \vec{x}_v, \vec{y}_L, \vec{z}_L)$ relative to $\mathcal{R}_v$ after a rotation around the common axis $\vec{x}_v$, as shown in figure 3.2.

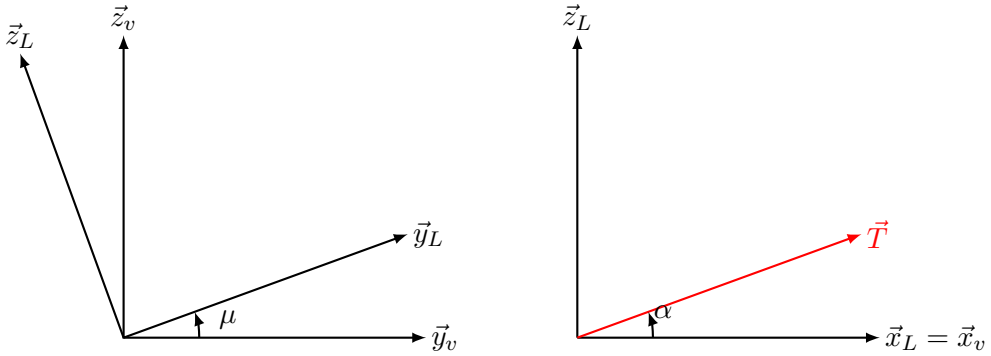


Figure 3.2 – $\mathcal{R}_L$ rotation relative to $\mathcal{R}_v$.

We assume that the lift $\vec{L}$ is directed along $\vec{z}_L$. The thrust $\vec{T}$ is assumed to be contained in the plane $(\vec{x}_v, \vec{z}_L)$, the *angle of attack* α defining now its position relative to $\vec{x}_v$ inside such a plane (see the right panel of figure 3.2). With these notations, the axis $\vec{y}_L$ is aligned with the aircraft's wing tips instead of $\vec{y}_v = \vec{y}_i$. In this context, $\vec{L}$ and $\vec{T}$ projections onto $\mathcal{R}_v$ have changed to

$$\begin{aligned}
\vec{L} &= L\vec{z}_L = -L\sin(\mu)\vec{y}_v + L\cos(\mu)\vec{z}_v \\
\vec{T} &= T\cos(\alpha)\vec{x}_v + T\sin(\alpha)\vec{z}_L \\
&= T\cos(\alpha)\vec{x}_v - T\sin(\alpha)\sin(\mu)\vec{y}_v + T\sin(\alpha)\cos(\mu)\vec{z}_v.
\end{aligned} \tag{3.2.11}$$

The new system of equations writes:

$$\begin{cases} m\dot{V}_r &= T\cos\alpha - D - mg\sin\gamma - m\dot{W}_{xv} \\ 0 &= -\left(T\sin\alpha + L\right)\sin\mu - m\dot{W}_{yv} \\ mV_r\dot{\gamma} &= \left(T\sin\alpha + L\right)\cos\mu - mg\cos\gamma - m\dot{W}_{zv} \end{cases}$$

We may notice here that the additional term in the second equation accounts for bank manoeuvres which allow the aircraft to compensate for lateral wind in order to stay in the vertical plane. Such manoeuvres result in some loss of a factor $\cos\mu$ of vertical ascending force, as seen in the third equation.

3.2.5 3D dynamics

We no longer assume that the aircraft is bound to fly in a vertical plane. We denote by $\mathcal{R}_h = (S, \vec{x}_h, \vec{y}_s, \vec{z}_s)$ an intermediary frame of reference such that S is the projection of G on the ground, $\vec{z}_h = \vec{z}_i$, $\vec{y}_h = \vec{y}_v$ and $\vec{x}_h$ points toward the direction of $\vec{V}_r$ projected on the ground (see figures 3.3 and 3.4). The speed frame of reference $\mathcal{R}_v$ rotates of the angle γ around $\vec{y}_v$ relative to $\mathcal{R}_h$. Let now ψ denote the rotation angle of $\mathcal{R}_h$ relative to $\mathcal{R}_i$ around the vertical axis $\vec{z}_i$, as depicted in figures 3.3 and 3.4.

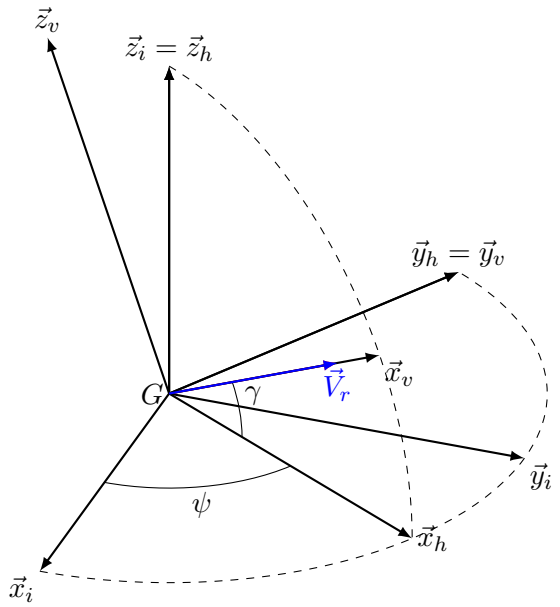


Figure 3.3 – $\mathcal{R}_v$ rotation relative to $\mathcal{R}_i$ in the case of a 3D trajectory with flat Earth assumption.

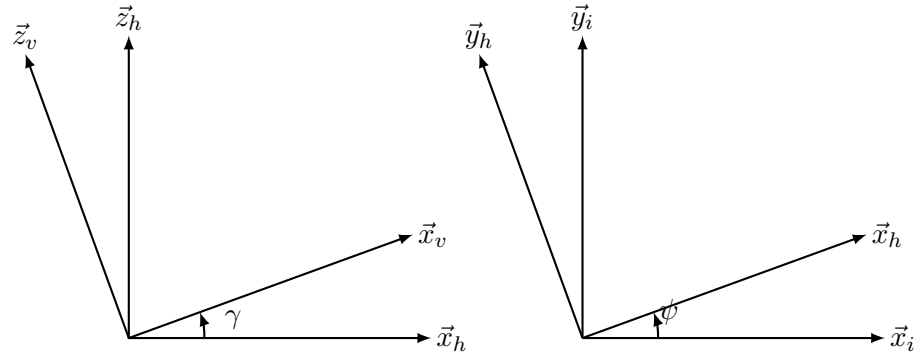


Figure 3.4 – $\mathcal{R}_v$ rotation relative to $\mathcal{R}_i$ in the case of a 3D trajectory with flat Earth assumption (2D representation).

In this case, the angular velocity vector of $\mathcal{R}_v$ relative to $\mathcal{R}_i$ writes

$$\vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_i} = -\dot{\gamma}\vec{y}_v + \dot{\psi}\vec{z}_i \tag{3.2.12}$$

and the true airspeed derivative from equation (3.2.5) becomes:

$$\begin{aligned}\left(\frac{d\vec{V}_r}{dt}\right)_{\mathcal{R}_i} &= \left(\frac{d\vec{V}_r}{dt}\right)_{\mathcal{R}_v} + \vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_i} \times \vec{V}_r \\ &= \dot{V}_r\vec{x}_v + V_r\dot{\psi}\cos(\gamma)\vec{y}_v + V_r\dot{\gamma}\vec{z}_v.\end{aligned} \tag{3.2.13}$$

Concerning the wind, we assumed we had access to its components in the inertial frame $\{W_x, W_y, W_z\}$. As we now have the additional ψ angle around $\vec{z}_i$ axis determining the direction towards which the aircraft is moving on the ground, the new projection of the wind vector in the speed coordinates system is:

$$\begin{bmatrix} W_{xv} \\ W_{yv} \\ W_{zv} \end{bmatrix} = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} \tag{3.2.14}$$

$$= \begin{bmatrix} W_x\cos\psi\cos\gamma + W_y\sin\psi\cos\gamma + W_z\sin\gamma \\ -W_x\sin\psi + W_y\cos\psi \\ -W_x\cos\psi\sin\gamma - W_y\sin\psi\sin\gamma + W_z\cos\gamma \end{bmatrix}. \tag{3.2.15}$$

The new dynamics that we obtain in 3D are:

$$\begin{cases} m\dot{V}_r = T\cos\alpha - D - mg\sin\gamma - m\dot{W}_{xv} \\ mV_r\dot{\psi}\cos\gamma = -(T\sin\alpha + L)\sin\mu - m\dot{W}_{yv} \\ mV_r\dot{\gamma} = (T\sin\alpha + L)\cos\mu - mg\cos\gamma - m\dot{W}_{zv}, \end{cases} \tag{3.2.16}$$

where the $\mathcal{R}_v$ coordinates of $\left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_i}$ write

$$\begin{cases} \dot{W}_{xv} = \dot{W}_x\cos\psi\cos\gamma + \dot{W}_y\sin\psi\cos\gamma + \dot{W}_z\sin\gamma \\ \dot{W}_{yv} = -\dot{W}_x\sin\psi + \dot{W}_y\cos\psi \\ \dot{W}_{zv} = -\dot{W}_x\cos\psi\sin\gamma - \dot{W}_y\sin\psi\sin\gamma + \dot{W}_z\cos\gamma. \end{cases} \tag{3.2.17}$$

3.2.6 Spheric rotating earth dynamics

We no longer assume the earth to be flat and immobile in this subsection. Hence, we have to define a new inertial frame with relation to which the dynamics can be written. Let $\mathcal{R}_i = (O, \vec{x}_i, \vec{y}_i, \vec{z}_i)$ be our new inertial frame of reference. We assume O to be the center of the planet, $\vec{x}_i$ to be aligned with Earth's rotation axis and $\vec{y}_i$ to point towards $Gr(t_0)$ the intersection between the Greenwich meridian and the Equator at time t_0 (see figure 3.5). Any immobile point on the surface of the planet rotates around $\vec{x}_i$ at the constant speed $\vec{\Omega}_e = \Omega_e\vec{x}_i$, with $\Omega_e \simeq \frac{2\pi}{24\times3600} = 7.3\ 10^{-5} rad/s$.

Let E be an arbitrary point on the Equator. E can be located with relation to the Greenwich meridian using the *longitude* angle ℓ. At time $t > t_0$, E makes an angle $\Omega_e(t - t_0) + \ell$ with $\vec{y}_i$ around $\vec{x}_i$. We consider now S to be the projection at time t of the center of mass of the aircraft G on the surface of the planet. We chose E to be the only point on the Equator of same longitude as S, as depicted in figure 3.5. Let $\mathcal{R}_s = (S, \vec{x}_s, \vec{y}_s, \vec{z}_s)$ be the frame of reference centered on S, such that the plane $(\vec{x}_s, \vec{y}_s)$ is horizontal (tangent to Earth's surface), $\vec{z}_s$ is vertical (orthogonal to Earth's surface) and $\vec{x}_s$ points to the North pole along the meridian passing by S. The angle between S and E around O is φ, called the *latitude* (see figure 3.6). Hence, the angular velocity vector between the surface frame of reference $\mathcal{R}_s$ and the inertial frame of reference $\mathcal{R}_i$ is

$$\vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i} = (\Omega_e + \dot{\ell})\vec{x}_i - \dot{\varphi}\vec{y}_s. \tag{3.2.18}$$

The aircraft TAS projected on the planet's surface makes an angle ψ with relation to $\vec{x}_s$, as in the last section (except for the translation, $\mathcal{R}_s$ is the equivalent of the $\mathcal{R}_i$ of

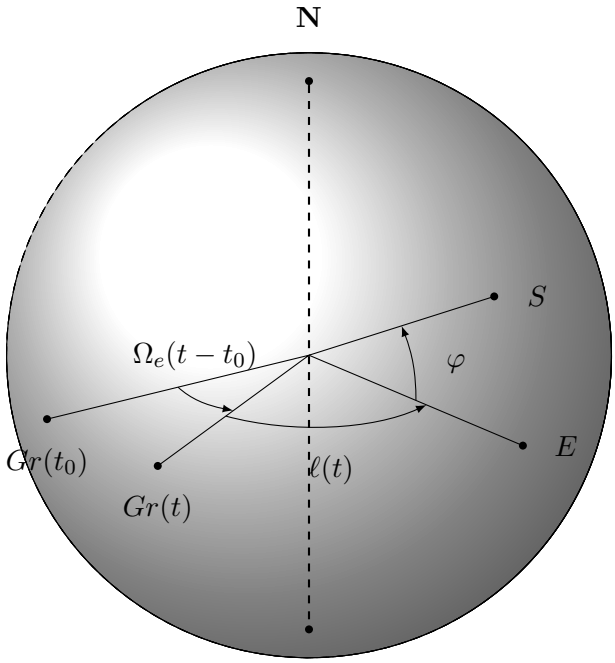


Figure 3.5 – Earth localization angles

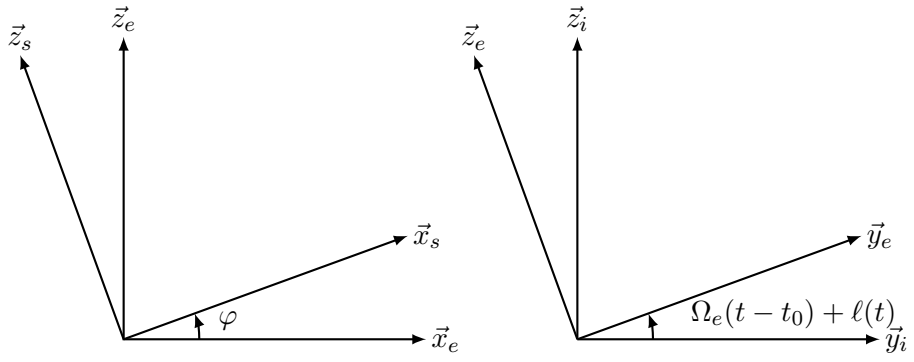


Figure 3.6 – $\mathcal{R}_s$ rotation relative to $\mathcal{R}_i$ in spheric Earth.

the previous subsections from a rotation point of view). In this context, with $\vec{x}_s$ defined as pointing to the North, ψ is commonly called the *heading angle*. We may then derive the rotation of the speed frame of reference $\mathcal{R}_v$, defined as in the previous sections, with relation to $\mathcal{R}_s$: $\vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_s} = -\dot{\gamma}\vec{y}_v + \dot{\psi}\vec{z}_s$. By rotation composition, we have that

$$\vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_i} = \vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_s} + \vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i}. \tag{3.2.19}$$

So now we have to recompute the TAS derivative in the inertial frame. As in (3.2.13), we have

$$\vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_s} \times \vec{V}_r = V_r\dot{\psi}\cos(\gamma)\vec{y}_v + V_r\dot{\gamma}\vec{z}_v. \tag{3.2.20}$$

According to (3.2.19), we still have to compute $\vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i} \times \vec{V}_r$. We have

$$\vec{x}_i = \cos\varphi\vec{x}_s - \sin\varphi\vec{z}_s \tag{3.2.21}$$

$$\vec{x}_s = \cos\psi(\cos\gamma\vec{x}_v - \sin\gamma\vec{z}_v) - \sin\psi\vec{y}_v \tag{3.2.22}$$

$$\vec{z}_s = \cos\gamma\vec{z}_v + \sin\gamma\vec{x}_v, \tag{3.2.23}$$

so

$$\begin{aligned}\vec{x}_i \times \vec{x}_v = &(\cos\varphi\cos\psi\sin\gamma - \sin\varphi\cos\gamma)\vec{y}_v \\ &+ \cos\varphi\sin\psi\vec{z}_v.\end{aligned} \tag{3.2.24}$$

Similarly,

$$\vec{y}_s = \cos\psi\vec{y}_v + \sin\psi(\cos\gamma\vec{x}_v - \sin\gamma\vec{z}_v) \tag{3.2.25}$$

so

$$\vec{y}_s \times \vec{x}_v = -\sin\psi\sin\gamma\vec{y}_v - \cos\psi\vec{z}_v. \tag{3.2.26}$$

Hence, by injecting relations (3.2.24) and (3.2.26) in (3.2.18), we get

$$\begin{aligned}\vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i} \times \vec{V}_r &= V_r\left[(\Omega_e + \dot{\ell})\vec{x}_i \times \vec{x}_v - \dot{\varphi}\vec{y}_s \times \vec{x}_v\right] \\ &= V_r\{(\Omega_e + \dot{\ell})[(\cos\varphi\cos\psi\sin\gamma - \sin\varphi\cos\gamma)\vec{y}_v \\ &\quad + \cos\varphi\sin\psi\vec{z}_v] + \dot{\varphi}(\sin\psi\sin\gamma\vec{y}_v + \cos\psi\vec{z}_v)\} \\ &= V_r[(\Omega_e + \dot{\ell})(\cos\varphi\cos\psi\sin\gamma - \sin\varphi\cos\gamma) + \\ &\quad \dot{\varphi}\sin\psi\sin\gamma]\vec{y}_v + V_r[(\Omega_e + \dot{\ell})\cos\varphi\sin\psi + \dot{\varphi}\cos\psi]\vec{z}_v\end{aligned} \tag{3.2.27}$$

and

$$\begin{aligned}\left(\frac{d\vec{V}_r}{dt}\right)_{\mathcal{R}_i} &= \left(\frac{d\vec{V}_r}{dt}\right)_{\mathcal{R}_v} + \vec{\Omega}_{\mathcal{R}_v/\mathcal{R}_s} \times \vec{V}_r + \vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i} \times \vec{V}_r \\ &= \dot{V}_r\vec{x}_v \\ &+ V_r[\dot{\psi}\cos\gamma + (\Omega_e + \dot{\ell})(\cos\varphi\cos\psi\sin\gamma - \sin\varphi\cos\gamma) \\ &\quad + \dot{\varphi}\sin\psi\sin\gamma]\vec{y}_v \\ &+ V_r\left[\dot{\gamma} + (\Omega_e + \dot{\ell})\cos\varphi\sin\psi + \dot{\varphi}\cos\psi\right]\vec{z}_v.\end{aligned} \tag{3.2.28}$$

As in the previous section, we have access to the wind coordinates $\{W_x, W_y, W_z\}$ and its derivatives in the surface frame of reference $\mathcal{R}_s$. But as $\mathcal{R}_s$ is not inertial, $\{\dot{W}_x, \dot{W}_y, \dot{W}_z\}$ does not correspond to the wind derivatives in $\mathcal{R}_i$ any more. Instead, we have that

$$\left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_i} = \left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_s} + \vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i} \times \vec{W}. \tag{3.2.29}$$

with

$$\begin{aligned}\vec{\Omega}_{\mathcal{R}_s/\mathcal{R}_i} \times \vec{W} = &[-\dot{\varphi} W_z + (\Omega_e + \dot{\ell}) \sin\varphi W_y]\vec{x}_s \\ &- (\Omega_e + \dot{\ell})(\sin\varphi W_x + \cos\varphi W_z)\vec{y}_s \\ &+ [(\Omega_e + \dot{\ell}) \cos\varphi W_y + \dot{\varphi} W_x]\vec{z}_s.\end{aligned} \tag{3.2.30}$$

This brings us to

$$\begin{aligned}\dot{W}_x^i &= \dot{W}_x - \dot{\varphi} W_z + (\Omega_e + \dot{\ell}) \sin\varphi W_y \\ \dot{W}_y^i &= \dot{W}_y - (\Omega_e + \dot{\ell})(\sin\varphi W_x + \cos\varphi W_z) \\ \dot{W}_z^i &= \dot{W}_z + (\Omega_e + \dot{\ell}) \cos\varphi W_y + \dot{\varphi} W_x,\end{aligned} \tag{3.2.31}$$

with $\left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_i} = \{\dot{W}_x^i, \dot{W}_y^i, \dot{W}_z^i\}$ in the coordinate system of $\mathcal{R}_s$. In $\mathcal{R}_v$ we have

$$\begin{cases}\dot{W}_{xv}^i = \dot{W}_x^i \cos\psi \cos\gamma + \dot{W}_y^i \sin\psi \cos\gamma + \dot{W}_z^i \sin\gamma \\ \dot{W}_{yv}^i = -\dot{W}_x^i \sin\psi + \dot{W}_y^i \cos\psi \\ \dot{W}_{zv}^i = -\dot{W}_x^i \cos\psi \sin\gamma - \dot{W}_y^i \sin\psi \sin\gamma + \dot{W}_z^i \cos\gamma,\end{cases} \tag{3.2.32}$$

and the aircraft dynamics becomes:

$$\begin{cases}m\dot{V}_r = T\cos\alpha - D - mg\sin\gamma - m\dot{W}_{xv}^i \\ mV_r[\dot{\psi}\cos\gamma + (\Omega_e + \dot{\ell})(\cos\varphi\cos\psi\sin\gamma - \sin\varphi\cos\gamma) \\ \qquad +\dot{\varphi}\sin\psi\sin\gamma] = -(T\sin\alpha + L)\sin\mu - m\dot{W}_{yv}^i \\ mV_r\left[\dot{\gamma} + (\Omega_e + \dot{\ell})\cos\varphi\sin\psi + \dot{\varphi}\cos\psi\right] \\ \qquad = (T\sin\alpha + L)\cos\mu - mg\cos\gamma - m\dot{W}_{zv}^i\end{cases} \tag{3.2.33}$$

3.2.7 Projection of the wind speed onto the ground frame of reference

In subsections 3.2.3 to 3.2.6, we assumed that we had access to the wind components $\{W_x, W_y, W_z\}$ and derivatives $\{\dot{W}_x, \dot{W}_y, \dot{W}_z\}$ in a frame of reference attached to the ground. In the later two subsections, this frame of reference is also assumed to be aligned with the South-North and West-East directions. In fact, as explained in section 2.1, we usually have access to the wind speed norm W and to its direction relative to the North in the horizontal plane ψ_w. Although the data available corresponds to horizontal wind only, we may also consider the case that we have access to 3D wind, making an angle γ_w relative to the horizontal plane (see figure 3.7). In this case, how do we compute $\{W_x, W_y, W_z\}$ and $\{\dot{W}_x, \dot{W}_y, \dot{W}_z\}$ using W, ψ_w and γ_w ?

For this, let $\mathcal{R}_w = (G, \vec{x}_w, \vec{y}_w, \vec{z}_w)$ be the air frame of reference, which is not only translating relative to $\mathcal{R}_s$ ($\mathcal{R}_i$ in section 3.2.5) as we had before in subsection 3.2.3. We assume that $\vec{z}_w$ is still parallel to $\vec{z}_s$, but now $\vec{x}_w$ gives the direction towards which the wind is blowing in the horizontal plane $(\vec{x}_s, \vec{y}_s)$ and γ_w defines the angle between $\vec{W}$ and $\vec{x}_w$ around $\vec{y}_w$. So by projection, we have

$$\vec{W} = W \cos\gamma_w \vec{x}_w + \sin\gamma_w \vec{z}_w \tag{3.2.34}$$

and

$$\begin{aligned} W_x &= W \cos\gamma_w \cos\psi_w \\ W_y &= W \cos\gamma_w \sin\psi_w \\ W_z &= W \sin\gamma_w. \end{aligned} \tag{3.2.35}$$

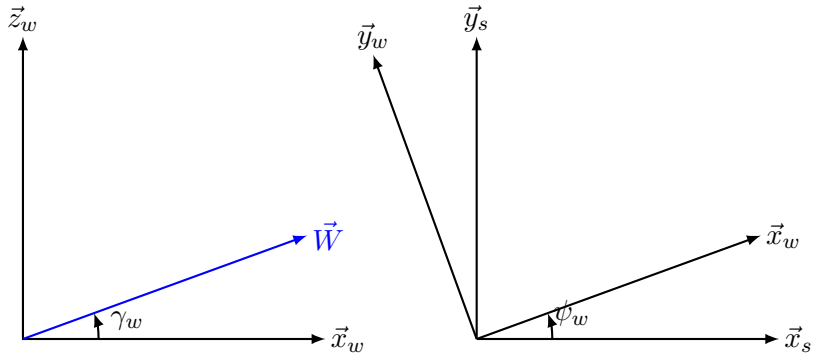


Figure 3.7 – $\mathcal{R}_w$ rotation relative to $\mathcal{R}_s$ (= $\mathcal{R}_i$ in subsection 3.2.5).

By differentiating relation (3.2.34) we get

$$\begin{aligned} \left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_w} &= (\dot{W}\cos\gamma_w - W\dot{\gamma}_w \sin\gamma_w)\vec{x}_w \\ &\quad + (\dot{W}\sin\gamma_w + W\dot{\gamma}_w \cos\gamma_w)\vec{z}_w. \end{aligned} \tag{3.2.36}$$

As done previously,

$$\begin{aligned}\left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_i} &= \left(\frac{d\vec{W}}{dt}\right)_{\mathcal{R}_w} + \vec{\Omega}_{\mathcal{R}_w/\mathcal{R}_i} \times \vec{W} \\ &= (\dot{W}\cos\gamma_w - W\dot{\gamma}_w \sin\gamma_w)\vec{x}_w \\ &+ (\dot{W}\sin\gamma_w + W\dot{\gamma}_w\cos\gamma_w)\vec{z}_w \\ &+ \dot{\psi}_w W\cos\gamma_w \underbrace{\vec{z}_s \times \vec{x}_w}_{\vec{y}_w}.\end{aligned} \tag{3.2.37}$$

Hence, by projecting on $\mathcal{R}_s$ we obtain

$$\begin{aligned}\dot{W}_x &= (\dot{W}\cos\gamma_w - W\dot{\gamma}_w\sin\gamma_w)\cos\psi_w - \dot{\psi}_w W\cos\gamma_w\sin\psi_w \\ \dot{W}_y &= (\dot{W}\cos\gamma_w - W\dot{\gamma}_w\sin\gamma_w)\sin\psi_w + \dot{\psi}_w W\cos\gamma_w\cos\psi_w \\ \dot{W}_z &= \dot{W}\sin\gamma_w + W\dot{\gamma}_w\cos\gamma_w.\end{aligned} \tag{3.2.38}$$

Under the assumption of horizontal wind only ($\gamma_w = \dot{\gamma}_w = 0$), we have

$$\begin{aligned}W_x &= W\cos\psi_w \\ W_y &= W\sin\psi_w \\ W_z &= 0,\end{aligned} \tag{3.2.39}$$

and

$$\begin{aligned}\dot{W}_x &= \dot{W}\cos\psi_w - \dot{\psi}_w W\sin\psi_w \\ \dot{W}_y &= \dot{W}\sin\psi_w + \dot{\psi}_w W\cos\psi_w \\ \dot{W}_z &= 0.\end{aligned} \tag{3.2.40}$$

3.3 Description of our identification problem

In the last section we listed several possibilities to model the dynamic behavior of an aircraft. The last models, i.e. those presented in sections 3.2.5 and 3.2.6, seem excessively complex to integrate, which is why we did not consider identifying them in our work. As we had not access to measurements of bank angles, nor to the variables necessary to derive it, we did not consider identifying the dynamics model presented in section 3.2.4 either. This left us with the flat Earth dynamics in a vertical plane with and without wind, which are summarized in equations (3.2.6) and (3.2.9). As we only considered the identification during the climb phase, the altitude plays an important role in our case. Hence, we added to these systems of equations the following kinematic equality linking the altitude to the true airspeed and path angle:

$$\dot{h} = V\sin\gamma + W_z. \tag{3.3.1}$$

In addition to this, we also added an equation translating the fact that the fuel consumption C_f is responsible for the aircraft mass variations:

$$\dot{m} = C_f = C_{sp}T, \tag{3.3.2}$$

where C_{sp} denotes the specific fuel consumption, which is an indicator of the engines performance (how much fuel is necessary to produce one unit of thrust in one unit of time).

The differential system obtained when we assume that there is wind writes

$$\begin{cases} \dot{h} = & V\sin\gamma + W_z, & (3.3.3)\\ \dot{V} = & \dfrac{T\cos\alpha - D - mg\sin\gamma - m\dot{W}_{xv}}{m}, & (3.3.4)\\ \dot{\gamma} = & \dfrac{T\sin\alpha + L - mg\cos\gamma - m\dot{W}_{zv}}{mV}, & (3.3.5)\\ \dot{m} = & -C_{sp}T, & (3.3.6) \end{cases}$$

where

$$\begin{cases} \dot{W}_{xv} = \dot{W}_x\cos\gamma - \dot{W}_z\sin\gamma \\ \dot{W}_{yv} = \dot{W}_y \\ \dot{W}_{zv} = \dot{W}_z\cos\gamma + \dot{W}_x\sin\gamma \end{cases} \tag{3.3.7}$$

The dynamics without wind is obtained by setting $W_x, W_y, W_z, \dot{W}_x, \dot{W}_y, \dot{W}_z$ to zero in (3.3.3)-(3.3.7). In this chapter, we call these two models respectively the *wind-dynamics* and *no-wind-dynamics*. They are compared in the numerical simulations presented in section 3.8. Note that the parametrization presented in section 3.4 stay the same for both dynamics, as well as all identification procedures described in section 3.6.

In system (3.3.3)-(3.3.6), we recognize the state variables ($\boldsymbol{x} = (h, V, \gamma, m)$) listed in section 3.1. In these equations, the elements T, D, L and C_{sp} (see table 3.1 for notations) are unknown and assumed to be functions of the state and control variables $(\boldsymbol{x}, \boldsymbol{u})$. Given an aircraft for which a sufficient amount of flight data is available, we aim to identify with good accuracy some model of its dynamics, which is equivalent to estimate the four functions: $T(\boldsymbol{x}, \boldsymbol{u}), D(\boldsymbol{x}, \boldsymbol{u}), L(\boldsymbol{x}, \boldsymbol{u})$ and $C_{sp}(\boldsymbol{x}, \boldsymbol{u})$. We insist that our intention is to get customized dynamics for each individual aircraft, as opposed to one general model for a whole class of airplanes. In the next section we describe the parametric models considered for the four functions to be identified.

3.4 Parametric hidden models

3.4.1 Restriction to parametric models

It is quite common in flight mechanics to assume the following dependencies based on physical considerations:

$$\begin{cases} T & \text{function of } M, \rho, N_1, \\ D & \text{function of } M, \rho, \alpha, V, \\ L & \text{function of } M, \rho, \alpha, V, \\ C_{sp} & \text{function of } M, h, SAT. \end{cases} \tag{3.4.1}$$

In order to apply identification methods to infer such functions, we need to assume that they belong to some function space, which is equivalent to choosing the form of the models. As explained in 3.1, the final aim of our identification task is to plug the learned dynamics into an optimal control problem, which will be solved using *direct transcription* methods. These type of methods involve time discretization of the original continuous problem, transforming it into an approximate finite dimensional nonlinear optimization problem. The obtained surrogate problem can then be solved using for example *sequential quadratic programming* or *interior point* algorithms. As these methods make use of the gradients of the objective and constraints of (6.2.1), optimal control solvers based on this type of techniques, such as BOCOP [Bonnans et al., 2017], rely on automatic differentiation tools. This means that our dynamics model, and hence the models of T, D, L and C_{sp}, need to be compatible with automatic differentiation. This made us limit the scope of our study to parametric models. Indeed, as nonparametric models are equivalent to the training data itself, they seemed too complex to be handled by this kind of tools.

3.4.2 Flight mechanics models

As stated in section 3.4.1, the thrust T, the drag D, the lift L and the specific fuel consumption C_{sp} are assumed to be parametric functions of the physical variables listed in (3.4.1). Many parametric models of these quantities can be found in the flight mechanics literature. In the case of the thrust and the specific consumption, our models were inspired by [Roux, 2005, chapter 2.2.1]:

$$T(M, \rho, N_1) = N_1 \rho^{0.6} \left(\boldsymbol{\theta}_1^T M^3 + \boldsymbol{\theta}_2^T \right), \tag{3.4.2}$$

$$C_{sp}(M, h, SAT) = \boldsymbol{\theta}_1^{csp} h + SAT^{\frac{1}{2}} \left(\boldsymbol{\theta}_2^{csp} + \boldsymbol{\theta}_3^{csp} h + \boldsymbol{\theta}_4^{csp} M + \boldsymbol{\theta}_5^{csp} hM \right), \tag{3.4.3}$$

where $\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \boldsymbol{\theta}_1^{csp}, \ldots, \boldsymbol{\theta}_5^{csp}$ are parameters to be estimated. As explicited in (3.4.4)-(3.4.5), these models are linear w.r.t their parameters:

$$T = X_T \cdot \boldsymbol{\theta}_T, \tag{3.4.4}$$

$$C_{sp} = X_{csp} \cdot \boldsymbol{\theta}_{csp}, \tag{3.4.5}$$

where

$$\begin{aligned} X_T &= N_1 \rho^{0.6} \left(M^3, 1\right), & \boldsymbol{\theta}_T &= \left(\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T\right), \\ X_{csp} &= \left(h, SAT^{\frac{1}{2}}, SAT^{\frac{1}{2}}h, SAT^{\frac{1}{2}}M, SAT^{\frac{1}{2}}hM\right), & \boldsymbol{\theta}_{csp} &= (\boldsymbol{\theta}_1^{csp}, \dots, \boldsymbol{\theta}_5^{csp}). \end{aligned}$$

and $\cdot$ denotes the standard dot product in Euclidean spaces.

Concerning the aerodynamic forces, we use the common model

$$\begin{cases} L(M, \rho, V, \alpha) = \frac{1}{2}\rho V^2 S C_z(\alpha, M), \\ D(M, \rho, V, \alpha) = \frac{1}{2}\rho V^2 S C_x(\alpha, M), \end{cases} \tag{3.4.6}$$

where S is the wing planform area, C_x is the drag coefficient and C_z is the lift coefficient [see e.g. Hull, 2007, p. 50]. Ignoring the value of S, we model the products SC_x and SC_z as polynomials of the couple of variables (α, M). Given $d \in \mathbb{N}^*$, we consider a family of $r = \begin{pmatrix} d+2 \\ 2 \end{pmatrix}$ monomials of degree at most d:

$$\begin{aligned} X^a = (X_1^a, \dots, X_r^a) &= \left(\alpha^k M^{j-k} : j = 0, \dots, d \quad ; \quad k = 0, \dots, j\right) \\ &= (1, \alpha, M, \alpha^2, \alpha M, \dots), \end{aligned} \tag{3.4.7}$$

The aerodynamic coefficients SC_x and SC_z are then modeled as linear combinations of these new features:

$$\begin{cases} SC_x(\alpha, M) = X^a \cdot \boldsymbol{\theta}_D, \\ SC_z(\alpha, M) = X^a \cdot \boldsymbol{\theta}_L, \end{cases} \tag{3.4.8}$$

where $\boldsymbol{\theta}_D, \boldsymbol{\theta}_L$ denote the vectors of parameters of D and L.

3.4.3 Remark on monomials selection

Most aerodynamic models found in the literature are polynomials as (4.3.1), but they are usually made of a few monomials. If the model which generated the data was of this sort, using a model which includes all monomials could lead to *overfitting* [see e.g. Hastie et al., 2009, chapter 7.2]. This motivates the search for a sparse structure of the parameter vectors $\boldsymbol{\theta}_D, \boldsymbol{\theta}_L$, which is commonly known in statistical learning as *feature selection*. This part of the study is postponed to chapter 4, although it has been used in the results presented in section 3.8.

In the next section we provide an overview of standard identification methods used in the aviation sector.

3.5 Aircraft systems identification state-of-the-art

3.5.1 The Output-Error Method

The *Output-Error Method* (OEM) [Maine and Iliff, 1985, 1986] is one the most widely used parametric system identification techniques in the aerospace and aeronautics industry. It falls into the "dynamical" class of identification approaches, which means that the system of ODE's describing the aircraft behavior is integrated several times in the estimation process. In the remaining of this subsection, we give a brief description of this method, while more details can be found in [Jategaonkar, 2006, chapter 4] and [Klein and Morelli, 2006, chapter 6.2] for example. This technique is closely related to what is called *parameter estimation* in the numerical control community. The reader is referred to Betts [2010] for a detailed description of these techniques in a more general setting.

Suppose that our dynamics depend on some parameters $\boldsymbol{\theta} \in \mathbb{R}^p$, $p \in \mathbb{N}^*$:

$$\dot{\boldsymbol{x}}(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}). \tag{3.5.1}$$

In this section we denote by $\boldsymbol{y} \in \mathcal{C}(\mathbb{R}_+, \mathbb{R}^{d_y})$ the vector-valued function representing measured physical quantities (listed in section 2.1 for our case). They are connected to the state and control variables through some mapping ϕ, depending on $\boldsymbol{\theta}$:

$$\boldsymbol{y}(t) = \phi(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}). \tag{3.5.2}$$

Equation (3.5.2) is commonly called *observation equation*. Note that $\boldsymbol{y}$ is a continuous function, whereas, in a practical setting, we only observe the system's outputs at discrete times $\{t_1, \dots, t_N\}$. Let z denote the function returning the measurements of $\boldsymbol{y}$ at these discrete times, assumed to be corrupted by some random measurement errors, denoted by v:

$$z(t_k) = \boldsymbol{y}(t_k) + v(t_k), \qquad k = 1, \dots, N. \tag{3.5.3}$$

The main assumptions made about this system are that the measurement errors $\{v(t_k)\}$ are independently drawn from a Gaussian distribution and satisfy for any $1 \leq j, k \leq N$

$$\mathbb{E}\left[v(t_k)\right] = 0 \tag{3.5.4}$$

$$\mathbb{E}\left[v(t_k)v(t_j)^\top\right] = R\delta_{kj}, \tag{3.5.5}$$

with R denoting the measurement error covariance matrix and δ_{kj} the Kronecker operator. We also assume that one has access to one or more sequences of controls and measurements $\{(u(t_k), z(t_k))\}_{k=1}^N$ from real flights. Given this assumptions, the approach falls into the following steps, illustrated in Figure 3.8:

1. Choose suitable initial values for the unknown parameters $\boldsymbol{\theta}_0$;

2. For some known flight controls $\{\boldsymbol{u}(t_k)\}_{k=1}^N$, integrate the state equation (3.5.1) using $\boldsymbol{\theta}_0$;

3. With the obtained states $\{\hat{\boldsymbol{x}}(t_k)\}_{k=1}^N$, the outputs can be defined as a function of the parameters $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{y}}(\boldsymbol{\theta}, t_k) := \phi(\hat{\boldsymbol{x}}(t_k), \boldsymbol{u}(t_k), \boldsymbol{\theta}); \tag{3.5.6}$$

4. Minimize the negative log-likelihood over $\boldsymbol{\theta}$ and R

$$\begin{aligned} \min_{\boldsymbol{\theta}, R} J(\boldsymbol{\theta}, R) = \frac{1}{2} \sum_{k=1}^{N} (z(t_k) - \hat{\boldsymbol{y}}(\boldsymbol{\theta}, t_k))^\top R^{-1} (z(t_k) - \hat{\boldsymbol{y}}(\boldsymbol{\theta}, t_k)] \\ + \frac{N}{2} \ln[\det R)] + \frac{N d_y}{2} \ln(2\pi), \end{aligned} \tag{3.5.7}$$

which summarizes to solving

$$\hat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}} \ \det(\hat{R}(\boldsymbol{\theta})), \tag{3.5.8}$$

$$\text{with } \hat{R}(\boldsymbol{\theta}) := \frac{1}{N} \sum_{k=1}^{N} (z(t_k) - \hat{\boldsymbol{y}}(\boldsymbol{\theta}, t_k))(z(t_k) - \hat{\boldsymbol{y}}(\boldsymbol{\theta}, t_k))^\top; \tag{3.5.9}$$

5. Stop if the desired accuracy is obtained in terms of $J(\hat{\boldsymbol{\theta}}, \hat{R})$, otherwise repeat steps 2 to 4 with the updated parameters $\hat{\boldsymbol{\theta}}$.

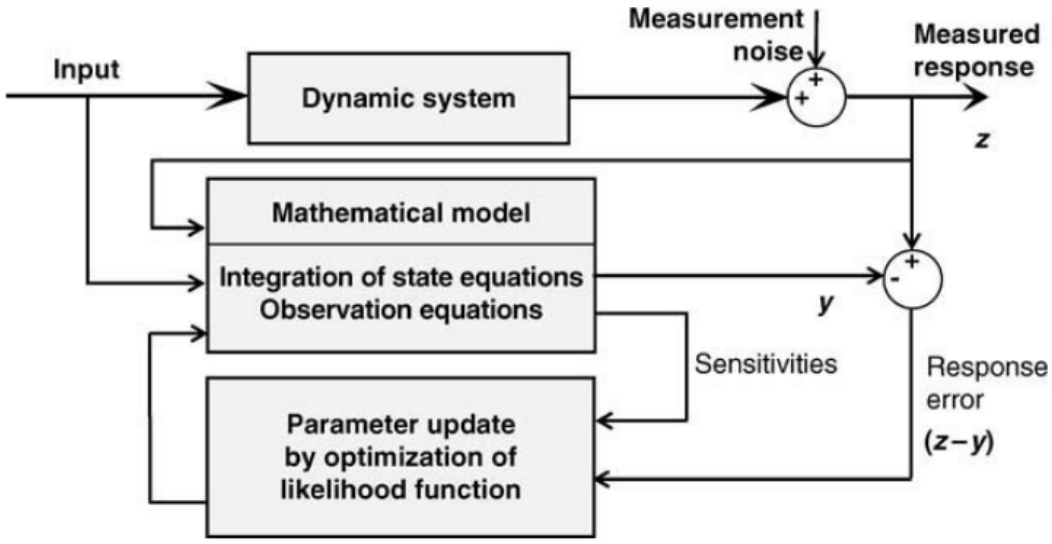


Figure 3.8 – Block schematic of Output-Error method (figure from [Jategaonkar, 2006, p.85]).

Step 4 corresponds to updating the parameters through maximum likelihood estimation. Further details explaining why (3.5.8)-(3.5.9) are equivalent to solving (3.5.7) are

given in section 3.6.3. Concerning step 2, several types of discrete integration schemes can be used. The fact that each iteration of this method involves integrating the whole dynamic system makes it computationally costly, specially when data from several flights are available.

3.5.2 The Filter-Error Method

The steps of the *Filter-Error Method* (FEM) have a really similar structure to those of the *Output-Error Method* (OEM), the main difference lying in the state integration in step 2. While the OEM considers that only measurement noise corrupts the data, in the FEM the system dynamics is supposed to be stochastic, i.e. perturbed by some process noise $w(t)$:

$$\dot{\boldsymbol{x}}(t) = g(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}) + w(t). \tag{3.5.10}$$

Hence, instead of integrating the dynamics (3.5.1), some *state estimator*, such as an extended Kalman filter [Peyada et al., 2008] or a neural network [Peyada and Ghosh, 2009], is used to filter the process noise and compute an approximation of the state sequence given some parameters $\boldsymbol{\theta}$ and a control sequence $\{\boldsymbol{u}(t_k)\}$. The steps of this approach are summarized in figure 3.9, while more details can be found in [Jategaonkar, 2006, chapter 5] or [Klein and Morelli, 2006, chapter 6.1].

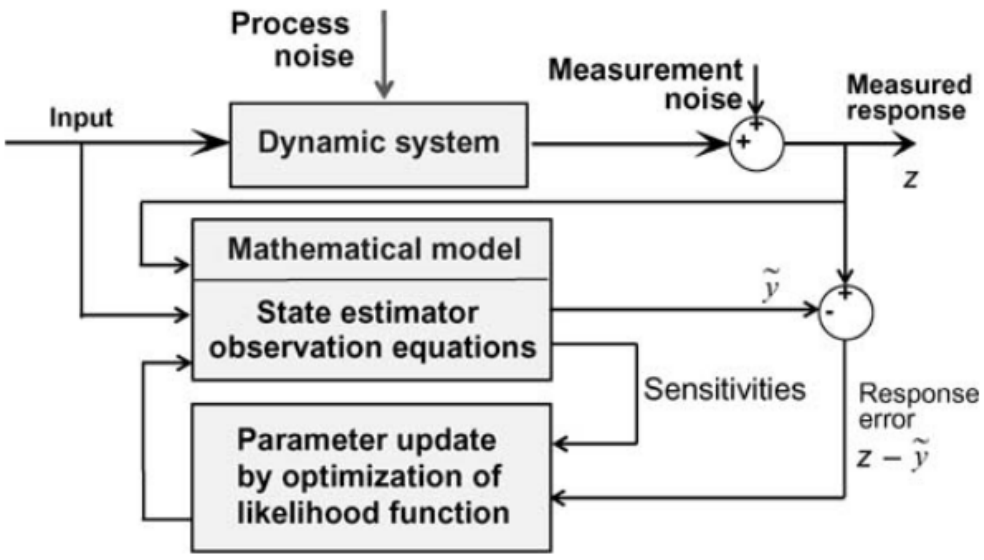


Figure 3.9 – Block schematic of Filter-Error method

While this approach is usually known by the control community as off-line parameter estimation using Kalman filters, from a statistical learning point of view, it may be interpreted as the application of an EM algorithm. Indeed, the E-step corresponds to the state filtering with fixed parameters and the M-step corresponds to the maximum likelihood

estimation of the parameters given the estimated state sequence [see e.g. Einicke et al., 2010].

While the FEM accounts for more types of errors when compared to the OEM, it is more difficult to tune and more computationally intensive. Another drawback is that, while many theoretical guarantees exits for the state filtering using Kalman filters when the system dynamics are linear, no optimal state estimator is known for nonlinear dynamics [Jategaonkar, 2006, p.146].

3.5.3 The Equation-Error Method

The *Equation-Error Method* adopts a completely different viewpoint compared to the former approaches. It supposes that the states and states derivatives are measured (or precomputed using measurements) and assumes that some additive noise models both errors in the measurements of $\{\dot{\boldsymbol{x}}(t_k)\}$ and random perturbations of the dynamics:

$$\dot{\boldsymbol{x}}(t_k) = g(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), \boldsymbol{\theta}) + \varepsilon(t_k), \qquad k = 1, \dots, N. \tag{3.5.11}$$

This approach originally disregards the temporal connection between observations of $\boldsymbol{x}, \boldsymbol{u}$ and $\dot{\boldsymbol{x}}$ and interprets the parameters estimation problem as a regression problem where the outputs are the state derivatives and the regressors are the states and controls

$$Y = (\dot{\boldsymbol{x}}(t_1), \dots, \dot{\boldsymbol{x}}(t_N)), \qquad X = ((\boldsymbol{u}(t_1), \boldsymbol{x}(t_1), \dots, (\boldsymbol{u}(t_N), \boldsymbol{x}(t_N)). \tag{3.5.12}$$

Initial uses of this approach fit the parameters in (3.5.11) using least-squares. Even when the dynamics are nonlinear, the model of g can be such that the dependence in the parameters $\boldsymbol{\theta}$ is linear:

$$Y = A(X) \cdot \boldsymbol{\theta} + \boldsymbol{e}, \tag{3.5.13}$$

where

$$\boldsymbol{e} = (\varepsilon(t_1), \dots, \varepsilon(t_N)). \tag{3.5.14}$$

In this case, the linear least-squares procedure reduces to a one-shot estimation based on a single matrix inversion:

$$\hat{\boldsymbol{\theta}} = (A^\top A)^{-1} A^\top Y. \tag{3.5.15}$$

In the case of a nonlinear dependence on the parameters, a gradient-based iterative procedure such as gradient descent [Cauchy, 1847], Gauss-Newton [see e.g. Bonnans, 2006, chapter 6.1.5] or Levenberg-Marquardt [Levenberg, 1944, Marquardt, 1963] can be used to minimize the least-squares criterion:

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^{N} \left\| \dot{\boldsymbol{x}}(t_k) - g(\boldsymbol{u}(t_k), \boldsymbol{x}(t_k), \boldsymbol{\theta}) \right\|_2^2. \tag{3.5.16}$$

Figure 3.10 illustrates this approach.

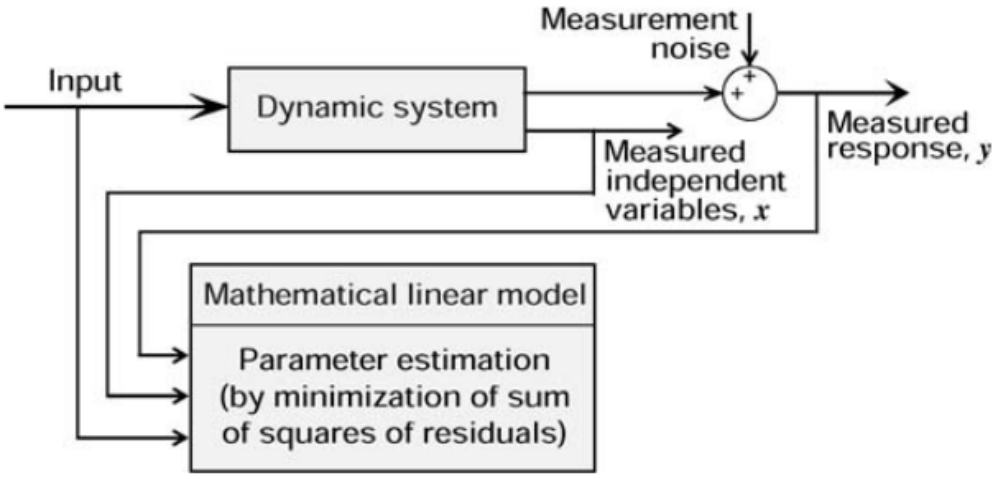


Figure 3.10 – Block schematic of Equation-Error method (from [Jategaonkar, 2006, p.179]).

This procedure can still be interpreted as *maximum likelihood* parameter estimation, just like OEM and FEM. The difference between the maximum likelihood estimations done in OEM-FEM and those performed in EEM relies only on the assumptions made regarding the noise type. A more detailed explanation on why the least-squares fitting in (3.5.16) can be interpreted as maximum likelihood estimation is given in section 3.6.

The *Equation-Error Method* [Greenberg, 1951] is older than the *Output-Error Method* and the *Filter-Error Method*. It was replaced by the later among the practitioners as more computing power started to be available, making possible multiple state integrations/estimations. Indeed, the OEM and FEM have the advantage over EEM of taking into account the time dependence in the data. They are also believed to lead to better accuracy in terms of system outputs $\boldsymbol{y}$, as these are the quantities which are fitted in these methods, while EEM fits the states derivatives $\dot{\boldsymbol{x}}$. Moreover, it is well-known that least-squares implementations of the EEM are more sensitive then the later to the quality of the data used.

In spite of these remarks, renewed interest for this older class of methods can be observed in the literature over the last decades [Morelli, 2000, 2017] for many possible reasons. Firstly, the measurement quality has considerably improved since the invention of the OEM, reducing the robustness and accuracy issues of the EEM from the past. The use of better preprocessing tools, prior to estimation, also helped coping with these difficulties. New versions of the method were also proposed in recent years. By making use of Fourier transformations, it was showed that the *frequency domain* EEM is able to produce unbiased estimations, even in a context of random design and high correlations

among regressors [Morelli, 2006].

Moreover, some EEM properties make it more suitable than OEM and FEM in many contexts. The EEM has the advantage over the OEM of accounting for both measurement and process noise, while having a simple implementation compared to the FEM. It is also easily extended to the case where multiple flights are processed simultaneously [Morelli, 2006], unlike OEM and FEM methods, which rely on state integration/estimation. For this same reason, EEM is a lot more scalable than the later, making it the privileged choice in a big data context. These reasons made us explore this framework for our aircraft dynamics identification problem.

3.6 Nonlinear multi-task regression

In this section, we present four possible instances of the *Equation-Error Method* based on *maximum likelihood* estimation. They will be compared to each other in section 3.8. For simplicity, all calculations in this section will be based on the *no-wind-dynamics* presented in section 3.3, but the same approach can be applied to the *wind-dynamics*.

3.6.1 Baseline regression method: Ordinary Least-Squares

The approach presented in this section is the most straightforward one. It will be used as a reference for comparison with all other approaches in the following sections. The first step to apply the EEM is to derive a set of regression problems from system (3.3.3)-(3.3.6) (where $W_z, \dot{W}_x, \dot{W}_y, \dot{W}_z$ have all been set to 0 for simplicity). We recall these equations here:

$$\begin{cases} \dot{h} = & V \sin\gamma, & (3.6.1) \\ \dot{V} = & \dfrac{T\cos\alpha - D - mg\sin\gamma}{m}, & (3.6.2) \\ \dot{\gamma} = & \dfrac{T\sin\alpha + L - mg\cos\gamma}{mV}, & (3.6.3) \\ \dot{m} = & -C_{sp}T, & (3.6.4) \end{cases}$$

We see that equation (3.6.1) does not contain any of the unknown functions listed in section 3.4, which means it is not useful here. While function T appears in (3.6.2)-(3.6.4), note that D, L and C_{sp} take part in a single equation each. Equation (3.6.4) is clearly the problematic one, because of the presence of a product between the unknowns C_{sp} and T. This means that we do not have linearity on the parameters of the r.h.s. of this equations, but also that it cannot be used alone to determine both elements separately ; only the product of them is *identifiable* here:

Definition 3.6.1 *Identifiable model [Walter and Pronzato, 1997, chapter 2.6.1]*

Let $\mathcal{M} : \Theta \to \mathcal{Y}$ be a model defined on some parameters space Θ into some output

space $\mathcal{Y}$. $\mathcal{M}$ *is said to be* identifiable *if for any* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$,

$$\mathcal{M}(\boldsymbol{\theta}_1) = \mathcal{M}(\boldsymbol{\theta}_2) \Rightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2. \tag{3.6.5}$$

The simplest way of overcoming such difficulties is to choose not to infer C_{sp} and use a general model of it instead. Even though this idea does not solve the real problem, its simplicity led us to test it. For this, we chose the specific consumption model from [Roux, 2005, Chapter 2.1.2], that we shall denote C_{sp}^{ER} hereafter. Thus, in this approach, only T, D and L are to be estimated based on equations (3.6.2)-(3.6.4).

By isolating only a single unknown term in the right-hand side of each equation, the system may be rewritten as follows:

$$\begin{cases} -m\dot{V} - mg\sin\gamma + \dfrac{C_f}{C_{sp}^{ER}}\cos\alpha = D, \\ mV\dot{\gamma} + mg\cos\gamma - \dfrac{C_f}{C_{sp}^{ER}}\sin\alpha = L, \\ \dfrac{C_f}{C_{sp}^{ER}} = T. \end{cases} \tag{3.6.6}$$

In system (3.6.6) we make two assumptions: (i) the mass rate $\dot{m}$ has been replaced by the negative total fuel flow $-C_f$ and (ii) that T is assumed to be strictly equal to C_f/C_{sp}^{ER}. Substituting the expressions of T, D and L in (3.6.6) by the parametric models from section 3.4, we obtain the following set of regression problems:

$$\begin{cases} Y_D = X_D \cdot \boldsymbol{\theta}_D + \boldsymbol{\theta}_{D0} + \varepsilon_D, \\ Y_L = X_L \cdot \boldsymbol{\theta}_L + \boldsymbol{\theta}_{L0} + \varepsilon_L, \\ Y_T = X_T \cdot \boldsymbol{\theta}_T + \boldsymbol{\theta}_{T0} + \varepsilon_T, \end{cases} \tag{3.6.7}$$

with

$$\begin{aligned} Y_D &= -m\dot{V} - mg\sin\gamma + \frac{C_f}{C_{sp}^{ER}}\cos\alpha, \\ Y_L &= mV\dot{\gamma} + mg\cos\gamma - \frac{C_f}{C_{sp}^{ER}}\sin\alpha, \\ Y_T &= \frac{C_f}{C_{sp}^{ER}}, \qquad \text{and} \qquad X_D = X_L = \frac{1}{2}\rho V^2 X^a. \end{aligned} \tag{3.6.8}$$

The *intercepts* $\boldsymbol{\theta}_{D0}, \boldsymbol{\theta}_{L0}$ and $\boldsymbol{\theta}_{T0}$, which are also estimated, should account for possible offsets in the dynamics formulation, and $\varepsilon_D, \varepsilon_L, \varepsilon_T$ are random variables representing the measurement and process errors.

Remark Had we different assumptions for the dynamics of our aircraft, we would simply

add some wind terms in the left-hand side of (3.6.6), which would change the target to be fit (3.6.8) but not the structure of the regression problems (3.6.7).

We consider now only one of this regression problems taken separately, indexed by $\ell \in \{D, L, T\}$. Given a sample $\{(x_{\ell,i}, y_{\ell,i})\}_{i=1}^{N}$ of $N \in \mathbb{N}^*$ observations drawn independently from the joint distribution of (X_ℓ, Y_ℓ), we may rewrite each of the regression problems in (3.6.7) in matrix form:

$$\boldsymbol{Y}_\ell = \boldsymbol{X}_\ell \boldsymbol{\theta}_\ell + \boldsymbol{\varepsilon}_\ell, \tag{3.6.9}$$

where $\boldsymbol{Y}_\ell = (y_{\ell,1}, \dots, y_{\ell,N})$ and $\boldsymbol{\varepsilon}_\ell = (\varepsilon_{\ell,1}, \dots, \varepsilon_{\ell,N})$ are vectors of $\mathbb{R}^N$ and $\boldsymbol{X}_\ell = (x_{\ell,1}, \dots, x_{\ell,N})$ is a $N \times p_\ell$ matrix of $\mathcal{M}_{N,p_\ell}(\mathbb{R})$. Note that each $x_{\ell,i}$ is a column vector of size p_ℓ (number of features of the linear model ℓ). Also note that the intercepts $\boldsymbol{\theta}_{\ell,0}$ have been included in the parameters vectors $\boldsymbol{\theta}_\ell$, by adding a column of ones to the matrices $\boldsymbol{X}_\ell$. One possible estimator of these parameters is the *Ordinary Least-Squares* estimators:

$$\hat{\boldsymbol{\theta}}_\ell = (\boldsymbol{X}_\ell^\top \boldsymbol{X}_\ell)^{-1} \boldsymbol{X}_\ell^\top \boldsymbol{Y}_\ell. \tag{3.6.10}$$

Many implementations based on SVD or Cholesky decompositions exist, allowing to solve (3.6.10) very efficiently, even in the case of large samples. Furthermore, according to the Gauss-Markov Theorem [see e.g. Bonnans, 2006, p.215], estimator (3.6.10) is the *Best Linear Unbiased Estimator*[1] under the following assumptions:

- the covariates observations $\{x_{\ell,i}, i = 1, \dots, N\}$ are known (not drawn from a random variable);
- the noises have zero mean: $\mathbb{E}[\varepsilon_{\ell,i}] = 0$, for $i = 1, \dots, N$;
- they are *homoscedastic*: $\mathrm{Var}[\varepsilon_{\ell,i}] = \sigma^2 < \infty$, for $i = 1, \dots, N$;
- and are mutually *uncorrelated*: $\mathrm{Cov}[\varepsilon_{\ell,i}, \varepsilon_{\ell,j}] = 0$, for any $i \neq j$.

Such assumptions are not all verified in practice. We know for example that there is some uncertainty in our covariates observations, that the noise is probably not homoscedastic and that the autocorrelation between data points is quite significant. Considering this, extended versions of this estimator may be better choices in our case, such as the *Weighted Least-Squares*, the *Generalized Least-Squares* [Aitken, 1936] or the *Total Least-Squares* [see e.g. Jategaonkar, 2006, chapter 6.5]. These options were not studied however, because they would still not allow us to infer C_{sp}, which means we would still not be solving the initial problem.

3.6.2 Multi-task framework

As stated in section 3.6.1, two of the main difficulties of the identification problem treated here are:

1. "*Best*" means of minimum variance here.

1. the *non-identifiability* (see definition 3.6.1) of the product of T and C_{sp} in equation (3.3.6) and

2. the fact that T is present in (3.3.4)-(3.3.6), which means that we have no chance of obtaining the same results three times if we used theses equations separately to identify it.

These remarks leads naturally to consider solving all three regressions jointly. This framework is usually called *multi-task regression* [Caruana, 1997] in the statistical learning community, although it is not a new concept. One potential benefit of multi-task learning for our case is that it allows us to enforce all equations to share the same thrust function T, solving difficulty 2. By doing so, we expect that information concerning T gathered while identifying equations (3.3.4) and (3.3.5) will be transferred to equation (3.3.6) during the estimation process, which should help to reduce the non-identifiability issue 1.

Unlike what was done in (3.6.6) for the Single-task Ordinary Least-Squares, here we will leave T in the r.h.s. of equations (3.3.4)-(3.3.6), since we want to use all of them to estimate it:

$$\begin{cases} m\dot{V} + mg\sin\gamma = T\cos\alpha - D, \\ mV\dot{\gamma} + mg\cos\gamma = T\sin\alpha + L, \\ C = C_{sp}T. \end{cases} \tag{3.6.11}$$

As before, by injecting T, D, L and C_{sp} as defined in (3.4.4)-(3.4.6) and (4.3.1), we build the set of regression problems:

$$\begin{cases} Y_1 = X_1 \cdot \boldsymbol{\theta}_1 + \varepsilon_1, \\ Y_2 = X_2 \cdot \boldsymbol{\theta}_2 + \varepsilon_2, \\ Y_3 = (X_T \cdot \boldsymbol{\theta}_T)(X_{csp} \cdot \boldsymbol{\theta}_{csp}) + \varepsilon_3, \end{cases} \tag{3.6.12}$$

where

$$\begin{aligned} Y_1 &= m\dot{V} + mg\sin\gamma, \\ Y_2 &= mV\dot{\gamma} + mg\cos\gamma, \\ Y_3 &= C, \end{aligned} \tag{3.6.13}$$

and

$$X_1 = \begin{pmatrix} X_T\cos\alpha \\ -X_D \end{pmatrix}, X_2 = \begin{pmatrix} X_T\sin\alpha \\ X_L \end{pmatrix}, \boldsymbol{\theta}_1 = \begin{pmatrix} \boldsymbol{\theta}_T \\ \boldsymbol{\theta}_D \end{pmatrix}, \boldsymbol{\theta}_2 = \begin{pmatrix} \boldsymbol{\theta}_T \\ \boldsymbol{\theta}_L \end{pmatrix}. \tag{3.6.14}$$

Remark In the third equation of (3.6.12), we can still add an *intercept* as in section 3.6.1, but it cannot be done by augmenting vector X_T or X_{csp}. In this case, we adapt the strategy by fitting the centered targets $\{\tilde{Y}_i = Y_i - \bar{Y}_i\}_{i=1}^3$ without intercepts and setting them *a posteriori* to be equal to the targets means: $\boldsymbol{\theta}_{i0} = \bar{Y}_i$, for $i = 1, \ldots, 3$.

As previously explained, we will consider the three regression problems from (3.6.12) as a single one, called *multi-task regression problem*

$$Y = f(X, \boldsymbol{\theta}) + \varepsilon, \tag{3.6.15}$$

where

$$Y = (Y_1, Y_2, Y_3), \quad \varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3) \in \mathbb{R}^3, \tag{3.6.16}$$

$$X = (X_T, X_{csp}, X_L, X_D) \in \mathbb{R}^p, \quad \boldsymbol{\theta} = (\boldsymbol{\theta}_T, \boldsymbol{\theta}_{csp}, \boldsymbol{\theta}_L, \boldsymbol{\theta}_D) \in \mathbb{R}^p. \tag{3.6.17}$$

Note that, for any $x \in \mathbb{R}^p$, function $f(x, \cdot) : \boldsymbol{\theta} \mapsto f(x, \boldsymbol{\theta})$ is nonlinear. We will call hereafter a *task* each regression problem from system (3.6.12). The number of *tasks* solved by a multi-task regression is hence given by the dimension of the outputs vector Y.

3.6.3 Maximum Likelihood Estimators

Multivariate Gaussian likelihood assumption Let us assume here for more generality that we are dealing with $K \in \mathbb{N}^*$ *tasks*, i.e. $Y, \varepsilon \in \mathbb{R}^K$. Considering a training set $\{(x_i, y_i)\}_{i=1}^N$, we are assuming through our regression model that there is some $\boldsymbol{\theta} \in \mathbb{R}^p$ such that, for any $i = 1, \ldots, N$

$$y_i = f(x_i, \boldsymbol{\theta}) + \varepsilon_i. \tag{3.6.18}$$

For some symmetric covariance matrix $\boldsymbol{\Sigma} \in \mathcal{M}_p(\mathbb{R})$, we assume here that the noises $\{\varepsilon_i\}_{i=1}^N$ are drawn independently from the same centered multivariate Gaussian distribution $\mathcal{N}(0, \boldsymbol{\Sigma})$. Given $\boldsymbol{\theta} \in \mathbb{R}^p$, the samples likelihood writes

$$\mathcal{L}_{ML}(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} [(2\pi)^K \det \boldsymbol{\Sigma}]^{-\frac{1}{2}} \exp\left(-\frac{1}{2} e_i(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} e_i(\boldsymbol{\theta})\right), \tag{3.6.19}$$

where $e_i(\boldsymbol{\theta}) = y_i - f(\boldsymbol{\theta}, x_i) \in \mathbb{R}^K$ denotes the residue's i^{th} component. With these assumptions, the maximum likelihood estimation is obtained by maximizing (3.6.19), which is equivalent to minimize the *negative log-likelihood* criteria:

$$-\log L(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \frac{NK}{2} \log(2\pi) + \frac{N}{2} \log \det \boldsymbol{\Sigma} + \frac{1}{2} \sum_{i=1}^{N} e_i(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} e_i(\boldsymbol{\theta}). \tag{3.6.20}$$

Multi-task weighted least-squares If we knew the covariance matrix $\boldsymbol{\Sigma}$, the Maximum Likelihood estimator under such assumptions would be obtained by maximizing

criterion (3.6.20) with relation to $\boldsymbol{\theta}$, which is equivalent to solving the following *weighted least-squares* problem:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} e_i(\boldsymbol{\theta})^\top \boldsymbol{\Sigma}^{-1} e_i(\boldsymbol{\theta}). \tag{3.6.21}$$

It becomes clear through expression (3.6.21) that, under the assumption that the K tasks have mutually independent noises of same variance σ (i.e. $\boldsymbol{\Sigma} = \sigma I_K$, with I_K the $K \times K$ Identity matrix), our estimator is equivalent to the standard *nonlinear least-squares* estimator cited in section 3.5.3:

$$\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^p} \sum_{i=1}^{N} e_i(\boldsymbol{\theta})^\top e_i(\boldsymbol{\theta}) = \sum_{i=1}^{N} \|y_i - f(x_i, \boldsymbol{\theta})\|_2^2. \tag{3.6.22}$$

Problem (3.6.22) is an unconstrained nonlinear optimization problem, which is usually non-convex. Typical algorithms used to solve it are standard gradient descent [Cauchy, 1847], Gauss-Newton [see e.g. Bonnans, 2006, chapter 6.1.5] or Levenberg-Marquardt [Levenberg, 1944, Marquardt, 1963], although they are not guaranteed to lead to a global minimum.

General case As we do not know $\boldsymbol{\Sigma}$ in our case, we will keep it as a variable in our optimization problem, which becomes after eliminating the constant term:

$$\min_{\boldsymbol{\theta}, \boldsymbol{\Sigma}} J(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \frac{N}{2} \log\det \boldsymbol{\Sigma} + \frac{1}{2} \sum_{i=1}^{N} e_i^\top \boldsymbol{\Sigma}^{-1} e_i. \tag{3.6.23}$$

First order optimality condition with regard to $\boldsymbol{\Sigma}$ gives

$$\frac{\partial}{\partial \boldsymbol{\Sigma}} J(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \frac{N}{2} \frac{\partial}{\partial \boldsymbol{\Sigma}} (\log\det \boldsymbol{\Sigma}) + \frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\Sigma}} \left(e_i^\top \boldsymbol{\Sigma}^{-1} e_i \right) \tag{3.6.24}$$

$$= \frac{N}{2} \boldsymbol{\Sigma}^{-1} - \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1} e_i e_i^\top \boldsymbol{\Sigma}^{-1} \tag{3.6.25}$$

$$= \frac{1}{2} \boldsymbol{\Sigma}^{-1} \left(N I_K - \left[\sum_{i=1}^{N} e_i e_i^\top \right] \boldsymbol{\Sigma}^{-1} \right) = 0_K, \tag{3.6.26}$$

where 0_K is the $K \times K$ zero matrix. This leads to the classic covariance estimator

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^{N} e_i e_i^\top = \frac{1}{N} \sum_{i=1}^{N} \Big(y_i - f(\boldsymbol{\theta}, x_i) \Big) \Big(y_i - f(\boldsymbol{\theta}, x_i) \Big)^\top. \tag{3.6.27}$$

Using such estimator, we may note that

$$\sum_{i=1}^{N} e_i^\top \hat{\boldsymbol{\Sigma}}^{-1} e_i = \operatorname{Tr}\left\{ \left(\sum_{i=1}^{N} e_i e_i^\top \right) \hat{\boldsymbol{\Sigma}}^{-1} \right\} \tag{3.6.28}$$

$$= N \operatorname{Tr}(I_K) = NK, \tag{3.6.29}$$

which allows us to simplify our log likelihood criterion (3.6.23) into

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \hat{\boldsymbol{\Sigma}}) = \log\det \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}). \tag{3.6.30}$$

Further details on this derivation can be found for example in [Goodwin and Payne, 1977, p.263] or [Walter and Pronzato, 1997, p.46].

Problem (3.6.30) is unconstrained and non-convex, as (3.6.22), but is also highly non-linear. We add to this problem the constraint that $\hat{\boldsymbol{\Sigma}}$ is positive-definite. Standard algorithms which can be used to solve it are *interior point methods* [see e.g. Boyd and Vandenberghe, 2004, chapter 11] or *sequential quadratic programming* [see e.g. Bonnans et al., 2013, chapter 15.1].

Cholesky relaxation As we assumed that $\hat{\boldsymbol{\Sigma}}$ is positive definite, this means that it has a unique modified Cholesky decomposition

$$\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}) = LDL^\top, \tag{3.6.31}$$

where L is a unit lower triangular matrix and D is a positive diagonal matrix. We see that if we replace $\hat{\boldsymbol{\Sigma}}$ by its decomposition in (3.6.30), then its determinant $\det \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta})$ is the product of the diagonal terms of D, which are all positive by assumption:

$$\det \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}) = \prod_{j=1}^{K} D_{jj} > 0. \tag{3.6.32}$$

In this case, optimization problem (3.6.30) becomes

$$\min_{\boldsymbol{\theta}, L, D} \sum_{j=1}^{K} \log D_{jj}$$
$$\text{such that} \begin{cases} D_{jj} > 0, \quad \forall j = 1, \dots, K \\ \hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta}) = LDL^\top. \end{cases} \tag{3.6.33}$$

Compared to (3.6.30), the unknown variables of this new problem lie in a space of higher dimension, which makes it seem more complex than the former. However, the objective function has been greatly simplified in the process, which led us to compare this formulation to (3.6.30) in our numerical simulations.

3.7 Quality criteria design

Before comparing the methods presented above, a set of quality criteria must be chosen. We consider the case in which the methods are assessed using a data set of $q > 0$ flights.

3.7.1 Static criterion

Given a randomly chosen flight $1 \leq i \leq q$, an estimator of the parameters $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_T, \hat{\boldsymbol{\theta}}_{csp}, \hat{\boldsymbol{\theta}}_L, \hat{\boldsymbol{\theta}}_D)$ can be built using the $q-1$ flights left. We call the i^{th} flight the *test flight* and denote the $N_i \in \mathbb{N}^*$ observations of its state, control and state derivatives by $\{(\boldsymbol{x}^i(t_k), \boldsymbol{u}^i(t_k), \dot{\boldsymbol{x}}^i(t_k))\}_{k=1}^{N_i}$. By plugging the parameters estimates in the desired models of the dynamics and functions T, D, L, C_{sp} (sections 3.3-3.4), we can compute pointwise estimations of the state derivatives of the test flight:

$$\hat{\dot{\boldsymbol{x}}}^i(t_k) := \left(\hat{\dot{h}}(t_k), \hat{\dot{V}}(t_k), \hat{\dot{\gamma}}(t_k), \hat{\dot{m}}(t_k)\right) = g\left(\boldsymbol{x}^i(t_k), \boldsymbol{u}^i(t_k), \hat{\boldsymbol{\theta}}\right). \tag{3.7.1}$$

From a regression point of view, a standard way of assessing our estimations is by computing the *mean squared-error*:

$$\mathcal{C}_1^i = \frac{1}{N_i} \sum_{k=1}^{N_i} \|\dot{\boldsymbol{x}}^i(t_k) - \hat{\dot{\boldsymbol{x}}}^i(t_k)\|_{\dot{\boldsymbol{x}}}^2, \tag{3.7.2}$$

where $\|\cdot\|_{\dot{\boldsymbol{x}}}$ denotes a scaling norm which brings all the components of $\dot{\boldsymbol{x}}$ to the same order of magnitude before applying the usual Euclidean norm in $\mathbb{R}^4$. If we repeat this procedure q times by permuting the test flight chosen, we may average the obtained scores, leading to a *leave-one-out cross-validation* score [Stone, 1974]:

$$\mathcal{C}_1 = \frac{1}{q} \sum_{i=1}^{q} \mathcal{C}_1^i. \tag{3.7.3}$$

As explained in section 2.2.2, this type of criterion allows to approximate the *expected prediction error*. Unlike the score obtained by estimating the parameters and evaluating the mean squared-error using the same dataset, this type of procedure provides us from misinterpreting *overfitting* as good performance.

3.7.2 Dynamic criterion

The issue with the previous criterion is that it is static: it does not incorporate the fact that the observations are time dependent. However, as stated in section 3.1, our ambition is to estimate an accurate model of the aircraft dynamics in order to plug it in an optimal control problem (6.2.1). For solving such a problem by *direct transcription*, the identified dynamic system will be integrated several times using different controls in order to look for the admissible control sequence which minimizes the objective function of (6.2.1). By

integrating estimated state derivatives of the test flight $\{\hat{\dot{\boldsymbol{x}}}^i(t_k)\}_{k=1}^{N_i}$ using the observed controls $\{\boldsymbol{u}^i(t_k)\}_{k=1}^{N_i}$, we should be able to get a sequence of states $\{\hat{\boldsymbol{x}}^i(t_k)\}_{k=1}^{N_i}$ supposed to be close to the observed states $\{\boldsymbol{x}^i(t_k)\}_{k=1}^{N_i}$ if our estimation procedure is good. This remark points towards the direction of a criterion of the form

$$\mathcal{C}_2^i = \frac{1}{N_i}\sum_{k=1}^{N_i} \|\boldsymbol{x}^i(t_k) - \hat{\boldsymbol{x}}^i(t_k)\|_{\boldsymbol{x}}^2, \tag{3.7.4}$$

where $\|\cdot\|_{\boldsymbol{x}}$ is a scaling Euclidean norm of $\boldsymbol{x}$, analogous to $\|\cdot\|_{\dot{\boldsymbol{x}}}$. In practice, we cannot use this assessment approach. Indeed, the small errors on the estimations of $\{\hat{\dot{\boldsymbol{x}}}^i(t_k)\}_{k=1}^{N_i}$ accumulate over the integration steps, leading systematically to integrated states $\{\hat{\boldsymbol{x}}^i(t_k)\}_{k=1}^{N_i}$ which move away quite rapidly from the measured ones $\{\boldsymbol{x}^i(t_k)\}_{k=1}^{N_i}$. This behavior is illustrated on figure 3.11.

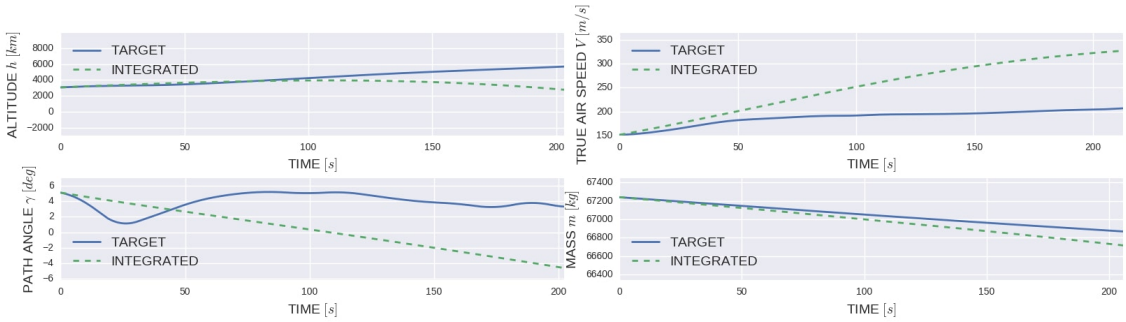


Figure 3.11 – (Dashed green) Direct integration of predicted dynamics of the beginning of a flight (using single-task OLS). (Blue) Observed states derivatives.

To counter such phenomenon, we try to find, for a given test flight, a control sequence not very far from the measured one which can bring the integrated state sequence as close as possible to the real state sequence. This idea can be translated into the following optimal control problem:

$$\begin{aligned} \min_{(\boldsymbol{u},\boldsymbol{x})\in\mathbb{U}\times\mathbb{X}} & \int_0^{t_{N_i}} \left(\|\boldsymbol{u}(t) - \boldsymbol{u}^i(t)\|_{\boldsymbol{u}}^2 + \|\boldsymbol{x}(t) - \boldsymbol{x}^i(t)\|_{\boldsymbol{x}}^2\right) dt \\ & \text{s.t.} \quad \dot{\boldsymbol{x}} = g(\boldsymbol{x}, \boldsymbol{u}, \hat{\boldsymbol{\theta}}), \end{aligned} \tag{3.7.5}$$

where the norm $\|\,.\,\|_{\boldsymbol{u}}$ is an analog of $\|\,.\,\|_{\boldsymbol{x}}$. If we denote $(\boldsymbol{u}^{*,i}, \boldsymbol{x}^{*,i})$ a solution of problem (3.7.5), we can define the following *dynamic quality criterion*:

$$\mathcal{C}_3^i = \frac{1}{N_i}\sum_{k=1}^{N_i} \left(\|\boldsymbol{x}^i(t_k) - \boldsymbol{x}^{*,i}(t_k)\|_{\boldsymbol{x}}^2 + \|\boldsymbol{u}^i(t_k) - \boldsymbol{u}^{*,i}(t_k)\|_{\boldsymbol{u}}^2\right). \tag{3.7.6}$$

As $\mathcal{C}_1$, this last criterion can be cross-validated across all possible permutations of test

flights:

$$\mathcal{C}_3 = \frac{1}{q}\sum_{i=1}^{q}\mathcal{C}_3^i. \tag{3.7.7}$$

3.8 Numerical benchmark

Implementation details

We compared the identification approaches described in section 3.6 using real QAR data from a single medium haul aircraft, preprocessed by the procedure described in chapter 2. The dataset used contained recordings from $q = 424$ flights, which corresponds to 334 531 point observations. The models used for T, C_{sp}, L and D are those presented in section 3.4 and the dynamics models are the *no-wind-dynamics* and *wind-dynamics* from section 3.3. The monomial integrating the models of L and D were selected prior to the parameters estimations. The description of the approach used and the results are postponed to section 4.3 in the next chapter.

Our baseline for the estimation part was the single-task ordinary least-squares (section 3.6.1), denoted *single-task OLS* hereafter. The other methods compared were the multi-task nonlinear least-squares (3.6.22), the multi-task Gaussian maximum likelihood with unknown covariance (3.6.30) and its Cholesky relaxation (3.6.33). They were respectively denoted by *multi-task NLS*, *multi-task ML* and *Cholesky ML* hereafter. The optimization problems were solved using functions of Python's SCIPY library [Jones et al., 2001–], as listed in table 3.2. Since the single-task OLS is the only method which is not iterative, its solution was used to initialize the multi-task NLS, whose solution was used to initialize the other two appoaches. In order to compute the dynamic scores $\mathcal{C}_3$, problem (3.7.5) was solved performed using BOCOP [Bonnans et al., 2017, Team Commands, 2017].

Table 3.2 – Optimization algorithms used

Estimation approach	Function used	Algorithm
Single-task OLS	LINALG.LSTSQ	SVD decomposition
Multi-task NLS	OPTIMIZE.LEASTSQ	Levenberg-Marquardt
Multi-task ML	OPTIMIZE.FMIN_SLSQP	Dieter Kraft's SQP algorithm Kraft [1988]
Cholesky ML	OPTIMIZE.FMIN_SLSQP	Dieter Kraft's SQP algorithm Kraft [1988]

Results

Figure 3.12 depicts the observed state derivatives for one flight (blue full line), as well as the estimations obtained using each one of the four EEM instances. The *no-wind-dynamics* was used for these estimations. As explained in section 3.7, all parameters were identified using data from the other recorded flights. We observe that all approaches have more trouble to estimate the path angle derivative $\dot{\gamma}$, but are nonetheless able to get reasonably close to all targets. We obtained similar results for other flights, with some quite systematic

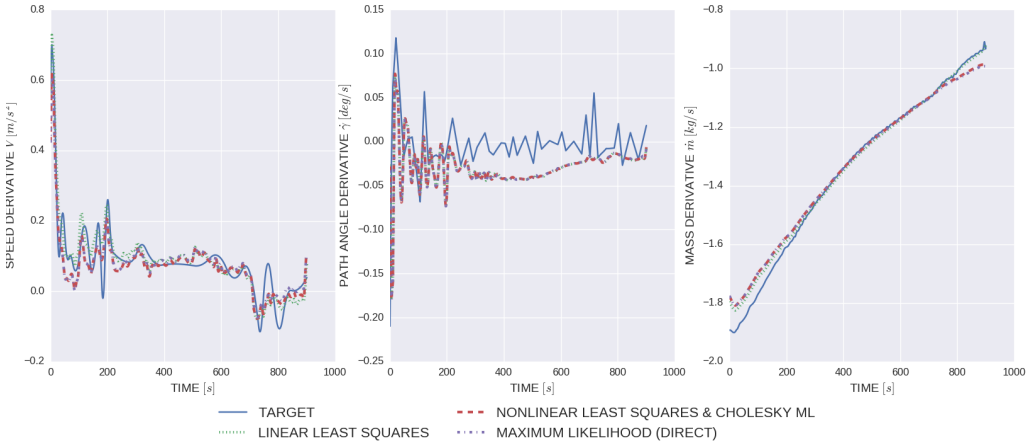


Figure 3.12 – Static results for single-task OLS (dotted green - baseline), multi-task NLS/Cholesky ML (dashed red) and multi-task ML (dot-dash purple).

biases such as. For example, all approaches seem to overestimate $\dot{m}$ at the beginning of the flights and underestimate of it at the end of the flights. Another systematic behavior observed was that the Cholesky ML, which is initialized at the solution given by the multi-task NLS, always stops after a single iteration since it is not able to improve the initial guess (reason why there is only on curve for both).

Table 3.3 contains the mean value and standard deviation of the cross-validated criterion $\mathcal{C}_1$ presented in section 3.7. First, we notice that better (lower) mean scores have been obtained using the wind dynamics for all approaches, with no impact on the standard deviation. Furthermore, although the three multi-task approaches outperform in average the single-task baseline, the differences does not seem statistically significant since the standard deviations obtained are quite high relatively to the mean scores gaps. In spite of this remark, the best averaged accuracy and lower variance was obtained by the multi-task ML approach, which is only 6 times slower to evaluate than the one-shot OLS, being the fastest multi-task algorithm tested.

Table 3.3 – Cross-validated $\mathcal{C}_1$ criterion results

Method	**No wind dynamics**		**Wind dynamics**		**Training time*** $[s]$	
	Mean	**Std.**	**Mean**	**Std.**	**Mean**	**Std.**
Single-task OLS	1.103	0.261	1.101	0.262	0.8	0.1
Multi-task NLS	1.025	0.208	0.989	0.205	32.8	4.4
Multi-task ML	1.023	0.207	0.988	0.204	4.8	0.6
Cholesky ML	1.025	0.208	0.989	0.205	672.0	49.7

*For learning sets of size 298846 ± 154, on 2.5 GHz and 24.6 GB RAM

Concerning the dynamic criterion, figure 4.7 shows the solution of problem (3.7.5) for the same flight depicted on figure 3.12. They were obtained using the *no-wind-dynamics*

and allow to compare the single-task OLS to the multi-task NLS approach. Notice that the curves are plotted as functions of the altitude, which is possible since the latter is monotonic over the climb phase. We see that the predicted dynamics were able to reproduce quite accurately the recorded flight in this case with a minor correction on the angle-of-attack α and a 20% maximum correction on the engines speed N_1. Similar results were obtained for other flights, which seems promising for the use of such techniques for defining trajectory optimization problems. we observe that almost identical results are obtained using both approaches, except for the N_1 variable for which the multi-task predictions seem slightly closer.

Table 3.4 contains the cross-validated dynamic scores $\mathcal{C}_3$ over all flights. Despite that the multi-task NL has a better mean score, its standard deviation is higher than the OLS. The scores seem to indicate that both approaches are statistically equivalent wrt to this criterion.

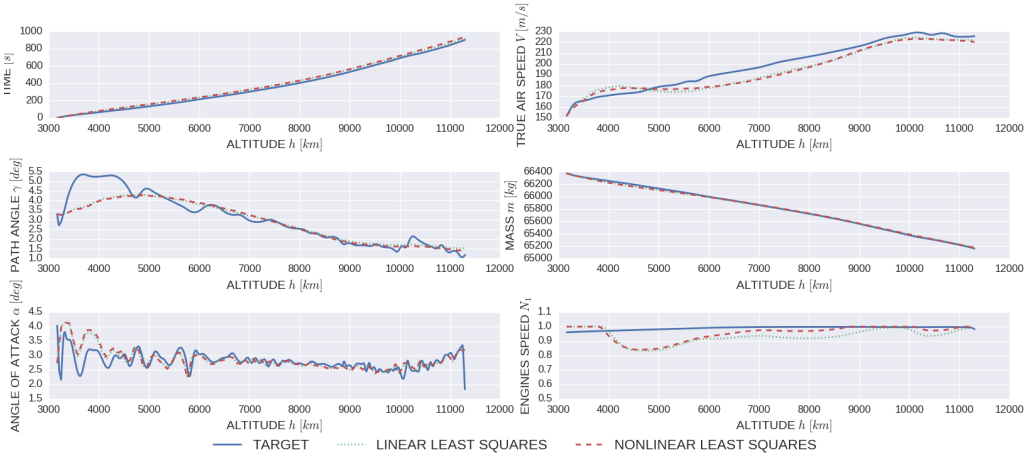


Figure 3.13 – Solution of problem (3.7.5) using dynamics estimated by single-task OLS (dotted green) and multi-task NLS (dashed red), in comparison to a recorded flight (full blue line).

Table 3.4 – Cross-validated $\mathcal{C}_3$ criterion results, with *no-wind-dynamics*

Method	**Mean**	**Std.**
Single-task OLS	1.266	0.120
Multi-task NLS	1.245	0.168

3.9 Conclusion

We presented in this chapter four different maximum likelihood estimators, which can be classified under the broad category of the *Equation-Error Method* for aircraft system

identification. A discussion led to the construction of 2 different criteria to assess and compare the performance of such methods: a static criterion and a dynamic one. The results of the static criterion seem to indicate that all the estimators presented here are able to approximate with good accuracy the dynamics of an aircraft using its historic QAR data, including in a *big data* framework. The results from the dynamic criterion show that these methods are suited to be used *a priori* to infer the dynamics defining an optimal control problem. Although the multi-task formulations proposed seem to be equivalent to a straightforward single-task formulation in terms of accuracy, they are shown to attain equivalent performance while allowing to estimate all unknown functions of the problem (T, L, D and C_{sp}), unlike the former.

Furthermore, two different possibilities for the dynamic system expression were considered: one taking into account the wind and the other one not. Our numerical results demonstrate that a better performance can be obtained for all estimators using the more complex dynamics, at no additional computational cost.

Chapter references

A. C. Aitken. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936.

J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control and dynamics*, 21(2):193–207, 1998.

J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM, 2010.

J. F. Bonnans. *Optimisation Continue*. Dunod, 2006.

J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2013.

J. F. Bonnans, D. Giorgi, V. Grelard, B. Heymann, S. Maindrault, P. Martinon, O. Tissot, and J. Liu. Bocop – A collection of examples. Technical report, INRIA, 2017. URL http://www.bocop.org.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

A. Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 25(1847):536–538, 1847.

G. A. Einicke, G. Falco, and J. T. Malos. EM algorithm state matrix estimation for navigation. *IEEE Signal Processing Letters*, 17(5):437–440, 2010.

G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Developmental Psychology Series. Academic Press, 1977. ISBN 9780122897504. URL https://books.google.fr/books?id=Iu9QAAAAMAAJ.

H. Greenberg. A survey of methods for determining stability parameters of an airplane from dynamic flight measurements. Technical report, NACA Ames Aeronautical Laboratory, 1951.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

D. G. Hull. *Fundamentals of Airplane Flight Mechanics*. Springer, 2007.

R. V. Jategaonkar. *Flight Vehicle System Identification: A Time Domain Methdology*. AIAA, 2006.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/.

V. Klein and E. A. Morelli. *Aircraft System Identification*. AIAA, 2006.

D. Kraft. *A Software Package for Sequential Quadratic Programming*. DFVLR, 1988. URL https://books.google.fr/books?id=4rKaGwAACAAJ.

K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

R. E. Maine and K. W. Iliff. *Identification of Dynamic Systems: Theory and Formulation*. NASA, STIB, 1985.

R. E. Maine and K. W. Iliff. *Application of Parameter Estimation to Aircraft Stability and Control: The Output-error Approach*. NASA, STIB, 1986.

D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

E. A. Morelli. Real-time parameter estimation in the frequency domain. *Journal of Guidance, Control and Dynamics*, 23(5):812–818, 2000.

E. A. Morelli. Practical aspects of the equation-error method for aircraft parameter estimation. In *AIAA Atmospheric Flight Mechanics Conference*. AIAA, 2006.

E. A. Morelli. Dynamic modeling from flight data with unknown time skews. *Journal of Guidance, Control and Dynamics*, pages 1–9, 2017.

N. K. Peyada and A. K. Ghosh. Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method. *The Aeronautical Journal*, 113(1142):243–252, 2009.

N. K. Peyada, A. Sen, and A. K. Ghosh. Aerodynamic characterization of hansa-3 aircraft using equation error, maximum likelihood and filter error methods. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2008.

E. Roux. *Pour une approche analytique de la dynamique du vol*. PhD thesis, SUPAERO, 2005. URL http://elodieroux.com/ReportFiles/TheseElodie.pdf.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

Inria Saclay Team Commands. Bocop: an open source toolbox for optimal control, 2017.

E. Walter and L. Pronzato. *Identification of Parametric Models from Experimental Data*. Springer, 1997.

Chapter 4

STRUCTURED MULTI-TASK FEATURE SELECTION

Abstract: *In this chapter we conserve the multi-task learning viewpoint from chapter 3. We present in this setting a class of estimators that we call block sparse Lasso, which conserves the structure between some groups of variables, while promoting sparsity within them. An implementation leading to consistent feature selection is suggested, allowing to obtain accurate models, which are suitable for trajectory optimization. An additional regularizer is also proposed to help in recovering hidden representations of the initial dynamical system. We illustrate our method with numerical results based on real flight data from 25 medium haul aircraft, totaling 8 millions observations.*

Contents

4.1 Introduction 68
4.2 Feature selection 68
4.2.1 A feature selection categorization 68
4.2.2 The Lasso 69
4.2.3 Inconsistency in high correlation setting 70
4.2.4 A resampling adaptation: the Bolasso 71
4.2.5 An efficient implementation: LARS 72
4.3 Application to aircraft dynamics identification in a single-task setting 73
4.4 Block-sparse estimators 76
4.4.1 Structured feature selection 76
4.4.2 Linear multi-task regression framework 77
4.4.3 Hidden functions parametric models 78
4.4.4 Block-sparse Lasso 79

4.4.5 Bootstrap implementation 81
4.5 Identifiability enhancement 81
4.5.1 The issue with the predicted hidden functions 81
4.5.2 Additional L2 Regularization 82
4.5.3 Block coordinate descent 83
4.6 Numerical results 83
4.6.1 Experiments design 83
4.6.2 Results 84
4.7 Conclusion 84

4.1 Introduction

In the last chapter, we presented the scope of the aircraft identification problem to be solved and proposed standard maximum likelihood based variations of the Equation-Error Method. The connection between these well-established techniques and the recently developped statistical framework of multi-task learning was made explicit. The numerical experiments presented seemed to indicate that the use of the least-squares criterion was equivalent to more complex maximum likelihood formulations.

In this chapter, we try to improve the previous results by casting a new linear formulation of the identification problem (section 4.4.2). As implied in the last chapter, we will introduce new polynomial models, customized to each different aircraft thanks to feature selection techniques (section 4.2). After applying these techniques to aerodynamic models in section 4.3, a new structured strategy will be proposed in sections 4.4.4 and 4.4.5. The sparsity patterns obtained will not only bring our models to comply with prior knowledge from the flight mechanics field, it will also render our results more interpretable and faster to evaluate. In addition, we hope that better overall accuracy and identifiability will be achieved by adding well chosen regularizations to a multi-task least-squares problem similar to the previously introduced (section 4.5).

4.2 Feature selection

4.2.1 A feature selection categorization

In Guyon and Elisseeff [2003], a review of the existing types of feature selection techniques is presented. They are mainly classified under three categories: *filter* methods, *wrapper* methods or *embedded* methods.

Filter methods are based on the idea of ranking all the features with regard to a chosen criteria, such as the correlations with the outputs for example. This allows not only to select the K most informative feature relatively to the chosen criterion, it is also useful

to identify variables which should not be considered at all. In this sense, filter methods are an additional preprocessing step, which is particularly interesting from a computation viewpoint in the case of a large number of variables. Although some of these methods were proven to be optimal in precise contexts, their main drawback seems to be that their selection is made independently of the choice of the prediction algorithm.

Wrapper methods, however, look for a subset of variables which maximize the prediction error for a chosen estimator. Instead of looking at all possible combinations of features, which would be intractable, *forward* (resp. *backward*) *wrapper* methods, such as *forward stepwise regression* (FSR), start with an empty (resp. full) set of selected variables and than adds (resp. drops) new variables iteratively. In these methods, a desired number of variables to select has to be set a priori.

Embedded methods, in the other hand, solve the regression problem *and* select the most informative features simultaneously. Well-known examples of these methods are the *classification and regression trees* [Breiman et al., 1984] and the *Lasso*. We describe the latter as well as some of his alternate versions in the following sections.

4.2.2 The Lasso

Given a linear regression model

$$Y = X \cdot \boldsymbol{\theta} + \varepsilon, \tag{4.2.1}$$

where $Y \in \mathbb{R}$, $X \in \mathbb{R}^p$ and $\varepsilon \in \mathbb{R}$ are random variables and $\boldsymbol{\theta} \in \mathbb{R}^p$ is an unknown parameters vector, a very popular *embedded* feature selection method is the *Lasso* [Tibshirani, 1994]. Given $t > 0$ and some sample $\{(x_i, y_i)\}_{i=1}^N$ of N observations drawn from the joint distribution of (X, Y), the Lasso is a regularization approach (see section 2.2.1), which consists in solving the least-squares optimization problem under an inequality constraint on the L^1-norm of the parameters vector:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} (y_i - x_i \cdot \boldsymbol{\theta})^2 \qquad \text{s.t.} \quad \|\boldsymbol{\theta}\|_1 \leq t, \tag{4.2.2}$$

It is however better known under its Lagrangian formulation:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} (y_i - x_i \cdot \boldsymbol{\theta})^2 + \lambda \|\boldsymbol{\theta}\|_1, \tag{4.2.3}$$

or in matrix form as

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \tag{4.2.4}$$

where $\lambda > 0$, $\boldsymbol{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$ and $\boldsymbol{X} = (x_1, \ldots, x_N) \in \mathcal{M}_{N,p}(\mathbb{R})$.

As the *Ridge regression* [Horel, 1962], which uses the squared L^2-norm penalty instead

(known as the Tikhonov regularizer [Tikhonov, 1943]),

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2, \tag{4.2.5}$$

the Lasso approach is aimed at improving the *Ordinary Least-Squares* (OLS) estimator by *shrinking* the parameters of the linear model, i.e. downsizing the importance of uninformative features. However, unlike the Ridge penalty, the L^1-norm of the Lasso induces sparse parameter vectors, which is why it is very popular now-a-days for feature selection. Some intuition on why the L^1 constraint brings some parameters to be exactly 0 can be found when looking at the shape of such a constraint: a cross-polytope. In a 2-dimensional Euclidean space, this would be a rotated square, as illustrated in figure 4.1. Under the standard assumption that $\boldsymbol{X}^\top\boldsymbol{X}$ is definite-positive, the least-squares criteria of some candidate $\boldsymbol{\theta}$ corresponds, up to a constant C, to an ellipsoid centered at the OLS solution $\hat{\boldsymbol{\theta}}_{LS} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$:

$$\|\boldsymbol{y} - \boldsymbol{X}\cdot\boldsymbol{\theta}\|_2^2 = (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{LS})^\top\boldsymbol{X}^\top\boldsymbol{X}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{LS}) + C. \tag{4.2.6}$$

The Lasso solution being the first $\boldsymbol{\theta}$ leading the ellipsoid to touch the constraint, it is likely that such contact point will be at one of the corners, which correspond to one of the parameters being equal to zero.

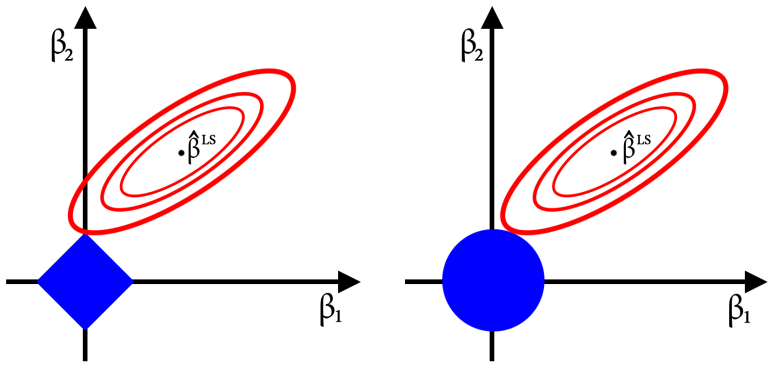


Figure 4.1 – The left frame shows the shape of the L^1 constraint of the Lasso in blue, the contours of the least-squares criterion being represented by the ellipsoids. The right frame represents the same, but with the squared L^2 constraint from the Ridge regression. The OLS solutions are denoted by $\hat{\beta}^{LS}$ here.

4.2.3 Inconsistency in high correlation setting

If the data was really generated by a subset of the features under analysis, the Lasso was proven to select the correct variables only under harsh conditions [van de Geer, 2010, Zhao and Yu, 2006]. One of the main drawbacks of this approach seems that it is known to deliver inconsistent selections when some of the variables are highly correlated. Indeed,

in this case, the variable selection is very sensitive to the training data used and makes a random choice among a couple of correlated features. One of the first attempts to solve this kind of instability was the *elastic net* [Zou and Hastie, 2005], which interpolates between the Lasso and Ridge regression:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2 + \lambda_1\|\boldsymbol{\theta}\|_1 + \lambda_2\|\boldsymbol{\theta}\|_2^2, \tag{4.2.7}$$

with $\lambda_1, \lambda_2 > 0$. The addition of the strongly convex squared L^2 penalty allows to stabilize the selection by improving the problem conditioning. In this case, if two variables are strongly correlated, the elastic net method will pick both.

Another extension of the Lasso to the high correlations setting is the *adaptive Lasso* [Zou, 2006], which adds data-dependent weights $\{\hat{w}_i\}_{i=1}^p$ inside the L^1 regularizer:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2 + \lambda \sum_{i=1}^{p} \hat{w}_i|\boldsymbol{\theta}_i|. \tag{4.2.8}$$

Suitable weights are, for example, $\hat{w}_i := 1/\hat{\boldsymbol{\theta}}_i^{LS}$, where $\hat{\boldsymbol{\theta}}_i^{LS}$ is the OLS estimate of the i^{th} parameter. Although it was shown that the adaptive Lasso can consistently select the correct features with correlated variables, it was proven that it produced less accurate predictions than the Lasso [van de Geer, 2010].

In the next section we describe another adaptation of the Lasso based on resampling, capable of selecting the correct features in a high correlations context.

4.2.4 A resampling adaptation: the Bolasso

Assume that the feature selection is being carried among a set of variables which include those having generated the data from model (4.2.1). It was shown in Bach [2008] that, even with correlated features, the Lasso *support* (set of selected variables) always includes the correct features under mild assumptions. Hence, he suggests to perform the Lasso repeatedly over several *bootstrap replications* of the initial data set (i.e. samples of size N drawn with replacement using a uniform distribution). The selected variables J are then given by the intersection of the supports over all Lasso executions. This procedure is summarized in the following algorithm, where x_J denotes the restriction of a feature vector $x \in \mathbb{R}^p$ to the indexes contained in J:

Let $J^* = \{j, \boldsymbol{\theta}_j \neq 0\}$ be the sparsity pattern of $\boldsymbol{\theta}$, having generated the data. For a sufficient number of replications, this method has been proved to select the correct variables with probability tending exponentially fast to one, under the following assumptions:

(A1) The cumulant generating functions $\mathbb{E}\left[\exp(s\|X\|_2^2)\right]$ and $\mathbb{E}\left[\exp(s\|Y\|_2^2)\right]$ are finite for some $s > 0$.

(A2) The joint matrix of second order moments $Q = \mathbb{E}\left[XX^\top\right] \in \mathbb{R}^{p\times p}$ is invertible.

Algorithm 1 Bolasso - [Bach, 2008]

Require: training data $\mathcal{T} = \{(x^i, y^i)\}_{i=1}^N$,
number of bootstrap replicates m,
L^1 penalization parameter λ,
for $k = 1$ **to** m **do**
Generate bootstrap sample $\mathcal{T}^k$,
Compute Lasso estimate $\hat{\boldsymbol{\theta}}^k$ from $\mathcal{T}^k$,
Compute support $J_k = \{j, \hat{\boldsymbol{\theta}}_j^k \neq 0\}$,
end for
Compute $J = \bigcap_{k=1}^m J_k$,
Compute $\hat{\boldsymbol{\theta}}_J$ by applying the OLS to $\mathcal{T}_J = \{(x_J^i, y^i)\}_{i=1}^N$.

(A3) $\mathbb{E}[Y|X] = X \cdot \boldsymbol{\theta}$ and $\mathrm{Var}[Y|X] = \sigma^2$ a.s. for some $\boldsymbol{\theta} \in \mathbb{R}^p$ and $\sigma \in \mathbb{R}_+^*$.

This result is summarized in theorem 4.2.1:

Theorem 4.2.1 (Bolasso consistency - [Bach, 2008]) *Assume **(A1-3)** and* $\lambda = \lambda_0 N^{-\frac{1}{2}}$, $\lambda_0 > 0$. *Then the probability that the Bolasso does not exactly select the correct model, i.e., for all* $m > 0$, $\mathbb{P}[J \neq J^*]$ *has the following upper bound:*

$$\mathbb{P}[J \neq J^*] \leq m A_1 e^{-A_2 N} + A_3 \frac{\log N}{N^{1/2}} + A_4 \frac{\log m}{m}, \tag{4.2.9}$$

where $A_1, A_2, A_3, A_4 > 0$.

In practice, Bach [2008] suggests that a softer intersection is likely to work better, such as keeping the variables presenting a selection frequency above 90%.

4.2.5 An efficient implementation: LARS

The Bolasso being a method based on consecutive executions of the Lasso, it is important to have an efficient algorithm for solving it. However, because of the L^1-norm, criterion (4.2.3) is not differentiable, making classical first order optimization methods, such as gradient descent, inapplicable. Furthermore, unlike the OLS and Ridge regression, the Lasso does not admit a closed-form solution, unless the features are orthonormal $\boldsymbol{X}^\top \boldsymbol{X} = I_p$. Possible iterative ways for solving it are subgradient techniques [Shor, 1985], coordinate descent [see e.g. Wu and Lange, 2008] or proximal gradient methods [Beck and Teboulle, 2009].

We chose to use the *Least Angle Regression* algorithm proposed in Efron et al. [2004], also known as LARS. It is a different feature selection method, initially unrelated to the Lasso, which can be seen as an adaptation of the forward stepwise regression (FSR) [see e.g. Hastie et al., 2009, chapter 3.3], since both of them start with an empty *support* (set of chosen variables, also called *active set*) and select features sequentially, updating their corresponding parameters. The main difference between LARS and FSR is that the latter

updates the parameters at each iteration through full ordinary least-squares estimations, while LARS is less *greedy* and proceeds as explained hereafter.

Suppose the target vector $\boldsymbol{y} \in \mathbb{R}^N$ has been centered and features $\boldsymbol{X} \in \mathbb{R}^N \times \mathbb{R}^p$ have been standardized (centered and scaled to unit variance). We denote by $\boldsymbol{x}_m \in \mathbb{R}^N$ the m's feature vector, corresponding to the m's column of $\boldsymbol{X}$. At the beginning of the algorithm, all parameters are set to zero $\boldsymbol{\theta}_m = 0$, $m = 1, \dots, p$, and the residuals vector is equal to the target:

$$r(\boldsymbol{\theta}) = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y}. \tag{4.2.10}$$

The variable which is most correlated to the residuals is selected (added to the support)

$$j = \sup\{\text{corr}(\boldsymbol{x}_m, r(\boldsymbol{\theta})) = \boldsymbol{x}_m^\top r(\boldsymbol{\theta}); m = 1, \dots, p\}. \tag{4.2.11}$$

Its corresponding coefficient, $\boldsymbol{\theta}_j$, is then updated, which modifies the residuals. The update is performed as follows: $\boldsymbol{\theta}_j$ is shifted in the direction which lowers the correlation between $r(\boldsymbol{\theta})$ and $\boldsymbol{x}_j$ and the shift length is so as to make a new feature $\boldsymbol{x}_i$ as correlated to the residuals as $\boldsymbol{x}_j$:

$$\boldsymbol{x}_j^\top r(\boldsymbol{\theta}) = \boldsymbol{x}_i^\top r(\boldsymbol{\theta}). \tag{4.2.12}$$

This new variable is then added to the support as well. The next step is to shift both variables' parameters in a direction decreasing the residuals correlation with them, while keeping them tied together, until a new variable catches up and is added to the support. This procedure is repeated $k \leq p$ times in order to select k features, and if it keeps going until $k = p$, the OLS solution is obtained.

Based on the result from Tibshirani [1994] stating that the Lasso *regularization path*[1] is piecewise linear, Efron et al. [2004] proved that, with a simple modification, the LARS leads to the same results as the Lasso. This makes the LARS a really efficient way to solve the Lasso problem compared to the other possibilities, since it only needs p iterations to compute the whole regularization path, which is equivalent to the computational complexity of the OLS.

4.3 Application to aircraft dynamics identification in a single-task setting

In section 3.4.2 of the previous chapter, we introduced polynomial models of the aerodynamic coefficients which lead to the following lift and drag structures:

$$\begin{cases} D = \frac{1}{2}\rho V^2 X^a \cdot \boldsymbol{\theta}_D, \\ L = \frac{1}{2}\rho V^2 X^a \cdot \boldsymbol{\theta}_L, \end{cases} \tag{4.3.1}$$

1. Parameters values for all L^1 penalty weights in a given range.

where

$$X^a = (X_1^a, \dots, X_r^a) = \left(\alpha^k M^{j-k} : j = 0, \dots, d \quad ; \quad k = 0, \dots, j\right), \tag{4.3.2}$$

and $\cdot$ denotes the inner product. Such models seeming artificial and excessively complex, we motivated in section 3.4.3 the use of feature selection techniques aimed to keep only a small subset of the monomials in X^a. However, the *interaction features*[2] in X^a are highly correlated in practice, as commonly experienced in polynomial regression. In this chapter we present numerical results obtained by applying the Bolasso in a single-task manner to the baseline linear models presented in section 3.6.1. The *feature selection* for the models of L and D was carried out previous to the estimations presented in section 3.8.

The data set used for this simulations contained 10 471 flights of 25 different medium haul aircraft of the same type. The maximum degree of the monomials selected was $d = 3$. Prior to the application of algorithm 1, the data set was split into a *model selection set* - 33% of it - and a training set. The L^1 penalty parameter λ was selected a single time for all the 128 bootstrap replications of the Bolasso using the *model selection set* only. For this, 100 values for λ were compared using the *mean square error* in a 50-fold *cross-validation* scheme.

In practice, the Lasso estimations needed for the feature selection part were carried out using the LassoLars and LassoLarsCV functions of Python's scikit-learn.linear_model library [Pedregosa et al., 2011], which implement the modified LARS.

Figure 4.2 shows the cross-validated mean square error path of both $L1$-penalty parameters λ used for the aerodynamic features selection described. The values chosen were at the minimum of these curves, i.e. $1.2\ 10^{-1}$ for SC_x and $6.2\ 10^{-5}$ for SC_z.

The results of the feature selection using the *no-wind-dynamics* (introduced in section 3.3) are presented in figure 4.3. For both matrices, each row corresponds to a different aircraft and each column to a possible feature to be selected. The color of the cells encodes the selection frequency of given feature for given aircraft among the 128 executions of the Lasso. Columns which are almost entirely black (respectively white) should indicate that the corresponding feature has a high (resp. low) probability of being relevant for the aerodynamic model of this type of aircraft. Taking this into account, we see that the lift coefficient features seem easier to select than the drag coefficient ones. Given this results, the following features were kept: $1, M, \alpha, M\alpha^2, M^3$ for the drag coefficient and $1, M, \alpha, \alpha^2, M^2\alpha, M^3, \alpha^3$ for the lift.

Figure 4.4 shows the feature selection results obtained using the *wind dynamics*. The matrices obtained in this case present more homogeneous columns than in the previous one, specially for the drag coefficient (the most influenced by the wind). These seem to indicate that the selection has been improved by the addition of a wind term to the

2. This is how the monomials are originally called in the polynomial regression literature.

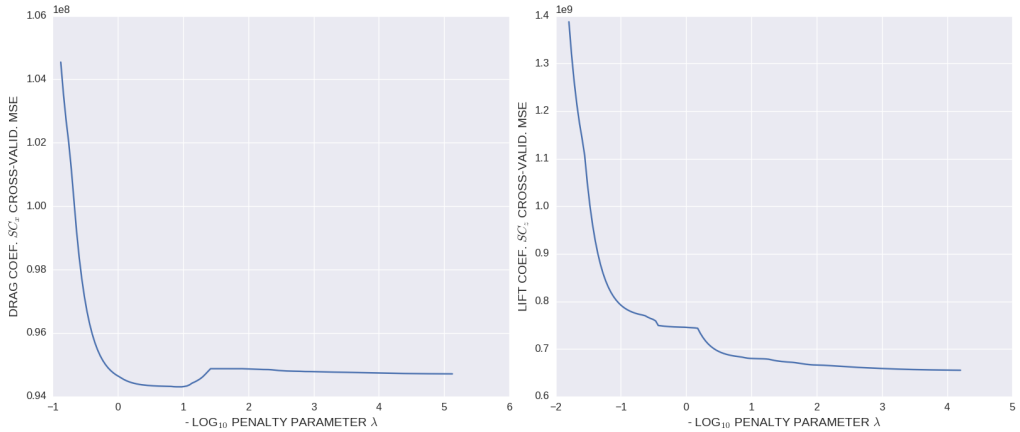
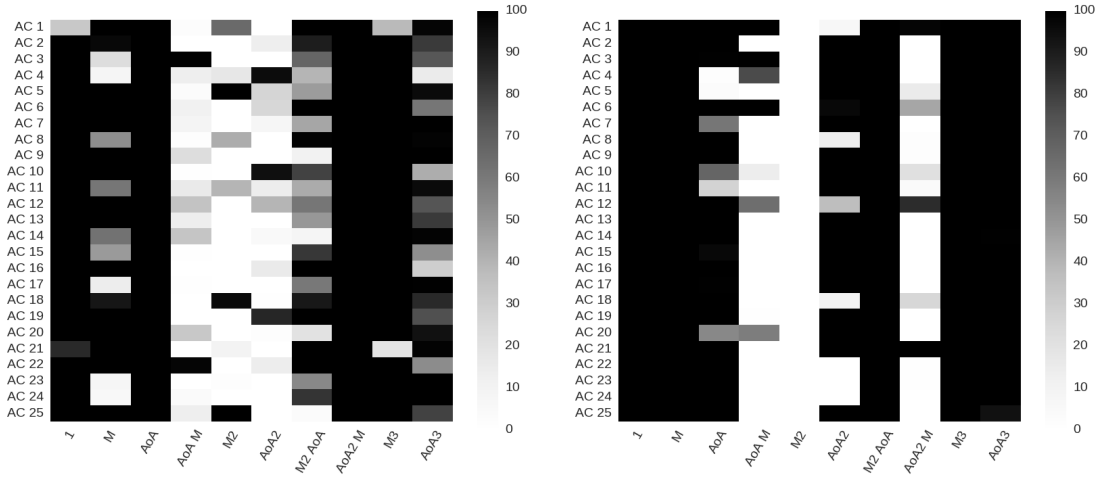


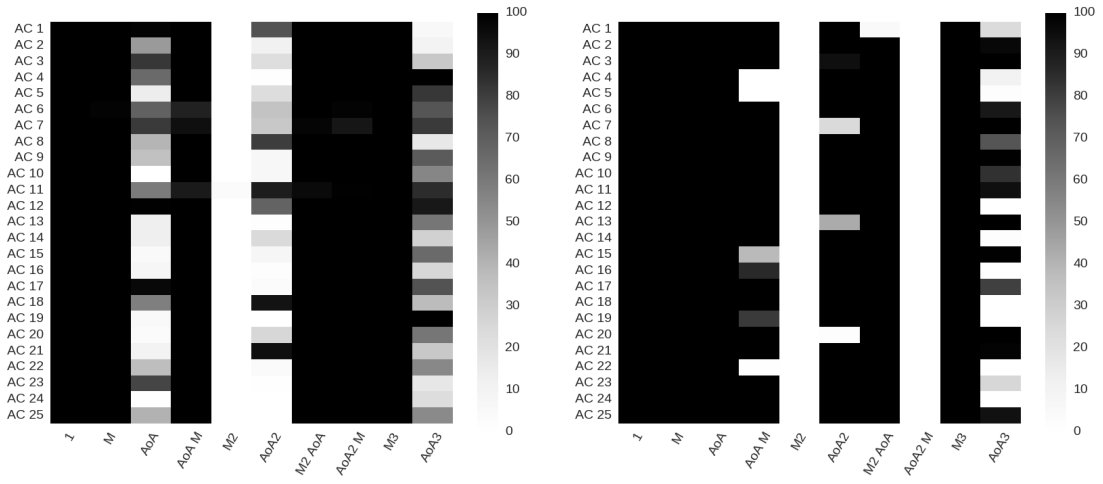
Figure 4.2 – Single-task Bolasso penalty parameter selection via cross-validation (no-wind-dynamics)



(a) SC_x Bolasso results (b) SC_z Bolasso results

Figure 4.3 – Single-task aerodynamic features selection results using no-wind-dynamics.

dynamics model. This is quite surprising since one could imagine that the wind data could add more noise to the target signal. The aerodynamic features kept in this case were $1, M, \alpha M, \alpha M^2, \alpha^2 M, M^3$ for the drag coefficient and $1, M, \alpha, \alpha M, \alpha^2, \alpha M^2, M^3$ for the lift coefficient.



(a) SC_x Bolasso results

(b) SC_z Bolasso results

Figure 4.4 – Single-task aerodynamic features selection results with wind-dynamics.

The results confirm that this approach is quite robust and can be applied in this context. One possible draw back is that the procedure was carried in a single-task setting, D and L features being selected separately. In the next section we suggest a way to incorporate the dynamics structure into the feature selection procedure.

4.4 Block-sparse estimators

4.4.1 Structured feature selection

Despite the great usefulness of the Lasso for inducing sparsity in statistical models, no structural information based on prior knowledge about the problem is encoded in its L^1 penalty. The *group Lasso* [Yuan and Lin, 2005] was probably one of the first of its extensions to cope with this limitation. It selects entire groups of variables by penalizing the sum (i.e. L^1-norm) of the L^2-norms of subsets forming a partition of the features. Many other variations of the Lasso going in this directions have been proposed since then. For example, the *latent group Lasso* [Obozinski et al., 2011] also selects unions of groups of variables in cases where the features subsets overlap. Similarly, variations of the *multi-task Lasso* [Argyriou et al., 2008, Obozinski et al., 2006] make use of a mixed $L^{2,1}$-norm to find common representations for all tasks of a regression problem with multiple outputs. Furthermore, additionally to the selection of groups of variables, some applications also require sparsity within the groups. This can be achieved using the *sparse-group Lasso*,

proposed by Friedman et al. [2010].

In the following sections we present a structured multi-task Lasso adaptation tailored to our identification problem. It relates to the *sparse-group Lasso* in the sense that it promotes sparsity within predefined groups of variables, but unlike the former, the group sparsity pattern is known a priori and hard-coded. It also differs from standard *multi-task Lasso*, as we do not assume that all tasks share the same features.

4.4.2 Linear multi-task regression framework

In order to cast our identification problem into a regression problem, we first rewrite equations (3.3.4)-(3.3.6) as follows, in order to isolate the functions to be learned in the right-hand side of the system:

$$\begin{cases} m\dot{V} + mg\sin\gamma = T(\boldsymbol{x},\boldsymbol{u})\cos\alpha - D(\boldsymbol{x},\boldsymbol{u}), & (4.4.1)\\ mV\dot{\gamma} + mg\cos\gamma = T(\boldsymbol{x},\boldsymbol{u})\sin\alpha + L(\boldsymbol{x},\boldsymbol{u}), & (4.4.2)\\ 0 = T(\boldsymbol{x},\boldsymbol{u}) + \dot{m}I_{sp}(\boldsymbol{x},\boldsymbol{u}). & (4.4.3) \end{cases}$$

The quantity I_{sp} introduced in (4.4.3) corresponds to the inverse of C_{sp} and is commonly called *specific impulse* in the aerospace literature:

$$I_{sp} = \frac{1}{C_{sp}}. \tag{4.4.4}$$

Assuming a linear model for I_{sp}, as well as for T, D and L, the introduction of this new function allows us to derive a set of three linear regression models, at the cost of having a target constantly equal to 0 in one of them:

$$\begin{cases} Y_1 = X_{T1}\cdot\boldsymbol{\theta}_T - X_D\cdot\boldsymbol{\theta}_D + \varepsilon_1, & (4.4.5)\\ Y_2 = X_{T2}\cdot\boldsymbol{\theta}_T + X_L\cdot\boldsymbol{\theta}_L + \varepsilon_2, & (4.4.6)\\ 0 = X_T\cdot\boldsymbol{\theta}_T + X_{Ispm}\cdot\boldsymbol{\theta}_{Isp} + \varepsilon_3, & (4.4.7) \end{cases}$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3$, are random errors of mean 0 and

$$\begin{gathered} Y_1 = m\dot{V} + mg\sin\gamma, \quad Y_2 = mV\dot{\gamma} + mg\cos\gamma,\\ X_{T1} = X_T\cos\alpha,\\ X_{T2} = X_T\sin\alpha, \qquad X_{Ispm} = \dot{m}X_{Isp}. \end{gathered} \tag{4.4.8}$$

As motivated before, we naturally chose to solve the three problems jointly in a multi-task regression framework:

$$Y = X\boldsymbol{\theta} + \varepsilon, \tag{4.4.9}$$

where $Y = (Y_1, Y_2, 0) \in \mathbb{R}^3$, $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3) \in \mathbb{R}^3$, $\boldsymbol{\theta} = (\boldsymbol{\theta}_T, \boldsymbol{\theta}_D, \boldsymbol{\theta}_L, \boldsymbol{\theta}_{Isp}) \in \mathbb{R}^p$ and X is a linear operator represented by the following sparse random matrix:

$$X = \begin{pmatrix} X_{T1}^\top & -X_D^\top & 0 & 0 \\ X_{T2}^\top & 0 & X_L^\top & 0 \\ X_T^\top & 0 & 0 & X_{Ispm}^\top \end{pmatrix} \in \mathbb{R}^{3\times p}. \tag{4.4.10}$$

By doing so, we enforced all three components of the final estimator to share the same $\hat{T}$ function, which complies with the dynamics structure (3.3.3)-(3.3.6). We also hoped to leverage the resulting coupling between tasks to increase the predictive accuracy, as observed in many other multi-task learning applications [Caruana, 1997, Evgeniou et al., 2005]. The next section gives further details concerning the linear models of T, D, L and I_{sp}.

4.4.3 Hidden functions parametric models

In section 4.3 we performed feature selection with polynomial models of the aerodynamic forces. The estimations conducted in chapter 3 used these data-dependent designs jointly with designs of T and C_{sp} extracted from the flight mechanics literature. We propose in this chapter to extend what was done for the aerodynamic forces to all four unknown functions T, D, L and I_{sp}.

Looking at the old models proposed in section 3.4.2, we realized that each of these four functions depend on different sets of 3 features, which can be obtained from the initial inputs of our problem $(\boldsymbol{x}, \boldsymbol{u})$ through standard flight mechanics formulas. These transformations can be seen as four different nonlinear mappings:

$$\varphi_\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^3, \tag{4.4.11}$$

where $\ell \in \{T, D, L, I_{sp}\}$. More precisely, we have in our case:

$$\begin{aligned} \varphi_T(\boldsymbol{x}, \boldsymbol{u}) &= (N1, \rho(h), M(h, V)), \\ \varphi_D(\boldsymbol{x}, \boldsymbol{u}) &= (q(h, V), \alpha, M(h, V)), \\ \varphi_L(\boldsymbol{x}, \boldsymbol{u}) &= (q(h, V), \alpha, M(h, V)), \\ \varphi_{I_{sp}}(\boldsymbol{x}, \boldsymbol{u}) &= (SAT(h), h, M(h, V)). \end{aligned} \tag{4.4.12}$$

Furthermore, most of T, D, L and I_{sp} models found in the literature were polynomials on the features listed in (4.4.12). Indeed, models from section 3.4.2 consist more or less in the product between one of these variables and a linear combination of some monomials of the other two. This led us to define the additional feature mapping, inspired by vector

X^a defined in (4.3.2):

$$\begin{array}{rcl} \mathbb{R}^3 & \to & \mathbb{R}^r \\ \Phi_d : (z_1, z_2, z_3) & \mapsto & \left(z_1 z_2^k z_3^{j-k}; \begin{array}{l} j = 0, \dots, d \\ k = 0, \dots, j \end{array} \right), \end{array} \tag{4.4.13}$$

with $d \in \mathbb{N}^*$ and $r = \begin{pmatrix} d+2 \\ 2 \end{pmatrix}$. By picking $d > 1$, this last transformation allows to project our initial features into a higher dimensional space, potentially increasing the explanatory power of our model. Hence, for some $d_T, d_D, d_L, d_{Isp} > 1$, we assume the following linear structures:

$$\begin{aligned} T(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_T) &= X_T \cdot \boldsymbol{\theta}_T, \\ D(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_D) &= X_D \cdot \boldsymbol{\theta}_D, \\ L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_L) &= X_L \cdot \boldsymbol{\theta}_L, \\ I_{sp}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_{Isp}) &= X_{Isp} \cdot \boldsymbol{\theta}_{Isp}, \end{aligned} \tag{4.4.14}$$

where

$$X_\ell = \Phi_{d_\ell}(\varphi_\ell(\boldsymbol{x}, \boldsymbol{u})) \in \mathbb{R}^{r_\ell}, \tag{4.4.15}$$

for $\ell \in \{T, D, L, I_{sp}\}$, and $\boldsymbol{\theta}_T, \boldsymbol{\theta}_D, \boldsymbol{\theta}_L, \boldsymbol{\theta}_{Isp}$ denote vectors of parameters.

Remark Despite the linearity of models (4.4.14), we can notice by looking at the feature mappings (4.4.15) and at equations (4.4.1)-(4.4.3) that our final statistical model of the aircraft dynamics has a complex network structure, connecting the states derivatives $\dot{\boldsymbol{x}}$ to the pairs of controls and states $(\boldsymbol{u}, \boldsymbol{x})$:

FIGURE A INSERER ICI

Notice that most of the parameters involved in this model are fixed a priori using domain specific knowledge, while the identification problem consists here in estimating those of the last hidden layer. Figure REF illustrates why we say hereafter that T, D, L and I_{sp} are *hidden* or *latent*. Indeed, they are nested features of our model, which are unobservable. One of the caveats of this hidden nature is presented in section 4.5.

4.4.4 Block-sparse Lasso

We consider in this section that we have access to N random observations of $\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{x}}$, allowing to build through the feature mappings (4.4.15), (4.4.8) and (4.4.10) a training set $\{(X^i, Y^i)\}_{i=1}^N$, where the X^i's are sampled from the distribution of the random matrix (4.4.10). We assume this sample to be i.i.d and that all these random variables are centered and scaled to 1-standard deviation, except for the third component of the targets Y_3^i, which

is assumed to be sampled from a Dirac distribution in 0. In this context, the empirical risk minimization for the squared loss writes:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|Y^i - X^i\boldsymbol{\theta}\|_2^2, \tag{4.4.16}$$

which is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\theta}_T,\boldsymbol{\theta}_D,\boldsymbol{\theta}_L,\boldsymbol{\theta}_{Isp}} & \|\boldsymbol{Y}_1 - \boldsymbol{X}_{T1}\boldsymbol{\theta}_T + \boldsymbol{X}_D\boldsymbol{\theta}_D\|_2^2 \\ & + \|\boldsymbol{Y}_2 - \boldsymbol{X}_{T2}\boldsymbol{\theta}_T - \boldsymbol{X}_L\boldsymbol{\theta}_L\|_2^2 \\ & + \|\boldsymbol{X}_T\boldsymbol{\theta}_T + \boldsymbol{X}_{Ispm}\boldsymbol{\theta}_{Isp}\|_2^2, \end{aligned} \tag{4.4.17}$$

where $\boldsymbol{X}_\ell$, $\ell \in \{T1, T2, T, D, L, I_{spm}\}$, (resp. $\boldsymbol{Y}_1, \boldsymbol{Y}_2$) denote matrices whose N rows correspond to the observations of X_ℓ (resp. observations of Y_1, Y_2).

Considering that the models from the literature used as inspiration for building our linear structures (4.4.14) were often composed of only a few monomials, feature selection seemed necessary here to try to avoid overfitting with an excessively complex model. Thus, the addition of an L^1-penalization to problem (4.4.16) was considered:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|Y^i - X^i\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1, \qquad \lambda > 0. \tag{4.4.18}$$

Problem (4.4.18) can be seen as a structured multi-task version of the Lasso, where the residuals are in $\mathbb{R}^3$ and the features vector has been replaced by the sparse features matrix (4.4.10). Despite this differences, we can solve (4.4.18) with all the available algorithms for solving the Lasso (including LARS), by noticing that the three terms forming the criterion from (4.4.17) can be written as a single-task least-squares criterion

$$\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2, \tag{4.4.19}$$

where the targets observations have been stacked

$$\boldsymbol{Y} = (Y_1^1, \ldots, Y_1^N, Y_2^1, \ldots Y_2^N, 0, \ldots, 0), \tag{4.4.20}$$

and the features observations have been used to build the block-sparse design matrix:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{T1} & -\boldsymbol{X}_D & 0 & 0 \\ \boldsymbol{X}_{T2} & 0 & \boldsymbol{X}_L & 0 \\ \boldsymbol{X}_T & 0 & 0 & \boldsymbol{X}_{Ispm} \end{pmatrix} \in \mathbb{R}^{3N \times p}. \tag{4.4.21}$$

For this reason, we call the described procedure the *block sparse Lasso* in the remaining of the chapter. Notice that, although $\boldsymbol{X}$ has a structure similar to the random matrix X

from (4.4.10), it is not random and it has $3N$ rows instead of 3.

4.4.5 Bootstrap implementation

As in section 4.3, the features of our polynomial models are highly correlated, which means that the selection performed by the L^1 regularizer is not consistent. Hence, we suggest to stabilize it using bootstrap resampling, as in the Bolasso:

Algorithm 2 Block sparse Bolasso

Require: training data $\mathcal{T} = \{(X^i, Y^i)\}_{i=1}^N$,
number of bootstrap replicates m,
L^1 penalization parameter λ,
for $k = 1$ **to** m **do**
Generate bootstrap sample $\mathcal{T}^k$,
Compute block sparse Lasso estimate $\hat{\boldsymbol{\theta}}^k$ from $\mathcal{T}^k$ (4.4.18),
Compute support $J_k = \{j, \hat{\boldsymbol{\theta}}_j^k \neq 0\}$,
end for
Compute $J = \bigcap_{k=1}^m J_k$,
Compute $\hat{\boldsymbol{\theta}}_J$ from $\mathcal{T}_J = \{(X_J^i, Y^i)\}_{i=1}^N$ using (4.4.16).

In the previous algorithm, X_J^i denotes the matrix formed by the columns of X^i indexed by J.

As the *block sparse Lasso* is equivalent to the Lasso with observations stacked in a certain way, the consistency result from theorem 4.2.1 also applies to algorithm 2. In our setting, assumption **(A1)** is equivalent to requiring finite cumulant generating functions for Y_1, Y_2, X_ℓ, $\ell \in \{T1, T2, T, D, L, I_{spm}\}$. This is the case when $\varepsilon_1, \varepsilon_2, X_\ell$, have compact support, which is verified in our case due to the physical nature of our data. Assumption **(A2)** summarizes here to each non-zero element of X having invertible second order moments, which has to be verified if we want our model to be identifiable. Assumption **(A3)** is of course equivalent to assuming the regression model (4.4.9).

4.5 Identifiability enhancement

4.5.1 The issue with the predicted hidden functions

With the estimated parameters $\hat{\boldsymbol{\theta}}$ obtained by applying algorithm 2, one can use equations (4.4.14) to have access to predictions of the hidden functions $\hat{T}, \hat{D}, \hat{L}$ and $\hat{I}_{sp}$. In practice, one may observe that every execution of the *block sparse Lasso* (4.4.18) leads to $\hat{T}$ and $\hat{I}_{sp}$ systematically equal to 0, all their parameters being rejected by the feature selection procedure. This seems to occur because we are regressing a function constantly equal to 0 in the last task (4.4.7), whose trivial solution is indeed setting all parameters of T and I_{sp} to 0. This seems to be allowed by our model because the group of features of T are highly correlated to the features of D and L. Hence, it is possible to find solutions to problems

(4.4.16) and (4.4.18) where targets Y_1 and Y_2 are completely explained by D and L, their parameters having compensated the absence of T. This is coherent with the fact that, even when the L^1 penalty is omitted and plain least-squares is applied (4.4.16), the predicted thrust and specific impulse $\hat{T}, \hat{I}_{sp}$ obtained are positive but really small compared to the known orders of magnitude of these physical quantities. Evidence of this is given in section 5.5.5.

4.5.2 Additional L2 Regularization

In order to avoid the previously described behavior, we introduce an additional regularization to our model. For this, we assume the availability of some prior estimator for any of the hidden functions T, D, L or I_{sp}. A simples prior estimator can be in this case a constant function equal to the known order of magnitude of given quantity. Without loss of generality, we suppose in what follows that $\tilde{I}_{sp}$ is such a prior estimator of I_{sp} and we define the *regularized block sparse Lasso* problem:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|Y^i - X^i\boldsymbol{\theta}\|_2^2 + \lambda_2\|\tilde{I}_{sp}^i - X_{Isp}^i \cdot \boldsymbol{\theta}_{Isp}\|_2^2 + \lambda_1\|\boldsymbol{\theta}\|_1, \qquad \lambda_1, \lambda_2 > 0, \tag{4.5.1}$$

where, for $i = 1, \ldots, N$, we denote $\tilde{I}_{sp}^i = \tilde{I}_{sp}(\boldsymbol{x}^i, \boldsymbol{u}^i)$ the prediction of the prior estimator for some sample observations of states and controls.

Up to a scaling factor λ_2, the additional L^2 regularization term in (4.5.1) can be interpreted as if we had added one more task to system (4.4.5)-(4.4.7):

$$\tilde{I}_{sp} = X_{Isp} \cdot \boldsymbol{\theta}_{Isp} + \varepsilon_4, \tag{4.5.2}$$

where ε_4 is a random error with 0 mean, representing the innovation of our prediction compared to the prior. Hence, algorithm 2 can still be used to stabilize (4.5.1), by replacing X^i and Y^i by

$$\tilde{X}^i = \begin{pmatrix} (X_{T1}^i)^\top & -(X_D^i)^\top & 0 & 0 \\ (X_{T2}^i)^\top & 0 & (X_L^i)^\top & 0 \\ (X_T^i)^\top & 0 & 0 & (X_{Ispm}^i)^\top \\ 0 & 0 & 0 & \lambda_2(X_{Isp}^i)^\top \end{pmatrix}, \qquad \tilde{Y}^i = \begin{pmatrix} Y_1^i \\ Y_2^i \\ 0 \\ \lambda_2\tilde{I}_{sp}^i \end{pmatrix}. \tag{4.5.3}$$

Empirical results presented in section 5.5.5 show that predictions of the hidden functions with the correct orders of magnitude can be obtained with this technique.

4.5.3 Block coordinate descent

4.6 Numerical results

4.6.1 Experiments design

Feature selection We present in this section numerical experiments carried using real flight data, extracted from Quick Access Recorder devices (QAR). The datasets used for the feature selection part are the same as in section 4.3, i.e. 8 261 619 observations from 10 471 different flights made by 25 different medium haul aircrafts of the same type.

In order to assess the quality of the feature selection performed by the *block sparse Bolasso* presented in section 4.4.5, algorithm 1 was run separately on the datasets of each aircraft. The sparsity patterns obtained were then compared. The features used to build the linear models of T, D, L and I_{sp} were obtained using the feature mapping (4.4.15), where $d_T = 4$ and $d_D = d_L = d_{Isp} = 3$. The regularization parameter λ_2 introduced in section 4.5 was set to 200, which is justified in section 5.5.5. The prior model $\tilde{I}_{sp}$ used here to define such regularization was found in Roux [2005]. The L^1 penalty parameter λ_1 of the regularized block sparse Lasso (4.5.1) was set using 30-fold cross-validation based on the squared-loss. This was performed separately for each dataset, on 33%-validation sets, and done a single time, prior to the m bootstrap replications. For all experiments, the number of replications was set to $m = 128$. Solving the multiple Lasso problems was done using the Least Angle Regression algorithm, implemented in Python's SCIKIT-LEARN.LINEAR_MODEL library Pedregosa et al. [2011].

Quality of the state equation estimations The quality of the estimators obtained using the *block sparse Bolasso* algorithm was assessed using a subset of $q = 424$ flights, corresponding to a single aircraft and comprising 334 531 observations. It corresponds to the same dataset used in section 3.8 of the last chapter. Our model was trained on $q - 1$ flights, leaving out one flight (chosen randomly) for testing. The L^1 parameter was selected using cross validation, prior to the execution of algorithm 1, as previously explained for the feature selection assessment. The calibration of λ_2 was done by solving (4.5.1) with varying parameter values. We kept the parameter leading to predicted $\hat{T}, \hat{D}, \hat{L}$ and $\hat{I}_{sp}$ agreeing with the physical context. The quality criteria used to assess the predictions are those described in section 3.7. For the dynamic criterion, we recall the optimal control problem used to define it:

$$\begin{aligned} \min_{(\boldsymbol{u},\boldsymbol{x})\in\mathbb{U}\times\mathbb{X}} & \int_0^{t_{N_i}} \left(\|\boldsymbol{u}(t) - \boldsymbol{u}^i(t)\|_{\boldsymbol{u}}^2 + \|\boldsymbol{x}(t) - \boldsymbol{x}^i(t)\|_{\boldsymbol{x}}^2 \right) dt \\ & \text{s.t.} \quad \dot{\boldsymbol{x}} = g(\boldsymbol{x}, \boldsymbol{u}, \hat{\boldsymbol{\theta}}), \end{aligned} \tag{4.6.1}$$

This problem was solved using BOCOP [Bonnans et al., 2017] here again.

4.6.2 Results

The results of the feature selection performed on the 25 datasets are depicted on figure 4.5. Each row of these matrices corresponds to a different aircraft and each column to a different feature. The cells colors encode the frequency of selection of given feature for given aircraft across all the *block sparse Lasso* executions. It can be observed that most dark columns are quite homogeneous, which indicates that similar features were selected for the majority of aircraft. This seems to validate our approach for this kind of data, as one would expect that airplanes of the same type should have physical models for the thrust, drag, lift and specific impulse with similar structures. As an example, a common sparse model for the thrust force T would be here made of the features: $X_T = N_1(1, \rho M^2, M^4, \rho^2 M^2, \rho^3 M, \rho^4)$.

Figure 4.6 illustrates the effect of the regularization presented in section 4.5 on the test flight predictions obtained with our estimator. Left frames show that it affects mainly $\hat{T}$ and $\hat{I}_{sp}$, which mirror each other because of equation (4.4.7). The drag D seems to vary slightly when the value of λ_2 is changed, while L is unchanged. It is well-known that during climb, the thrust force of commercial aircraft decreases. However, it may be observed that for a small value of $\lambda_2 = 20$, $\hat{T}$ and $\hat{I}_{sp}$ are flat, which seems to indicate that more regularization is needed. A high value of $\lambda_2 = 2000$ brought our predicted thrust to increase quickly at the end of the climb, which does not seem correct either. We seem to have constrained to much our model of $\hat{I}_{sp}$ in this case. Hence, $\lambda_2 = 200$ seems like the best choice here. Right frames of figure 4.6 show that the regularization parameter λ_2 has little effect on the final predictions of the state function estimator $\hat{g}$. These predictions seem to be relatively good approximations of the observed states derivatives for our purposes. This can be seen for instance in figure 4.7, which shows the obtained solution for problem (4.6.1), illustrating that it is possible to resimulate the test flight using our estimator.

4.7 Conclusion

We derived in this chapter a new method for aircraft dynamics identification based on a multi-task learning framework. As it has been designed for trajectory optimization purposes, it delivers interpretable models, which agree with the flight mechanics common knowledge and can be quickly evaluated and differentiated by an optimization solver. Our estimator is able to obtain consistent and structured sparsity, thanks to an original regression formulation and the use of a previously suggested bootstrap strategy: the Bolasso. We also showed that such estimator could be trained using available, well-known and efficient algorithms. The numerical experiments that we carried using real data from 25 different aircraft strongly suggest that our approach performs reliable feature selection. Furthermore, although this was not our main design criterion, satisfactory accuracy was evidenced by our results, at the cost of less than 10 minutes training using a laptop and a data set of 334 531 observations. The main drawback of our approach seems to be the

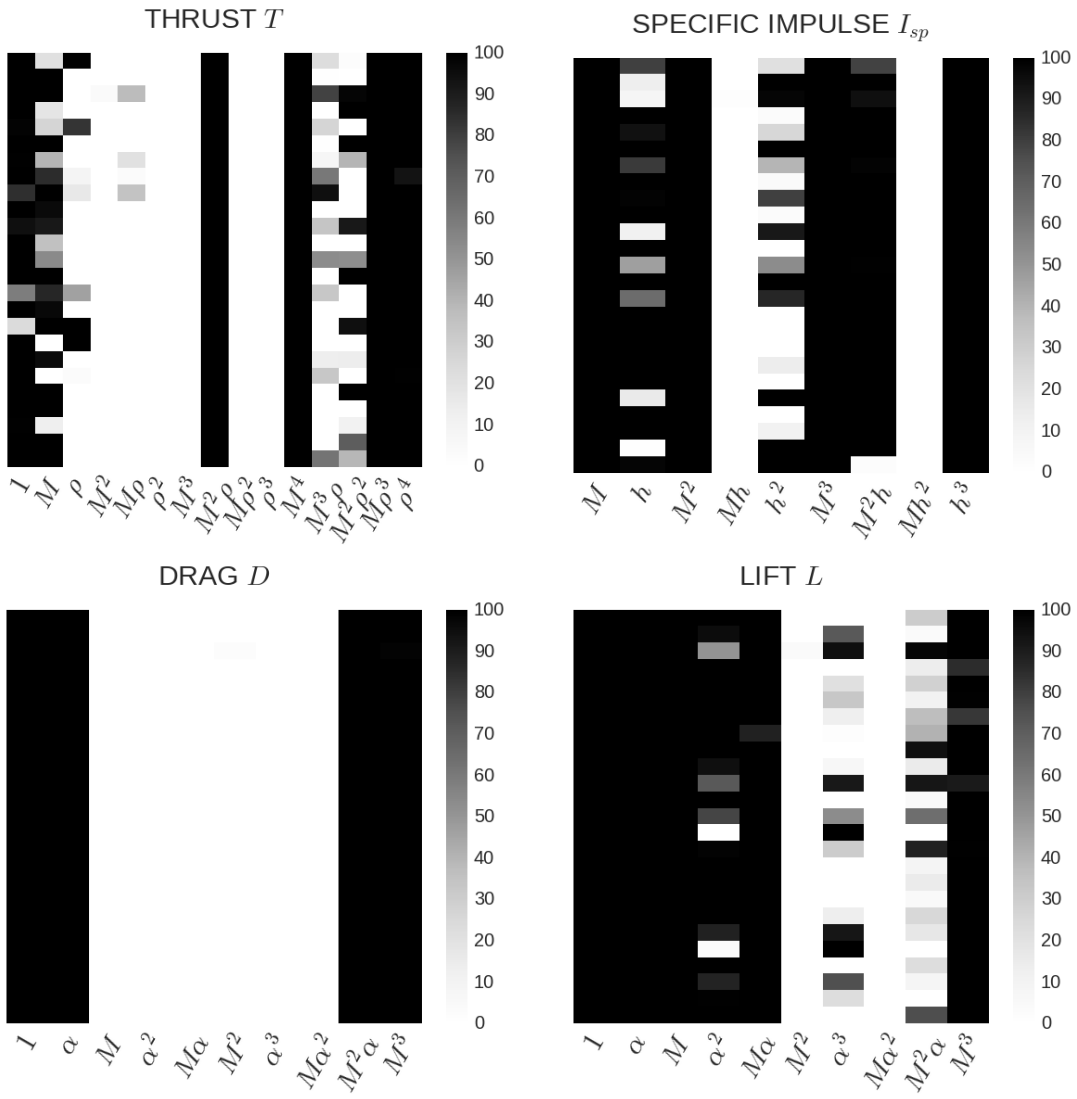


Figure 4.5 – Block sparse Bolasso selections for the thrust, drag, lift and specific impulse models. The columns correspond to possible features for T/N_1, D/q, L/q and I_{sp}/SAT.

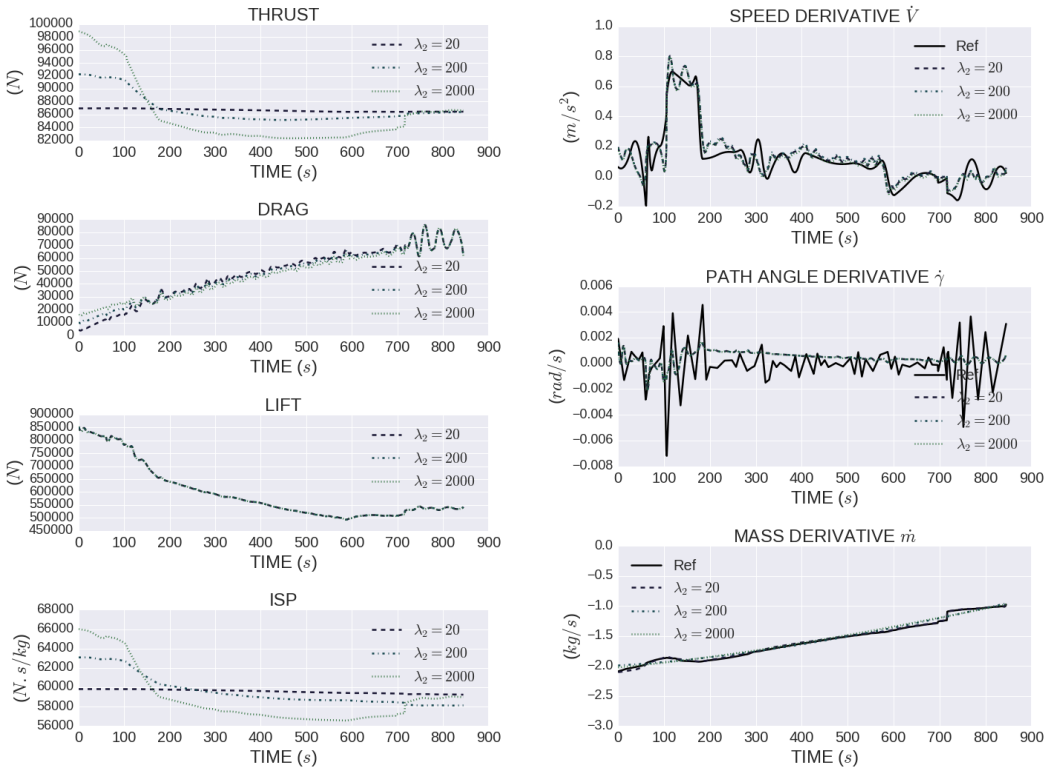


Figure 4.6 – (*Left*) Thrust, Drag, Lift and Specific impulse predicted for the test flight with different regularization parameters λ_2. (*Right*) Respective states derivatives predictions.

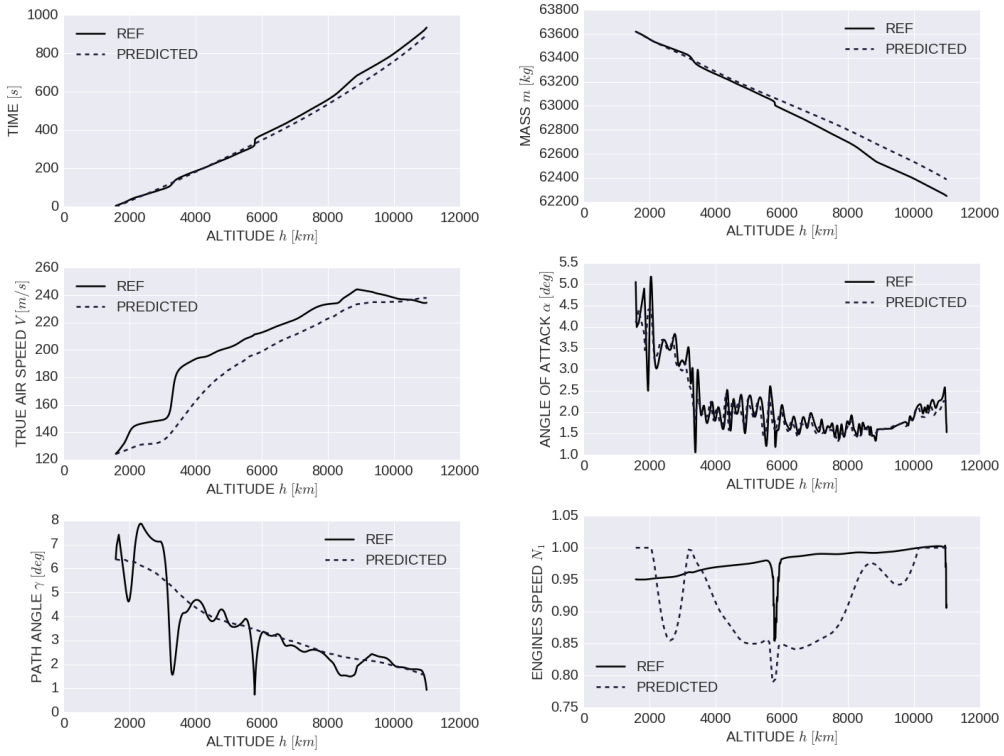


Figure 4.7 – Resimulated trajectory obtained by solving problem (4.6.1) with BOCOP.

need of a prior estimator of one of the hidden functions.

Considering the many issues caused by the strong correlations among features of T, D, L and I_{sp}, the use of different models (e.g. using orthogonal polynomials) could be investigated in future work. Moreover, another interesting question raised by this approach is whether the trajectories optimized using these types of estimators lie close to the observed flights used to compute the model. This could give an indication on the reliability of such trajectories and is investigated in the next part of the manuscript.

Chapter references

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning Journal*, 2008.

F. Bach. Bolasso: model consistent Lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40, 2008.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

J. F. Bonnans, D. Giorgi, V. Grelard, B. Heymann, S. Maindrault, P. Martinon, O. Tissot, and J. Liu. Bocop – A collection of examples. Technical report, INRIA, 2017. URL http://www.bocop.org.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth Advanced Books and Software, 1984.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32:407–499, 2004.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal Machine Learning Research.*, 6:615–637, 2005.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the Group Lasso and a Sparse group Lasso. arXiv:1001.0736, 2010.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

A. E. Horel. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.

G. Obozinski, B. Taskar, and M. I. Jordan. Multi-task feature selection. In *ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.

G. Obozinski, L. Jacob, and J.-P. Vert. Group Lasso with Overlaps: the Latent Group Lasso approach. Technical report, 2011. arXiv:1110.0413.

F. Pedregosa et al. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

E. Roux. *Pour une approche analytique de la dynamique du vol.* PhD thesis, SUPAERO, 2005. URL http://elodieroux.com/ReportFiles/TheseElodie.pdf.

N. Z. Shor. *Minimization methods for non-differentiable functions*. Springer Science & Business Media, 1985.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1994.

A. N. Tikhonov. On the stability of inverse problems. In *Doklady Akademii Nauk SSSR*, volume 39, pages 195–198, 1943.

S. van de Geer. $\ell 1$-regularization in high-dimensional statistical models. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010)*, volume 4, pages 2351–2369, 2010.

T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68:49–67, 2005.

P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

H. Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Part II

SIMULATED TRAJECTORIES ASSESSMENT

Chapter 5

PROBABILISTIC TRAJECTORY ASSESSMENT

Abstract: *In this chapter, we tackle the general problem of quantifying the closeness of a newly observed curve to a given sample of random functions, supposed to have been sampled from the same distribution. We define a probabilistic criterion for such a purpose, based on the marginal density functions of an underlying random process. For practical applications, a class of estimators based on the aggregation of multivariate density estimators is introduced and proved to be consistent. We use a dataset of real aircraft trajectories to illustrate the effectiveness of our estimators, as well as the practical usefulness of the proposed criterion, in a general setting and for our particular application.*

Contents

5.1 Motivation 94
5.2 The trajectory acceptability problem 94
5.3 Marginal Likelihood estimators 95
5.3.1 Mean Marginal Likelihood 95
5.3.2 Empirical Version 97
5.3.3 Consistency of the marginal density estimations 98
5.3.4 Proof of the consistency of the marginal density estimation (theorem 5.3.6) 100
5.4 Choice of the Marginal Density Estimator 105
5.4.1 Parametric or nonparametric ? 105
5.4.2 Kernel estimator 106
5.4.3 Self-consistent kernel estimator 113
5.5 Application to the Assessment of Optimized Aircraft Trajectories 120

5.5.1 Experiments Motivation 120
5.5.2 Experiments Design 121
5.5.3 Alternate Approaches Based on Standard Methods 121
5.5.4 Algorithms Settings 122
5.5.5 Results and Comments 125
5.6 Conclusions 127

5.1 Motivation

Aircraft trajectory optimization is a complex class of problems that has been extensively studied with respect to different aspects, such as fuel consumption [Cots et al., 2018], flight time [Nguyen, 2006] and noise reduction [Khardi et al., 2010], as well as in collision avoidance [Cafieri et al., 2018]. In the last chapters, a particular framework has been considered in which the aircraft dynamics have been estimated from previous flights data. This setting raises the question of whether the optimized trajectory does not deviate too much from the validity region of the dynamics model, which corresponds to the area occupied by the data used to build it. Moreover, for practical purposes and namely acceptance by pilots and Air Traffic Control, optimized trajectories that do not look too unfamiliar are preferable. These questions may both be addressed by quantifying the closeness between the optimization solution and the set of real flights used to identify the model. In this chapter, a general method relying on estimations of the probability density functions of some flight variables conditionally to the aircraft altitude is proposed for this matter. We adopt a generic viewpoint in most of the chapter, postponing the application to aircraft trajectories to section 5.5. More details concerning the our motivations are given in chapter 6.

5.2 The trajectory acceptability problem

Functional Data Analysis (FDA) has received an increasing amount of attention in the last years [e.g. Ramsay and Silverman, 2007], this kind of data being present in many fields of application, such as speech recognition [Ferraty and Vieu, 2003], radar waveforms classification [Dabo-Niang et al., 2007] and aircraft trajectories classification [Gregorutti et al., 2015, Nicol, 2013]. In this paper we are interested in the general problem of quantifying how close some newly observed random curve is to a set of random functions, being mainly motivated by the practical task of assessing optimized aircraft trajectories. To our knowledge, this problem has not been studied in the literature.

We choose to adopt a probabilistic point of view, interpreting the original problem as the estimation of the likelihood of observing the new curve, given the sample of previously

observed functions. This problem is hence related to the estimation of the probability density of a random variable valued on a function space.

Density estimation has been a longstanding problem in statistics and machine learning. Many parametric and nonparametric techniques have been proposed ever since to address it in a finite-dimensional setting. For functional data, density estimation has been studied for example by Dabo-Niang [2004], who proposed an extension of the well-known kernel density estimator. Similarly, Prakasa Rao [2010b] developed a delta-sequence method for functional density estimation.

As the distribution of a functional random variable is hard to grasp and does not present good topological properties because it is defined on sets of a space which is "too large" (see e.g. Bosq [2012], Jacod [2007]), alternatives were proposed for casting this problem into a finite-dimensional setting. For example, Prakasa Rao [2010a] proposed to project the random curves on basis functions, while Hall and Heckman [2002] suggested to study the structure of the distribution of a functional random variable by estimating its modes and density ascent lines. We propose another finite-dimensional approach, by estimating an aggregation of the marginal densities, which reduces to a finite sequence of multivariate density estimation problems.

After introducing our method in section 5.3.1, an empirical version of it is presented for practical applications (section 5.3.2). The obtained statistic is built using marginal density estimators which are shown to be consistent in section 5.3.3. In section 5.4.3, we propose an implementation of our approach using the self-consistent kernel density estimator from Bernacchia and Pigolotti [2011] and extending it to the functional data context. We illustrate the usefulness of our approach for aircraft trajectory optimizationon a dataset of real aircraft trajectories and compare it to more standard approaches (section 5.5).

5.3 Marginal Likelihood estimators

5.3.1 Mean Marginal Likelihood

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $\mathbb{T} = [0; t_f]$ be an interval of $\mathbb{R}$. We denote by E a compact subset of $\mathbb{R}^d$, $d \in \mathbb{N}^*$, endowed with the Borel σ-field $\mathcal{B}$. Let $Z = (Z_t)_{t\in\mathbb{T}}$ be a random variable valued in $\mathcal{C}(\mathbb{T}, E)$, the set of continuous functions from $\mathbb{T}$ to E, and suppose that a training set of m observations of Z is available: $\mathcal{T} = \{\boldsymbol{z}^1, \dots, \boldsymbol{z}^m\} \subset \mathcal{C}(\mathbb{T}, E)$. We denote by μ_t the marginal distribution of Z_t for any $t \in \mathbb{T}$, and we assume that it has a density f_t relative to the Lebesgue measure on $\mathbb{R}^d$. We assume that $(t, z) \in \mathbb{T} \times E \mapsto f_t(z)$ is continuous. Let $\boldsymbol{y} \in \mathcal{C}(\mathbb{T}, E)$ be some arbitrary new curve that we would like to assess.

Given $t \in \mathbb{T}$, we can interpret the quantity $f_t(\boldsymbol{y}(t))$ as the likelihood of observing $Z_t = \boldsymbol{y}(t)$. By summarizing in some way the infinite collection $\{f_t(\boldsymbol{y}(t)) : t \in \mathbb{T}\}$, which we call the *marginal likelihoods* of $\boldsymbol{y}$ hereafter, we hope to build a global and simple

likelihood indicator. The first idea for aggregating these quantities is to average them with respect to time:

$$\frac{1}{t_f}\int_0^{t_f} f_t(\boldsymbol{y}(t))dt. \tag{5.3.1}$$

The main problem with this criterion is that it mixes elements from densities which may have very different shapes. Indeed, density values of likely observations at two times $t_1, t_2 \in \mathbb{T}$ may have completely different orders of magnitude. For this reason, we propose to use some continuous scaling map $\psi : L^1(E, \mathbb{R}_+) \times E \to [0; 1]$ prior to averaging:

$$\mathrm{MML}(Z, \boldsymbol{y}) = \frac{1}{t_f}\int_0^{t_f} \psi\left[f_t, \boldsymbol{y}(t)\right] dt, \tag{5.3.2}$$

and we call the obtained quantity the *mean marginal likelihood* of $\boldsymbol{y}$ given Z. (see figure 5.1).

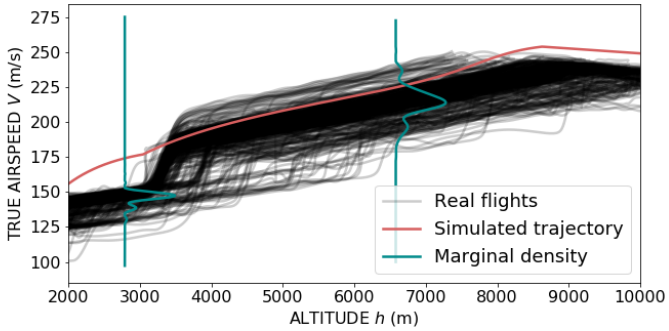


Figure 5.1 – Illustration of the marginal likelihoods.

A natural choice for ψ is simply the function that normalizes $f_t(\boldsymbol{y}(t))$ over E:

$$\psi[f_t, \boldsymbol{y}(t)] = \frac{f_t(\boldsymbol{y}(t))}{\max\limits_{z\in E} f_t(z)}. \tag{5.3.3}$$

However, we can also consider more meaningful scaling maps, such as the *confidence level* at $\boldsymbol{y}(t)$:

Definition 5.3.1 *Let X be a continuous random variable on E of density function $f \in \mathcal{C}(E)$ and let $a \in \mathbb{R}_+$. We call the* confidence level *of f at a the probability that X lies in a region where its density is lower or equal to a:*

$$\psi[f, a] = \int_E f(x)\mathbf{1}_{\{f(x)\leq f(a)\}}dx = \mathbb{P}\left(f(X) \leq f(a)\right). \tag{5.3.4}$$

In this case, $\psi[f_t, \boldsymbol{y}(t)]$ corresponds to the probability of Z_t falling outside the smallest confidence region containing $\boldsymbol{y}(t)$, as illustrated in figure 5.2.

A numerical comparison of these two scalings can be found in section 5.5, while a class

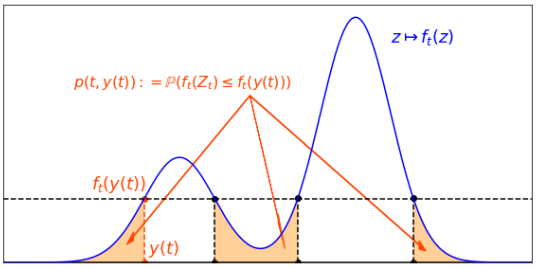


Figure 5.2 – Illustration of the confidence level in the case of a univariate bimodal distribution

of estimators of the mean marginal likelihood is presented in the next section.

5.3.2 Empirical Version

Usually in FDA, one only has access to discrete observations of the random functions under study. We assume to be in this context: for $1 \leq r \leq m$, each path $\boldsymbol{z}^r$ of the training set $\mathcal{T}$ is assumed to be observed at $n \in \mathbb{N}^*$ discrete times $\{t_1^r < t_2^r < \cdots < t_n^r\} \subset \mathbb{T}$, drawn independently from some random variable T, supposed independent of Z. Hence, we denote by $\mathcal{T}^D$ the set of all discrete observations:

$$\mathcal{T}^D = \{(t_j^r, z_j^r)\}_{\substack{1 \leq j \leq n \\ 1 \leq r \leq m}} \subset \mathbb{T} \times E, \tag{5.3.5}$$

where $z_j^r = \boldsymbol{z}(t_j^r)$. Likewise, we assume that the new curve $\boldsymbol{y}$ is observed at $\tilde{n} \in \mathbb{N}$ discrete times $\{\tilde{t}_j\}_{j=1}^{\tilde{n}} \subset \mathbb{T}$ and we denote these observations by

$$\mathcal{Y} = \{(\tilde{t}_j, y_j)\}_{j=1}^{\tilde{n}} \subset \mathbb{T} \times E, \tag{5.3.6}$$

where $y_j = \boldsymbol{y}(\tilde{t}_j)$.

Had we enough observations of $\boldsymbol{y}$, we could approximate the integral in (5.3.2) by a Riemann sum:

$$\frac{1}{t_f} \sum_{j=1}^{\tilde{n}} \psi[f_j, y_j] \Delta \tilde{t}_j, \tag{5.3.7}$$

where $f_j := f_{\tilde{t}_j}$, $\Delta \tilde{t}_j := \tilde{t}_j - \tilde{t}_{j-1}$ and $\tilde{t}_0 = 0$. Yet, the marginal densities $\{f_j\}_{j=1}^{\tilde{n}}$ are unknown and need to be estimated for practical use.

Our approach is based on the idea of partitioning $\mathbb{T}$ into q_m intervals, or *bins*, of same length $b_m = t_f / q_m$ (see figure 5.3). For $1 \leq \ell \leq q_m$, let $\tau_\ell := \tau_0 + \ell b_m$, where $\tau_0 = \inf \mathbb{T} = 0$. We denote by $B_\ell := [\tau_{\ell-1}; \tau_\ell)$ the ℓ^{th} bin for $\ell = 1, \ldots, q_m - 1$ and $B_{q_m} := [\tau_{q_m - 1}; \tau_{q_m}]$. Similarly, $\mathcal{T}_\ell = \{(t_j^r, z_j^r) : t_j^r \in B_\ell\} \subset \mathcal{T}^D$ denotes the set of obser-

vations whose sampling time fall into B_ℓ. For some m and $1 \leq j \leq \tilde{n}$, let ℓ be such that $\tilde{t}_j \in B_\ell$. For b_m small enough, we estimate f_j by building a density estimator with the partial data contained in $\mathcal{T}_\ell$. This is done by applying a common statistic $\Theta : \mathcal{S} \to L^1(E, \mathbb{R}_+)$ to $\mathcal{T}_\ell$, where $\mathcal{S}\{(z_k)_{k=1}^N \in E^N : N \in \mathbb{N}^*\}$ denotes the set of finite sequences valued on $E \subset \mathbb{R}^d$: $\hat{f}_j := \Theta[\mathcal{T}_\ell]$. Hence, we consider a single density estimator per bin, averaging along the times in it. We denote this estimated quantities $\{\hat{f}_j\}_{j=1}^{\tilde{n}}$ and by summing them we can build the following plug-in estimator, called the *Empirical Mean Marginal Likelihood* hereafter:

$$\mathrm{EMML}_m(Z, \boldsymbol{y}) := \frac{1}{t_f} \sum_{j=1}^{\tilde{n}} \psi[\hat{f}_j, y_j] \Delta \tilde{t}_j. \tag{5.3.8}$$

In the following subsection 5.3.3, sufficient conditions are given for the consistent estimation of the marginal densities f_t, while section 5.4.3 presents a possible class of kernel density estimators to compute $\{\hat{f}_j\}_{j=1}^{\tilde{n}}$.

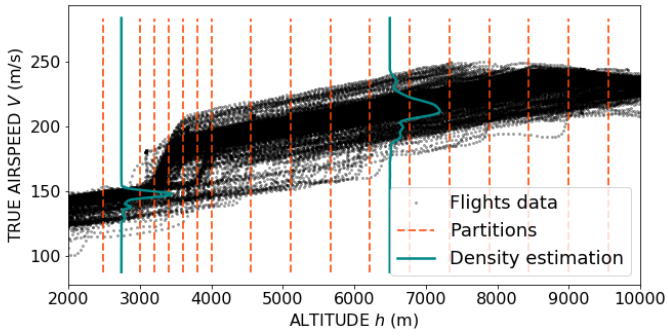


Figure 5.3 – Illustration of the marginal density estimation.

5.3.3 Consistency of the marginal density estimations

General case In this section we state that by using some well chosen statistic to build density estimators in the bins described in section 5.3.2 we obtain pointwise consistent estimations of the marginal densities of Z. We describe the main ideas of the proof here, while the technical details are postponed to the next section. Our consistency result is summarized in theorem 5.3.6 and relies on the following 4 assumptions:

Assumption 5.3.2 *The random variable T is absolutely continuous and $\nu \in L^\infty(E, \mathbb{R}_+)$, its density relative to the Lebesgue measure, satisfies:*

$$\nu_+ := \operatorname*{ess\,sup}_{t \in \mathbb{T}} \nu(t) < \infty, \qquad \nu_- := \operatorname*{ess\,inf}_{t \in \mathbb{T}} \nu(t) > 0. \tag{5.3.9}$$

Assumption 5.3.3 *The function defined by*

$$(t, z) \in \mathbb{T} \times E \mapsto f_t(z) \tag{5.3.10}$$

is continuous on both variables and Lipschitz in time with constant $L > 0$: for any $z \in E$ and $t_1, t_2 \in \mathbb{T}$

$$|f_{t_1}(z) - f_{t_2}(z)| \leq L|t_1 - t_2|. \tag{5.3.11}$$

Assumption 5.3.4 *The homogeneous partition $\{B_\ell^m\}_{\ell=1}^{q_m}$ of $\mathbb{T} = [0; t_f]$, where the bins have size $b_m := t_f/q_m$, is such that*

$$\lim_{m\to\infty} b_m = 0, \tag{5.3.12}$$

$$\lim_{m\to\infty} m b_m = \infty. \tag{5.3.13}$$

Let $\mathcal{S} = \{(z_k)_{k=1}^N \in E^N : N \in \mathbb{N}^*\}$ be the set of finite sequences with values in the compact set $E \subset \mathbb{R}^d$. We also need to assume that the statistic $\Theta : \mathcal{S} \to L^1(E, \mathbb{R}_+)$ used to build the density estimators leads to uniformly consistent density estimations in a standard i.i.d setting, which is summarized in the following assumption:

Assumption 5.3.5 *Let $\mathcal{G}$ be an arbitrary family of probability density functions on E.Given a density $\rho \in \mathcal{G}$, let S_ρ^N be an* i.i.d *sample of size N valued in $\mathcal{S}$. The estimator obtained by applying Θ to S_ρ^N, denoted by*

$$\hat{\rho}^N := \Theta[S_\rho^N] \in L^1(E, \mathbb{R}_+), \tag{5.3.14}$$

is a (pointwise) consistent density estimator, uniformly in ρ:

$$\begin{gathered}\text{For all } z \in E, \varepsilon > 0, \alpha_1 > 0, \text{ there is } N_{\varepsilon,\alpha_1} > 0 \text{ such that, for any } \rho \in \mathcal{G}, \\ N \geq N_{\varepsilon,\alpha_1} \Rightarrow \mathbb{P}\left(\left|\hat{\rho}^N(z) - \rho(z)\right| < \varepsilon\right) > 1 - \alpha_1.\end{gathered} \tag{5.3.15}$$

For $m \in \mathbb{N}^*$, let $\ell^m : \mathbb{T} \to \mathbb{N}^*$ be the function mapping any point $t \in \mathbb{T} = [0; t_f]$ to the index of the bin containing it:

$$\ell^m(t) := \left\lceil \frac{t}{b_m} \right\rceil. \tag{5.3.16}$$

We denote by $\hat{f}^m_{\ell^m(t)}$ the estimator obtained by applying Θ to the subset of data points $\mathcal{T}^m_{\ell^m(t)}$ whose sampling times fall in the bin containing t.

Theorem 5.3.6 *Under assumptions 5.3.2 to 5.3.5, for any $z \in E$ and $t \in \mathbb{T}$, $\hat{f}^m_{\ell^m(t)}(z)$ consistently approximates the marginal density $f_t(z)$ as the number of curves m grows:*

$$\forall \varepsilon > 0, \quad \lim_{m\to\infty} \mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f_t(z)| < \varepsilon\right) = 1. \tag{5.3.17}$$

Remark Note that, unlike assumption 5.3.5, the convergence in theorem 5.3.6 is written in terms of m. This is a big difference since the number of observation points used is supposed to be controlled in the general setting of assumption 5.3.5, while it is a random variable in theorem 5.3.6 (number of observations falling in the bin $\mathcal{T}^m_{\ell^m(t)}$).

Before explaining the proof of such a theorem, notice that the observations falling into a certain bin for a given number of curves m follow some distribution whose density function can be explicitly derived. Indeed, for $\mathcal{V} \subset E$ and $B \subset \mathbb{T}$ two compact sets, we have

$$\mathbb{P}\left(Z_T \in \mathcal{V} | T \in B\right) = \frac{\mathbb{P}\left(\{Z_T \in \mathcal{V}\} \bigcap \{T \in B\}\right)}{\mathbb{P}\left(T \in B\right)} = \frac{\int_{\mathcal{V}} \int_B f_t(z)\nu(t)dt \; dz}{\int_B \nu(t)dt}. \tag{5.3.18}$$

This shows that $(Z_T | Z_T \in \mathcal{T}^m_{\ell^m(t)}) = (Z_T | T \in B^m_{\ell^m(t)})$ follows a distribution of density

$$f^m_{\ell^m(t)}(z) := \frac{\int_{\tau^m_{\ell^m(t)-1}}^{\tau^m_{\ell^m(t)}} f_t(z)\nu(t)dt}{\int_{\tau^m_{\ell^m(t)-1}}^{\tau^m_{\ell^m(t)}} \nu(t)dt}. \tag{5.3.19}$$

The proof of theorem 5.3.6 relies on the fact that $f^m_{\ell^m(t)}$ converges pointwise to the marginal density f_t as m tends to infinity. It is indeed quite straightforward to show this by using the assumptions that f_t is Lipschitz in time (5.3.11) and that the bin sizes b_m tend to 0 (5.3.12). From there, the idea is to try to apply the consistency result from assumption 5.3.5 to show that $\hat{f}^m_{\ell^m(t)}$ converges pointwise in probability to $f^m_{\ell^m(t)}$. However, two main difficulties arise here:

1. $\hat{f}^m_{\ell^m(t)}$ is trained using the observations from $\mathcal{T}^m_{\ell^m(t)}$ and the number of elements contained in this subset, denoted by $N^m_{\ell^m(t)}$, is random;
2. we need to train $\hat{f}$ on i.i.d observations whose number tend to infinity in order to apply (5.3.15).

The first difficulty can be tackled by conditioning on $N^m_{\ell^m(t)}$. For the second one, we use the fact that, as the bin size tend to 0 and as the number n of observations per curve is fixed with respect to m, than each training subset has, with high probability, at most one observation per curve asymptotically. Hence, because the curves are independent observations of Z, the observations contained in $\mathcal{T}^m_{\ell^m(t)}$ for m large enough will be independently drawn from $f^m_{\ell^m(t)}$ with probability 1. Furthermore, we can show that if the bin size does not decrease too fast, as required by (5.3.13), than $N^m_{\ell^m(t)}$ diverges to $+\infty$ in probability. The detailed proof of theorem 5.3.6 can be found in the following section.

5.3.4 Proof of the consistency of the marginal density estimation (theorem 5.3.6)

In this section we prove theorem 5.3.6 from section 5.3.3. The proof relies on 5 lemmas stated and proved hereafter.

Lemma 5.3.7 *Let $t \in \mathbb{T}$ and $z \in E$. Under assumptions 5.3.3 and 5.3.4, $f^m_{\ell^m(t)}(z)$ converges to $f_t(z)$:*

$$\lim_{m \to \infty} |f_t(z) - f^m_{\ell^m(t)}(z)| = 0. \tag{5.3.20}$$

Proof

$$\begin{aligned} |f_t(z) - f^m_{\ell^m(t)}(z)| &= \left| f_t(z) - \frac{\int_{\mathcal{B}^m_{\ell^m(t)}} f_s(z)\nu(s)ds}{\int_{B^m_{\ell^m(t)}} \nu(s)ds} \right|, \\ &= \frac{\left| \int_{\mathcal{B}^m_{\ell^m(t)}} (f_t(z) - f_s(z))\nu(s)ds \right|}{\int_{B^m_{\ell^m(t)}} \nu(s)ds}, \\ &\leq \frac{\int_{\mathcal{B}^m_{\ell^m(t)}} |f_t(z) - f_s(z)|\nu(s)ds}{\int_{B^m_{\ell^m(t)}} \nu(s)ds}. \end{aligned} \tag{5.3.21}$$

According to assumption 5.3.3,

$$|f_t(z) - f_s(z)| \leq L|t-s| \leq L|\tau^m_{\ell^m(t)} - \tau^m_{\ell^m(t)-1}| = Lb_m. \tag{5.3.22}$$

Hence,

$$|f_t(z) - f^m_{\ell^m(t)}(z)| \leq Lb_m \frac{\int_{\mathcal{B}^m_{\ell^m(t)}} \nu(s)ds}{\int_{B^m_{\ell^m(t)}} \nu(s)ds} = Lb_m. \tag{5.3.23}$$

Since $b_m \to 0$ by assumption 5.3.4, the conclusion follows. ∎

Lemma 5.3.8 *Let $m \in N^*$ and $t \in \mathbb{T}$. Under assumption 5.3.4, the probability that $\mathcal{T}^m_{\ell^m(t)}$ (the subset of training points whose sampling time fall in the bin containing t) contains at most one observation point per curve is asymptotically equal to 1, meaning that for large enough m, the observations in $\mathcal{T}^m_{\ell^m(t)}$ will be independent with high probability:*

$$\lim_{m\to\infty} \mathbb{P}(N^m_{r,\ell^m(t)} \leq 1) = 1, \quad r = 1, \ldots, m, \tag{5.3.24}$$

where $N^m_{r,\ell^m(t)}$ denotes the number of observations of $\boldsymbol{z}^r$ in $\mathcal{T}^m_{\ell^m(t)}$.

Proof Let $1 \leq r \leq m$ and

$$\mathcal{T}^m_{r,\ell^m(t)} := \{(t^r_k, z^r_k) : t^r_k \in B^m_{\ell^m(t)}; 1 \leq k \leq n\} \tag{5.3.25}$$

be the set of observations of the r^{th} curve with times lying in the $\ell^m(t)^{th}$ bin $B^m_{\ell^m(t)}$. Let $N^m_{r,\ell^m(t)}$ be the number of elements in $\mathcal{T}^m_{\ell^m(t)}$. The random variable $N^m_{r,\ell^m(t)}$ follows a binomial law $\mathcal{B}(n, P^m_{\ell^m(t)})$, where

$$P^m_{\ell^m(t)} = \int_{B^m_{\ell^m(t)}} \nu(t)dt \tag{5.3.26}$$

is the probability of a new observation of the r^{th} curve falling in $\mathcal{T}^m_{r,\ell^m(t)}$. The probability

of $N^m_{r,\ell^m(t)}$ being at most equal to 1 writes

$$\begin{aligned}\mathbb{P}(N^m_{r,\ell^m(t)} \leq 1) &= \sum_{k=0}^{1} \begin{pmatrix} n \\ k \end{pmatrix} (P^m_{\ell^m(t)})^k (1 - P^m_{\ell^m(t)})^{n-k}, \\ &= (1 - P^m_{\ell^m(t)})^n + n P^m_{\ell^m(t)} (1 - P^m_{\ell^m(t)})^{n-1}, \\ &= (1 - P^m_{\ell^m(t)})^{n-1} (1 + (n-1) P^m_{\ell^m(t)}).\end{aligned} \tag{5.3.27}$$

As

$$P^m_{\ell^m(t)} = \int_{B^m_{\ell^m(t)}} \nu(t) dt \leq b_m \nu_+, \tag{5.3.28}$$

we have

$$\mathbb{P}(N^m_{r,\ell^m(t)} \leq 1) \geq (1 - P^m_{\ell^m(t)})^{n-1} \geq (1 - b_m \nu_+)^{n-1}. \tag{5.3.29}$$

Since $b_m \to 0$ according to assumption 5.3.4, we obtain

$$\lim_{m \to \infty} \mathbb{P}(N^m_{r,\ell^m(t)} \leq 1) = 1, \quad r = 1, \ldots, m. \tag{5.3.30}$$

∎

Lemma 5.3.9 *Under assumption 5.3.4 and for any $t \in \mathbb{T}$, the number $N^m_{\ell^m(t)}$ of observations falling in $\mathcal{T}^m_{\ell^m(t)}$ diverges in probability to $+\infty$:*

$$\forall M > 0, \quad \lim_{m \to \infty} \mathbb{P}\left(N^m_{\ell^m(t)} > M\right) = 1. \tag{5.3.31}$$

Proof Let $M > 0$. As in the proof of lemma 5.3.8, we have $N^m_{\ell^m(t)} \sim B(mn, P^m_{\ell^m(t)})$ and hence,

$$\mathbb{P}\left(N^m_{\ell^m(t)} \leq M\right) = \sum_{k=1}^{\lfloor M \rfloor} \binom{nm}{k} (P^m_{\ell^m(t)})^k (1 - P^m_{\ell^m(t)})^{nm-k}. \tag{5.3.32}$$

As the sum has is finite when we fix M, the limit of expression (5.3.32) when m tends to infinity is the sum of the limits of its terms.

$$\begin{aligned}&\binom{nm}{k} (P^m_{\ell^m(t)})^k (1 - P^m_{\ell^m(t)})^{nm-k} \\ &\qquad \leq \binom{nm}{k} (\nu_+ b_m)^k (1 - \nu_- b_m)^{nm-k}, \\ &\qquad \underset{m \to \infty}{\sim} (\nu_+ nm b_m)^k (1 - \nu_- b_m)^{nm}, \\ &\qquad = (nm b_m)^k \exp\left[nm \log\left(1 - \nu_- b_m\right)\right], \\ &\qquad \underset{m \to \infty}{\sim} (nm b_m)^k \exp\left[-\nu_- nm b_m + o(m b_m)\right].\end{aligned} \tag{5.3.33}$$

Using (5.3.13) from assumption 5.3.4, we obtain that (5.3.33) tends to 0 which proves that

$$\lim_{m\to\infty} \mathbb{P}\left(N^m_{\ell^m(t)} > M\right) = 1 - \lim_{m\to\infty} \mathbb{P}\left(N^m_{\ell^m(t)} \leq M\right) = 1.$$

∎

In what follows, for any $M > 0$ we denote C_M the following event:

$$C_M := \{N^m_{\ell^m(t)} > M\} \bigcap_{r=1}^{m} \{N^m_{r,\ell^m(t)} \leq 1\}. \tag{5.3.34}$$

Lemma 5.3.10 *For any $t \in \mathbb{T}$, if assumption 5.3.4 holds,*

$$\forall M > 0, \quad \lim_{m\to\infty} \mathbb{P}\left(C_M\right) = 1. \tag{5.3.35}$$

Proof Let $M > 0$. We have by definition of the conditional probability

$$\mathbb{P}\left(C_M\right) = \mathbb{P}\left(N^m_{\ell^m(t)} \middle| N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m\right) \times \mathbb{P}\left(N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m\right). \tag{5.3.36}$$

As the variables $N^m_{r,\ell^m(t)}$ are independent

$$\mathbb{P}\left(N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m\right) = \prod_{r=1}^{m} \mathbb{P}\left(N^m_{r,\ell^m(t)} \leq 1\right) \tag{5.3.37}$$

which tends to 1 when m grows according to lemma 5.3.8.

Furthermore, as $N^m_{\ell^m(t)} = \sum_{r=1}^m N^m_{r,\ell^m(t)} \sim \mathcal{B}(nm, P^m_{\ell^m(t)})$, we know that the random variable $(N^m_{\ell^m(t)} | N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m)$ is also a binomial random variable where at least $m(n-1)$ Bernouilli have failed, leaving m trials. Hence, we know that $(N^m_{\ell^m(t)} | N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m) \sim \mathcal{B}(m, P^m_{\ell^m(t)})$. As in the proof of lemma 5.3.9,

$$\begin{aligned}\mathbb{P}\left(N^m_{\ell^m(t)} \leq M \middle| N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m\right) = \\ \sum_{k=1}^{\lfloor M \rfloor} \binom{m}{k} (P^m_{\ell^m(t)})^k (1 - P^m_{\ell^m(t)})^{m-k},\end{aligned} \tag{5.3.38}$$

whose limit is the sum of the limits of the sum terms. As in the proof of lemma 5.3.9, we have

$$\binom{m}{k} (P^m_{\ell^m(t)})^k (1 - P^m_{\ell^m(t)})^{nm-k} \simeq (mb_m)^k \exp\left[-\nu_- m b_m + o(mb_m)\right], \tag{5.3.39}$$

and hence, by using (5.3.13) from assumption 5.3.4,

$$\lim_{m\to\infty} \mathbb{P}\left(N^m_{\ell^m(t)} \leq M \middle| N^m_{r,\ell^m(t)} \leq 1; r = 1, \ldots, m\right) = 0. \tag{5.3.40}$$

Combining (5.3.36), (5.3.37) and (5.3.40) we obtain (5.3.35). ∎

Lemma 5.3.11 *For any $z \in E$ and $t \in \mathbb{T}$, if assumptions 5.3.4 and 5.3.5 hold for some function Θ used to compute $\hat{f}^m_{\ell^m(t)}$, then*

$$\forall \varepsilon > 0, \qquad \lim_{m\to\infty} \mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| < \varepsilon\right) = 1. \tag{5.3.41}$$

Proof The randomness of $\hat{f}^m_{\ell^m(t)}(z)$ comes from both the sample of observations drawn from the bin's density $f^m_{\ell^m(t)}$ and the random number of observations falling in the bin $N^m_{\ell^m(t)}$. Hence, the idea here is to separate these two sources of randomness by conditioning on one of them. As we would like to use the result from assumption 5.3.5, it makes sense to condition on $N^m_{\ell^m(t)}$ here.

Indeed, for any $m > 0$, $\hat{f}^m_{\ell^m(t)}(z) = \Theta[\mathcal{T}^m_{\ell^m(t)}](z)$, where $\mathcal{T}^m_{\ell^m(t)}$ is a sample of $N^m_{\ell^m(t)}$ observations drawn from $f^m_{\ell^m(t)}$. Hence, according to assumption 5.3.5, for any $\varepsilon > 0$, for any $\alpha_1 > 0$, there is some $N_{\varepsilon,\alpha_1,m}$ such that

$$\mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| < \varepsilon \middle| C_{N_{\varepsilon,\alpha_1,m}}\right) > 1 - \alpha_1. \tag{5.3.42}$$

As seen in lemma 5.3.8, the conditioning on $N^m_{r,\ell^m(t)} \leq 1$ comes from the fact that when $\mathcal{T}^m_{\ell^m(t)}$ contains at most one observation per curve, those observations are independent. Furthermore, for ε, α_1 fixed, let $M := \sup_{m'}\{N_{\varepsilon,\alpha_1,m'}\}$ (which exists according to assumption 5.3.5). In this case, when assumption 5.3.4 holds, we have from lemma 5.3.10 that for any $\alpha_2 > 0$,there is some $m_{\varepsilon,\alpha_1,\alpha_2}$ such that,

$$m \geq m_{\varepsilon,\alpha_1,\alpha_2} \quad \Rightarrow \quad \mathbb{P}(C_M) > 1 - \alpha_2. \tag{5.3.43}$$

Hence, as $N^m_{\ell^m(t)} > M \Rightarrow N^m_{\ell^m(t)} > N_{\varepsilon,\alpha_1,m}$, we have

$$m \geq m_{\varepsilon,\alpha_1,\alpha_2} \quad \Rightarrow \quad \mathbb{P}\left(C_{N_{\varepsilon,\alpha_1,m}}\right) > 1 - \alpha_2. \tag{5.3.44}$$

Moreover, we can also write the following useful inequality:

$$\begin{aligned}
&\mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| < \varepsilon\right) \\
&\quad > \mathbb{P}\left(\{|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| < \varepsilon\} \bigcap C_{N_{\varepsilon,\alpha_1,m}}\right) \\
&\quad = \mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| < \varepsilon \middle| C_{N_{\varepsilon,\alpha_1,m}}\right) \times \mathbb{P}\left(C_{N_{\varepsilon,\alpha_1,m}}\right).
\end{aligned} \tag{5.3.45}$$

By combining (5.3.42), (5.3.44) and (5.3.45) we get that for any $\varepsilon, \alpha_1, \alpha_2 > 0$, there is $m_{\varepsilon,\alpha_1,\alpha_2}$ such that

$$m \geq m_{\varepsilon,\alpha_1,\alpha_2} \Rightarrow \mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| < \varepsilon\right) > 1 - \alpha_3, \tag{5.3.46}$$

with $\alpha_3 = \alpha_1 + \alpha_2 - \alpha_1\alpha_2$. ∎

Using these lemmas we can finally prove theorem 5.3.6:

Proof of Theorem 5.3.6 Let $(t, z) \in \mathbb{T} \times E$ and let $\varepsilon > 0$. According to lemma 5.3.7, under assumptions 5.3.3 and 5.3.4, there is $m_\varepsilon \in \mathbb{N}^*$ such that for any $m \geq m_\varepsilon$,

$$|f^m_{\ell^m(t)}(z) - f_t(z)| < \varepsilon \tag{5.3.47}$$

As

$$\begin{aligned} |\hat{f}^m_{\ell^m(t)}(z) - f_t(z)| &\leq |\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| + |f^m_{\ell^m(t)}(z) - f_t(z)|, \\ &\leq |\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| + \varepsilon, \end{aligned} \tag{5.3.48}$$

we have

$$\{|\hat{f}^m_{\ell^m(t)}(z) - f_t(z)| > 2\varepsilon\} \subset \{|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| > \varepsilon\},$$

and

$$\mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f_t(z)| > 2\varepsilon\right) \leq \mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| > \varepsilon\right). \tag{5.3.49}$$

According to lemma 5.3.11 we have that

$$\lim_{m\to\infty} \mathbb{P}\left(|\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z)| > \varepsilon\right) = 0, \tag{5.3.50}$$

which allow us to obtain (5.3.17) from (5.3.49). ∎

An even stronger convergence (in the L^2 sense) is proven in section 5.4.2 for standard kernel density estimators.

5.4 Choice of the Marginal Density Estimator

5.4.1 Parametric or nonparametric ?

In the previous section we presented a general estimator of some discrepancy used to quantify how close a certain curve is to a set of other curves, called the *mean marginal likelihood*. As explained, such plug-in estimator is based on the aggregation of other consistent density estimators trained on uniform bins. One may wonder what local density estimator to use in this situation.

As most statistical learning problems, density estimation can be tackled in a parametric or a nonparametric setting. In the first case, a specific class of density functions has to be fixed *a priori*, a finite set of unknown parameters needing to be tuned using information contained in the data. Such approaches, as for example *maximum likelihood* estimation, are known to be fast to train and evaluate, they have the best learning rate [1] attainable

1. In statistical learning, we usually call *learning rate* the convergence speed of an estimator in terms of its accuracy, with regard to the number of observations used to build it.

and are usually very scalable and accurate if the model assumptions are correct. However, the nonparametric density learning techniques make little to no assumptions on the shape of the density to be estimated. As explained in section 5.3.1, the marginal densities at different times may greatly vary in shape, which is why we preferred to consider nonparametric estimators in this chapter and more precisely the popular *kernel density estimators* [Parzen, 1962, Rosenblatt, 1956].

5.4.2 Kernel estimator

Classical setting

For d-dimensional data, kernel density estimators (KDE) have the following general form when trained on N i.i.d observations $\{x_k\}_{k=1}^N$ of a random variable X of density f:

$$\hat{f}^N(x) := \frac{1}{N \det H} \sum_{k=1}^{N} K\left(H^{-1}(x - x_k)\right). \tag{5.4.1}$$

In (5.4.1), the function $K : \mathbb{R}^d \to \mathbb{R}_+$ is called a *smoothing kernel* and $H \in GL_d(\mathbb{R})$ is the kernel's *bandwidth* matrix.

As seen in expression (5.4.1), the idea of this estimation method is to place a smoothing kernel at each observation point, which generalizes the idea of a histogram (see figure 5.4). By representing the empirical distribution of the sample as a sum of Dirac functions [2] centered on the observation points

$$\frac{1}{N} \sum_{k=1}^{N} \delta_{x_k}(x), \tag{5.4.2}$$

the KDE from (5.4.1) writes as the convolution of this function with the smoothing kernel:

$$\hat{f}^N(x) = K_H * \left(\frac{1}{N} \sum_{k=1}^{N} \delta_{x_k}(x) \right), \tag{5.4.3}$$

where $K_H(x) := K(H^{-1}x)/\det H$. From this viewpoint, KDE can be seen as a *moving-average method* which smooths the empirical distribution using the kernel as a window.

MML setting

In our case, we would like to understand how estimator (5.4.1) would behave when trained on the data contained in one of the subsets $\mathcal{T}^m_{\ell^m(t)}$ built in section 5.3.2. Such an estimator writes as follows:

$$\hat{f}^m_{\ell^m(t)}(z) = \frac{1}{\sigma N^m_{\ell^m(t)}} \sum_{z_k \in \mathcal{T}^m_{\ell^m(t)}} K\left(\frac{z_k - z}{\sigma}\right) = \frac{1}{N^m_{\ell^m(t)}} \sum_{z_k \in \mathcal{T}^m_{\ell^m(t)}} K_\sigma(z_k - z). \tag{5.4.4}$$

2. equal to 1 at the sample point and zero elsewhere

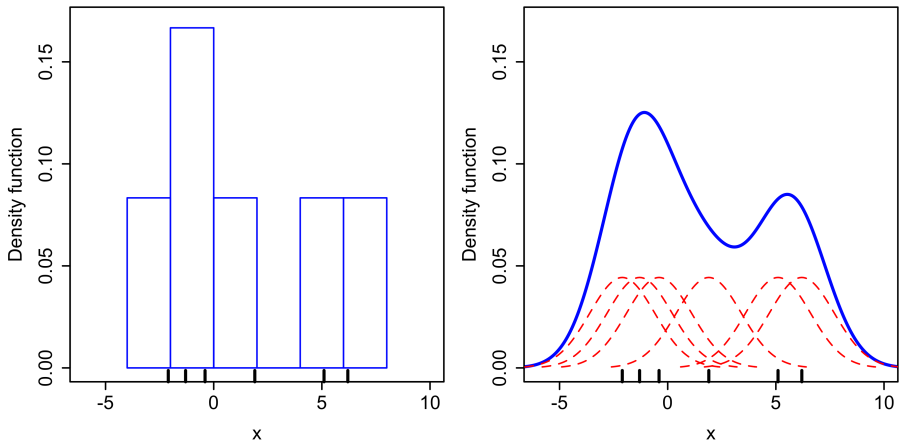


Figure 5.4 – Illustration of a histogram density estimation (*left*) and a kernel density estimation (*right*) built using the same samples. The red dashed curves correspond to the kernels centered at the data points.

where $K : \mathbb{R}^d \to \mathbb{R}$ is symmetric kernel summing to 1 and $\sigma > 0$ is the bandwidth, chosen to be scalar here for simplicity.

In the next paragraph, we state a consistency result for this particular setting which is stronger than theorem 5.3.6.

Pointwise consistency of the deterministic kernel marginal density estimator

We denote the second moments of the random variables of density K_σ and K_σ^2 respectively by

$$\sigma_{K_\sigma}^2 = \int w^2 K_\sigma(w) dw = \sigma^2 \int w^2 K(w) dw = \sigma^2 \sigma_K^2, \tag{5.4.5}$$

$$\sigma_{K_\sigma^2}^2 = \int w^2 K_\sigma(w)^2 dw = \sigma \int w^2 K(w)^2 dw = \sigma \sigma_{K^2}^2, \tag{5.4.6}$$

and we denote the kernel risk by

$$R(K_\sigma) = \int K_\sigma(w)^2 dw = \frac{1}{\sigma} \int K(w)^2 dw = \frac{1}{\sigma} R(K). \tag{5.4.7}$$

For this particular setting, we state in theorem 5.4.2 that, under certain conditions, $\hat{f}^m_{\ell^m(t)}(z)$ approximates $f_t(z)$ consistently in expected squared-error, which is stronger than the convergence in probability stated in theorem 5.3.6:

Assumption 5.4.1 *The function $(t, z) \in \mathbb{T} \times E \mapsto f_t(z)$ is $\mathcal{C}^4(E)$ in z and $\mathcal{C}^1(\mathbb{T})$ in t ; the Lipschitz constant of the function*

$$t \mapsto \frac{d^2 f_t}{dz^2}(z) := f_t''(z) \tag{5.4.8}$$

is denoted by $L'' > 0$: for any $z \in E$ and $t_1, t_2 \in \mathbb{T}$,

$$|f''_{t_1}(z) - f''_{t_2}(z)| \leq L''|t_1 - t_2|. \tag{5.4.9}$$

Theorem 5.4.2 *Under assumptions 5.3.2, 5.3.4 and 5.4.1, if $\hat{f}^m_{\ell^m(t)}$ is a kernel estimator of the form (5.4.4) where the kernel K and the bandwidth $\sigma := \sigma_m$ are deterministic (i.e. do not depend on the data), such that $\sigma_K < \infty$, $\sigma_{K^2} < \infty$, $R(K) < \infty$ and if*

$$\lim_{m\to\infty} \sigma_m = 0, \qquad \lim_{m\to\infty} m b_m \sigma_m = +\infty, \tag{5.4.10}$$

then

$$\lim_{m\to\infty} \mathbb{E}\left[(\hat{f}^m_{\ell^m(t)}(z) - f_t(z))^2\right] = 0. \tag{5.4.11}$$

As an example, according to (5.3.13) from assumption 5.3.4, theorem 5.4.2 applies to the case of a Gaussian kernel and a bandwidth $\sigma_m = 1/\sqrt{m b_m}$. Unfortunately, it does not apply to the marginal density estimator presented in the next section, whose kernel is random.

Proof of theorem 5.4.2

We assume here that $\hat{f}^m_{\ell^m(t)}$ is a standard kernel density estimator of the form (5.4.4).

Remark Such an estimator is only defined for $N^m_{\ell^m(t)} > 0$. This is why, in the remaining section, we consider the conditioned random variables $(N^m_{\ell^m(t)} | N^m_{\ell^m(t)} > 0)$ and $(\hat{f}^m_{\ell^m(t)} | N^m_{\ell^m(t)} > 0)$, which are still denoted $N^m_{\ell^m(t)}$ and $\hat{f}^m_{\ell^m(t)}$ for simplicity. Furthermore,

$$\mathbb{P}\left(\cdot | N^m_{\ell^m(t)} > 0\right) = \frac{\mathbb{P}\left(\cdot \bigcap \{N^m_{\ell^m(t)} > 0\}\right)}{\mathbb{P}\left(N^m_{\ell^m(t)} > 0\right)}, \tag{5.4.12}$$

and, according to lemma 5.3.9, $\lim_{m\to\infty} \mathbb{P}\left(N^m_{\ell^m(t)} > 0\right) = 1$.

In the following, we use the following notations for the conditional expectation and variance:

$$\tilde{\mathbb{E}}[\cdot] := \mathbb{E}\left[\cdot | N^m_{\ell^m(t)} > 0\right], \qquad \tilde{\mathrm{Var}}[\cdot] := \mathrm{Var}\left[\cdot | N^m_{\ell^m(t)} > 0\right]. \tag{5.4.13}$$

In the remaining of this section we derive the conditions under which $\hat{f}^m_{\ell^m(t)}$ converges in expected squared-error to f_t. We recall that a sufficient condition for this is having its bias and variance tending to 0. The proof was greatly inspired by the derivations presented in Scott [2015] for the standard multivariate case.

Similarly to lemma 5.3.7, we can prove the following convergence result:

Lemma 5.4.3 *Under assumption 5.4.1, for any* $(t,z) \in \mathbb{T} \times E$,

$$\frac{d^2 f^m_{\ell^m(t)}}{dz^2}(z) := \left(f^m_{\ell^m(t)}\right)''(z) = \frac{\int_{\tau^m_{\ell-1}}^{\tau^m_\ell} f''_t(z)\nu(t)dt}{\int_{\tau^m_{\ell-1}}^{\tau^m_\ell} \nu(t)dt}, \tag{5.4.14}$$

and

$$\lim_{m\to\infty} \left| \left(f^m_{\ell^m(t)}\right)''(z) - f''_t(z) \right| = 0. \tag{5.4.15}$$

Proof Similar argument to lemma 5.3.7. ∎

Furthermore, the following bounds of the estimator's bias and variance hold under the same assumption:

Lemma 5.4.4 *If assumption 5.4.1 holds,*

$$\left| \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] - f^m_{\ell^m(t)}(z) \right| \leq \frac{1}{2} \left\| \left(f^m_{\ell^m(t)}\right)'' \right\|_\infty \tilde{\mathbb{E}}\left[\sigma^2_{K_\sigma}\right], \tag{5.4.16}$$

$$\begin{aligned} \tilde{\mathrm{Var}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] \leq\ & f^m_{\ell^m(t)}(z)\tilde{\mathbb{E}}\left[\frac{R(K_\sigma)}{N^m_{\ell^m(t)}}\right] + \\ & \frac{1}{2} \left\| \left(f^m_{\ell^m(t)}\right)'' \right\|_\infty \tilde{\mathbb{E}}\left[\frac{\sigma^2_{K^2_\sigma}}{N^m_{\ell^m(t)}}\right] + \frac{1}{4} \left\| \left(f^m_{\ell^m(t)}\right)'' \right\|^2_\infty \tilde{\mathbb{E}}\left[(\sigma^2_{K_\sigma})^2\right]. \end{aligned} \tag{5.4.17}$$

Proof From the law of total expectation [e.g. Wasserman, 2004, p.55]

$$\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] = \tilde{\mathbb{E}}\left[\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z) | N^m_{\ell^m(t)}\right]\right]. \tag{5.4.18}$$

For large enough m, $\hat{f}^m_{\ell^m(t)}$ writes as an empirical average of independent identically distributed random variables $Z_{m,t}$ of density $f^m_{\ell^m(t)}$ and hence

$$\begin{aligned} \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z) | N^m_{\ell^m(t)}\right] &= \tilde{\mathbb{E}}\left[K_\sigma(z - Z_{m,t}) | N^m_{\ell^m(t)}\right] = \tilde{\mathbb{E}}\left[K_\sigma(z - Z_{m,t})\right], \\ &= \int K_\sigma(z-x) f^m_{\ell^m(t)}(x)dx = \int K_\sigma(w) f^m_{\ell^m(t)}(z-w)dw. \end{aligned} \tag{5.4.19}$$

The second order integral Taylor expansion of $f^m_{\ell^m(t)}$ around z gives

$$f^m_{\ell^m(t)}(z-w) = f^m_{\ell^m(t)}(z) - \left(f^m_{\ell^m(t)}\right)'(z)w + \int_0^1 (1-x)\left(f^m_{\ell^m(t)}\right)''(z-xw)w^2 dx, \tag{5.4.20}$$

which leads to

$$\begin{aligned} \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z) | N^m_{\ell^m(t)}\right] = f^m_{\ell^m(t)}(z) \int K_\sigma(w)dw - \left(f^m_{\ell^m(t)}\right)'(z) \int wK_\sigma(w)dw \\ + \int \int_0^1 (1-x)\left(f^m_{\ell^m(t)}\right)''(z-xw)w^2 K_\sigma(w)dx\ dw. \end{aligned} \tag{5.4.21}$$

As K_σ is a symmetric probability density function, we have

$$\int K_\sigma(w)dw = 1, \qquad \int wK_\sigma(w)dw = 0, \tag{5.4.22}$$

which leads to

$$\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right] = f^m_{\ell^m(t)}(z) + \int\int_0^1 (1-x)\left(f^m_{\ell^m(t)}\right)''(z-xw)w^2K_\sigma(w)dx\ dw. \tag{5.4.23}$$

By combining (5.4.18) and (5.4.23) we obtain the following bias expression

$$\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] - f^m_{\ell^m(t)}(z) = \tilde{\mathbb{E}}\left[\int\int_0^1 (1-x)\left(f^m_{\ell^m(t)}\right)''(z-xw)w^2K_\sigma(w)dx\ dw\right], \tag{5.4.24}$$

proving bound (5.4.16).

Concerning the variance, we apply the law of total variance [e.g. Wasserman, 2004, p.55]:

$$\tilde{\mathrm{Var}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] = \tilde{\mathbb{E}}\left[\tilde{\mathrm{Var}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right] + \tilde{\mathrm{Var}}\left[\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right]. \tag{5.4.25}$$

As in (5.4.19), we may express the conditional variance of $\hat{f}^m_{\ell^m(t)}(z)$ using the conditional variance of the kernel, which brings the first term in (5.4.25) to

$$\tilde{\mathbb{E}}\left[\tilde{\mathrm{Var}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right] = \tilde{\mathbb{E}}\left[\frac{1}{N^m_{\ell^m(t)}}\tilde{\mathrm{Var}}\left[K_\sigma(z-Z_{m,t})|N^m_{\ell^m(t)}\right]\right]. \tag{5.4.26}$$

Furthermore, the kernel variance can be developped as follows

$$\tilde{\mathrm{Var}}\left[K_\sigma(z-Z_{m,t})|N^m_{\ell^m(t)}\right] = \tilde{\mathbb{E}}\left[K_\sigma(z-Z_{m,t})^2|N^m_{\ell^m(t)}\right] - \tilde{\mathbb{E}}\left[K_\sigma(z-Z_{m,t})|N^m_{\ell^m(t)}\right]^2, \tag{5.4.27}$$

which is hence smaller than the first term, i.e. the kernels second moment. Using integral Taylor expansion of $f^m_{\ell^m(t)}$ around z (5.4.20) truncated to the first order, such a quantity can be written as follows:

$$\begin{aligned}
\tilde{\mathbb{E}}\left[K_\sigma(z-Z_{m,t})^2|N^m_{\ell^m(t)}\right] &= \int K_\sigma(z-\lambda)^2 f^m_{\ell^m(t)}(\lambda)d\lambda \\
&= \int K_\sigma(w)^2 f^m_{\ell^m(t)}(z-w)dw \\
&= f^m_{\ell^m(t)}(z)R(K_\sigma) - \left(f^m_{\ell^m(t)}\right)'(z)\underbrace{\int wK_\sigma(w)^2dw}_{=0} \\
&\quad + \int\int_0^1 (1-x)w^2\left(f^m_{\ell^m(t)}\right)''(z-xw)K_\sigma(w)^2dx\ dw,
\end{aligned} \tag{5.4.28}$$

where we used the fact that K_σ^2 is an even function. By using (5.4.26)-(5.4.27) and (5.4.28)

we get

$$\tilde{\mathbb{E}}\left[\tilde{\mathrm{Var}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right] \leq f^m_{\ell^m(t)}(z)\tilde{\mathbb{E}}\left[\frac{R(K_\sigma)}{N^m_{\ell^m(t)}}\right] + \frac{1}{2}\left\|\left(f^m_{\ell^m(t)}\right)''\right\|_\infty \tilde{\mathbb{E}}\left[\frac{\sigma^2_{K^2_\sigma}}{N^m_{\ell^m(t)}}\right]. \quad (5.4.29)$$

We still need to bound the second term in (5.4.25). We have

$$\begin{aligned}&\tilde{\mathrm{Var}}\left[\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right]\\ &\quad= \mathbb{E}\left[\left(\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right] - \underbrace{\tilde{\mathbb{E}}\left[\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right]}_{\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right]}\right)^2\right]. \end{aligned} \quad (5.4.30)$$

By denoting

$$G_m = \int\int_0^1 (1-x)\left(f^m_{\ell^m(t)}\right)''(z-xw)w^2K_\sigma(w)dx\ dw \quad (5.4.31)$$

and plugging (5.4.23)-(5.4.24) in (5.4.30), we obtain

$$\tilde{\mathrm{Var}}\left[\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right] = \tilde{\mathbb{E}}\left[\left(G_m - \tilde{\mathbb{E}}\left[G_m\right]\right)^2\right] \leq \tilde{\mathbb{E}}\left[G_m^2\right]. \quad (5.4.32)$$

Furthermore,

$$\begin{aligned}\tilde{\mathbb{E}}\left[G_m^2\right] &= \tilde{\mathbb{E}}\left[\left(\int\int_0^1 (1-x)\left(f^m_{\ell^m(t)}\right)''(z-xw)w^2K_\sigma(w)dx\ dw\right)^2\right],\\ &\leq \frac{1}{4}\left\|\left(f^m_{\ell^m(t)}\right)''\right\|^2_\infty \tilde{\mathbb{E}}\left[\left(\int w^2K_\sigma(w)dw\right)^2\right],\end{aligned} \quad (5.4.33)$$

which leads to the following bound of the second term in (5.4.25)

$$\tilde{\mathrm{Var}}\left[\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)|N^m_{\ell^m(t)}\right]\right] \leq \frac{1}{4}\left\|\left(f^m_{\ell^m(t)}\right)''\right\|^2_\infty \tilde{\mathbb{E}}\left[(\sigma^2_{K_\sigma})^2\right], \quad (5.4.34)$$

and proves inequality (5.4.17). ∎

As a direct consequence, we get the following bounds for the simpler case where kernel and bandwidth are deterministic:

Lemma 5.4.5 *If $\sigma = \sigma_m$ depends on m and K is fixed, both deterministic (do not depend on the sample), then under assumption 5.4.1:*

$$\left|\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] - f^m_{\ell^m(t)}(z)\right| \leq \frac{1}{2}\left\|\left(f^m_{\ell^m(t)}\right)''\right\|_\infty \sigma^2_K\sigma^2_m, \quad (5.4.35)$$

$$\begin{aligned}\tilde{\mathrm{Var}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] &\leq \tilde{\mathbb{E}}\left[\frac{1}{N^m_{\ell^m(t)}}\right]\left(\frac{f^m_{\ell^m(t)}(z)R(K)}{\sigma_m} + \left\|\left(f^m_{\ell^m(t)}\right)''\right\|_\infty \frac{\sigma^2_{K^2}\sigma_m}{2}\right)\\ &\quad + \frac{1}{4}\left\|\left(f^m_{\ell^m(t)}\right)''\right\|^2_\infty (\sigma^2_K\sigma^2_m)^2.\end{aligned} \quad (5.4.36)$$

Proof Obtained directly by using (5.4.5)-(5.4.7) together with lemma 5.4.4. ∎

We know from lemmas 5.3.7 and 5.4.3 that, under assumptions 5.3.4 and 5.4.1:

$$\lim_{m\to\infty} f^m_{\ell^m(t)}(z) = f_t(z) < \infty, \tag{5.4.37}$$

$$\lim_{m\to\infty} \left(f^m_{\ell^m(t)}\right)''(z) = f_t''(z) < \infty. \tag{5.4.38}$$

Hence, by looking at expressions (5.4.35) and (5.4.36), it seems clear that the convergence of the bias and the variance to zero will strongly depend on the asymptotics of $\tilde{\mathbb{E}}\left[1/N^m_{\ell^m(t)}\right]$. This motivates the following lemma:

Lemma 5.4.6 *If assumption 5.3.2 to 5.3.4 hold, as* $m \to \infty$

$$\tilde{\mathbb{E}}\left[\frac{1}{N^m_{\ell^m(t)}}\right] = O\left(\frac{1}{nmP^m_{\ell^m(t)}}\right) = O\left(\frac{1}{mb_m}\right). \tag{5.4.39}$$

Proof We recall that $N^m_{\ell^m(t)} \sim \mathcal{B}(nm, P^m_{\ell^m(t)})$, with $P^m_{\ell^m(t)} = \int_{B^m_{\ell^m(t)}} \nu(t)dt$. According to the main theorem proved in Cribari-Neto et al. [2000],

$$\begin{aligned} S_1 := \mathbb{E}\left[\frac{1}{1+N^m_{\ell^m(t)}}\right] &= \sum_{k=0}^{nm} \binom{nm}{k} \frac{(P^m_{\ell^m(t)})^k (1-P^m_{\ell^m(t)})^{nm-k}}{1+k} \\ &= O\left(\frac{1}{nmP^m_{\ell^m(t)}}\right). \end{aligned} \tag{5.4.40}$$

As noted in remark 5.4.2, $\hat{f}^m_{\ell^m(t)}$ is not defined for $N^m_{\ell^m(t)} = 0$. This motivates the conditioning of the variable $\frac{1}{N^m_{\ell^m(t)}}$ by the event $N^m_{\ell^m(t)} > 0$ and we have

$$S_0 := \mathbb{E}\left[\frac{1}{N^m_{\ell^m(t)}} \middle| N^m_{\ell^m(t)} > 0\right] = \sum_{k=1}^{nm} \binom{nm}{k} \frac{(P^m_{\ell^m(t)})^k (1-P^m_{\ell^m(t)})^{nm-k}}{k}. \tag{5.4.41}$$

By computing the difference between the S_1 and S_0 we obtain the following bound:

$$\begin{aligned} S_0 - S_1 &= -\binom{nm}{0}(1-P^m_{\ell^m(t)})^{nm} + \sum_{k=1}^{nm} \binom{nm}{k} \frac{(P^m_{\ell^m(t)})^k (1-P^m_{\ell^m(t)})^{nm-k}}{k(k+1)} \\ &\leq \sum_{k=1}^{nm} \binom{nm}{k} \frac{(P^m_{\ell^m(t)})^k (1-P^m_{\ell^m(t)})^{nm-k}}{k+1} \leq S_1. \end{aligned} \tag{5.4.42}$$

We conclude that $S_0 \leq 2S_1$ and hence

$$\mathbb{E}\left[\frac{1}{N^m_{\ell^m(t)}} \middle| N^m_{\ell^m(t)} > 0\right] = O\left(\frac{1}{nmP^m_{\ell^m(t)}}\right). \tag{5.4.43}$$

Finally, the last equality in (5.4.39) comes from the fact that $P^m_{\ell^m(t)} \geq \nu_- b_m$. ∎

In conclusion, we can now prove theorem 5.4.2 stating that the kernel marginal density estimator will be consistent in expected squared-error (which implies convergence in probability stated in theorem (5.3.6)).

Proof of Theorem 5.4.2 By (5.4.35) and (5.4.10), the bias of $\hat{f}^m_{\ell^m(t)}(z)$ converges to 0:

$$\lim_{m\to\infty} \left| \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] - f^m_{\ell^m(t)}(z) \right| = 0. \tag{5.4.44}$$

Similarly, as σ_m converges to 0, the two last terms in (5.4.36) shrink. Concerning the first term, we can conclude from lemma 5.4.6 and from condition (5.4.10) that

$$\lim_{m\to\infty} \tilde{\mathbb{E}}\left[\frac{1}{N^m_{\ell^m(t)}}\right] \frac{1}{\sigma_m} = 0, \tag{5.4.45}$$

which means that the variance of $\hat{f}^m_{\ell^m(t)}(z)$ also converges to 0:

$$\lim_{m\to\infty} \tilde{\mathbb{V}\text{ar}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] = 0. \tag{5.4.46}$$

From the bias-variance decomposition of the expected squared-error between $\hat{f}^m_{\ell^m(t)}(z)$ and $f^m_{\ell^m(t)}(z)$ we have

$$\begin{aligned}
\tilde{\mathbb{E}}\left[(\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z))^2\right] &= \tilde{\mathbb{E}}\left[\left(\hat{f}^m_{\ell^m(t)}(z) - \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right]\right.\right. \\
&\qquad \left.\left. + \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] - f^m_{\ell^m(t)}(z)\right)^2\right], \\
&= \tilde{\mathbb{E}}\left[\left(\hat{f}^m_{\ell^m(t)}(z) - \tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right]\right)^2\right] \\
&\qquad + \tilde{\mathbb{E}}\left[\left(\tilde{\mathbb{E}}\left[\hat{f}^m_{\ell^m(t)}(z)\right] - f^m_{\ell^m(t)}(z)\right)^2\right].
\end{aligned}$$

Furthermore, by Jensen inequality we have that,

$$\frac{1}{2}\tilde{\mathbb{E}}\left[(\hat{f}^m_{\ell^m(t)}(z) - f_t(z))^2\right] \leq \tilde{\mathbb{E}}\left[(\hat{f}^m_{\ell^m(t)}(z) - f^m_{\ell^m(t)}(z))^2\right] + (f^m_{\ell^m(t)}(z) - f_t(z))^2. \tag{5.4.47}$$

Finally, by using lemma 5.3.7 in conjunction with (5.4.44), (5.4.46) and (5.4.47), we obtain that both terms in (5.4.47) tend to 0, leading to result (5.4.11). ∎

5.4.3 Self-consistent kernel estimator

Motivation

In general, the main drawback of kernel density estimators (5.4.1) lies on the subjective choice of the kernel K and bandwidth H. However, it is well-known folklore in the density estimation literature [see e.g. Wasserman, 2004, chapter 20.3] that KDE's accuracy is not really sensitive to the choice of K and depends mainly on the bandwidth H used (see

figure 5.5).

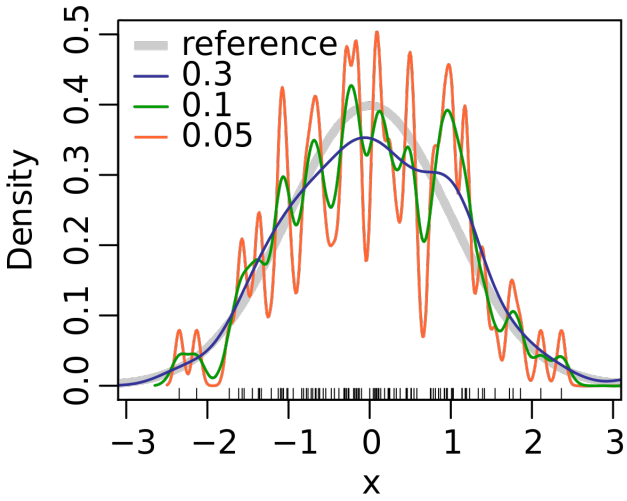


Figure 5.5 – Example of kernel density estimation using different bandwidth. The sample was drawn from a standard Gaussian distribution.

Several rules and heuristics have been suggested since then to choose such a parameter, but they are usually based on quite strict assumptions, such as Silverman's rule of thumb for the estimation of a 1-dimensional Gaussian density [Silverman, 1986]. Another possibility is to select H through cross-validation [Stone, 1984] but this approach is computationally intensive, specially if d is larger than 1. For these reasons, we decided to consider a similar method proposed by Bernacchia and Pigolotti [2011], called the *self-consistent density estimator*. It consists indeed in a KDE whose kernel incorporates the bandwidth and is learned directly from the data, hence not requiring any parameter tuning. Its derivation is based on the use of a fixed-point equation to approximate the optimal kernel estimator in the sense of the Mean Integrated Squared-Error (MISE). The obtained estimator takes the form of the Fourier transform of the following characteristic function estimator:

$$\hat{\Phi}_{sc}(s) := \frac{N\Delta(s)}{2(N-1)}\left(1+\sqrt{1-\frac{(\Delta_N^{min})^2}{|\Delta(s)|^2}}\right)\mathbf{1}_{A_N}(s),$$

where s is the Fourier variable, N is the number of training observations $\{x_k\}_{k=1}^N$, Δ is the empirical characteristic function

$$\Delta(s) := \frac{1}{N}\sum_{k=1}^{N} e^{ix_k\cdot s}, \qquad \Delta_N^{min} := \sqrt{\frac{4(N-1)}{N^2}}, \tag{5.4.48}$$

$i=\sqrt{-1}$ and $\mathbf{1}_{A_N}$ denotes the indicator function over an arbitrary subset $A_N \subset \mathbb{S}_N$ of the frequencies in

$$\mathbb{S}_N := \left\{s : |\Delta(s)|^2 \geq (\Delta_N^{min})^2\right\}. \tag{5.4.49}$$

Bernacchia and Pigolotti [2011] proved for 1D data that, under mild assumptions on A_N, the self-consistent estimator converges almost-surely to the true density f as the number of observations N grows and is hence (strongly) consistent. This result is summarized in the following theorem:

Theorem 5.4.7 ([Bernacchia and Pigolotti, 2011]) *Let $[-t^*; t^*] = A_N \subset \mathbb{S}_N$ be an interval of frequencies in $\mathbb{R}$. Assuming that f is $L^2(E, \mathbb{R}_+)$ and its Fourier transform $\Phi = \mathcal{F}[f]$ is $L^1(\mathbb{R}, \mathbb{R}^+)$, if the bounds of A_N are such that*

$$\lim_{N\to\infty} t^* = \infty, \qquad \lim_{N\to\infty} \frac{t^*}{\sqrt{N}} = 0, \tag{5.4.50}$$

then the density estimator defined by

$$\hat{f}^N_{sc}(x) := \mathcal{F}^{-1}[\hat{\Phi}_{sc}](x), \quad \forall x \in E \tag{5.4.51}$$

converges almost surely to f as N tends to infinity:

$$\mathbb{P}\left(\lim_{N\to\infty} \hat{f}^N_{sc}(x) = f(x)\right) = 1, \quad \forall x \in E. \tag{5.4.52}$$

It has been demonstrated through extensive numerical experiments in Bernacchia and Pigolotti [2011], O'Brien et al. [2014], O'Brien et al. [2016] that the self-consistent estimator achieves state-of-the-art MISE accuracy for many types of underlying densities. Furthermore, modern implementations of this estimator proposed by O'Brien et al. [2014], O'Brien et al. [2016]are shown to be several times faster to compute than a regular KDE. This is achieved thanks to the smart use of the Non Uniform Fast Fourier Transform [Greengard and Lee, 2004] to compute the empirical characteristic function Δ. This property is particularly important in our case because of the potentially large number of trainings needed to compute the EMML, which is equal to the number of bins of $\mathbb{T}$'s partition.

For all these reasons, all EMML numerical experiments presented in section 5.5 make use of this density estimator. More details concerning its derivation and implementation are given in the following subsections.

Optimal Kernel Density Estimator

Let $S^N = \{z_k\}_{k=1}^N \subset E$ be a sample of N observations drawn from a common density function f. We suppose that $f \in L^2(E, \mathbb{R}_+)$ and that $N > 0$ is deterministic. We consider a kernel estimator of f

$$\hat{f}(z) := \frac{1}{N}\sum_{k=1}^{N} K(z - z_k), \qquad \forall z \in E, \tag{5.4.53}$$

where $K : \mathbb{R}^d \to \mathbb{R}$ is a smoothing kernel in $L^2(\mathbb{R}^d, \mathbb{R})$. One can interpret (5.4.53) as a kernel density estimator with an implicit bandwidth $H \in GL_d(\mathbb{R}_+)$ hidden inside the expression of the kernel function $K(z) = \tilde{K}(H^{-1}z)/\det H$, $\forall z \in \mathbb{R}^d$.

Denote by $\mathbb{E}$ the expectation relative to the random sample S^N, defined for any deterministic function $\varphi : E^N \to \mathbb{R}$ by

$$\mathbb{E}\left[\psi(S^N)\right] := \int_E \dots \int_E \varphi(z_1, \dots, z_N) f(z_1) \dots f(z_N) dz_1 \dots dz_N. \tag{5.4.54}$$

In this context, a common quality measure of density estimators is the Mean Integrated Squared Error:

$$MISE := \mathbb{E}\left[\int_E (\hat{f}(z) - f(z))^2 dz\right]. \tag{5.4.55}$$

As we will show, it becomes relatively easy to minimize such a criterion with regard to the choice of the kernel K once we've shifted it to the Fourier domain. Hence, for any function $v \in L^2(\mathbb{R}^d, \mathbb{R})$, we define its Fourier transform hereafter with the following convention

$$\mathcal{F}[v](s) := \int_{\mathbb{R}^d} v(z) e^{iz\cdot s} dz, \qquad \forall s \in \mathbb{R}^d, \tag{5.4.56}$$

where $i = \sqrt{-1}$, its inverse being defined by

$$\mathcal{F}^{-1}[v](z) := \frac{1}{2\pi} \int_{\mathbb{R}^d} v(s) e^{-iz\cdot s} ds, \qquad \forall z \in \mathbb{R}^d. \tag{5.4.57}$$

As $f, \hat{f} \in L^2$, Plancherel's theorem gives that

$$MISE = \frac{1}{2\pi} \mathbb{E}\left[\int_{\mathbb{R}^d} |\hat{\Phi}(s) - \Phi(s)|^2 ds\right], \tag{5.4.58}$$

where $\Phi := \mathcal{F}[f]$, usually called the *characteristic function*, and $\hat{\Phi} := \mathcal{F}[\hat{f}]$. By noticing that $\hat{f}$ can be seen as the convolution between the kernel and a sum of Dirac functions centered on the data points

$$\hat{f}(z) = \left(K * \left(\frac{1}{N} \sum_{k=1}^N \delta_{z_k}\right)\right)(z), \tag{5.4.59}$$

it follows that

$$\hat{\Phi}(s) = \kappa(s)\Delta(s), \tag{5.4.60}$$

where

$$\kappa(s) := \mathcal{F}[K](s) \tag{5.4.61}$$

$$\Delta(s) := \mathcal{F}\left[\frac{1}{N} \sum_{k=1}^N \delta_{z_k}\right](s) = \frac{1}{N} \sum_{k=1}^N e^{iz_k\cdot s} \in \mathbb{C}. \tag{5.4.62}$$

The function Δ is commonly called the *empirical characteristic function* (ECF).

Plugging (5.4.60) in the MISE expression (5.4.58) and expanding the square gives:

$$\begin{aligned} MISE = \frac{1}{2\pi} \int_{\mathbb{R}^d} \Big[& |\kappa|^2 \, \mathbb{E}\left[|\Delta|^2\right] + |\Phi|^2 \\ & -\kappa(\mathbb{E}\left[\Delta\right] \Phi^* + \mathbb{E}\left[\Delta^*\right] \Phi) \Big] ds, \end{aligned} \tag{5.4.63}$$

the arguments s being omitted for lighter notation and c^* denoting the complex conjugate of any $c \in \mathbb{C}$. Furthermore, as shown in [Tsybakov, 2008, section 1.3, lemma 1.2] for example, the ECF is an unbiased estimator of the characteristic function

$$\begin{aligned} \mathbb{E}\left[\Delta(s)\right] &= \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\left[e^{is \cdot z_k}\right] \\ &= \frac{1}{N} \sum_{k=1}^{N} \int_{-\infty}^{+\infty} e^{is \cdot z_k} f(z_k) dz_k \\ &= \Phi(s), \end{aligned} \tag{5.4.64}$$

and its second moment is

$$\mathbb{E}\left[|\Delta(s)|^2\right] = \mathbb{E}\left[\Delta(s)\Delta(s)^*\right] = \mathbb{E}\left[\Delta(s)\Delta(-s)\right] \tag{5.4.65}$$

$$= \mathbb{E}\left[\frac{1}{N^2} \sum_{j,k:j\neq k} e^{is \cdot (z_j - z_k)}\right] + \frac{1}{N} \tag{5.4.66}$$

$$= \frac{1}{N^2} \sum_{j,k:j\neq k} \mathbb{E}\left[e^{is \cdot z_j} e^{-is \cdot z_k}\right] + \frac{1}{N} \tag{5.4.67}$$

$$= \frac{1}{N^2} \sum_{j,k:j\neq k} \mathbb{E}\left[e^{is \cdot z_j}\right] \mathbb{E}\left[e^{-is \cdot z_k}\right] + \frac{1}{N} \tag{5.4.68}$$

$$= \frac{1}{N^2} \sum_{j,k:j\neq k} \Phi(s)\Phi(-s) + \frac{1}{N} \tag{5.4.69}$$

$$= \frac{N-1}{N} \Phi(s)\Phi(-s) + \frac{1}{N} \tag{5.4.70}$$

$$= \frac{N-1}{N} |\Phi(s)|^2 + \frac{1}{N}. \tag{5.4.71}$$

Passing from line (5.4.67) to (5.4.68) is based on the assumption that the random variables $\{z_k\}_{k=1}^N$ are independent.

Replacing (5.4.64) and (5.4.71) in (5.4.63), we obtain

$$MISE = \frac{1}{2\pi} \int_{\mathbb{R}^d} \frac{|\kappa|^2}{N} (1 - |\Phi|^2) + |\Phi|^2 (1 - \kappa)^2 ds. \tag{5.4.72}$$

As initially shown in Watson and Leadbetter [1963], expression (5.4.72) can be minimized with respect to the transformed kernel κ. Indeed, for any $s \in \mathbb{R}^d$, we get the following first order optimality condition by differentiating the quadratic integrand in (5.4.72) relative

to $\kappa(s)$:

$$\frac{1}{N}\kappa(1-|\Phi|^2)-|\Phi|^2(1-\kappa)=0, \tag{5.4.73}$$

leading to the *optimal transformed kernel*

$$\kappa_{opt}(s) := \frac{N}{N-1+|\Phi(s)|^{-2}}, \qquad \forall s \in \mathbb{R}^d. \tag{5.4.74}$$

Hence the optimal density estimator relative to the MISE is given by

$$\boxed{\hat{f}_{opt}(z) = \mathcal{F}^{-1}[\hat{\Phi}_{opt}](z),} \tag{5.4.75}$$

where

$$\boxed{\hat{\Phi}_{opt}(s) = \kappa_{opt}(s)\Delta(s) = \frac{N\Delta(s)}{N-1+|\Phi(s)|^{-2}}.} \tag{5.4.76}$$

Self-Consistent Estimator

The practical problem with estimator (5.4.76) is that it depends on the true characteristic function Φ, which is unknown. Hence, the solutions to the fixed-point equation (5.4.77) was suggested by Bernacchia and Pigolotti [2011] to approximate $\hat{\Phi}_{opt}$:

$$\hat{\Phi}^N_{sc} = \frac{N\Delta}{N-1+|\hat{\Phi}^N_{sc}|^{-2}}. \tag{5.4.77}$$

This is justified by the fact that the optimal estimator $\hat{\Phi}_{opt}$ should be very close to the true characteristic function Φ, as illustrated by the MISE criterion (5.4.58).

Equation (5.4.77) can be transformed into a second order equation in $|\hat{\Phi}^N_{sc}|$,

$$(N-1)|\hat{\Phi}^N_{sc}|^2 - N|\Delta||\hat{\Phi}^N_{sc}| + 1 = 0, \tag{5.4.78}$$

which admits a solution in $\mathbb{R}_+$ provided that

$$|\Delta(s)|^2 \geq (\Delta^{min}_N)^2 := \frac{4(N-1)}{N^2}. \tag{5.4.79}$$

When inequality (5.4.79) holds, the two possible solutions of (5.4.78) are

$$|\hat{\Phi}^+| := \frac{N|\Delta|}{2(N-1)}\left(1+\sqrt{1-\frac{(\Delta^{min}_N)^2}{|\Delta|^2}}\right), \tag{5.4.80}$$

$$|\hat{\Phi}^-| := \frac{N|\Delta|}{2(N-1)}\left(1-\sqrt{1-\frac{(\Delta^{min}_N)^2}{|\Delta|^2}}\right). \tag{5.4.81}$$

After some analysis, we can show that $\hat{\Phi}^+$ is a stable fixed-point, while $\hat{\Phi}^-$ is unstable. Bernacchia and Pigolotti [2011] suggest to keep only the stable one, which brings us to

the self-consistent estimator of the characteristic function:

$$\boxed{\hat{\Phi}^N_{sc}(s) := \frac{N\Delta(s)}{2(N-1)}\left(1+\sqrt{1-\frac{(\Delta^{min}_N)^2}{|\Delta(s)|^2}}\right)\mathbf{1}_{A_N}(s),} \tag{5.4.82}$$

where $\mathbf{1}_{A_N}$ denotes the indicator function over an arbitrary subset $A_N \subset \mathbb{S}_N$ of the frequencies in

$$\mathbb{S}_N := \left\{s : |\Delta(s)|^2 \geq (\Delta^{min}_N)^2\right\}. \tag{5.4.83}$$

Hence our new density estimator writes

$$\boxed{\hat{f}^N_{sc}(z) := \mathcal{F}^{-1}\left[\hat{\Phi}^N_{sc}\right](z), \qquad \forall z \in E.} \tag{5.4.84}$$

Practical Considerations

Heuristics for choosing A_N were proposed in Bernacchia and Pigolotti [2011], O'Brien et al. [2014] for the univariate case and in O'Brien et al. [2016] in a multivariate setting.

One practical problem with the self-consistent estimator is that $\hat{f}^N_{sc} = \mathcal{F}^{-1}[\hat{\Phi}^N_{sc}]$ is not lower-bounded by zero. This can be corrected by translating $\hat{f}^N_{sc}$ downwards until the positive part integrates to one and then setting the negative part to 0. Indeed, it was proven by Glad et al. [2003] that such a transformation induces no cost in terms of MISE accuracy.

Another practical drawback with estimator $\hat{\Phi}^N_{sc}$ is that the direct computation of the empirical characteristic function Δ can be expensive: $O(N \cdot M)$ exponential evaluations, where M is the number of frequency points $s \in \mathbb{R}^d$. Noting from definition (5.4.62) that the expression of Δ is equivalent to some Discrete Fourier Transform

$$\Delta(s) = \frac{1}{N}\sum_{k=1}^{N} a_k e^{is\cdot z_k}, \tag{5.4.85}$$

where the Fourier coefficients a_k are all equal to 1, the idea of using the Fast Fourier Transform algorithm (FFT) proposed by Cooley and Tukey [1965] seems natural. However, the latter only applies to the case of uniformly spaced data, which is not the case of $\{z_k\}_{k=1}^N$. For this reason, O'Brien et al. [2014] proposed to use an implementation of Nonuniform Fast Fourier Transform (NUFFT) developped by Greengard and Lee [2004].

It consists in interpolating the original data points $\{z_k\}_{k=1}^N$ on a new equispaced grid $\{\tilde{z}_j\}_{j=1}^{\tilde{N}_\ell}$ by using another Gaussian kernel density estimator:

$$\begin{aligned}\tilde{f}(\tilde{z}_j) &:= \frac{1}{N}\sum_{k=1}^{N} K_G(z_k - \tilde{z}_j),\\ &= K_G * \left(\frac{1}{N}\sum_{k=1}^{N}\delta_{z_k}\right)(\tilde{z}_j),\end{aligned} \tag{5.4.86}$$

with $K_G(z) := \exp\left(-\frac{z^2}{\sigma^2}\right)$, $\forall z \in \mathbb{R}^d$, and $\sigma \in \mathbb{R}_+^*$. The FFT can than be used to approximate $\tilde{\Phi}(s) := \mathcal{F}[\tilde{f}](s)$, and by dividing it by the transformed Gaussian kernel $\kappa_G(s) := \mathcal{F}[K_G](s)$, we obtain the ECF evaluation:

$$\Delta(s) = \tilde{\Phi}(s) \cdot [\kappa_G(s)]^{-1} . \tag{5.4.87}$$

As computing $\{\tilde{f}(\tilde{z}_j)\}_{j=1}^{\tilde{N}_\ell}$ still takes $O(\tilde{N}_\ell \cdot N)$ operations, Dutt and Rokhlin [1993] suggested to use only $N^c < N$ surrounding points from $\{z_k\}_{k=1}^N$ to evaluate each new grid point $\tilde{z}_j$. We obtain an overall complexity of $O(N^c \cdot \tilde{N}_\ell + M \log M)$ which, in the case where $N^c < M \leq N$, is better than the original DFT formulation $O(N \cdot M)$. The analysis conducted in Greengard and Lee [2004] indicates that simple precision can be achieved in (5.4.87) by normalizing the data $\{z_k\}_{k=1}^N$ to the range $[0, 2\pi]$ and setting $2N^c = 12$, $\tilde{N}_\ell = 2N_\ell$ and $\sigma^2 = 24/N^2$. Hence, for a desired precision, this step of the algorithm introduces no additional hyperparameter to be tuned (see Greengard and Lee [2004] for the double-precision settings).

5.5 Application to the Assessment of Optimized Aircraft Trajectories

5.5.1 Experiments Motivation

In this section we illustrate our approach on real data recorded from $m = 424$ flights of the same medium haul aircraft $\mathcal{T} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^m\}$, which corresponds to 334 531 observation points. These trajectories are used to estimate the differential system describing the aircraft dynamics. Then, by numerically solving an optimal control problem defined using the estimated aircraft dynamics, a new trajectory $\boldsymbol{y}$ is obtained, supposed to minimize the overall fuel consumption for some future flight [see e.g. Rommel et al., 2017].
Note that:

1. The dynamics model is not guaranteed to be valid outside of the region occupied by the data used to estimate it. Hence, it is natural to want the optimized trajectory to avoid going too far from its validity region.

2. Furthermore, it is desirable for the proposed trajectory not to be too unusual compared to standard climb profiles for better acceptance by the pilots and Air Traffic Control.

The two previous points motivate the need for an indicator of closeness between the optimized trajectory and the set of recorded flights.

5.5.2 Experiments Design

Training set The training data used for our experiments were extracted from the *Quick Access Recorder* (QAR) of the same aircraft, whose sampling rate is of one measurement per second. We only used the recordings of 5 variables, which are the altitude h, the true airspeed V, the path angle γ, the angle of attack α and the throttling position N_1. These variables are not all directly accessible and some were computed from other measurements using standard formulas from flight mechanics [see e.g. Rommel et al., 2017]. Only the portion corresponding to the climb phase of these signals was kept for our experiments, i.e. data corresponding to altitudes between 1524 m = 5000 ft and the *top of climb* (cruise altitude, specific to each flight). The training set of curves obtained by the described procedure is displayed on figure 5.6a.

Test set In order to evaluate the estimated mean marginal likelihood on relevant examples, the following test flights were considered:

1. 50 real flights, extracted from the training set before training;
2. 50 simulated trajectories which were optimized as described in section 5.5.1, with constraints keeping the resulting speed V and N_1 between reasonable operational bounds;
3. and another 50 simulated trajectories optimized without the operational constraints.

We evaluated the likelihood of these 150 test trajectories using the EMML and the competing methods described in the following section in order to assess and compare their discriminative power and computation efficiency.

5.5.3 Alternate Approaches Based on Standard Methods

The problem of quantifying how close a newly observed random curve is with respect to a set of observations from the same stochastic process has not been treated by the statistical learning literature to our knowledge. However, this problem is related to other more standard approaches from Functional Data Analysis and Conditional Density Estimation, which could be adapted quite straightforwardly for this purpose. For this reason, we discuss in the following paragraphs the characteristics of two of these other existing methods, before comparing them to our approach in the numerical results of section 5.5.5.

Functional Principal Components Analysis

Functional Principal Components Analysis (FPCA) is a standard tool in FDA capable of building a small number of descriptors which summarize the structure of a set of random functions. As explained for example in Nicol [2013], this dimension reduction method can

be used to project the train set of infinite-dimensional random trajectories into a finite (low) dimensional space.

Following the same reasoning used to derive the MML, our idea here consists in estimating the density function of these low dimensional representations of the training set. Then, after projecting the new trajectory $\boldsymbol{y}$ into the same descriptors, we can evaluate the density estimate at it and obtain an approximation of its likelihood.

Least-Squares Conditional Density Estimation

From a completely different point of view, we could forget for a moment that we are considering a random process Z and look at (T, Z_T) as a pair of standard random variables valued on the finite dimensional space $\mathbb{T} \times E$. In this case, we could see the marginal densities f_t as the conditional probability density functions

$$f_{Z_T|T}(t,z) = \frac{f_{(T,Z_T)}(t,z)}{f_T(t)}, \qquad (t,z) \in \mathbb{T} \times E. \tag{5.5.1}$$

We could hence estimate (5.5.1) at the observed points of the new trajectory $\boldsymbol{y}$ and use them to compute the EMML indicator (5.3.8). It is however well-known in density ratio estimation that approximating $f_{(T,Z_T)}(t,z)$ and $f_T(t)$ separately before building the ratio in (5.5.1) is not a good idea because it magnifies the errors. For this reason, Sugiyama et al. [2010] proposed to use a linear model for this purpose

$$f_{Z_T|T}(t,z) = \theta^\top \phi(t,z), \tag{5.5.2}$$

where $\theta = (\theta_1, \ldots, \theta_p)$ is a vector of scalar parameters and $\phi(t,z) = (\phi_1(t,z), \ldots, \phi_p(t,z))$ is a family of nonnegative basis functions. The parameters θ are then chosen so as to minimize a L^2-penalized least-squares criterion, which is shown to have a closed-form solution. This method was coined *Least-Squares Conditional Density Estimation* (LSCDE) by the authors, and is also known as *Unconstrained Least-Squares Importance Fitting* (uLSIF) in the density ratio estimation literature [Kanamori et al., 2009]. The extensive numerical results presented in Sugiyama et al. [2010] indicate that this approach have state-of-the-art accuracy in conditional density estimation.

5.5.4 Algorithms Settings

For all the methods tested, the altitude h played the role of "time". This is a natural assumption made when optimizing the climb profile of a civil airliner, since the altitude is an increasing function of the time and every other variable depends on it. This allowed us to reduce the dimension of our problem from 5 to 4.

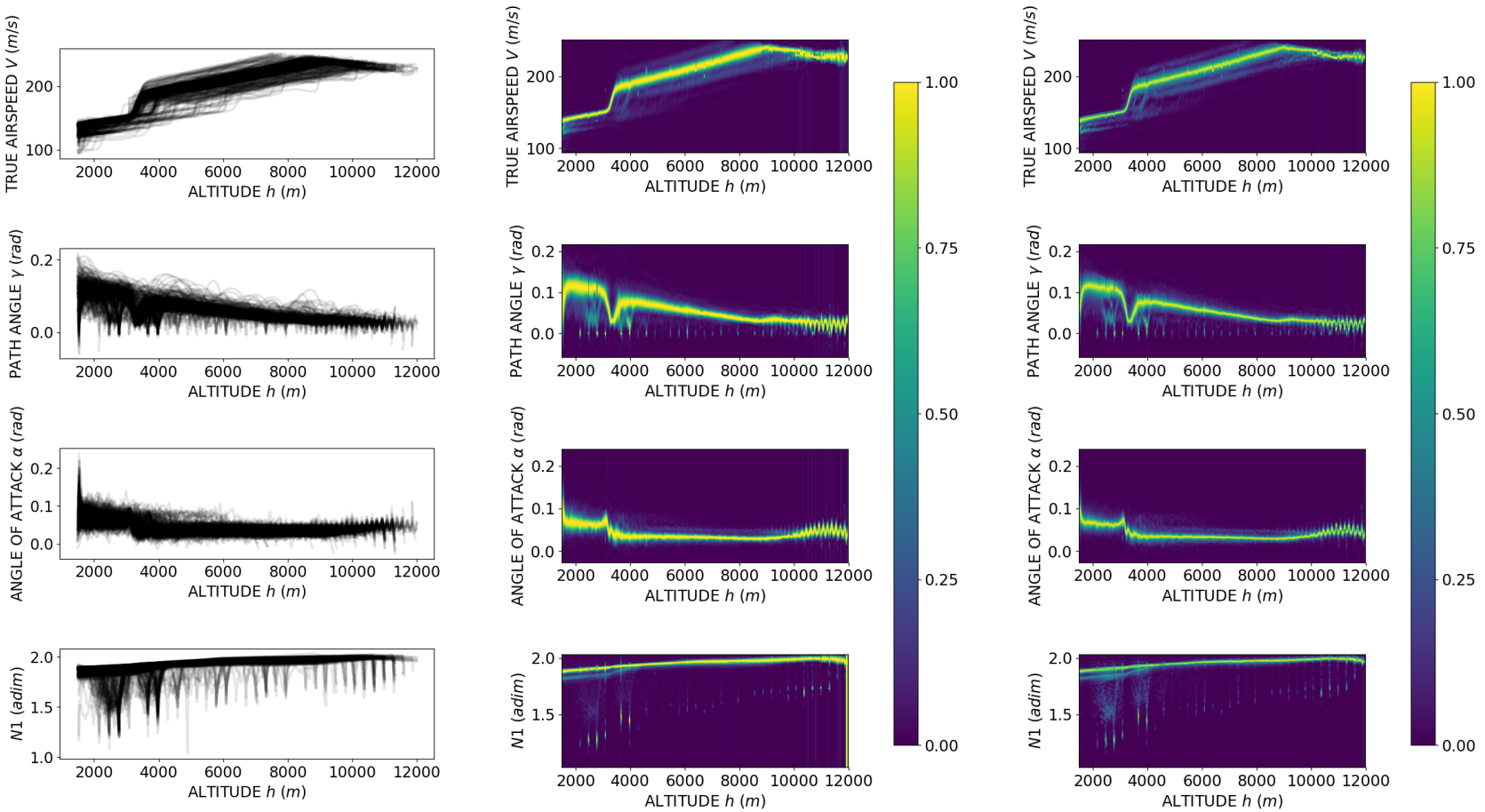
MML with Self-Consistent Kernel Estimator settings The python library fastKDE [O'Brien et al., 2014, O'Brien et al., 2016] was used to compute the marginal densities from the bins data. It contains the implementation of the Self-Consistent kernel estimator described in section 5.4.3. The precision of the density estimations were set to single. The *confidence levels* were approximated by numerical integration using the trapezoidal rule over a fine grid of approximately 300 points per bin.

Concerning bin sizes, we chose to use an uneven partition in our experiments. The reason for this are the *climb-steps* visible in the trajectories between 3000 and 4000 m, which correspond to phases during which the aircraft decreases considerably its ascent speed, leading to slowly increasing altitudes. Such behaviors translate into rapidly increasing speeds V with respect to the altitude, as well as into plummeting values of γ and N_1 (see figure 5.6a). This brought us to consider tighter bins around these climb-step altitudes:

- between 1524 and 3000m and between 4000 and 12000m, we partitioned the altitudes homogeneously into bins of size $b_m^{(1)} = 21\text{m} \simeq 1/\sqrt{m}$ (which satisfies assumption 5.3.4);
- between 3000 and 4000m, we used a bin size twice smaller $b_m^{(2)} = 10\text{m} \simeq b_m^{(1)}/2$;

FPCA settings Concerning the Functional Principal Components Analysis method, all training and testing flights were resampled on an equispaced grid of altitudes, using a step size of 5m. The trajectories were then centered and decomposed into a basis of 128 cubic B-splines. The SVD decomposition was carried using the `PCA` class from `scikit-learn` python library [Pedregosa et al., 2011]. We kept 4 components for each variable (V, γ, α and N_1), which was enough to explain more than $90\%, 65\%, 60\%$ and 75% of their respective variance. A Gaussian mixture model was used to estimate the density of the training trajectory scores obtained by the projection into the principal functions. The model was trained using a standard EM algorithm, implemented in `scikit-learn` as well. The number of components was selected between 1 and 5 using the Bayesian information criterion (BIC).

LS-CDE settings For the Least-Squares Conditional Density Estimation, the python package `densratio` [Makiyama, 2016], implementing the uLSIF method from Kanamori et al. [2009] was used and adapted. The basis functions chosen were $p = 100$ Gaussian kernels with the same variance σ and different centers. These centers were randomly drawn from the training data points using a uniform distribution, as suggested in Sugiyama et al. [2010]. The variance σ, as well as the L^2 penalty weight λ needed for minimizing the least-squares criterion were selected by cross-validation.



(a) Training set of 424 trajectories

(b) Normalized density heatmap

(c) Confidence level heatmap

Figure 5.6 – Estimated marginal densities using two types of scaling functions

5.5.5 Results and Comments

Figures 5.6b and 5.6c show heatmaps encoding the estimated marginal likelihoods using the normalized density (5.3.3) and the confidence level (5.3.4). We notice that both figures are similar and seem to catch the shape of the plot from figure 5.6a, including the multi-modalities visible for N_1 below $h = 4000$m for example.

Table 5.1a contains the estimated Mean Marginal Likelihood scores averaged over each test flight category. The training was carried on each dimension separately and the average total training time was of 5 seconds on a laptop (2.30 GHz, 7.7 GB). First of all, we notice for both types of scaling functions that the test flight categories are nicely separated by three really distinct ranges of scores. Furthermore, the higher differences between the two types of optimized flights can be seen for the variables V and N_1, which makes sense since those are the variables which are left free for *Opt2* flights and constrained for *Opt1* flights. As expected from figures 5.6b and 5.6c, the performances of both confidence level and normalized density based MML are comparable in terms of discrimination power and seem adequate for the task of assessing optimized aircraft climb trajectories.

Table 5.1b contains the estimated Mean Marginal Likelihood scores in a 2-dimensional setting, where the pairs (V, γ) and (α, N_1) have been treated together. The average training time needed here was 16 times larger than in the 1D case, i.e. 1 minute 20 seconds. The scores observed are globally really low and the test flight categories are not well separated. Moreover, we expected to obtain large scores for the real flights, since we used the marginal densities of their category to build the criterion, but this is not the case here. We conclude that the MML criteria based on the self-consistent estimator does not work so well in higher dimension and we suspect this to be related to the curse of dimensionality. Indeed, it is well-known [see e.g. Wasserman, 2004, chapter 21.3] that as the dimension grows, the amount of data needed to attain a given accuracy with kernel density estimators skyrockets, which may explain this poor performance.

From a practical point of view, a reference value or threshold is needed if one wanted to use our method to determine automatically whether a given optimized flight should be accepted or not. In such a context, a quite straightforward solution would be to use a leave-one-out cross-validation approach: compute the MML score of each real flight leaving it out of the training data and then averaging over the obtained scores. These reference values have been computed for our dataset and are summarized in table 5.2. We note that the values obtained are very close to the average scores of the real flights showed in table 5.1a.

Table 5.3a contains the scores obtained using the Functional PCA based method presented in section 5.5.3. The training time needed here was of 20 seconds in average. As for the MML in 2D, the real flights' scores are surprisingly low and the two types of simulated trajectories are not well discriminated by the criterion. This might be caused by the fact that this method encodes each training trajectory by a single point in the 4-dimensional

Table 5.1 – Average and standard deviation of the Mean Marginal Likelihood scores using confidence level and normalized density for 50 real flights (*Real*), 50 optimized flights with operational constraints (*Opt1*) and 50 optimized flights without constraints (*Opt2*).

(a) 1-dimensional case

Var.	Confidence level			Normalized Density		
	Real	Opt1	Opt2	Real	Opt1	Opt2
V	0.52 ± 0.16	0.38 ± 0.14	0.15 ± 0.09	0.63 ± 0.16	0.45 ± 0.16	0.17 ± 0.10
γ	0.54 ± 0.09	0.24 ± 0.12	0.22 ± 0.09	0.67 ± 0.09	0.33 ± 0.17	0.29 ± 0.11
α	0.53 ± 0.06	0.08 ± 0.05	0.02 ± 0.01	0.65 ± 0.07	0.10 ± 0.06	0.02 ± 0.01
N_1	0.47 ± 0.24	0.71 ± 0.00	0.03 ± 0.01	0.57 ± 0.27	0.83 ± 0.01	0.04 ± 0.02
Mean	0.52 ± 0.07	0.35 ± 0.06	0.10 ± 0.02	0.63 ± 0.07	0.43 ± 0.08	0.13 ± 0.02

(b) 2-dimensional case

Var.	Confidence level			Normalized Density		
	Real	Opt1	Opt2	Real	Opt1	Opt2
(V, γ)	0.09 ± 0.05	0.11 ± 0.05	0.03 ± 0.03	0.05 ± 0.03	0.06 ± 0.03	0.01 ± 0.01
(α, N_1)	0.03 ± 0.02	0.01 ± 3e-3	0.01 ± 2e-3	0.02 ± 0.01	4e-3 ± 2e-3	3e-3 ± 1e-3
Mean	0.06 ± 0.03	0.06 ± 0.02	0.02 ± 0.01	0.03 ± 0.02	0.03 ± 0.02	0.01 ± 0.01

Table 5.2 – Leave-one-out cross-validated MML scores of the training trajectories.

Var.	Confidence level	Normalized Density
V	0.50 ± 0.19	0.60 ± 0.22
γ	0.51 ± 0.13	0.63 ± 0.15
α	0.51 ± 0.15	0.62 ± 0.17
N_1	0.51 ± 0.22	0.61 ± 0.24
Mean	0.51 ± 0.21	0.62 ± 0.22

space spanned by the principal functions. The training set obtained is hence of $m = 424$ points, which might be too small too attain sufficient accuracy from the Gaussian mixture density estimator in such a high dimension. The principal functions used and scatter plots of the projected trajectories can be found in the online version.

Concerning the LS-CDE approach, because the algorithm needs large Gram matrices (of size $O(nm^2)$) to be stored, we encountered several memory problems when trying to run it on our dataset of 334 531 observation points. For this reason, our results were obtained by applying it to 100 uniform batches. These batches were obtained by partitioning the data according to the altitude. Although the three categories are well-separated by this method, as shown on table 5.3b, the total time needed to train the estimators on every batch was approximately 14 hours.

We didn't test both alternate methods in the 2D setting since the problems observed in 1D (curse of dimensionality for FPCA and memory/time for LS-CDE) would be aggra-

vated.

Table 5.3 – Average and standard deviation of the normalized density scores using Functional PCA and Least-Squares Conditional Density Estimation of 50 real flights (*Real*), 50 optimized flights with operational constraints (*Opt1*) and 50 optimized flights without constraints (*Opt2*).

(a) FPCA

Var.	Real	Opt1	Opt2
V	0.15 ± 0.22	4.9e-04 ± 9.0e-04	2.1e-05 ± 8.1e-05
γ	0.20 ± 0.22	9.3e-03 ± 1.5e-02	1.4e-02 ± 2.2e-02
α	0.28 ± 0.28	1.2e-05 ± 1.8e-05	7.0e-08 ± 1.7e-07
N_1	7.6e-03 ± 6.1e-03	1.6e-02 ± 2.3e-04	1.3e-06 ± 6.7e-07
Mean	0.16 ± 0.12	6.4e-03 ± 3.8e-03	3.6e-03 ± 5.4e-03

(b) LS-CDE

Var.	Real	Opt1	Opt2
V	0.81 ± 0.13	0.63 ± 0.11	0.40 ± 0.23
γ	0.65 ± 0.05	0.55 ± 0.10	0.53 ± 0.08
α	0.91 ± 0.02	0.74 ± 0.03	0.68 ± 0.01
N_1	0.72 ± 0.10	0.79 ± 0.01	0.35 ± 0.05
Mean	0.77 ± 0.05	0.68 ± 0.04	0.49 ± 0.06

In conclusion, our numerical results indicate that the MML criterion has better discriminative power than FPCA and LS-CDE for the task of assessing curves relatively to a set of "good" examples. Furthermore, the training time and memory needed for using LS-CDE in datasets of this size seems crippling. Concerning the FPCA method, it does not seem to be applicable to datasets with so few curves and present a higher training time than MML.

5.6 Conclusions

In this paper we proposed a new approach for a problem which seemed unaddressed by the statistical learning community: quantifying the closeness from a curve to a set of random functions. We introduced a class of probabilistic criteria for this context called the Mean Marginal Likelihood (MML), and analyzed two possible scaling functions used to build them. We also derived a class of estimators of our criteria, which make use of local density estimators proved to consistently approximate the marginal densities of a random process. For practical applications, we suggested a particular flexible density estimator believed to have the right properties needed in this setting, called the self-consistent kernel estimator.

Numerical experiments using real aircraft data were carried to compare the MML with other well-established approaches from Functional Data Analysis and Conditional Density Estimation. The results show that, although the MML does not take into account the

temporal structure of the data as other standard functional data analysis methods, it is a good candidate for the type of applications suggested. This seems to be especially the case if the number of training trajectories is too small for using FPCA or if the total number of observation points is too large for conditional density estimation. Moreover, the training times obtained for MML are by far the shortest among the compared methods, which confirms the relevance of the self-consistent kernel estimator. Furthermore, the ease to visualize, localize and interpret discrepancy zones allowed by MML make it a good exploratory analysis tool for functional data (see e.g. figure 5.6). We also note that our method does not perform as well in a multidimensional setting, but that the training time should not be an obstacle. In future work we intend to test the MML with parametric density estimators, which should be less affected by the curse of dimensionality.

Chapter references

A. Bernacchia and S. Pigolotti. Self-consistent method for density estimation. *Journal of the Royal Statistical Society*, 73(3):407–422, 2011.

D. Bosq. *Nonparametric statistics for stochastic processes: estimation and prediction*, volume 110. Springer Science & Business Media, 2012.

S. Cafieri, L. Cellier, F. Messine, and R. Omheni. Combination of optimal control approaches for aircraft conflict avoidance via velocity regulation. *Optimal Control Applications and Methods*, 39(1):181–203, 2018.

J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965.

O. Cots, J. Gergaud, and D. Goubinat. Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft. *Optimal Control Applications and Methods*, 39(1):281–301, 2018.

F. Cribari-Neto, K. L. P. Vasconcellos, and N. L. Garcia. A note on inverse moments of binomial variates. *Brazilian Review of Econometrics*, 20:269–277, 2000.

S. Dabo-Niang. Kernel density estimator in an infinite-dimensional space with a rate of convergence in the case of diffusion process. *Applied mathematics letters*, 17(4):381–386, 2004.

S. Dabo-Niang, F. Ferraty, and P. Vieu. On the using of modal curves for radar waveforms classification. *Computational Statistics & Data Analysis*, 51(10):4878–4890, 2007.

A. Dutt and V. Rokhlin. Fast Fourier Transforms for Nonequispaced Data. *SIAM Journal on Scientific Computing*, 14(6):1368–1393, 1993.

F. Ferraty and P. Vieu. Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44(1-2):161–173, 2003.

I. K. Glad, N. L. Hjort, and N. G. Ushakov. Correction of density estimators that are not densities. *Scandinavian Journal of Statistics*, 30(2):415–427, 2003.

L. Greengard and J.-Y. Lee. Accelerating the Nonuniform Fast Fourier Transform. *SIAM Review*, 46(3):443–454, 2004.

B. Gregorutti, B. Michel, and P. Saint-Pierre. Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis*, 90:15–35, 2015.

P. Hall and N. E. Heckman. Estimating and depicting the structure of a distribution of random functions. *Biometrika*, 89(1):145–158, 2002.

J. Jacod. Lecture notes on "Mouvement brownien et calcul stochastique", 2007. Cours de M2R Univ. Pierre et Marie Curie.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009.

S. Khardi, L. Abdallah, O. Konovalova, and M. Houacine. Optimal approach minimizing aircraft noise and fuel consumption. *Acta acustica united with Acustica*, 96(1):68–75, 2010.

K. Makiyama. densratio, A Python Package for Density Ratio Estimation, Dec. 2016. URL https://github.com/hoxo-m/densratio_py. Dowloaded on may 4th 2018.

N. Nguyen. Singular arc time-optimal climb trajectory of aircraft in a two-dimensional wind field. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6598, 2006.

F. Nicol. Functional principal component analysis of aircraft trajectories. In *Proceedings of the 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management (ISIATM)*, 2013.

T. A. O'Brien, W. D. Collins, S. A. Rauscher, and T. D. Ringler. Reducing the Computational Cost of the ECF Using a nuFFT: A Fast and Objective Probability Density Estimation Method. *Computational Statistics & Data Analysis*, 79:222–234, 2014.

T. A. O'Brien, K. Kashinath, N. R Cavanaugh, W. D. Collins, and J. P. O'Brien. A fast and objective multidimensional kernel density estimation method: fastkde. *Computational Statistics & Data Analysis*, 101:148–160, 2016.

E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

F. Pedregosa et al. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

B. L. S. Prakasa Rao. Nonparametric density estimation for functional data via wavelets. *Communications in Statistics—Theory and Methods*, 39(8-9):1608–1618, 2010a.

B. L. S. Prakasa Rao. Nonparametric density estimation for functional data by delta sequences. *Brazilian Journal of Probability and Statistics*, 24(3):468–478, 2010b.

J. O. Ramsay and B. W. Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007.

C. Rommel, J. F. Bonnans, B Gregorutti, and P. Martinon. Aircraft dynamics identification for optimal control. In *Proceedings of the 7th European Conference for Aeronautics and Aerospace Sciences*, 2017.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837, 1956.

D. W. Scott. *Multivariate density estimation: theory, practice, and visualization.* John Wiley & Sons, 2015.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall, 1986.

C. J. Stone. An asymptotically optimal window selection rule for kernel density estimates. *The Annals of Statistics*, 12(4):1285–1297, 1984.

M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya, and D. Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788, 2010.

A. B. Tsybakov. *Introduction to Nonparametric Estimation.* Springer, 1st edition, 2008.

L. Wasserman. *All of statistics: a concise course in statistical Inference.* Springer texts in statistics. Springer, 2004.

G. S. Watson and M. R. Leadbetter. On the estimation of the probability density. *The Annals of Mathematical Statistics*, 34(2):480–491, 1963.

Chapter 6

OPTIMIZATION OF ACCEPTABLE TRAJECTORIES

Abstract: *We consider the task of solving an aircraft trajectory optimization problem where the system dynamics have been estimated from recorded data. Additionally, we want to avoid optimized trajectories that go too far away from the domain occupied by the data, since the model validity is not guaranteed outside this region. This motivates the need for a proximity indicator between a given trajectory and a set of reference trajectories. In this presentation, we propose such an indicator based on a parametric estimator of the training set density. We then introduce it as a penalty term in the optimal control problem. Our approach is illustrated with an aircraft minimal consumption problem and recorded data from real flights. We observe in our numerical results the expected trade-off between the consumption and the penalty term.*

Contents

6.1 Introduction 134

6.2 Aircraft Trajectory Optimization as an Identified Optimal Control Problem 134

6.3 A Parametric Marginal Density Estimator 135

6.3.1 The Gaussian Mixture Model 135

6.3.2 The Expectation-Maximization Algorithm 137

6.3.3 More details on the derivation of the EM algorithm 138

6.4 Application to an Aircraft Minimal-Consumption Problem . . 139

6.4.1 Experiments Description 139

6.4.2 Data Description 139

6.4.3 Choice of the “time” variable 140

6.4.4 Algorithm Settings 140

6.4.5 Numerical Comparison Between MML using Gaussian Mixture and Self-Consistent Kernel Estimators 141
6.4.6 Penalized Optimization Results 142
6.5 Conclusion 143

6.1 Introduction

The suggested indicator, called the Mean Marginal Likelihood (MML), is estimated in practice by aggregating several kernel density estimators (KDE).

As pointed out in this paper, although a good discriminating power can be achieved with the MML in one dimension, using the KDE as marginal density estimator leads to poor performance when applied to more than one variable. Building the MML with a parametric density estimator could help dealing with such undesired behavior, since these are known to be less sensitive to dimensionality increase.

Furthermore, instead of just assessing simulated trajectories after the optimization, one could want to use such a criterion to penalize the optimal control problem, in order to obtain more acceptable trajectories directly. From a practical perspective, the nonparametric nature of the KDE-based MML from Rommel et al. [2018b] makes it unsuitable for generic optimal control solvers such as BOCOP [Bonnans et al., 2017], which require the objective and constraint functions to be compatible with automatic differentiation tools. It is in this context that we suggest an adaptation of the Mean Marginal Likelihood using Gaussian mixture models as marginal density estimator.

In what follows, we will first introduce the optimal control problem with identified dynamics and add to it a penalty term based on the parametric Mean Marginal Likelihood. After recalling the derivation of such a quantity, more details concerning the estimation of the Gaussian mixture models will be given. Finally, numerical results based on real aircraft data will be used to compare the performances of our MML extension to the original nonparametric version from Rommel et al. [2018b]. They will also serve to illustrate the usefulness of our approach as a new trajectory optimization penalty.

6.2 Aircraft Trajectory Optimization as an Identified Optimal Control Problem

We consider an aircraft in climb phase, modeled as a dynamical system of state variables $\boldsymbol{x} \in \mathbb{X} \subset W^{1,\infty}(0, t_f; \mathbb{R}^{d_x})$ and control variables $\boldsymbol{u} \in \mathbb{U} \subset L^{\infty}(0, t_f; \mathbb{R}^{d_u})$. In such a context, the problem of optimizing the trajectory over a certain horizon $t_f > 0$ may be seen as a nonlinear constrained optimal control problem of the following form:

$$\min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{X}\times\mathbb{U}} \int_0^{t_f} C(t,\boldsymbol{u}(t),\boldsymbol{x}(t))dt,$$
$$\text{s.t.} \begin{cases} \dot{\boldsymbol{x}}(t) = \hat{g}(t,\boldsymbol{u}(t),\boldsymbol{x}(t)), \quad \text{for a.e. } t\in[0,t_f], \\ \boldsymbol{u}(t)\in U_{ad}, \quad \boldsymbol{x}(t)\in X_{ad}, \quad \text{for a.e. } t\in[0,t_f], \\ \Phi(\boldsymbol{x}(0),\boldsymbol{x}(t_f))\in K_\Phi, \\ c_j(t,\boldsymbol{u}(t),\boldsymbol{x}(t))\leq 0, \quad j=1,\dots,n_c, \quad \text{for all } t\in[0,t_f], \end{cases} \tag{6.2.1}$$

where $C\in\mathcal{C}^2$ is the running cost, $\hat{g}\in\mathcal{C}^1$ if the dynamics functions, $U_{ad}\subset\mathbb{R}^{d_u}$ and $X_{ad}\subset\mathbb{R}^{d_x}$ are the admissible (closed) sets for the controls and states, $\Phi\in\mathcal{C}^1$ is the initial-final state constraint function, K_Φ is a nonempty closed convex set of $\mathbb{R}^{n_\Phi}$ and $\{c_j\}_{j=1}^{n_c}\in(\mathcal{C}^1)^{n_c}$ are the path constraint functions.

The function $\hat{g}$ from (6.2.1) reflects the fact that we want the optimized trajectory to respect some estimated model of the true aircraft dynamics, denoted by g. Assuming that one has access to some dataset $\mathcal{D}=\{(\boldsymbol{x}_i,\boldsymbol{u}_i,\dot{\boldsymbol{x}}_i)\}_{i=1}^m$ of previous flights, it may be used to build $\hat{g}$ (see e.g. Rommel et al. [2017, 2018a]). Note that, in a general setting, the dynamics model $\hat{g}$ can only be guaranteed to accurately approximate g close to the region occupied by the data. Hence, it seems desirable to prevent the solution of (6.2.1) from going too far away from $\mathcal{D}$. This is even more true for practical applications with high safety requirements such as airplane flights, since the optimized trajectories will have to be accepted by the Air Traffic Control and by the pilots supposed to fly them. Indeed, avoiding solutions outside some flight envelope of "realistic" trajectories should help on that.

This can be achieved by adding to (6.2.1) a flight domain as a hard constraint on the states and controls $(\boldsymbol{x},\boldsymbol{u})$. Another more flexible possibility is to introduce a weight $\lambda>0$ and a penalty term $\text{Pen}_{\hat{g}}(\boldsymbol{u},\boldsymbol{x})$ to the objective function:

$$\min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{X}\times\mathbb{U}} \int_0^{t_f} C(t,\boldsymbol{u}(t),\boldsymbol{x}(t))dt + \lambda\,\text{Pen}_{\hat{g}}(\boldsymbol{u},\boldsymbol{x}),$$
$$\text{s.t.} \begin{cases} \dot{\boldsymbol{x}}(t) = \hat{g}(t,\boldsymbol{u}(t),\boldsymbol{x}(t)), \quad \text{for a.e. } t\in[0,t_f], \\ \boldsymbol{u}(t)\in U_{ad}, \quad \boldsymbol{x}(t)\in X_{ad}, \quad \text{for a.e. } t\in[0,t_f], \\ \Phi(\boldsymbol{x}(0),\boldsymbol{x}(t_f))\in K_\Phi, \\ c_j(t,\boldsymbol{u}(t),\boldsymbol{x}(t))\leq 0, \quad j=1,\dots,n_c, \quad \text{for all } t\in[0,t_f]. \end{cases} \tag{6.2.2}$$

6.3 A Parametric Marginal Density Estimator

6.3.1 The Gaussian Mixture Model

Marginal density estimation can be done by uniformly partitioning the space of times $\mathbb{T}$ into *bins* and building standard density estimators using the data points whose sampling times fall in each bin. A special instance of kernel density estimators (KDE) were proposed for this task in Rommel et al. [2018b], which are nonparametric. Despite the great flexibility

of the latter, parametric models may be better suited in some contexts, such as solving numerically penalized problems of the form of (6.2.2). For this reason, we considered in this paper the use of Gaussian mixture models (GMM) for the marginal density estimation.

Indeed, despite being straightforward parametric models, Gaussian mixtures are still quite flexible, being interpretable as simpler versions of Gaussian kernel density estimators [1]. These models are known for being well-suited for problems in which the data falls into a small number of clusters. In our case, we indeed observe that, for a given zone in time during climb phase, different trajectories from the same aircraft will often concentrate around a small number "standard" values.

In this model, for some time $t \in \mathbb{T}$, the marginal density is assumed to be a finite convex combination of K terms

$$\forall z \in E, \qquad f_h(z) = \sum_{k=1}^{K} w_{t,k}\phi(z, \mu_{t,k}, \Sigma_{t,k}), \qquad \text{with } \sum_{k=1}^{K} w_{t,k} = 1, \tag{6.3.1}$$

where all weights $w_{t,k}$ are nonnegative and $\phi(\cdot, \mu, \Sigma)$ denotes a d-dimensional Gaussian density function with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathcal{S}_d(\mathbb{R})$:

$$\phi(z, \mu, \Sigma) := \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} \exp\left(-\frac{1}{2}(z-\mu)^\top \Sigma^{-1}(z-\mu)\right). \tag{6.3.2}$$

Let $\{z_1, \dots, z_N\} \subset E$ denote the subsample of observations falling in the bin containing t, considered as a neighborhood. As the means and covariances of the K components are unknown, they need to be inferred using the data, which is usually done through maximum likelihood estimation. Had we one component, i.e. $K = 1$, the log-likelihood of a sample of observations $\{z_i\}_{i=1}^N$ would be

$$\sum_{i=1}^{N} -\frac{1}{2}\left(\log(2\pi) - \log\det\Sigma_{h,1} - (z_i - \mu_{h,1})^\top \Sigma_{h,1}^{-1}(z_i - \mu_{h,1})\right), \tag{6.3.3}$$

which is maximized by the sample mean and covariance:

$$\hat{\mu}_{h,1} = \frac{1}{N}\sum_{i=1}^{N} z_i, \qquad \hat{\Sigma}_{h,1} = \frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{\mu}_{h,1})(z_i - \hat{\mu}_{h,1})^\top. \tag{6.3.4}$$

In the general case $K > 1$ there is no closed-form for the maximum likelihood estimates, which is why an Expectation-Maximization (EM) algorithm is usually preferred.

1. While KDE's place a Gaussian density around each observation, GMM places fewer Gaussians around estimated centers.

6.3.2 The Expectation-Maximization Algorithm

The EM is a well-established optimization algorithm for maximum likelihood estimation originally proposed by Dempster et al. [1977]. The key of the EM procedure is to suppose the existence of a hidden random variable J valued on $\{1, \dots, K\}$, such that the i.i.d sample of its observations $\{J_i\}_{i=1}^N$, "labels" the observations of Z_t: if $J_i = k$, then the i^{th} observation z_i was drawn from the k^{th} Gaussian component. In this case, we can interpret $\phi(z_i, \mu_{t,k}, \Sigma_{t,k})$ as the probability density of $Z_{h,i}$ conditioned by $J_i = 1$ and the weight $w_{t,k}$ as the prior probability of being drawn from the k^{th} component $\mathbb{P}(J_i = k)$. Furthermore, the usefulness of the unobserved variables J_i is intuitive since, if we knew for each sample z_i which component generated it, we could just group them by component and compute the sample means and covariances separately as done in (6.3.4).

That being said, the EM falls into two main steps. Starting from an initial guess of parameters $\hat{\theta}_t = (\hat{w}_{t,k}, \hat{\mu}_{t,k}, \hat{\Sigma}_{t,k})_{k=1}^K$, the first step corresponds to the computation of the posterior probabilities of the labels $\mathbb{P}(J_i = k|\hat{\theta}_t, Z_h)$, usually called *responsibilities*:

$$\hat{\pi}_{k,i} := \mathbb{P}(J_i = k|\hat{\theta}_t, Z_h) = \frac{\hat{\mu}_{t,k}\phi(z_i, \hat{\mu}_{t,k}, \hat{\Sigma}_{t,k})}{\sum_{j=1}^N \hat{w}_{t,k}\phi(z_j, \hat{\mu}_{t,k}, \hat{\Sigma}_{t,k})}, \qquad k = 1, \dots, K, \quad i = 1, \dots, N. \tag{6.3.5}$$

This step is called the *Expectation* step since it estimates a discrete probability distribution which is later used to compute the expectation of the log-likelihood to be maximized. As explained in the appendix, this criteria is a lower bound of the log-likelihood, which is the main idea of the procedure. Its maximization is carried in the second step, called the *Maximization* step, which brings the weights to be approximated by the empirical mean of the responsibilities

$$\hat{w}_{t,k} = \frac{1}{N}\sum_{i=1}^N \hat{\pi}_{k,i}, \tag{6.3.6}$$

and the means and covariances to be the weighted maximum likelihood estimates:

$$\hat{\mu}_{t,k} = \frac{\sum_{i=1}^N \hat{\pi}_{k,i} z_i}{\sum_{i=1}^N \hat{\pi}_{k,i}}, \qquad \hat{\Sigma}_{t,k} = \frac{\sum_{i=1}^N \hat{\pi}_{k,i}(z_i - \hat{\mu}_{t,k})(z_i - \hat{\mu}_{t,k})^\top}{\sum_{i=1}^N \hat{\pi}_{k,i}}. \tag{6.3.7}$$

These steps are then repeated until convergence. For this reason, the EM can be seen as an alternating optimization algorithm in which we successively adjust the weights probabilities and the parameters values. It has been shown in Dempster et al. [1977] that such a procedure converges linearly to a local maximum of the likelihood and Xu and Jordan [1996] proved that it is competitive to other superlinear and quadratic optimization algorithms.

6.3.3 More details on the derivation of the EM algorithm

In this subsection we give more insight on the statistical origin of the EM algorithm. The explanations hereafter are strongly inspired by the excellent notes Sridharan [2014] towards which the reader is referred for further details.

We see from the expression of the Gaussian mixture density (6.3.1) that the log-likelihood of a sample of N observations $\{z_i\}_{i=1}^N$ writes:

$$\log f_h(z_1, \dots, z_N|\theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} w_{t,k} \phi(z_i, \mu_{t,k}, \Sigma_{t,k}) \right). \tag{6.3.8}$$

It is the presence of the sum inside the logarithm that is problematic here. However, when we add the hidden variable J, we have a simpler expression for the log-likelihood relative to the distribution of the joint variable (Z_h, J):

$$\log p_{Z_h,J} \left(\{(z_i, J_i)\}_{i=1}^N | \theta \right) = \sum_{i=1}^{N} \sum_{k=1}^{K} \delta_k(J_i) \left(\log \phi(z_i, \mu_{t,k}, \Sigma_{t,k}) \right), \tag{6.3.9}$$

where $\delta_k(x) = 1$ if $x = k$ and 0 otherwise, for $k = 1, \dots, K$. This corresponds to the remark that if we knew for each sample z_i which component generated it, we could just group them by component and compute the sample means and covariances separately as done in (6.3.4).

By writing f_h as the marginal over J of $p_{Z_h,J}$ and artificially introducing the density of p_J of J, we can write the log-likelihood (6.3.8) in terms of the expectation relative to J:

$$\begin{aligned}
\log f_h(z_1, \dots, z_N|\theta) &= \log \sum_{k=1}^{K} p_{Z_h,J} \left(\{(z_i, J_i = k)\}_{i=1}^N | \theta \right), \\
&= \log \sum_{k=1}^{K} \frac{p_{Z_h,J} \left(\{(z_i, J_i = k)\}_{i=1}^N | \theta \right)}{p_J(J_i = k)} p_J(J_i = k), \\
&= \log \mathbb{E}_J \left[\frac{p_{Z_h,J} \left(\{(z_i, J_i)\}_{i=1}^N | \theta \right)}{p_J(J_i)} \right], \\
&\geq \mathbb{E}_J \left[\log \frac{p_{Z_h,J} \left(\{(z_i, J_i)\}_{i=1}^N | \theta \right)}{p_J(J_i)} \right].
\end{aligned} \tag{6.3.10}$$

The lower-bound in (6.3.10) comes from Jensen's inequality. Using the definition of the conditional density, we can rewrite it as follows:

$$\begin{aligned}
\mathbb{E}_J \left[\log \frac{p_{Z_h,J} \left(\{(z_i, J_i)\}_{i=1}^N | \theta \right)}{p_J(J_i)} \right] &= \mathbb{E}_J \left[\log \frac{p_{J|Z_h} \left(J | \{z_i\}_{i=1}^N; \theta \right) f_h \left(\{z_i\}_{i=1}^N; \theta \right)}{p_J(J_i)} \right] \\
&= \log f_h \left(\{z_i\}_{i=1}^N; \theta \right) - \mathrm{KL} \left(p_J \,||\, p_{J|Z_h} \left(\cdot | \{z_i\}_{i=1}^N; \theta \right) \right),
\end{aligned} \tag{6.3.11}$$

where KL denotes the Kullbeck-Liebler divergence between the posterior distribution of J and its true distribution. This allows to interpret the Expectation step of the EM algorithm as the maximization of criteria (6.3.11) w.r.t. p_J, which is reached for $p_J = p_{J|Z_h}\left(\cdot|\{z_i\}_{i=1}^N;\theta\right)$.

Concerning the Maximization step, (6.3.10) can be written in the following way:

$$\mathbb{E}_J\left[\log\frac{p_{Z_h,J}\left(\{(z_i,J_i)\}_{i=1}^N|\theta\right)}{p_J(J_i)}\right] = \mathbb{E}_J\left[\log p_{Z_h,J}\left(\{(z_i,J_i)\}_{i=1}^N|\theta\right)\right] - \mathbb{E}_J\left[\log p_J(J_i)\right]. \tag{6.3.12}$$

We see that only the first term in (6.3.12) depends on θ and that it corresponds to the expectation of (6.3.9), which we know how to compute for give p_J.

6.4 Application to an Aircraft Minimal-Consumption Problem

6.4.1 Experiments Description

In this section, we test the proposed approach with real aircraft data. First, we compare the suggested GMM-based Mean Marginal Likelihood criterion to the original nonparametric kernel-based MML from Rommel et al. [2018b]. Then, we solve a series of trajectory optimization problems penalized with such an indicator and assess the obtained results.

6.4.2 Data Description

In all experiments presented, the MML estimators were computed based on a training set of 424 different flights from the same medium haul aircraft, corresponding to a total of 334 531 point observations. This dataset was extracted from the *Quick Access Recorder* (QAR) and is exactly the same as in Rommel et al. [2018b]. We only used signals sampled at 1 Hz of 5 variables, namely the altitude h, the true airspeed V (or TAS), the path angle γ, the angle of attack α and the average engines turbofan speed N_1. Some of these signals were not directly available and were computed from other measurements using standard flight mechanics formulas (see e.g. Rommel et al. [2017]). Furthermore, we kept only the portion of these signals corresponding to the climb phase, starting from 5000 ft. This same dataset was used to estimate the aircraft dynamics with the methods described in Rommel et al. [2017]. All optimization problems solved in this section used this same dynamics model.

Concerning the comparisons made between the parametric and nonparametric versions, the test set used to compute the MML scores was also the same as in Rommel et al. [2018b]:

1. 50 real flights, extracted from the training set prior to training;

2. 50 simulated trajectories obtained by using BOCOP Bonnans et al. [2017] to solve problem (6.2.1), with constraints keeping the resulting speed V and N_1 between reasonable operational bounds;

3. and another 50 simulated trajectories optimized without the operational constraints.

6.4.3 Choice of the "time" variable

As all the trajectories considered are climb profiles, a change of variable is used to parametrize all functions using the altitude h, which plays the role of "time" here. Indeed, this is quite a standard practice in trajectory optimization since h is nondecreasing during this phase of flight and all other variables strongly depend on it.

6.4.4 Algorithm Settings

For all MML scores computation, the partitioning of the altitude was carried exactly as in Rommel et al. [2018b], i.e.:

- between 1524 and 3000m and between 4000 and 12000m, the altitudes were divided homogeneously into intervals of size $b_m^{(1)} = 21\text{m} \simeq 1/\sqrt{m}$;

- between 3000 and 4000m we used an interval size $b_m^{(2)} = 10\text{m} \simeq b_m^{(1)}/2$.

The altitudes between 3000 and 4000m define a zone where climb-steps are common. Hence, faster variations w.r.t. the altitude can be observed in this region (see e.g. figures 6.2), which explains why a finer partition was preferred.

All self-consistent kernel density estimations were carried using the same implementation as in Rommel et al. [2018b], i.e. the python library FASTKDE [O'Brien et al., 2014, O'Brien et al., 2016]. The parameters of the Gaussian mixtures were learned using the SCIKIT-LEARN library Pedregosa et al. [2011], which implements the EM algorithm described in section 6.3.2. The number of components was chosen between 1 and 5 using the Bayesian information criteria (BIC) Schwarz [1978]. Default settings were kept for the initialization: initial weights, means and covariances were set by solving a K-means clustering problem 10 times using Lloyd's algorithm Lloyd [1982] initialized with centroids drawn randomly from the dataset.

Concerning the optimal control problems (6.2.1) and (6.2.2), both were set with the estimated dynamics and solved using BOCOP [Bonnans et al., 2017]. The altitude was used as independent variable (i.e 'time') with 1000 steps for all optimizations. The initial and final conditions of the penalized control problems were set to match 20 flights randomly drawn from the training dataset.

6.4.5 Numerical Comparison Between MML using Gaussian Mixture and Self-Consistent Kernel Estimators

Tables 6.1a and 6.1b contain the MML scores using normalized density (5.3.3) and confidence level (5.3.4) as scaling maps for both parametric and nonparametric formulations. These scores were computed in a 1-dimensional setting: each variable of interest (V, γ, α and N_1) was processed independently. We see that the values returned by both implementations of the MML estimator are statistically the same for all four variables considered, with both scaling maps. As a consequence, the discriminating power of the GMM-based method is as good as the KDE-based one.

Table 6.1 – Average and standard deviation of the Mean Marginal Likelihood scores in **1-dimensional setting** using confidence level and normalized density for 50 real flights (*Real*), 50 optimized flights with operational constraints (*Opt1*) and 50 optimized flights without constraints (*Opt2*).

(a) Self-consistent kernel estimator

Var.	Confidence level			Normalized Density		
	Real	Opt1	Opt2	Real	Opt1	Opt2
V	0.52 ± 0.16	0.38 ± 0.14	0.15 ± 0.09	0.63 ± 0.16	0.45 ± 0.16	0.17 ± 0.10
γ	0.54 ± 0.09	0.24 ± 0.12	0.22 ± 0.09	0.67 ± 0.09	0.33 ± 0.17	0.29 ± 0.11
α	0.53 ± 0.06	0.08 ± 0.05	0.02 ± 0.01	0.65 ± 0.07	0.10 ± 0.06	0.02 ± 0.01
N_1	0.47 ± 0.24	0.71 ± 0.00	0.03 ± 0.01	0.57 ± 0.27	0.83 ± 0.01	0.04 ± 0.02
Mean	0.52 ± 0.07	0.35 ± 0.06	0.10 ± 0.02	0.63 ± 0.07	0.43 ± 0.08	0.13 ± 0.02

(b) GMM

Var.	Confidence level			Normalized Density		
	Real	Opt1	Opt2	Real	Opt1	Opt2
V	0.51 ± 0.15	0.37 ± 0.13	0.15 ± 0.08	0.61 ± 0.16	0.44 ± 0.15	0.17 ± 0.09
γ	0.53 ± 0.09	0.24 ± 0.13	0.22 ± 0.08	0.67 ± 0.08	0.33 ± 0.17	0.29 ± 0.11
α	0.54 ± 0.06	0.08 ± 0.05	0.01 ± 0.01	0.67 ± 0.06	0.10 ± 0.07	0.02 ± 0.01
N_1	0.47 ± 0.23	0.71 ± 0.01	0.03 ± 0.01	0.55 ± 0.26	0.83 ± 0.01	0.02 ± 0.02
Mean	0.51 ± 0.06	0.35 ± 0.06	0.10 ± 0.02	0.63 ± 0.07	0.42 ± 0.08	0.12 ± 0.02

Tables 6.2a and 6.2b contain the scores in a 2-dimensional setting: variables were grouped in two pairs, (V, γ) and (α, N_1), so that densities of variables of the same group are estimated jointly. The first group contains state variables, supposed to vary slowly, while the second contains control variables. As noted in Rommel et al. [2018b], the scores obtained using the kernel estimator version are abnormally low, even for the real flights, and the three categories of test flights are not well separated. We suspect that this poor performance was caused by the high model-complexity of this nonparametric method, which makes it more sensitive to the curse of dimensionality. Following this line of thought, parametric models should have a better behavior in this context. The scores obtained by

the GMM version seem to confirm this assumption. Indeed, the three groups of flights are well discriminated by the MML scores, being higher for the real flight and lower for the flights optimized without reasonable constraints. Moreover, the training of the GMM version took in average only 1.9 seconds [2], which is more than 40 times faster than the kernel version (81.4 seconds in average). However, the variances observed in this 2-dimensional setting are larger than those obtained in dimension one. No discriminating advantage seems to exist in this context relative to the previous one, which is why variables scores were computed separately in the following section.

Table 6.2 – Average and standard deviation of the Mean Marginal Likelihood scores in **2-dimensional setting** using confidence level and normalized density for 50 real flights (*Real*), 50 optimized flights with operational constraints (*Opt1*) and 50 optimized flights without constraints (*Opt2*).

(a) Self-consistent kernel estimator

VAR.	CONFIDENCE LEVEL			NORMALIZED DENSITY		
	REAL	OPT1	OPT2	REAL	OPT1	OPT2
(V, γ)	0.09 ± 0.05	0.11 ± 0.05	0.03 ± 0.03	0.05 ± 0.03	0.06 ± 0.03	0.01 ± 0.01
(α, N_1)	0.03 ± 0.02	0.01 ± 3E-3	0.01 ± 2E-3	0.02 ± 0.01	4E-3 ± 2E-3	3E-3 ± 1E-3
MEAN	0.06 ± 0.03	0.06 ± 0.02	0.02 ± 0.01	0.03 ± 0.02	0.03 ± 0.02	0.01 ± 0.01

(b) GMM

VAR.	CONFIDENCE LEVEL			NORMALIZED DENSITY		
	REAL	OPT1	OPT2	REAL	OPT1	OPT2
(V, γ)	0.52 ± 0.17	0.22 ± 0.10	0.07 ± 0.03	0.59 ± 0.14	0.30 ± 0.09	0.08 ± 0.08
(α, N_1)	0.93 ± 0.03	0.84 ± 0.01	0.82 ± 0.01	0.63 ± 0.12	0.12 ± 0.06	0.01 ± 0.01
MEAN	0.73 ± 0.10	0.53 ± 0.05	0.45 ± 0.01	0.61 ± 0.12	0.21 ± 0.06	0.05 ± 0.04

6.4.6 Penalized Optimization Results

In order to check that the GMM-based Mean Marginal Likelihood can be used as a penalty term $\mathrm{Pen}_{\hat{g}}(\boldsymbol{u}, \boldsymbol{x})$ in the optimal control problem (6.2.2), we tested different penalty weights ranging from 0 to 1000 for 20 initial and final conditions. Only the 1-dimensional MML was computed using the normalized density as scaling map. This choice was motivated by the results from the previous section showing that 1D computations allow a better discrimination and that both scaling maps lead to similar results, while the normalized density is easier to compute. Moreover, only the state variables V and γ were penalized in the problem, since penalizing the control variables led to convergence difficulties. The total fuel consumption and Mean Marginal Likelihood scores of the optimized flights, averaged over all 20 initializations, are plotted on figure 6.1. First, we confirm that the

2. on a single 2.30 GHz core with 7.7 GB memory

more we penalize, the more we consume and the higher the Mean Marginal Likelihood of the trajectory. Furthermore, we see that an MML score comparable to a real flight (i.e. > 0.6) is attainable with enough penalization, with and without operational constraints. Of course , this is achieved at the cost of a higher fuel consumption, which illustrates that a trade-off must be found for practical applications. We can also observe that with the highest values of λ, similar consumptions and MML scores are reached for both types of constraints. Indeed, if we penalize too much we end up always finding a solution passing in the same narrow zone of highest densities, as depicted for one example on figure 6.2.

6.5 Conclusion

In this chapter we introduced a parametric version of the Mean Marginal Likelihood estimator from chapter 5. This new estimator was shown to lead to equivalent scores and discriminative power when used for quantifying the distance between a simulated climb trajectory and a set of real flights. It also overcomes the limitations of the former kernel-based estimator relative to multi-dimensional data. Furthermore, our numerical simulations show that our MML estimator is suitable for penalizing an optimal control problem, being a tool to obtain trajectories with good objective while still being acceptable for practical applications.

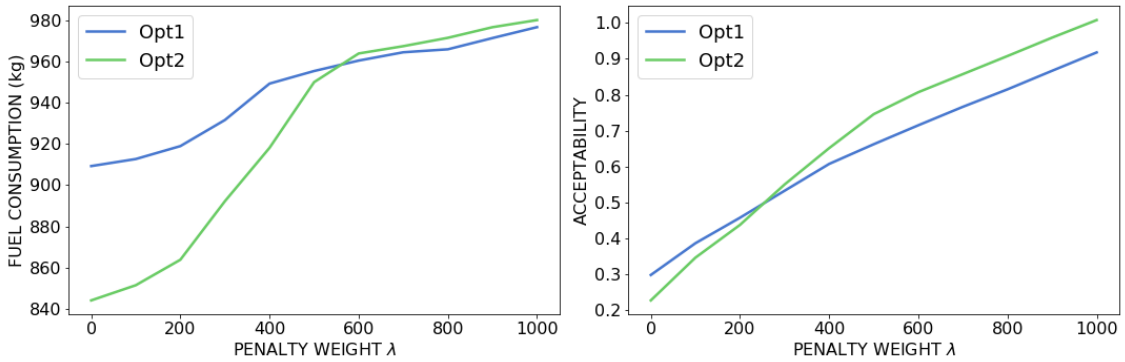
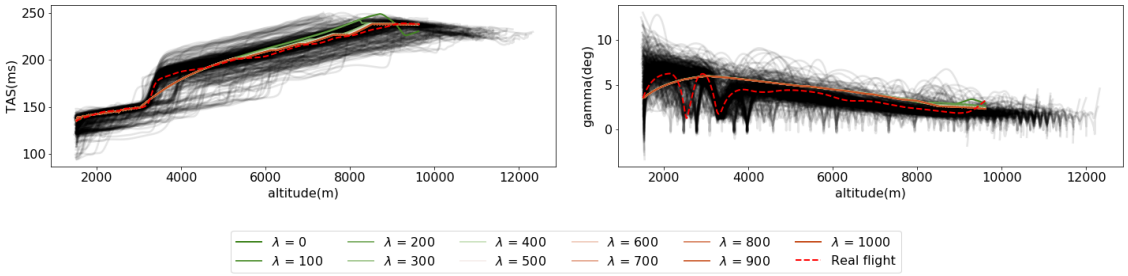
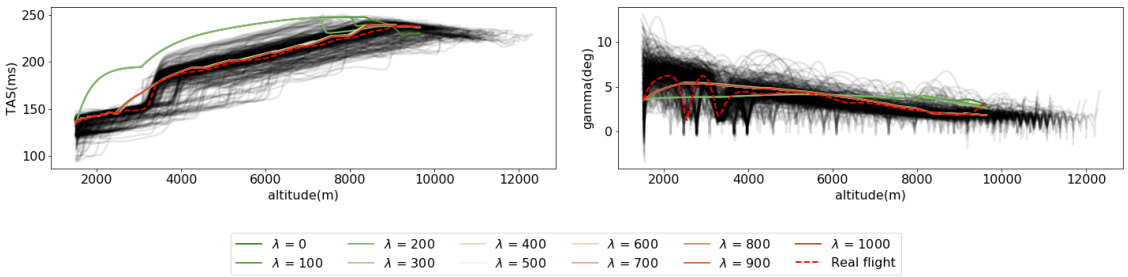


Figure 6.1 – Average over 20 flights of the total fuel consumption and MML score (called *acceptability* here) of optimized trajectories with varying MML-penalty weight λ.



(a) With operational constraints (OPT1).



(b) Without operational constraints (OPT2).

Figure 6.2 – Example of optimized flight with different MML-penalty weights λ. The dashed red lines correspond to the real trajectory used to define the initial and final conditions for the optimization.

Chapter references

J. F. Bonnans, D. Giorgi, V. Grelard, B. Heymann, S. Maindrault, P. Martinon, O. Tissot, and J. Liu. Bocop – A collection of examples. Technical report, INRIA, 2017. URL http://www.bocop.org.

S. Cafieri, L. Cellier, F. Messine, and R. Omheni. Combination of optimal control approaches for aircraft conflict avoidance via velocity regulation. *Optimal Control Applications and Methods*, 39(1):181–203, 2018.

O. Cots, J. Gergaud, and D. Goubinat. Direct and indirect methods in optimal control with state constraints and the climbing trajectory of an aircraft. *Optimal Control Applications and Methods*, 39(1):281–301, 2018.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

S. Khardi, L. Abdallah, O. Konovalova, and M. Houacine. Optimal approach minimizing aircraft noise and fuel consumption. *Acta acustica united with Acustica*, 96(1):68–75, 2010.

S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

N. Nguyen. Singular arc time-optimal climb trajectory of aircraft in a two-dimensional wind field. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6598, 2006.

T. A. O'Brien, W. D. Collins, S. A. Rauscher, and T. D. Ringler. Reducing the Computational Cost of the ECF Using a nuFFT: A Fast and Objective Probability Density Estimation Method. *Computational Statistics & Data Analysis*, 79:222–234, 2014.

T. A. O'Brien, K. Kashinath, N. R Cavanaugh, W. D. Collins, and J. P. O'Brien. A fast and objective multidimensional kernel density estimation method: fastkde. *Computational Statistics & Data Analysis*, 101:148–160, 2016.

F. Pedregosa et al. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

C. Rommel, J. F. Bonnans, B Gregorutti, and P. Martinon. Aircraft dynamics identification for optimal control. In *Proceedings of the 7th European Conference for Aeronautics and Aerospace Sciences*, 2017.

C. Rommel, J. F. Bonnans, B Gregorutti, and P. Martinon. Block sparse linear models for learning structured dynamical systems in aeronautics. 2018a.

C. Rommel, J. F. Bonnans, B Gregorutti, and P. Martinon. Quantifying the closeness to a set of random curves via the mean marginal likelihood. 2018b.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

R. Sridharan. An introduction to the EM algorithm, motivated by Gaussian mixture models, 2014. URL http://people.csail.mit.edu/rameshvs/content/gmm-em.pdf.

L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Exploration de données pour l'optimisation de trajectoires aériennes

Mots clés : optimisation de trajectoires, identification de systèmes dynamiques, selection de variables, apprentissage multi-tâches, estimation de densité, analyse de données fonctionnelles

Résumé : Cette thèse porte sur l'utilisation de données de vols pour l'optimisation de trajectoires de montée vis-à-vis de la consommation de carburant. Dans un premier temps nous nous sommes intéressé au problème d'identification de modèles de la dynamique de l'avion dans le but de les utiliser pour poser le problème d'optimisation de trajectoire à résoudre. Nous commençont par proposer une formulation statique du problème d'identification de la dynamique. Nous l'interpretons comme un problème de régression multi-tâche à structure latente, pour lequel nous proposons un modèle paramétrique. L'estimation des paramètres est faite par l'application de quelques variations de la méthode du maximum de vraisemblance. Nous suggérons également dans ce contexte d'employer des méthodes de sélection de variable pour construire une structure de modèle de régression polynomiale dépendant des données. L'approche proposée est une extension à un contexte multi-tâche structuré du bootstrap Lasso. Elle nous permet en effet de sélectionner les variables du modèle dans un contexte à fortes corrélations, tout en conservant la structure du problème inhérente à nos connaissances métier.

Dans un deuxième temps, nous traitons la caractérisation des solutions du problème d'optimisation de trajectoire relativement au domaine de validité des modèles identifiés. Dans cette optique, nous proposons un critère probabiliste pour quantifier la proximité entre une courbe arbitraire et un ensemble de trajectoires échantillonnées à partir d'un même processus stochastique. Nous proposons une classe d'estimateurs de cette quantitée et nous étudions de façon plus pratique une implémentation nonparamétrique basé sur des estimateurs à noyau, et une implémentation paramétrique faisant intervenir des mélanges Gaussiens. Ce dernier est introduit comme pénalité dans le critère d'optimisation de trajectoire dans l'objectif l'intention d'obtenir directement des trajectoires consommant peu sans trop s'éloigner des régions de validité.

Title : Data analysis for aircraft trajectory optimization

Keywords : trajectory optimization, system identification, structured feature selection, multi-task learning, density estimation, functional data analysis

Abstract: This thesis deals with the use of flight data for the optimization of climb trajectories with relation to fuel consumption. We first focus on methods for identifying the aircraft dynamics, in order to plug it in the trajectory optimization problem. We suggest a static formulation of the identification problem, which we interpret as a structured multi-task regression problem. In this framework, we propose parametric models and use different maximum likelihood approaches to learn the unknown parameters. Furthermore, polynomial models are considered and an extension to the structured multi-task setting of the bootstrap Lasso is used to make a consistent selection of the monomials despite the high correlations among them.

Next we consider the problem of assessing the optimized trajectories relatively to the validity region of the identified models. For this, we propose a probabilistic criterion for quantifying the closeness between an arbitrary curve and a set of trajectories sampled from the same stochastic process. We propose a class of estimators of this quantity and prove their consistency in some sense. A nonparemetric implementation based on kernel density estimators, as well as a parametric implementation based on Gaussian mixtures are presented. We introduce the later as a penalty term in the trajectory optimization problem, which allows us to control the trade-off between trajectory acceptability and consumption reduction.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France